Karlsruhe Institute of Technology

Eine neue Methode zum robusten Entwurf von Regressionsmodellen bei beschränkter Rohdatenqualität

Zur Erlangung des akademischen Grades
Doktor der Ingenieurwissenschaften
der KIT-Fakultät für Maschinenbau
Karlsruher Institut für Technologie (KIT)

genehmigte
Dissertation
von

Wolfgang Doneit, M.Sc.

Tag der mündlichen Prüfung:	20. Februar 2018
Hauptreferent:	apl. Prof. Dr.-Ing Ralf Mikut
Korreferent:	Prof. Dr. Rudolf Kruse
Korreferent:	Prof. Dr.-Ing. Veit Hagenmeyer
Korreferent:	PD Dr.-Ing. Markus Reischl

Zusammenfassung

Die gestiegene preiswert verfügbare Rechenleistung und die Akzeptanz komplexer datengetriebener Modelle führt inzwischen häufig zu einer Substitution regelbasierter Expertensysteme durch automatisiert entworfene, nicht interpretierbare Black-Box-Modelle. Beim Entwurf solcher Modelle bestimmt die Datenqualität maßgeblich die Anwendbarkeit der Modelle. Ein Aspekt der Datenqualität ist die Datenabdeckung. Sie beschreibt, ob die zum Modellentwurf verfügbare Datenbasis alle potentiellen Anwendungsfälle des zu bildenden Modells abdeckt, d.h. ob ein solches Modell in der Anwendung zuverlässig ist und nicht unvorhergesehen versagt. In dieser Arbeit wird der automatisierte Entwurf von Regressionsmodellen unter der besonderen Berücksichtigung eingeschränkter Datenabdeckung untersucht. Dazu werden zunächst Bewertungskriterien entwickelt, um unterschiedliche Phänomene einer eingeschränkten Datenabdeckung zu quantifizieren. Weiterhin werden neue Bewertungskriterien für Regressionsmodelle vorgestellt, die Modelle in Bereichen geringer Datenabdeckung lokal bewerten können. Als Erweiterung des Modellentwurfsprozesses wird zum einen gezeigt, wie Vorwissen über den Funktionsverlauf systematisch erfasst und in nichtlineare Regressionsprobleme mit Hilfe von Stützvektor-Regressionen integriert werden kann. Zum anderen wird ein neues automatisiertes Entwurfsverfahren für hybride Modelle vorgestellt. Die hybriden Modelle bestehen aus zwei unterschiedlich komplexen Modellen. Die nichtlineare Funktion, um die beiden Modelle in Abhängigkeit der lokalen Datenabdeckung zu wichten, gibt implizit eine Abschätzung der Vertrauenswürdigkeit des Modells. Die Praktikabilität der Bewertungskriterien sowie die Überlegenheit automatisiert entworfener hybrider Modelle gegenüber automatisiert selektierter herkömmlicher Modelle wird auf simulierten Datensätzen, bekannten Benchmark-Datensätzen sowie für reale Anwendungen gezeigt. Die realen Anwendungen stammen aus dem Bereich des Turbomaschinenbaus, der Energieinformatik sowie der Medizintechnik.

Vorwort

Die vorliegende Arbeit entstand während meiner Tätigkeit am Institut für Angewandte Informatik (IAI) am Karlsruher Institut für Technologie (KIT). Ich danke vielmals Herrn Prof. Dr.-Ing. Ralf Mikut, der mir ermöglicht hat mit vielen Freiheiten an einem so spannenden und vielfältigen Thema zu arbeiten. Die Arbeit wäre ohne die beispiellose fachliche und persönliche Betreuung nicht entstanden.

Ich möchte mich bei Herrn Prof. Prof. E.h. Dr.-Ing. habil. Dr. E.h. mult. Georg Bretthauer bedanken, der mein Interesse an der datengetriebenen Modellierung geweckt und mir die Arbeit als wissenschaftlicher Mitarbeiter ermöglicht hat. Herrn Prof. Dr.-Ing. Veit Hagenmeyer danke ich für die wertvollen Hinweise zu meinen Arbeiten.

Weiterhin danke ich Herrn PD Dr.-Ing. Markus Reischl für fachliche Ratschläge, unzählige Korrekturen und eine gesunde Portion Pragmatismus. Für seine Geduld und Unterstützung im Bereich restringierter Optimierung danke ich Herrn PD Dr.-Ing. Lutz Gröll. Ich bin sehr dankbar, dass sich Herr Prof. Dr. Rudolf Kruse als Gutachter eingebracht hat und dass sich Herr Prof. Dr.-Ing. Martin Gabi als Prüfungsvorsitzender angeboten hat.

Stellvertretend für alle am Projekt I-CARE Beteiligten danke ich Frau Jana Lohse sowie Frau Prof. Dr.-Ing. Tanja Schultz. Als Ingenieur in einem sozialwissenschaftlichen Umfeld mitwirken zu dürfen war fachlich und persönlich sehr bereichernd. Für die Zusammenarbeit am Projekt TELMYOS danke ich Herrn Dr.-Ing. Rüdiger Rupp, Herrn Andreas Kogut, Herrn Prof. Dr. med. David Liebetanz, Frau Leonie Schmalfuß, Herrn Manuel Hewitt und besonders Herrn Dr.-Ing. Michele René Tuga. Das Projekt begleitete mich bereits während des Studiums und begeisterte mich für die angewandte Informatik. Herrn Dr.-Ing. Tim Pychynski danke ich für die wertvollen Vorarbeiten und die Gespräche über die Modellierung des Durchflussverhaltens von Labyrinthdichtungen.

Zudem bedanke ich mich bei Herrn Baifan Zhou, Frau Luisa Stäb, Herrn Ajit Basarur, Herrn Peter Maucher sowie Herrn Roman Bruch, die durch Praktika und Abschlussarbeiten einen wichtigen Beitrag geleistet haben.

Bei meinen Kolleginnen und Kollegen Nicole Ludwig, Andreas Bartschat, Jorge Ángel González Ordiano, Benjamin Schott, Dr.-Ing. Johannes Stegmaier und Simon Waczowicz bedanke ich mich für die angenehme und freundschaftliche Atmosphäre.

Ich danke meinen Eltern Susanne und Wolfgang sowie meiner Freundin Julia für den Rückhalt und die Unterstützung.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Abkürzungsverzeichnis

Abkürzung	Bedeutung
1K-SVM	Ein-Klassen-Stützvektor-Maschine (engl. *one-class support vector machine*)
AIC	Akaike Informationskriterium (engl. *Akaike information criterion*)
ANN	künstliches neuronales Netz (engl. *artificial neural network*)
BIC	Bayessches Informationskriterium (engl. *Bayesian information criterion*)
CFD	numerische Strömungsmechanik (engl. *computational fluid dynamics*)
CNN	faltendes neuronales Netz (engl. *convolutional neural network*)
COD	Fluch der Dimensionalität (engl. *curse of dimensionality*)
CV	Kreuzvalidierung (engl. *crossvalidation*)
DNN	tiefes neuronales Netz (engl. *deep neural network*)
DT	Datentupel
ELM	Extreme-Lern-Maschine (engl. *extreme learning machine*)
EMG	Elektromyographie
FS	unscharfes System (engl. *fuzzy system*)
GPR	Gaußsche-Prozess-Regression
GSMO	generalisierte sequentielle minimale Optimierung (engl. *generalized sequential minimal optimization*)
HDNN	hierarchisches tiefes neuronales Netz (engl. *hierarchical deep neural Network*)
k-NN	k-Nächste-Nachbarn
LMN	lokales Modellnetz

Symbol	Bedeutung
LOF	lokaler Ausreißerfaktor (engl. *local outlier factor*)
LOLIMOT	lokal linearer Modellbaum (*local linear model tree*)
MAD	mittlere absolute Abweichung (engl. *mean absolute deviation*)
MAE	mittlerer absoluter Fehler (engl. *mean absolute error*)
MARS	multivariate adaptive Regressions-Splines (engl. *multivariate adaptive regression splines*)
MLP	mehrschichtiges Perzeptron (engl. *multilayer perceptron*)
MMS	Mensch-Maschine-Schnittstelle
PCA	Hauptkomponentenanalyse (engl. *principal component analysis*)
PLS	partielle Regression (engl. *partial least squares*)
PV	Photovoltaik
QP	quadratisches Programm
r	relative Feuchtigkeit (engl. *relative humidity*)
RBFN	Radiale-Basisfunktionen-Netz
RR	Ridge-Regression
RSS	Summe der Fehlerquadrate (engl. *residual sum of squares*)
SMO	sequentielle minimale Optimierung (engl. *sequential minimal optimization*)
SP	Luftdruck (engl. *surface pressure*)
SSRD	Solarstrahlung an der Oberfläche (engl. *surface solar radiation downwards*)
STRD	Wärmestrahlung an der Oberfläche (engl. *surface thermal radiation downwards*)
SV	Stützvektor (engl. *support vector*)
SVM	Stützvektor-Maschine (engl. *support vector machine*)
TCC	Bewölkung (engl. *total cloud cover*)
tciw	Eiswassergehalt (engl. *total column ice water*)
tclw	Flüssigwassergehalt (engl. *total column liquid water*)
TLS	orthogonale Regression (engl. *total least squares*)
TP	Niederschlag (engl. *total precipitation*)
TSR	Solarstrahlung oberhalb der Atmosphäre (engl. *top net solar radiation*)

Symbol	Bedeutung
VI	Gültigkeitsindex (engl. *validity index*)
VSM	Ansatz der virtuellen Datentupel (engl. *virtual samples method*)
VV	Verlaufsvalidierung
WLS	gewichtete Regression (engl. *weighted least squares*)

Symbolverzeichnis

Symbol	Bedeutung
$10p[t]$	Zeitreihe der absoluten Windgeschwindigkeit
$10u[t]$	Zeitreihe der zonalen Windgeschwindigkeit
$10v[t]$	Zeitreihe der meridionalen Windgeschwindigkeit
$2T[t]$	Zeitreihe der Lufttemperatur zwei Meter über dem Erdboden
a	Filterparameter
AIC	Akaike Informationskriterium
b	skalarer Offset einer SVM
$\hat{b}$	geschätzter skalarer Offset einer SVM
BIC	Bayessches Informationskriterium
c	Zentrum einer radialen Basisfunktion alternativ: Hilfsgröße zur Bestimmung des Bewertungskriteriums für Orthogonalität
C	Metaparameter zur Regularisierung in Stützvektor-Regressionen
C_{d}	Durchflussverhalten
c_j	Anzahl unterschiedlicher Ausprägungen von x_j
C_{p}	Mallow's Kriterium für Anpassung
C_{reg}	Regularisierungsparameter für reguläre Datentupel
$C_{\mathrm{R},i}$	Regularisierungsparameter für i-tes virtuelle Datentupel
$d^{u,v}_{\mathrm{Chebyshev}}$	Chebyshev-Distanz unter der Berücksichtigung der u-ten und v-ten Eingangsgröße
$\boldsymbol{d}_{\text{k-NN},x_j,x_l}$	Vektor der euklidischen Distanzen aller Datentupel zum k-ten nächsten Nachbarn unter Berücksichtigung von x_j und x_l

Symbol	Bedeutung
$d_{\text{k-NN},x_j,x_l,\max}$	maximale euklidische Distanz eines Datentupels zum k-ten nächsten Nachbarn unter Berücksichtigung von x_j und x_l
D_{Sim}	Simulierter Datensatz
$D_{\text{Sim,Lern,1}}$	erster simulierter Lerndatensatz
$D_{\text{Sim,Lern,2}}$	zweiter simulierter Lerndatensatz
$D_{\text{Sim,Lern,3}}$	dritter simulierter Lerndatensatz
$D_{\text{Sim,Test}}$	simulierter Testdatensatz
$\boldsymbol{e}$	Einheitsvektor
$g_{\text{g,E}}(\boldsymbol{x})$	gradueller Fehler des Extrapolationsmodells in hybriden Modellstrukturen
$g_{\text{g,I}}(\boldsymbol{x})$	gradueller Fehler des Interpolationsmodells in hybriden Modellstrukturen
$e_{j,\text{In},l}$	Streuung in x_j-Richtung der in $\mathcal{I}_{\text{In},l}$ enthaltenen Datentupel
$e_{j,\text{Out},l}$	Streuung in x_j-Richtung der in $\mathcal{I}_{\text{Out},l}$ enthaltenen Datentupel
$e_{l,\text{In},j}$	Streuung in x_l-Richtung der in $\mathcal{I}_{\text{In},j}$ enthaltenen Datentupel
$e_{l,\text{Out},j}$	Streuung in x_l-Richtung der in $\mathcal{I}_{\text{Out},j}$ enthaltenen Datentupel
f	Modellstruktur
$\mathcal{F}$	Merkmalsraum
f_{E}	Extrapolationsmodell
f_{Hybrid}	optimiertes hybrides Modell
f_{I}	Interpolationsmodell
f_{Ref}	Referenzmodell, ausgewählt gemäß herkömmlicher Modellselektion
F_{Rel}	relative Häufigkeit
$\hat{f}$	Regressionsmodell
$g(\cdot)$	allgemeine Funktion alternativ: Ein-Klassen-Klassifikator
$g_{\text{1K-SVM}}$	binäre Entscheidungsfunktion einer 1K-SVM
$\bar{g}$	mittlere Bewertung eines Datensatzes durch einen Ein-Klassen-Klassifikator
$\tilde{g}_{\text{1K-SVM}}$	kontinuierliche Entscheidungsfunktion einer 1K-SVM

Symbol	Bedeutung
HC_d	Honigwabendurchmesser einer Labyrinthdichtung
i_Opt	Index des aus einem Modellpool gewählten Interpolationsmodells $f_{\mathrm{I},i}$ in der Optimierung einer hybriden Modellstruktur
$i_\mathrm{Selektion}$	Index des aus einem Modellpool gewählten Modells f_i in einer Modellselektion
$\mathcal{I}_\mathrm{In}$	Indexmenge zur Bestimmung des Bewertungskriteriums für Orthogonalität
$\mathcal{I}_\mathrm{Out}$	Indexmenge zur Bestimmung des Bewertungskriteriums für Orthogonalität
$\mathcal{I}_{\boldsymbol{X}}$	Indexmenge aller in $\boldsymbol{X}$ enthaltener Eingangsvektoren
j_Opt	Index des aus einem Modellpool gewählten Extrapolationsmodells $f_{\mathrm{E},j}$ in der Optimierung einer hybriden Modellstruktur
k	diskreter Zeitpunkt
$k(\cdot)$	Kernelfunktion
$\boldsymbol{K}$	Kernelmatrix
l_Opt	Index des aus einem 1K-SVM-Pool gewählten Koordinators g_l in der Optimierung einer hybriden Modellstruktur
m	Filterparameter
MAE	mittlerer absoluter Fehler
n	Spitzenanzahl einer Labyrinthdichtung
N	Anzahl der Datentupel
n_1	Index von $\boldsymbol{x}_\mathrm{a}$ für die Verlaufsvalidierung
n_2	Index von $\boldsymbol{x}_\mathrm{b}$ für die Verlaufsvalidierung
N_CV	Anzahl der Teildatensätze einer Kreuzvalidierung
N_LM	Anzahl lokaler Modelle in einem LMN
N_R	Anzahl an Restriktionen, Straftermen oder Anforderungen
N_RBF	Anzahl radialer Basisfunktionen in einem RBFN
N_S	Nutversatz einer Labyrinthdichtung
N_W	Nutweite einer Labyrinthdichtung
$P[t]$	allgemeine Zeitreihe
$P[t]$	alternativ: Leistungszeitreihe

Symbol	Bedeutung
$\hat{P}[t]$	prognostizierte Leistungszeitreihe
p	Anzahl Eingangsgrößen
Q	Bewertungsergebnis
$Q(\boldsymbol{\theta})$	Gütefunktion in Abhängigkeit des Parametervektors
$q_{\text{Aus,Ind},x_j,x_l}$	Indikator für Ausreißer von x_j und x_l
$Q_{\text{Aus,min}}$	minimale Bewertung von Ausreißern
q_{Aus,x_j,x_l}	Bewertungskriterium für Ausreißer von x_j und x_l
$Q_{\text{Cluster,min}}$	minimale Bewertung von Clustern
$q_{\text{Cluster},x_j,x_l}$	Bewertungskriterium für Cluster von x_j und x_l
$Q_{\text{Konfig,min}}$	minimale Bewertung von Konfigurationen
q_{Konfig,x_j}	Bewertungskriterium für Konfigurationen von x_j
$Q_{\text{Korr,min}}$	minimale Bewertung von Korrelationen
q_{Korr,x_j,x_l}	Bewertungskriterium für Korrelationen von x_j und x_l
Q_{Opt}	Gütewert eines Optimierungsproblems
$Q_{\text{Ortho,min}}$	minimale Bewertung von Orthogonalität
q_{Ortho,x_j,x_l}	Bewertungskriterium für Orthogonalität von x_j und x_l
Q_{VV}	Aggregierte lokale Bewertung der Verlaufsvalidierung
$Q_{\text{VV},1}$	Kriterium der Verlaufsvalidierung hinsichtlich lokaler Extremwerte
$Q_{\text{VV},2}$	Kriterium der Verlaufsvalidierung hinsichtlich der Steigung
$Q_{\text{VV},3}$	Kriterium der Verlaufsvalidierung hinsichtlich der Abweichungen von einer linearen Interpolation
$q_{\text{VV,min}}$	Mindestbewertung der Verlaufsvalidierung
$\bar{Q}_{\text{Aus}}$	mittlere Bewertung von Ausreißern
$\bar{Q}_{\text{Cluster}}$	mittlere Bewertung von Clustern
$\bar{Q}_{\text{Konfig}}$	mittlere Bewertung von Konfigurationen
$\bar{Q}_{\text{Korr}}$	mittlere Bewertung von Korrelationen
$\bar{Q}_{\text{Ortho}}$	mittlere Bewertung von Orthogonalität
$\tilde{Q}$	Aggregierte Bewertung der Datenabdeckung
R^2	Bestimmtheitsmaß
r_n	Residuum beim Eingangsvektor $\boldsymbol{x}_n$
$r[t]$	Zeitreihe der relativen Feuchtigkeit

Symbol	Bedeutung
r_{x_j,x_l}	empirischer Korrelationskoeffizient zwischen x_j und x_l
RSS	Summe der Fehlerquadrate
s	Hilfsvariable im SMO-Algorithmus alternativ: Spaltweite einer Labyrinthdichtung
S	Anzahl an Schichten eines künstlichen neuronalen Netzes
S_{B}	Spitzenbreite einer Labyrinthdichtung
S_{H}	Spitzenhöhe einer Labyrinthdichtung
$SP[t]$	Zeitreihe des Luftdrucks
$SSRD[t]$	Zeitreihe der Solarstrahlung an der Oberfläche
ST_{H}	Stufenhöhe einer Labyrinthdichtung
$STRD[t]$	Zeitreihe der Wärmestrahlung an der Oberfläche
ST_{S}	Stufenshift einer Labyrinthdichtung
t	Teilung einer Labyrinthdichtung
t_0	Referenzzeitpunkt
T_{a}	Abtastintervall
$TCC[t]$	Zeitreihe der Bewölkung
$tciw[t]$	Zeitreihe des Eiswassergehalts
$tclw[t]$	Zeitreihe des Flüssigwassergehalts
$TP[t]$	Zeitreihe des Niederschlags
$TSR[t]$	Zeitreihe der Solarstrahlung oberhalb der Atmosphäre
v_{Dip}	Dip-Index
$v_{\mathrm{max,a,b}}$	maximale Steigung eines Modells zwischen $\boldsymbol{x}_a$ und $\boldsymbol{x}_b$
$v_{\mathrm{rot}}[k]$	Zeitreihe der rotatorischen Geschwindigkeit
$v_{\mathrm{trans}}[k]$	Zeitreihe der translatorischen Geschwindigkeit
w	Wichtungsfaktor
$\boldsymbol{w}$	Parametervektor einer Stützvektor-Maschine im Merkmalsraum (primäre Formulierung)
$\hat{\boldsymbol{w}}$	geschätzter Parametervektor einer Stützvektor-Maschine im Merkmalsraum (primäre Formulierung)
$\boldsymbol{W}$	Wichtungsmatrix einer gewichteten Regression
$W(\cdot)$	Zielfunktion im GSMO-Algorithmus
$W'(\theta_v)$	symmetrische Ableitung von W in Abhängigkeit von θ_v

Symbol	Bedeutung
$\boldsymbol{x}$	Eingangsvektor
$\boldsymbol{X}$	Datenmatrix
$\boldsymbol{x}^*$	transformierter Eingangsvektor
$\boldsymbol{x}_\mathrm{a}$	erster von zwei Eingangsvektoren, zwischen denen die Verlaufsvalidierung durchgeführt wird
$\boldsymbol{x}_\mathrm{b}$	zweiter von zwei Eingangsvektoren, zwischen denen die Verlaufsvalidierung durchgeführt wird
$x_{\mathrm{f},i}[k]$	Zeitreihe des gefilterten Signals des i-ten Sensors
$x_{\mathrm{f,max},i}$	Normierungsparameter des i-ten Sensors
$x_{\mathrm{f,min},1}$	Normierungsparameter des i-ten Sensors
$x_{\mathrm{I},i,j}$	Intention des i-ten Sensors im j-Kalibrierungsschritt
$x[k]$	Zeitreihe des Rohsignals
$\boldsymbol{X}_\mathrm{Lern}$	Datenmatrix des Lerndatensatzes
$x_{\mathrm{mean},i}$	Mittelwert-Gleichanteil des i-ten Sensors
$\boldsymbol{x}_n$	n-ter Eingangsvektor eines Datensatzes
$x_{\mathrm{n,bikalib},i,j}[k]$	Zeitreihe der normierten Signale des i-ten Sensors im j-ten Kalibrierungsschritt
$x_{\mathrm{n},i}[k]$	Zeitreihe des normierten Signals des i-ten Sensors
$x_{\mathrm{n,med},i,j}$	Median von $x_{\mathrm{n,bikalib},i,j}[k]$
$x_{\mathrm{r},i}[k]$	Zeitreihe des gleichgerichteten Signals des i-ten Sensors
$\boldsymbol{x}_\mathrm{R}$	Eingangsvektor einer Restriktion oder eines Strafterms
$\boldsymbol{X}_\mathrm{Sim,Lern,1}$	Datenmatrix der ersten Lerndaten des simulierten Datensatzes
$\boldsymbol{X}_\mathrm{Sim,Lern,2}$	Datenmatrix der zweiten Lerndaten des simulierten Datensatzes
$\boldsymbol{X}_\mathrm{Sim,Lern,3}$	Datenmatrix der dritten Lerndaten des simulierten Datensatzes
$\boldsymbol{X}_\mathrm{Sim,Test}$	Datenmatrix der Testdaten des simulierten Datensatzes
$\hat{x}_{\mathrm{I},i}[k]$	Zeitreihe der geschätzten Intention des i-ten Sensors
$\boldsymbol{y}$	Zielgrößenvektor
$\boldsymbol{y}_{i,\mathrm{Sim,Lern,1}}$	i-ter Zielgrößenvektor der ersten Lerndaten des simulierten Datensatzes
$\boldsymbol{y}_{i,\mathrm{Sim,Lern,2}}$	i-ter Zielgrößenvektor der zweiten Lerndaten des simulierten Datensatzes

Symbol	Bedeutung
$\boldsymbol{y}_{i,\mathrm{Sim,Lern,3}}$	i-ter Zielgrößenvektor der dritten Lerndaten des simulierten Datensatzes
$\boldsymbol{y}_{i,\mathrm{Sim,Test}}$	i-ter Zielgrößenvektor der Testdaten des simulierten Datensatzes
$\boldsymbol{y}_{\mathrm{Lern}}$	Zielgrößenvektor des Lerndatensatzes
$y_{\max}$	Maximalwert der Zielgröße
$y_{\min}$	Minimalwert der Zielgröße
y_n	n-ter Wert des Zielgrößenvektors
y_{R}	Zielgröße einer Restriktion oder eines Strafterms
$\boldsymbol{y}_{\mathrm{Val}}$	Zielgrößenvektor des Validierungsdatensatzes
$\hat{y}_{\mathrm{a},,j}$	geschätzte Zielgröße des untersuchten Modells an der j-ten diskreten Stelle zwischen $\boldsymbol{x}_{\mathrm{a}}$ und $\boldsymbol{x}_{\mathrm{b}}$
$\hat{y}_{\mathrm{lin},j}$	lokale lineare Interpolation an der j-ten diskreten Stelle der Verlaufsvalidierung
$\hat{y}_{\max}$	Maximalwert der geschätzten Zielgröße des untersuchten Modells zwischen $\boldsymbol{x}_{\mathrm{a}}$ und $\boldsymbol{x}_{\mathrm{b}}$
$\hat{y}_{\min}$	Minimalwert der geschätzten Zielgröße des untersuchten Modells zwischen $\boldsymbol{x}_{\mathrm{a}}$ und $\boldsymbol{x}_{\mathrm{b}}$
$\hat{y}_{\mathrm{Val}}$	geschätzter Zielgrößenvektor eines Regressionsmodells für den Validierungsdatensatz
z	innerer Zustand eines Neurons
γ	Wichtungsfaktor von Straftermen
ε	insensitiver Bereich einer Lossfunktion
ε_{R}	Minimum aller geforderten ε-Parameter für virtuelle Datentupel
$\varepsilon_{\mathrm{reg}}$	ε-Parameter für reguläre Datentupel
$\varepsilon_{\mathrm{R},i,\mathrm{gef}}$	geforderter ε-Parameter für i-tes virtuelles Datentupel
$\boldsymbol{\theta}$	Parametervektor der freien Parameter
θ_u	freier Parameter im Zwei-Variablen-Problem im GSMO-Algorithmus
θ_v	freier Parameter im Zwei-Variablen-Problem im GSMO-Algorithmus
$\hat{\boldsymbol{\theta}}$	geschätzter Parametervektor für eine Stützvektor-Maschine in der dualen Formulierung

Symbol	Bedeutung
$\hat{\boldsymbol{\theta}}_i$	geschätzter Parametervektor für die Kalibrierungsfunktion des i-ten Signalkanals
$\hat{\boldsymbol{\theta}}_{\mathrm{RR}}$	geschätzter Parametervektor einer Ridge-Regression
$\hat{\theta}_u$	geschätzter Parameter im Zwei-Variablen-Problem im GSMO-Algorithmus
$\hat{\theta}_v$	geschätzter Parameter im Zwei-Variablen-Problem im GSMO-Algorithmus
$\hat{\boldsymbol{\theta}}_{\mathrm{WLS}}$	geschätzter Parametervektor einer gewichteten Regression
λ_{RR}	Regularisierungsparameter einer Ridge-Regression
ν	Anzahl an diskreten Stellen des Eingangsraums, an denen ein Modell bei der Verlaufsvalidierung betrachtet wird alternativ: Metaparameter für 1K-SVMs
$\xi^{(*)}$	Schlupfvariable
π	Druckverhältnis zwischen Eintritts- und Austrittsdruck
$\rho(r)$	Lossfunktion
σ	Streuung eines Gaußkernels
$\boldsymbol{\Sigma}$	Kovarianzmatrix eines Gaußkernels
$\hat{\boldsymbol{\Sigma}}$	geschätzte Kovarianzmatrix eines Gaußkernels
$\boldsymbol{\Sigma}$	Kovarianzmatrix der Streuung eines Gaußkernels
σ_{RBF}	Streuung einer radialen Basisfunktion
$\tau_{\mathrm{Aus},i}$	Metaparameter des Bewertungskriteriums für Ausreißer
τ_{Cluster}	Metaparameter des Bewertungskriteriums für Cluster
τ_{Ortho}	Metaparameter des Bewertungskriteriums für Orthogonalität
ϕ	Basisfunktion

1 Einführung

1.1 Bedeutung der Arbeit

Moderne mathematische Verfahren erkennen Zusammenhänge in Daten und erstellen Modelle, um Vorhersagen zu generieren. Solche Vorhersagen bestimmen immer häufiger unseren Alltag. Sei es eine vorgeschlagene Veranstaltung in sozialen Netzwerken, eine Kaufempfehlung beim Online-Versandhandel oder die Einschätzung der Kreditwürdigkeit bei Banken. In den Ingenieurwissenschaften werden die Modelle beispielsweise verwendet, um die Zusammenhänge zwischen einstellbaren Parametern (Eingangsgrößen) eines technischen Systems und einer zu optimierenden Eigenschaft (Zielgröße) abzubilden. Dadurch kann die Anzahl zeit- und kostenintensiver Simulationen und Messungen reduziert werden. Die Besonderheit solcher Modelle ist, dass sie ausschließlich auf Daten basieren und nicht auf physikalischen, soziologischen oder wirtschaftswissenschaftlichen Zusammenhängen. Sind die tatsächlichen Zusammenhänge nicht vollständig bekannt, können solche *datengetriebenen Modelle* die Realität in einigen Fällen genauer abbilden, als von Experten entwickelte analytische Modelle. Zudem wird durch den (teil-)automatisierten Entwurf der datengetriebenen Modelle der Aufwand für Domänenexperten reduziert. Die Qualität der verwendeten Daten beeinflusst maßgeblich die Güte eines Modells und seiner Vorhersagen. Fehler in der Datenaufzeichnung, -speicherung und -übertragung oder die Aufzeichnung von nicht relevanten, bzw. redundanten Daten sind nur einige Gründe für schlechte Datenqualität und unzuverlässige Vorhersagen.

Ein gutes Modell erfüllt mehrere Bedingungen. Es bildet die Eingangsgrößen auf die zugeordnete Zielgröße ab. Es kann für noch nicht erfasste Ausprägungen der Eingangsgrößen Werte für die Zielgröße vorhersagen, die denen des realen Systems ähneln. Und es ist häufig wünschenswert, dass

das Modell und seine Parameter interpretierbar sind, um Rückschlüsse auf das reale System ziehen zu können.

Datengetriebene Modellbildung wird zum Beispiel in [1] dazu verwendet den Einfluss verschiedener geometrischer Parameter in Labyrinthdichtungen auf deren Durchflussverhalten zu untersuchen und den Durchfluss für neue Geometriekonfigurationen vorherzusagen. Dazu werden Daten aus Messungen an Labyrinthdichtungen verschiedener Geometriekonfigurationen verwendet.

Während die Labyrinthdichtung bei der Modellbildung als statisches System angesehen werden kann, gibt es im ingenieurwissenschaftlichen Umfeld erhöhten Bedarf an der Modellierung dynamischer (nichtlinearer) Systeme. In dieser Arbeit werden ausschließlich statische Modelle behandelt. Allerdings werden statische Modelle auch in der Abbildung dynamischer Systeme verwendet. So können beispielsweise dynamische nichtlineare Systeme durch die Hintereinanderschaltung von einem dynamischen linearen Modell und einem statischen nichtlinearen Modell abgebildet werden [2].

Die Auswahl einer Modellfamilie, der Modellstruktur und anderer Randbedingungen kann auf Grundlage domänenspezifischen Wissens (*White-Box-Modeling*), ausschließlich datengetrieben (*Black-Box-Modeling*) oder in einer Kombination beider Möglichkeiten (*Grey-Box-Modeling*) erfolgen. Individuelle Kalibrierungen von Mensch-Maschine-Schnittstellen (MMS) erfolgen beispielsweise kombiniert: Es werden nutzerindividuell Daten erhoben, die abgebildet werden, um die Schnittstelle dem Anwender anzupassen. Es werden jedoch auch Randbedingungen, z.B. aus Sicherheitsaspekten, in die Modellbildung und -anwendung in Form von Restriktionen, zu verwendender Modellstruktur oder nachverarbeitenden Schritten integriert [3].

Verschiedene Einschränkungen in der Datenqualität beeinflussen die Modellgüte. Ausreißer, die durch fehlerhafte Messungen oder Datenübertragungen entstehen können, haben häufig großen Einfluss auf die Modellbildung. Unterscheiden sich die Eingangsgrößen in der Modellanwendung maßgeblich von den Daten, die zur Modellbildung zur Verfügung standen, verlieren datengetriebene Modelle ihre Zuverlässigkeit. Das resultiert oftmals aus einer unzureichenden Datenabdeckung. Modelle, die auf der Grundlage von domänenspezifischem Wissen erstellt werden, sind in dem Fall zuverlässiger. Der Vorteil der datengetriebenen Modelle liegt in der möglichen Abbildung von Zusammenhängen, die dem Domänenexperten unbekannt

sind und der damit erhöhten Genauigkeit des Modells. Weiterhin können widersprüchliche Teildatensätze existieren oder als statisch betrachtete Zusammenhänge zeitvariante Änderungen erfahren, wodurch die Modelle ihre Zuverlässigkeit verlieren. Für eine erfolgreiche Anwendung datengetriebener Modelle ist es demnach unerlässlich, zuvor die zu erwartende Modellgüte, bzw. die voraussichtliche Zuverlässigkeit einzelner Vorhersagen zu bestimmen, welche unter anderem von der Datenbasis und ihrer Qualität abhängt.

Die vorliegende Arbeit stellt deshalb zunächst eine ausführliche Taxonomie für Datenqualität in der datengetriebenen Modellbildung vor. Die unzureichende Datenabdeckung als Einschränkung der Datenqualität wird anschließend ausführlich behandelt. Es werden entsprechende Kriterien zur Quantifizierung einer unzureichenden Datenabdeckung vorgestellt. Da die beschränkte Datenqualität nicht nur die Güte der Modelle beeinträchtigen, sondern auch die Aussagekraft bekannter Validierungsverfahren mindern kann, werden neue Bewertungskriterien und Validierungsverfahren zur Modellbewertung bei beschränkter Datenqualität entwickelt. Weiterhin werden neue Methoden zum Modellentwurf vorgestellt. Besondere Berücksichtigung erfährt dabei die Wahl einer geeigneten Modellstruktur und die Möglichkeit, Vorwissen in den Modellentwurf zu integrieren.

Zur Validierung der Methoden werden sowohl simulierte Benchmark-Datensätze als auch im Bereich der datengetriebenen Modellbildung bereits etablierte Benchmark-Datensätze verwendet. Weiterhin werden die Methoden auf Echtwelt-Probleme aus dem Turbomaschinenbau, der Energieinformatik und der Medizintechnik angewandt.

1.2 Darstellung des Entwicklungsstands

1.2.1 Regressionsprobleme

Liegen Daten in Form von Eingangsvektoren $\boldsymbol{x}_{\mathrm{n}}^{T} = (x_{\mathrm{n},1}, \ldots, x_{\mathrm{n,p}})$ und ihnen zugeordneten skalaren Werten einer Zielgröße y_{n} vor, dann wird

der Datensatz durch die *Datenmatrix* $\boldsymbol{X}$ und den *Zielgrößenvektor* $\boldsymbol{y}$ beschrieben:

$$\boldsymbol{X} = \begin{pmatrix} x_{1,1} & \dots & x_{1,\mathrm{p}} \\ \vdots & \ddots & \vdots \\ x_{\mathrm{N},1} & \dots & x_{\mathrm{N,p}} \end{pmatrix} \text{ und } \boldsymbol{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_{\mathrm{N}} \end{pmatrix} \tag{1.1}$$

Sie bilden den *Datensatz*, der aus N *Datentupeln* (DT) $(\boldsymbol{x}_n, y_n)$ besteht. Die Anzahl p der Eingangsgrößen definiert den Eingangsraum (*design space, input space*) $\mathbb{R}^p$, in dem jeder Eingangsvektor $\boldsymbol{x}_n^T, n = 1, \dots, N$ einen Punkt darstellt. Die Modellierung der Zusammenhänge zwischen p Eingangsgrößen und einer kontinuierlichen Zielgröße $y \in \mathbb{R}$ wird *Regression* genannt. Die Regression stellt eine formalisierte Problemstellung im Bereich des *Data Minings* dar und dient zur Abgrenzung von anderen Problemstellungen wie der *Klassifikation*, bei der von einer diskreten Zielgröße ausgegangen wird. In der Statistik wird bei einer Regression davon ausgegangen, dass der strukturelle Zusammenhang zwischen Eingangs- und Zielgröße bekannt ist und die Zielgröße mit einem signifikanten Störterm behaftet ist. Bei einer *Approximation* wird dagegen von einem unbekannten Zusammenhang und dem Fehlen des Störterms ausgegangen. Daher liegt streng genommen ein *Approgressionsproblem* vor [4], bei dem von einer unbekannten Struktur sowie einem signifikanten Störterm ausgegangen wird. In dieser Arbeit wird aber aufgrund der Verständlichkeit der Begriff „Regression“ verwendet. Die Modellbildung in der Regression wird *Entwurf eines Regressionsmodells* genannt. Die Vorhersagen eines Regressionsmodells heißen *Prädiktionen*.

Für Zeitreihenprädiktionen [5–7] müssen die Daten restrukturiert werden. Sei beispielsweise $P[k]$ der Wert einer Zeitreihe P zum Zeitpunkt k[1], die durch einen autoregressiven Prozess erster Ordnung modelliert wird und deren Werte für die aufeinanderfolgenden Zeitpunkte $k = 1, \dots, N+1$ bekannt sind, dann ergeben sich Datenmatrix und Zielgrößenvektor zu

$$\boldsymbol{X}^{N\times 1} = (P[1], \dots, P[N])^T \text{ und } \boldsymbol{y}^{N\times 1} = (P[2], \dots, P[N+1])^T . \tag{1.2}$$

Dadurch entsteht auch für Zeitreihenprädiktionen die vorgestellte Datenstruktur.

[1] k lässt sich mit dem Referenzzeitpunkt t_0 und dem Abtastintervall T_a als $k = t_0 + kT_a$ darstellen

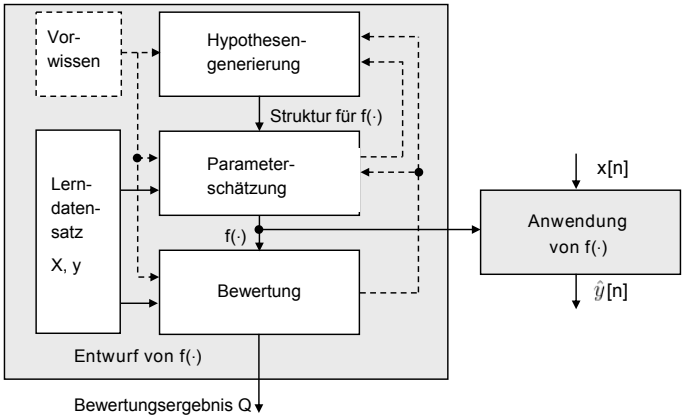

Abbildung 1.1: Entwurfsprozess von Regressionsmodellen [8]

Regressionsmodelle dienen dazu, die Zusammenhänge zwischen Eingangs- und Zielgröße derart abzubilden, dass beispielsweise Werte der Zielgröße für Punkte im Eingangsraum vorhergesagt werden können, die nicht im Datensatz vorhanden sind, oder Aussagen über globale Extremwerte und Ähnliches getroffen werden können. Beim Entwurf eines Regressionsmodells ist nicht nur die Näherung an die vorhandenen Daten wichtig, sondern auch die Sicherung einer Generalisierungsfähigkeit durch das Finden relevanter Zusammenhänge. Generell kann bei datengetriebenen Modellen aber davon ausgegangen werden, dass ihre Prädiktionsgüte mit zunehmender Entfernung von den zur Verfügung stehenden Daten sinkt. Solche Prädiktionen werden *Extrapolationen* genannt und die entsprechenden Bereiche des Eingangsraums *Extrapolationsbereiche*. Prädiktionen in Bereichen des Eingangsraums, in denen Lerndaten zum Modellentwurf vorlagen, werden *Interpolationen* und die entsprechenden Bereiche *Interpolationsbereiche* genannt.

Abbildung 1.1 zeigt den prinzipiellen Ablauf des Entwurfsprozesses von Regressionsmodellen nach [8]. In der *Hypothesengenerierung* wird eine Modellfamilie bzw. Modellstruktur gewählt. Dazu gehört auch die Selektion oder Transformation geeigneter Merkmale aus den Eingangsgrößen sowie die Festlegung von Metaparametern, die in Abhängigkeit der mit ihnen erreichten Modellgüte gewählt werden. Anschließend werden die freien

Parameter (Parametervektor $\boldsymbol{\theta}$) der Modellstruktur in der *Parameterschätzung*[2] an den Lerndatensatz angepasst und in der *Bewertung* wird ein Bewertungsergebnis Q für die Modellstruktur bestimmt. Mit Hilfe der Güte können unterschiedliche Modellstrukturen verglichen werden. Das für den Anwendungsfall am besten geeignete Modell kann dann zur Nutzung in der Anwendung verwendet werden. Zur Hypothesengenerierung, der Parameterschätzung und zur Bewertung kann *Vorwissen* genutzt werden, um die Modellbildung zu verbessern. Das Ergebnis des Entwurfsprozesses ist eine Funktion $\hat{f}(\cdot)$[3]. Im Weiteren werden bekannte Modellfamilien diskutiert.

Eine gängige Modellfamilie sind globale Polynome. Es handelt sich dabei oft um parameterlineare Modelle der Form

$$f(\boldsymbol{x}, \boldsymbol{\theta}) = \boldsymbol{\theta}^T \begin{pmatrix} 1 \\ \boldsymbol{x} \end{pmatrix} \tag{1.3}$$

Werden die Eingangsgrößen ohne Selektion oder Transformation (Hypothesengenerierung) in der Parameterschätzung verwendet, ergibt sich das lineare Modell

$$f(\boldsymbol{x}, \boldsymbol{\theta}) = \theta_0 + \theta_1 x_1 + \ldots + \theta_p x_p. \tag{1.4}$$

Werden die Eingangsgrößen in der Hypothesengenerierung nichtlinear transformiert, sodass z.B.

$$\boldsymbol{x}^* = \left(x_1, \ldots, x_p, x_1^2, \ldots, x_p^2\right)^T \tag{1.5}$$

gilt, dann ergibt sich ein entsprechendes nichtlineares Modell der Form

$$f(\boldsymbol{x}^*, \boldsymbol{\theta}) = \theta_0 + \theta_1 x_1 + \ldots + \theta_p x_p + \theta_{p+1} x_1^2 + \ldots + \theta_{2p} x_p^2, \tag{1.6}$$

das weiterhin parameterlinear ist. In Polynomansätzen gemäß (1.3) ist die Interpretierbarkeit des Modells aufgrund der Parameterlinearität sehr hoch. Komplexe nichtlineare Zusammenhänge können von Polynomansätzen nur bedingt abgebildet werden. Hinzu kommt oszillierendes Verhalten bei Polynomen hoher Grade, was ein unerwartetes Verhalten des Modells in Extrapolationsbereichen verursacht.

[2] Für die datengetriebene Suche eines geeigneten Parametervektors wird durchgehend der aus der Statistik bekannte Begriff Parameterschätzung verwendet, obwohl bei einigen Modellfamilien statistische Voraussetzungen verletzt werden.

[3] Das Symbol der statistischen Schätzung verdeutlicht, dass die Modellstruktur sowie die freien Parameter auf Basis von Daten angepasst wurden.

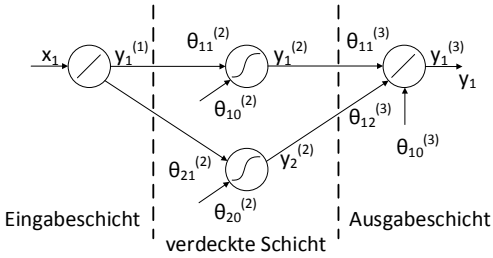

Abbildung 1.2: Aufbau eines vorwärtsgerichteten MLP-Netzes mit einer Eingangsgröße und zwei Neuronen in der verdeckten Schicht

Künstliche neuronale Netze (engl. *artificial neural networks*, ANN) stellen eine Modellfamilie dar, mit der beliebige nichtlineare Zusammenhänge abgebildet werden können. Bekannte Vertreter der ANNs sind Netze aus „mehrschichtigen Perzeptronen“ (engl. *multilayer perceptrons*, MLP) und die Radiale-Basisfunktionen-Netze (RBFN). Details zum Aufbau der verschiedenen Arten von ANNs finden sich in [9, 10].

Ein ANN besteht aus einzelnen Neuronen, die miteinander verknüpft sind. In vielen Netzen gehört jedes Neuron zu einer Schicht. Im Fall eines vorwärtsgerichteten MLP-Netzes sind die Eingänge eines Neurons einer Schicht nur mit den Neuronen der vorhergehenden Schicht verbunden und der Ausgang eines Neurons mit Neuronen der nachfolgenden Schicht.

Abbildung 1.2 zeigt den Aufbau eines solchen Netzes. Die Eingabeschicht (Schicht $s = 1$) dient der Aufnahme der Eingangsgrößen in das Netz, die Ausgänge der entsprechenden Neuronen berechnen sich meist mit einer linearen Aktivierungsfunktion. Die Ausgabeschicht (Schicht $s = S$) liefert den Ausgang des Netzes. Die Schichten dazwischen dienen der Steigerung der Komplexität und werden verdeckte Schichten genannt (Schicht $s = 2, \ldots, S-1$). Die Neuronen der verdeckten Schicht und der Ausgabeschicht verfügen über einen inneren Zustand z, der sich in MLP-

Netzen durch die gewichtete Summe der Ausgänge der Vorgängerschicht und einem Absolutterm berechnet[4]:

$$z_i^{(s)} = \theta_{\mathrm{i0}}^{(s)} + \sum_j \theta_{\mathrm{ij}}^{(s)} y_j^{(s-1)} \tag{1.7}$$

Der Ausgang der Neuronen der verdeckten Schichten und der Ausgabeschicht ergibt sich aus der Aktivierungsfunktion $f_i^{(s)}$ für den Zustand $z_i^{(s)}$ zu

$$y_i^{(s)} = f_i^{(s)}(z_i^{(s)}) \tag{1.8}$$

In den verdeckten Schichten werden meist sigmoidale, in der Ausgabeschicht lineare Aktivierungsfunktionen verwendet. Für das Netz aus Abbildung 1.2 ergibt sich damit der Zusammenhang:

$$\begin{aligned} f(\boldsymbol{x}, \boldsymbol{\theta}) = y_1^{(3)} &= f_1^{(3)}(z_1^{(3)}) = \theta_{11}^{(3)} y_1^{(2)} + \theta_{10}^{(3)} + \theta_{12}^{(3)} y_2^{(2)} \\ &= \theta_{11}^{(3)} \left(\frac{2}{1 + \exp\left(-2(\theta_{11}^{(2)} x_1 + \theta_{10}^{(3)})\right)} - 1 \right) + \theta_{10}^{(3)} \\ &+ \theta_{12}^{(3)} \left(\frac{2}{1 + \exp\left(-2(\theta_{21}^{(2)} x_1 + \theta_{20}^{(2)})\right)} - 1 \right) \end{aligned} \tag{1.9}$$

Die Größe der Eingabeschicht bestimmt sich aus der Merkmalsselektion der Hypothesengenerierung. Die Anzahl der Neuronen in der verdeckten Schicht und die Wahl der Aktivierungsfunktionen sind weitere vom Anwender zu wählende Metaparameter der Hypothesengenerierung. Die Gewichte $\theta_{\mathrm{ij}}^{(s)}$ stellen freie Parameter dar, die in der Parameterschätzung bestimmt werden. Sie wirken sich nichtlinear auf das Modell aus, was die Interpretationsfähigkeit mindert. Ansätze zur Steigerung der Interpretierbarkeit von ANNs finden sich in [11, 12]. Die Modellstruktur (1.9) zeigt weiterhin, dass ANNs bereits bei wenigen Neuronen in der verdeckten Schicht sehr komplex werden, was zu unvorhergesehenem Verhalten in Extrapolationsbereichen führen kann.

[4] Es handelt sich dabei um keinen Zustand im Sinne der Regelungstechnik. Der Begriff wird zur Sicherung der Konsistenz mit biologischen und künstlichen Neuronen mit interner Dynamik verwendet. Vorwärtsgerichtete MLPs bilden statische Systeme ab.

Sogenannte Extreme-Lern-Maschinen (engl. *extreme learning machines*, ELM) sind vorwärtsgerichtete MLP-Netze mit nur einer verdeckten Schicht. Bei ELMs werden die Gewichte zwischen Eingabeschicht und der verdeckten Schicht randomisiert gewählt. Lediglich die Gewichte zwischen der verdeckten Schicht und der Ausgabeschicht werden als freie Parameter in der Parameterschätzung an die Daten angepasst [13, 14].

Um dynamische Systeme abzubilden, können rekurrente Netze eingesetzt werden [15,16]. Bei rekurrenten Netzen können die Ausgänge von Neuronen einer Schicht auch mit den Eingängen von Neuronen der gleichen oder einer vorangegangenen Schicht verbunden sein.

Ein aktueller Trend für Zeitreihen und in der Bildverarbeitung ist das Verwenden von neuronalen Netzen mit vielen verdeckten Schichten, sogenannte tiefe neuronale Netze (engl. *deep neural networks*, DNN) [17–20]. Die Überlegenheit von DNNs gegenüber gewöhnlichen ANNs wurde für Klassifikationsprobleme gezeigt, für Regressionsprobleme mit vielen Eingangsgrößen sind DNNs häufig zu komplex. Hierarchische tiefe neuronale Netze (engl. *hierarchical deep neural networks*, HDNN) versuchen den Nachteil zu kompensieren [21]. Aufgrund der Aktualität solcher Entwicklungen mangelt es allerdings bisher an ausführlichen Untersuchungen. Weiterhin existieren für die Bildverarbeitung und Spracherkennung sogenannte faltende neuronale Netze (engl. *convolutional neural networks*, CNN) [22, 23].

Einen Sonderfall der MLP-Netze stellen die RBF-Netze dar. Sie haben nur eine verdeckte Schicht und die Berechnung des inneren Zustands der entsprechenden Neuronen erfolgt mit Hilfe von radialen Basisfunktionen[5]. Es werden lineare Aktivierungsfunktionen verwendet. Dadurch wirken sich die Neuronen lokal auf die resultierende Funktion f aus. RBF-Netze entsprechen damit einer Linearkombination von N_{RBF} Basisfunktionen

$$\phi_i(\boldsymbol{x}, \boldsymbol{c}_i, \sigma_{\mathrm{RBF,i}}) = \exp\left(-\frac{1}{2}\frac{||\boldsymbol{x} - \boldsymbol{c}_i||^2}{\sigma^2_{\mathrm{RBF,i}}}\right), i = 1, \ldots, N_{\mathrm{RBF}}. \tag{1.10}$$

[5]Eine radiale Basisfunktion ist eine Funktion, deren Wert ausschließlich vom zu definierenden Abstand von einem Vektor abhängt

Die Wahl der Anzahl N_{RBF} und die Lage der Zentren $\boldsymbol{c}_i$ sowie die Festlegung der Streuungen $\sigma_{\mathrm{RBF,i}}$ der Basisfunktionen können Bestandteil der Hypothesengenerierung sein. Dadurch ergibt sich die Modellstruktur

$$f(\boldsymbol{x}, \boldsymbol{\theta}) = \sum_{\mathrm{i}=1}^{N_{\mathrm{RBF}}} \theta_i \phi_i(\boldsymbol{x}, \boldsymbol{c}_i, \sigma_{\mathrm{RBF,i}}). \tag{1.11}$$

Werden die Streuungen oder die Lage der Zentren als freie Parameter in den Entwurf des Modells integriert, um den Anwender in der Hypothesengenerierung zu entlasten, wird die Parameterschätzung komplexer.

Die ursprünglich für Klassifikationsaufgaben entwickelten Stützvektor-Maschinen (engl. *support vector machines*, SVM) können auch für kontinuierliche Zielgrößen durch sogenannte Stützvektor-Regressionen (engl. *support vector regression*, SVR) eingesetzt werden. Das resultierende Modell der SVR hat die Form

$$f(\boldsymbol{x}) = \sum_{\mathrm{n}=1}^{N} \theta_n k(\boldsymbol{x}, \boldsymbol{x}_n) + b \tag{1.12}$$

und alle $\boldsymbol{x}_n | \theta_n \neq 0$ werden Stützvektoren genannt. Die Wahl und die Parametrierung der Kernelfunktion $k(\cdot)$ bestimmen maßgeblich die Komplexität des Modells. Die Kernelfunktion berechnet das Skalarprodukt von zwei Vektoren in einem transformierten, nicht explizit bekannten Kernelraum. Durch nichtlineare Kernelfunktionen lassen sich komplexe Zusammenhänge modellieren. Im Falle eines gaußschen Kernels entspricht das Modell einem RBF-Netz mit einer einheitlichen Streuung für alle Basisfunktionen. Erweiterungen der SVR behandeln die Integration von Vorwissen [24], die implizite Berechnung von Metaparametern in der Parameterschätzung [25], inkrementelle Parameterschätzungen [26] sowie die Verwendung multipler Kernel [27].

Anstatt nichtlineare Basisfunktionen zu überlagern, kann für Regressionsprobleme auch das Prinzip „teile-und-herrsche“ verwendet werden. Ziel ist dabei, den Eingangsraum aufzuteilen und lokal einfache, interpretierbare Modelle zu nutzen. Dabei kann es sich um konstante Werte handeln, um lineare Modelle [28] oder auch um nichtlineare Polynommodelle [29]. Konstante Werte als lokale Modelle werden häufig in Regressionsbäumen [30,31] verwendet. Besonderes Kennzeichen von Regressionsbäumen ist die scharfe

Trennung zwischen den lokalen Modellen. In der Regelungstechnik haben sich vor allem lokale Modellnetze (LMN) etabliert. In lokalen Modellnetzen wird der Eingangsraum durch Clusteranalysen [32] oder iterativ gemäß eines Gütekriteriums partitioniert und der Zusammenhang lokal z.B. linear modelliert. Um eine global stetige und differenzierbare Funktion zu erhalten, können die lokalen Modelle z.B. mit Hilfe normalisierter radialer Basisfunktionen gewichtet und überlagert werden [33, 34]. Es ergibt sich beispielhaft die Modellstruktur:

$$f(\boldsymbol{x}) = \sum_{i=1}^{N_{\mathrm{LM}}} \left(\frac{\phi_i(||\boldsymbol{x} - \boldsymbol{c}_i||)}{\sum_{j=1}^{N_{\mathrm{LM}}} \phi_j(||\boldsymbol{x} - \boldsymbol{c}_j||)} \cdot \left(\sum_{l=1}^{p} \theta_{i,l} x_l + \theta_{i,0} \right) \right) \tag{1.13}$$

mit N_{LM} lokalen linearen Modellen. Durch die Partitionierung und die Verwendung lokal einfacher Modelle bleibt das resultierende nichtlineare Modell zumindest lokal interpretierbar. Außerdem können bei geeigneten Partitionierungen die Modelle in unscharfe Systeme (engl. *fuzzy systems*, FS) [35] überführt werden. Es ergeben sich Takagi-Sugeno-Systeme [36, 37]. Deshalb finden lokale Modellnetze breite Anwendung in der Kennfeldapproximation und der nichtlinearen Systemidentifikation [38, 39] als Verfahren der *Computational Intelligence* [40, 41]. Sie wurden allerdings auch beispielsweise zur Zeitreihenprädiktion verwendet [42]. Die Partitionierung kann auch in einem Zustandsraum erfolgen, der durch Transformation des Eingangsraums erreicht wird. Praktische Anwendung findet die Trennung von Eingangsgrößen des Modells und Merkmale des Zustandsraums in der Systemidentifikation [43, 44]. Die Güte eines lokalen Modellnetzes hängt maßgeblich vom gewählten Partitionierungsverfahren und der verwendeten Heuristiken ab. Zum Beispiel kann die Partitionierung orthogonal zu den Achsen des Zustandsraums mit Hilfe des Verfahrens für lokal lineare Modellbäume (engl. *local linear model tree*, LOLIMOT) [38] oder des Verfahrens für multivariate adaptive Regressions-Splines (engl. *multivariate adaptive regression splines*, MARS) erfolgen. Alternative kann die Partitionierung auch schräg zu den Achsen [45] stattfinden.

Komitee-Maschinen (engl. *Committee Machines*) [46] stellen einen weiteren Mehrmodellansatz [47,48] dar. Hierbei werden verschiedene Modelle erstellt, um die gleiche Zielgröße zu prädizieren. Die Prädiktionen der Modelle werden anschließend mit Hilfe des Mittelwerts oder eines gewichteten Mittels aggregiert. Die Bildung der Modelle kann beispielsweise auf einem Bootstrap-Verfahren (*Bagging Predictors* [49]) beruhen.

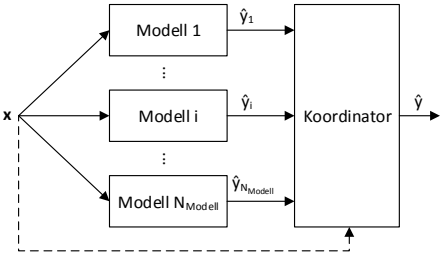

Abbildung 1.3: Schematische Anwendung von Mehrmodellansätzen

Alle Mehrmodellansätze haben eine ähnliche Struktur, deren Anwendung in Abbildung 1.3 dargestellt ist. Mehrere Modelle $f_i(\boldsymbol{x}), i = 1, \ldots, N_{\text{Modelle}}$ werden angewendet und ein Koordinator wichtet die Prädiktionen der Teilmodelle zu einer Gesamtprädiktion. Unterschiede bestehen in Anzahl und Struktur der Teilmodelle sowie in der Art des Koordinators. Weiterhin unterscheiden sich Mehrmodellansätze im Entwurfsverfahren.

Beim Verstärken (engl. *boosting*) [50, 51] bildet das erste Teilmodell den gesamten Zusammenhang ab. Alle weiteren Teilmodelle bilden die Fehlerterme der Superposition aller vorangegangenen Modelle ab, d.h. sie verstärken die vorangegangenen Modelle in ihrer Prädiktionsgüte. Der Koordinator addiert alle Teilmodelle. Solche Mehrmodellansätze werden auch Ensemblemethoden (engl. *Ensemble Methods*) genannt.

Die vorgestellten Verfahren bilden in der Regel stetig differenzierbare Funktionen, die für einen beliebigen Punkt $\boldsymbol{x}$ im Eingangsraum einen Wert für die Zielgröße liefern. Für punktuelle Aussagen über die Zielgröße gibt es einige weitere Verfahren, die lokal bzw. punktweise gültige Funktionen erstellen, aber kein globales Modell. Solche Verfahren werden den Bereichen *instance based learning* [52], *memory based learning* [53, 54] oder *lazy learning* [55] zugeordnet.

Ein Beispiel ist die k-Nächste-Nachbarn-Regression (k-NN-Regression), bei der die Zielgröße an einem Punkt im Eingangsraum auf Basis eines definierten Distanzmaßes vorhergesagt wird. Im Eingangsraum werden dazu die k nächsten Nachbarn der vorhandenen Daten bestimmt und

ihre zugeordneten Werte der Zielgröße gemittelt oder distanzgewichtet aggregiert. Die Parameterschätzung entfällt, da lediglich alle Datentupel gespeichert werden müssen. Die Anwendung des Modells ist zeitaufwändig, da im Gesamtdatensatz die nächsten Nachbarn gesucht werden. Die Anzahl der zu berücksichtigenden Nachbarn und ein geeignetes Distanzmaß sind in der Hypothesengenerierung zu wählen.

Lokale Regressionen basieren auf einem k-NN-Verfahren. Zunächst werden die k nächsten Nachbarn im Eingangsraum für den betrachteten Punkt gesucht. Anschließend wird eine Parameterschätzung für einen lokalen Polynomansatz durchgeführt.

Die Anwendung der vorgestellten Modelle liefern sogenannte Punktprädiktionen. Ziel einer Punktprädiktion ist für einen gegebenen Eingangsvektor den wahrscheinlichsten Wert der Zielgröße zu finden. Mit Hilfe von Prädiktionsintervallen können Unsicherheiten im Modellentwurf und der Rauschterm der gemessenen Zielgröße abgebildet werden. Prädiktionsintervalle liefern für einen gegebenen Eingangsvektor den Wertebereich, in dem sich die Zielgröße mit einer bestimmten Wahrscheinlichkeit befindet. Zur Berechnung von Prädiktionsintervallen existieren abhängig von der Modellfamilie verschiedene Ansätze [56, 57]. Die Gaußsche-Prozess-Regression (engl. *gaussian processes regression*, GPR) [58] impliziert die Berechnung des Prädiktionsintervalls bereits. Sie ist allerdings bei vielen Datentupeln und Eingangsgrößen sehr rechen- und speicherintensiv.

Abbildung 1.4 zeigt die Modellverläufe verschiedener Modellfamilien und die entsprechenden Lerndaten.

Um Modelle wie in Abbildung 1.4 zu erhalten, müssen die freien Parameter der jeweiligen Modellstruktur mit Hilfe der zur Verfügung stehenden Daten geschätzt werden.

Dazu kann die Methode der kleinsten Fehlerquadrate verwendet werden, die sich als folgendes Optimierungsproblem formuliert:

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \frac{1}{2} \sum_{\mathrm{n}=1}^{N} (y_n - f(\boldsymbol{x}_n, \boldsymbol{\theta}))^2. \tag{1.14}$$

Die Abweichung $(y_n - f(\boldsymbol{x}_n, \boldsymbol{\theta}))$ an der Stelle $\boldsymbol{x}_n$ wird Residuum r_n genannt und die Lossfunktion $\rho(r)$ beschreibt den Einfluss eines Residuums r auf die Parameterschätzung.

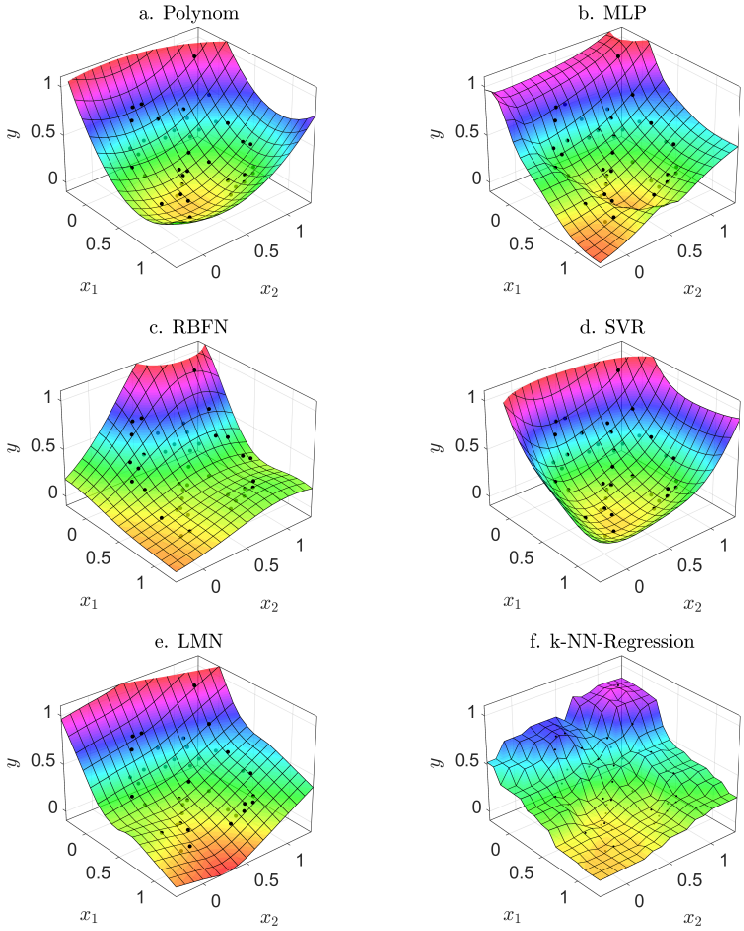

Abbildung 1.4: Modellverlauf a: eines Polynoms mit Grad 2. b: eines MLP-Netzes mit 4 Neuronen in der verdeckten Schicht. c: eines RBF-Netzes mit 4 Neuronen in der verdeckten Schicht. d: einer SVR mit $\sigma = 0.7$. e: eines achsenorthogonal partitionierten LMNs. f: einer k-NN-Regression mit $k = 2$. Die Helligkeit der Lerndaten zeigt an, ob sich die Lerndaten visuell vor oder hinter dem Modellverlauf befinden.

Für parameterlineare Ansätze wie in der Modellstruktur (1.3) liegt in der Parameterschätzung mit Hilfe der Methode der kleinsten Fehlerquadrate ein quadratisches Problem vor, das analytisch gelöst werden kann. Bei ANNs ist das Optimierungsproblem der Parameterschätzung aufgrund der nichtlinearen Einflüsse der freien Parameter nicht konvex und enthält lokale Minima. Die Parameterschätzung erfolgt u.a. mit Hilfe von Backpropagationverfahren oder des Levenberg-Marquardt-Algorithmus [59], das Ergebnis hängt aber auch von zufällig initialisierten Startwerten ab. Bei SVRs liegt bei der Parameterschätzung ein quadratisches Optimierungsproblem mit Randbedingungen vor, für das aufgrund seiner besonderen Struktur Verfahren wie die sequentielle minimale Optimierung (engl. *sequential minimal optimization*, SMO) [60] entwickelt wurden, um die Optimierung zu beschleunigen. Bei lokalen Modellnetzen ergeben sich analytisch lösbare Optimierungsprobleme für die lokalen Teilmodelle. Eine geeignete Partitionierung des Eingangs- oder Zustandsraums stellt jedoch ein nichtlineares Optimierungsproblem dar, das häufig heuristisch gelöst wird.

1.2.2 Vorwissen

In der Modellbildung existiert häufig Vorwissen. Das kann Wissen über die abzubildenden Zusammenhänge oder über die Qualität der Daten sein, die in vorverarbeitenden Schritten untersucht wurde. Vorwissen über Eigenschaften und den Verlauf der gesuchten Funktion kann mit Hilfe verschiedener Formulierungszugänge in die Modellbildung integriert werden [4], z.B. als:

- impliziter Zugang über Strukturansätze,
- expliziter Zugang über Restriktionen und
- Kompromisszugang über Strafterme.

In [4] dient als Beispiel das Vorwissen über die Nichtnegativität einer Funktion $f : \mathcal{D} \to \mathbb{R}$ über ihrer Definitionsmenge $\mathcal{D}$. Der *explizite Zugang über Restriktionen* $\forall x \in \mathcal{D} : f(\boldsymbol{x}; \boldsymbol{\theta}) \geq 0$ oder relaxiert durch eine endliche Anzahl an Restriktionen $f(\boldsymbol{x}_{\mathrm{R},i}) \geq 0; i = 1, \ldots, N_R$ bietet den Vorteil, dass die zu optimierende Funktion der Parameterschätzung unverändert bleibt. Es muss jedoch ein restringiertes Optimierungsproblem gelöst werden.

Beim *impliziten Zugang über einen Strukturansatz* $f(\boldsymbol{x};\boldsymbol{\theta}) = [g(\boldsymbol{x};\boldsymbol{\theta})]^2$, werden zwar keine Restriktionen benötigt, um die Nichtnegativität zu sichern, es kommt jedoch häufig zum Konvexitätsverlust des Optimierungsproblems. So gestaltet sich z.B. für $g(\boldsymbol{x};\boldsymbol{\theta}) = \theta_0 + \theta_1 x_2 + \theta_2 x_2$ das Problem zur Minimierung der Fehlerquadrate als Minimierung der Gütefunktion $Q(\boldsymbol{\theta}) = \sum_{\mathrm{n}=1}^{N} (y_n - (\theta_0 + \theta_1 x_{n,1} + \theta_2 x_{n,2})^2)^2$.

Als *Kompromisszugang* bietet die Gütefunktion

$$Q(\boldsymbol{\theta}) = \sum_{n=1}^{N} (y_n - f(x_n;\boldsymbol{\theta}))^2 + \sum_{i=1}^{N_\mathrm{R}} \gamma_i (\min\{0, f(x_{\mathrm{R},i};\boldsymbol{\theta})\})^2 \qquad (1.15)$$

die Möglichkeit, das Vorwissen ohne eine zwingende Einhaltung einzubeziehen. Den Grad der Einhaltung bestimmen hier die Wichtungsfaktoren γ_i. Wie am Beispiel der Nichtnegativität gesehen, können sich Eigenschaften auf Bereiche des Eingangsraums beziehen, lassen sich aber bei geeigneter Wahl der Stützstellen auch durch eine endliche Anzahl punktweiser Anforderungen sicherstellen. Die Anzahl, respektive die Dichte der Stützstellen, muss abhängig von der Funktionsstruktur ausreichend groß gewählt werden. Es bietet sich außerdem an, die Stützstellen im betrachteten Bereich beispielsweise äquidistant zu verteilen.

Das resultierende Optimierungsproblem muss anschließend gelöst, bzw. zuvor vereinfacht werden. Dazu können beispielsweise Reduktions- oder Erweiterungsmethoden verwendet werden. Mit Hilfe von Vorwissen und den vorgestellten Formulierungszugängen ist es beispielsweise möglich, die Modellqualität in Bereichen geringer Datendichte zu verbessern.

1.2.3 Modellqualität

Modellfamilien und entsprechende Verfahren zur Parameterschätzung unterscheiden sich unter anderem in der Interpretierbarkeit, dem Aufwand zur Strukturwahl und dem Rechenaufwand. Solche Faktoren dienen dem Ausschluss oder der Bevorzugung von Modellfamilien in der Hypothesengenerierung. Eine quantifizierbare Modellqualität, die in der Bewertung und Auswahl der verwendeten Modellfamilien und Modellstrukturen verwendet werden kann, ist die Prädiktionsfähigkeit. Sie beschreibt die Genauigkeit der Vorhersagen von Modellen. Grundvoraussetzung geeigneter Modelle

ist die Berücksichtigung aller relevanten Eingangsgrößen und die Erfüllung der Bedingung, dass die abzubildenden Zusammenhänge statisch, d.h. zeitinvariant, sind. Bei Nichterfüllung einer der Bedingungen kann das Modell zum Zeitpunkt seiner Anwendung deutlich vom realen Zusammenhang abweichen. Die Parameterschätzung für eine Modellstruktur wird mit Hilfe eines Lerndatensatzes $(\boldsymbol{X}_{\mathrm{Lern}}, \boldsymbol{y}_{\mathrm{Lern}})$ durchgeführt. Anschließend wird das Regressionsmodell für die Eingangsgrößen eines Validierungsdatensatzes angewandt. In der Bewertung werden die Prädiktionen $\hat{\boldsymbol{y}}_{\mathrm{Val}}$ mit den tatsächlichen Werten der Zielgröße $\boldsymbol{y}_{\mathrm{Val}}$ verglichen und ein Bewertungsergebnis Q berechnet. Die Modellanwendung erfolgt nicht auf den Lerndaten, um Überanpassung zu vermeiden. Überanpassung ist eine sehr genaue Abbildung der Lerndaten und gegebenenfalls sogar des Messrauschens oder anderer zufälliger Effekte, was zu einer verminderten Prädiktionsfähigkeit auf anderen Daten führt.

Ist der Datensatz klein, kann z.B. eine Kreuzvalidierung (engl. *crossvalidation*, CV) durchgeführt werden. Dazu wird der Lerndatensatz randomisiert in N_{CV} Teildatensätze aufgeteilt. Anschließend erfolgt die Parameterschätzung auf $N_{\mathrm{CV}} - 1$ der Teildatensätze, während der übrige Teildatensatz zur Validierung genutzt wird. Für die Kreuzvalidierung wird jeder der Teildatensätze einmal der Parameterschätzung vorenthalten und zur Validierung verwendet. Die mittlere Güte über alle Validierungsschritte bewertet die Modellstruktur.

Zum Vergleich verschiedener Modellfamilien wird ein Testdatensatz benötigt. Während innerhalb der Modellfamilien die beste Struktur mit Hilfe von Kreuzvalidierungen oder der Anwendung auf Validierungsdaten gewählt werden kann, können mit Hilfe der bis dato nicht verwendeten Testdaten die besten Modellstrukturen der jeweiligen Modellfamilien verglichen werden. Grundannahme ist, dass die Teildatensätze der gleichen Wahrscheinlichkeitsverteilung entstammen [8].

Zur Berechnung der Güte werden quantitative Bewertungskriterien benötigt. Beispiele sind der mittlere absolute Fehler (*mean absolut error*, MAE)

$$MAE = \frac{1}{N} \sum_{n=1}^{N} |y_i - \hat{f}(\boldsymbol{x}_i)| \tag{1.16}$$

die Summe der Fehlerquadrate (*residual sum of squares*, RSS)

$$RSS = \sum_{n=1}^{N} (y_n - \hat{f}(\boldsymbol{x}_n))^2 \tag{1.17}$$

oder das Bestimmtheitsmaß

$$R^2 = 1 - \frac{\sum_{n=1}^{N} (y_n - \hat{f}(\boldsymbol{x}_n))^2}{\sum_{n=1}^{N} (y_n - \overline{y})^2}. \tag{1.18}$$

Neben den Bewertungsmaßen, die Abweichungen des Modells von Testdaten bewerten, existieren sogenannte informationstheoretische Maße, um z.B. die Komplexität einer Modellstruktur in Abhängigkeit der Größe des Lerndatensatzes und der Abweichungen von den Lerndaten zu bewerten. Damit werden Modellstrukturen vermieden, die zu Überanpassung neigen.

Beispiele für informationstheoretische Maße sind das Akaike Informationskriterium (engl. *Akaike information criterion*, AIC) [61], das Bayessche Informationskriterium (engl. *Bayesian information criterion*, BIC) oder Mallows's C_p [62].

Mit Hilfe statistischer Tests, z.B. dem T-Test, können bei Modellen gemäß (1.3) freie Parameter bzw. ihre zugehörigen Regressoren auf ihre Signifikanz und damit auf ihre Relevanz getestet werden [63].

Eine detailliertere Untersuchung der Modellgüte kann durch die sogenannte Residualanalyse stattfinden. Sie dient der Bestätigung der angenommenen Modellstruktur, der Identifikation von Ausreißern sowie der Überprüfung weiterer getroffener Annahmen.

Abbildung 1.5 zeigt ein lineares Modell und die Darstellung der Residuen über der ersten Eingangsgröße.

Ein gutes Modell hat im Mittel ein Residuum nahe 0 und eine Streuung der Residuen, die von Eingangs- und Zielgröße unabhängig ist. Für das lineare Modell scheint ein funktionaler Zusammenhang zwischen der Eingangsgröße und dem Residuum zu bestehen. Das lässt auf eine ungeeignete Modellstruktur schließen.

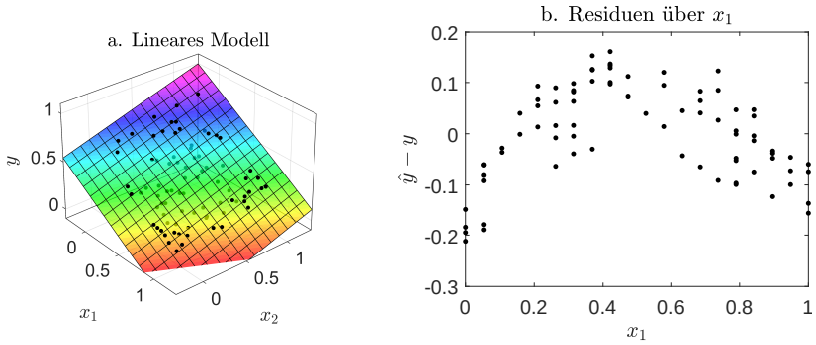

Abbildung 1.5: a. Verlauf eines linearen Modells und b. Verlauf der Residuen in Abhängigkeit einer Eingangsgröße

Ziel der vorgestellten Bewertungsmaße (1.16) - (1.18) ist die Abschätzung einer globalen Güte des Modells. Häufig ist es aber wichtig die Zuverlässigkeit eines Modells an einem bestimmten Punkt der Prädiktion anzugeben. Dazu können sogenannte Prädiktionsintervalle berechnet werden [56, 57].

Die Zuverlässigkeit eines Modells kann auch in Abhängigkeit von der Datendichte im Eingangsraum beschrieben werden. Die Prädiktionsgüte ist in Bereichen des Eingangsraums mit hoher Datendichte oft höher, als in Bereichen, in denen kaum Daten vorhanden sind [64, 65].

Unter Umständen können damit auch nichtbeachtete Zeitvarianzen erkannt werden, wenn sie sich auf die Ausprägungen der berücksichtigten Eingangsgrößen auswirken.

Solche Abschätzungen werden in [66] für künstliche neuronale Netze mit Hilfe der Parzen-Fenster-Methode [67] untersucht. Es zeigt sich, dass der Modellfehler mit abnehmender Datendichte tendenziell zunimmt. Kritische Aufgabe für den Anwender ist es, einen geeigneten Schwellwert der Datendichte zu finden, um den Übergang von hoher zu niedriger Modellgüte in der Anwendung des betrachteten Modells zu erkennen. In [68] werden zur Abbildung der Datendichte ANNs verwendet, wodurch die Zuverlässigkeit eines anderen ANNs gleicher Struktur abgeschätzt wird. Ein Schwellwert wird empirisch festgelegt. In [69, 70] wird das VI-Netz (engl. *validity in-*

dex net) vorgestellt, ein RBF-Netz, das seine eigene Vertrauenswürdigkeit berechnet.

Dichteschätzungen, aber auch die Berechnung der Vertrauenswürdigkeit im VI-Netz, ähneln den Problemstellungen Ausreißererkennung (engl. *outlier detection*), Anomalieerkennung (engl. *anomaly detection*), Neuheitserkennung (engl. *novelty detection*), oder Fehlererkennung (engl. *fault detection*), die sich in den verwendeten Methoden überschneiden und sich in ihrer Motivation häufig nur geringfügig unterscheiden. Es wird bestimmt, ob ein beliebiger Eingangsvektor in einem Bereich liegt, in dem der Modellbildung Lerndaten zur Verfügung standen oder nicht.

Das Problem kann auch als sogenannte Ein-Klassen-Klassifikation [71–73] verstanden werden. [74] gibt einen Überblick über aktuelle Verfahren. Neben der Erweiterung für Regressionsprobleme in [75] wird in [76] beispielsweise die Erweiterung der SVMs auf Ein-Klassen-Probleme vorgestellt, sogenannte Ein-Klassen-Stützvektor-Maschinen (engl. *one-class support vector machine*, 1K-SVM). In der Parameterschätzung der 1K-SVM werden möglichst alle Lerndaten im Merkmalsraum mit einer Hyperebene mit maximalem Abstand vom Ursprung getrennt. Einzelne Fehlklassifikationen werden zu Gunsten einer geringeren Komplexität der Hyperebene erlaubt. Fehlklassifikationen sind Vektoren der Lerndaten, die nicht durch die Hyperebene vom Ursprung getrennt werden. Der Anteil der bei der Parameterschätzung erlaubten Fehlklassifikationen wird durch den vom Anwender zu wählenden Metaparameter ν bestimmt[6].

Zur Anwendung der 1K-SVM wird die binäre Entscheidungsfunktion

$$g_{\text{1K-SVM}}(\boldsymbol{x}) = \operatorname{sgn}\left(\sum_{n=1}^{N} \theta_n k(\boldsymbol{x}, \boldsymbol{x}_n) - b\right), \theta_n \geq 0 \tag{1.19}$$

verwendet. Details zur Parameterschätzung finden sich in [76]. Als Metaparameter ist bei Verwendung eines gaußschen Kernels unter anderem die Streuung σ zu wählen, die sich maßgeblich auf die Empfindlichkeit der 1K-SVM auswirkt. Bei kleinem σ wird eine Extrapolation bereits erkannt,

[6] und wird möglichst klein gewählt, damit die Umgebungen aller Lerndaten auch tatsächlich als Interpolationsbereiche erkannt werden. Beispiel: $\nu = 0.001$

wenn sich der betrachtete Eingangsvektor nur geringfügig von den Lerndaten unterscheidet. Abbildung 1.6 zeigt die Klassengrenzen von 1K-SVMs mit unterschiedlicher Metaparametrierung und ihre Lerndaten.

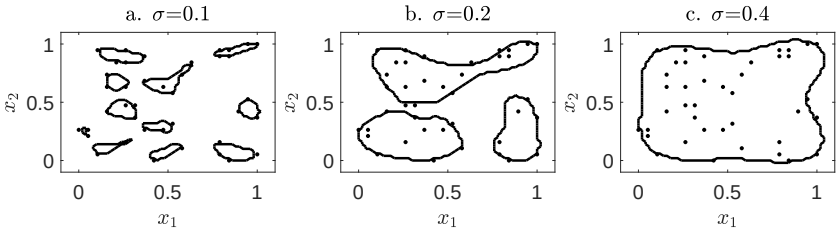

Abbildung 1.6: Beispiele für 1K-SVMs mit unterschiedlicher Metaparametrierung. Für die eingeschlossenen Bereiche gilt $g_{\text{1K-SVM}}(\boldsymbol{x}) = 1$. Außerhalb der Bereiche gilt $g_{\text{1K-SVM}}(\boldsymbol{x}) = -1$.

1.2.4 Datenqualität

Die systematische Erfassung und Behandlung von Datenqualität hat sich vor allem in der Verwaltung betrieblicher Datenbanken etabliert. In Regressionsproblemen wird Datenqualität vorwiegend im Sinne der statistischen multilinearen Regressionen interpretiert und behandelt. Im Folgenden werden fehlende Werte, heterogene Datendichte, fehlerhafte Verteilungsannahmen, unterschiedliche Skalierungen, Ausreißer, Multikollinearität, Heteroskedastizität und Fehler in den Eingangsgrößen als Einschränkungen in der Datenqualität diskutiert:

Bei *fehlenden Werten* wird davon ausgegangen, dass in der Datenmatrix Elemente als fehlende Werte kenntlich gemacht sind [77,78]. Fehlende Werte entstehen z.B. in der Datenerfassung, -speicherung oder -übertragung.

Eine *heterogene Datendichte* liegt vor, wenn Bereiche des Eingangsraums nicht oder nur spärlich von Daten erfasst sind. Solche nicht erfassten Bereiche resultieren aus einer nicht durchgeführten oder nicht durchführbaren statistischen Versuchsplanung. Die Versuchsplanung legt fest, für welche Werte der Eingangsgrößen und in welcher Reihenfolge Daten erfasst werden. Sie stellt sicher, dass ein Datensatz vollständig ist und Modelle mit

möglichst wenig Daten, d.h. bei minimalem Aufwand für die Datenerhebung, entworfen werden können [79]. Wenn Daten beispielsweise durch die passive Beobachtung eines Systems erfasst werden, kann eine Versuchsplanung häufig nicht realisiert werden. Dadurch werden Bereiche, die bestimmte Betriebspunkte darstellen, im Datensatz überrepräsentiert, während andere Betriebspunkte oder Übergangsverhalten nur mit einer geringen Datendichte oder gar nicht erfasst werden. Eine heterogene Datendichte resultiert auch aus dem Fluch der Dimensionalität (engl. *curse of dimensionality*, COD). Er äußert sich in zwei Phänomenen: Die benötigte Anzahl an Datentupeln, um in höherdimensionalen Räumen eine ähnliche Datendichte wie im eindimensionalen Fall zu erreichen, steigt exponentiell mit der Anzahl der Eingangsgrößen, wodurch entsprechend hochdimensionale Datensätze zwangsläufig eine geringe Datendichte aufweisen (engl. *sparsity*). Außerdem wird die Aussagekraft der von vielen Verfahren verwendeten euklidischen Distanz vermindert, da sich in hochdimensionalen Räumen die euklidische Distanz zum nächsten Nachbarn der Distanz zum entferntesten Nachbarn annähert. In der Modellbildung können individuelle Wichtungen der Daten die negativen Auswirkungen der heterogenen Datendichte mindern. Die Datendichte lässt sich neben den bereits in Abschnitt 1.2.3 vorgestellten Verfahren beispielsweise mit Hilfe von Kerndichteschätzern [80], dem lokalen Ausreißerfaktor (engl. *local outlier factor*, LOF) [81] oder k-NN-basierenden Verfahren [46] bestimmen.

Bei *fehlerhaften Verteilungsannahmen* wird von Verteilungen in den Eingangsgrößen ausgegangen, denen der Datensatz nicht gerecht wird. Das kann zu unerwartetem Versagen von Verfahren führen, die von entsprechenden Verteilungsannahmen ausgehen. Außerdem können die Eingangsgrößen unterschiedlich skaliert sein, was zu Problemen mit Distanzmaßen führt.

Ausreißer sind Datentupel, die sich deutlich von den übrigen Datentupeln unterscheiden. „Vertikale“ Ausreißer sind Datentupel, die sich hinsichtlich der Zielgröße deutlich von den Datentupeln in ihrer Nähe des Eingangsraums unterscheiden. Sie können in der Residualanalyse erkannt werden. Ausreißer hinsichtlich der Eingangsgrößen werden auch Hebelwerte [82] genannt. Eine Übersicht über Verfahren zur Erkennung von Ausreißern findet sich in [83–86]. Auch wenn methodische Überschneidungen mit der Neuheitserkennung bestehen, unterscheiden sich die beiden Ansätze jedoch in ihrer Motivation: In der Neuheitserkennung wird angenommen, dass alle verfügbaren Daten zu einer Klasse gehören. Bei der Ausreißerdetektion

wird angenommen, dass der verfügbare Datensatz einige wenige Datentupel beinhaltet, die sich vom Großteil des Datensatzes unterscheiden.

Multikollinearität beschreibt die Korrelation zwischen Eingangsgrößen eines Modells. Multikollinearität hat vor allem negative Auswirkungen auf Polynomansätze, da die Varianzen der geschätzten Parameter sehr groß werden. Das bedeutet, dass kleine Änderungen im Datensatz große Änderungen in der Parameterschätzung verursachen können. Damit wird die Interpretation von Parametern erschwert. Bei einer perfekten Kollinearität von Eingangsgrößen wird die rechnerische Durchführung der linearen Regressionsanalyse sogar unmöglich.

Heteroskedastizität liegt vor, wenn der Störterm der Zielgröße über dem Eingangsraum nicht konstant ist. Eine Bewertung hinsichtlich Heteroskedastizität in einem linearen Regressionsmodell findet sich in [87, 88].

Eine weitere Einschränkung der Datenqualität sind mit einem *Störterm behaftete Eingangsgrößen*. Bei der Parameterschätzung gemäß (1.14) wird davon ausgegangen, dass die Eingangsgrößen in den Experimenten zur Datenerhebung frei einstellbar sind[7], während die Zielgröße u.U. mit einem Rauschterm überlagert gemessen wird. Daher wird das Residuum der Verfahren mit Hilfe der univariaten Distanz in Richtung der Zielgröße berechnet. Handelt es sich bei den Eingangsgrößen jedoch auch um gemessene Größen, sind die Ergebnisse der Parameterschätzung gemäß der Methode der kleinsten Fehlerquadrate nicht optimal.

Einige Phänomene können durch eine visuelle Inspektion der Daten erkannt werden. Für hochdimensionale Daten eignen sich dabei z.B. Streuwolkendiagramm-Matrizen (engl. *Scatterplot Matrices*). Um den visuellen Eindruck zu quantifizieren, wurden in [89] die sogenannten *Scagnostics* eingeführt und in [90, 91] detailliert beschrieben. Die quantifizierten Phänomene wurden allerdings unabhängig von Regressionsproblemen gewählt.

Ein Datensatz wird als inkonsistent bezeichnet, wenn sich die Zusammenhänge zwischen erfassten Eingangsgrößen und der Zielgröße lokal stark unterscheiden oder sich Daten unterschiedlichen Verlaufs sogar im Eingangsraum überlappen. Lösungsstrategien für solche Problemklassen finden sich in [92, 93]. In [94] wird eine Methode zur Konsistenzanalyse unterschiedlicher Datenquellen durch Regressionsmodelle anhand des Beispiels

[7] bzw. bei der passiven Beobachtung eines Systems fehlerfrei aufgezeichnet werden

von ANNs vorgestellt. Sie basiert im Wesentlichen auf einer Kreuzvalidierung, bei der die Teildatensätze auf Konsistenz geprüft werden und daher nicht zufällig gewählt werden. Kriterien für Inkonsistenz sind u.a. der mittlere Fehler und der Korrelationskoeffizient der jeweiligen Validierungsdaten. Während eine solche Konsistenzanalyse z.B. den Ausschluss von Teildatensätzen zur Folge hat, um ein präzises Modell auf einem nicht vollständigen Datensatz zu erhalten, existieren Mehrmodellansätze (engl. *mixture models*), um die Wahrscheinlichkeitsverteilung der Zielgröße unter Berücksichtigung aller Daten mit Hilfe mehrerer Modelle abzubilden [95–97]. Beim inkrementellen Lernen [98], wie es zur Behandlung von Datenströmen verwendet wird, sind a priori keine potentiell inkonsistenten Teildatensätze gegeben. Stattdessen ändern sich die Zusammenhänge über die Zeit. Für neue Datentupel kann a posteriori z.B. durch Benutzer-Rückmeldungen überprüft werden, ob sie mit den bisherig modellierten Zusammenhängen konsistent sind oder ein sogenannter Konzept-Drift (engl. *concept drift*) vorliegt [99]. Während konsistente Datentupel aus Mangel an neuen Informationen im Lernprozess vernachlässigt werden können, um Speicher und Rechenleistung zu sparen, werden neue Datentupel im Falle eines Konzept-Drifts in den inkrementellen Lernprozess integriert.

1.2.5 Kompensationsmethoden

Für die Einschränkungen der Datenqualität existieren Verfahren, um die negativen Auswirkungen auf die datengetriebene Modellbildung zu kompensieren.

Fehlende Werte können z.B. mit Hilfe von Imputationsverfahren ersetzt werden [78].

In [100] wird festgestellt, dass in Echt-Welt-Datensätzen hoher Dimensionalität global zwar eine geringe Datendichte vorliegt, die Daten aber lokal häufig niederdimensional und dicht verteilt sind. Es wurde deshalb ein Verfahren vorgestellt, was auf einer Partitionierung des Eingangsraums, lokaler Dimensionsreduktionen und lokal gewichteter Regression beruht.

Unterschiedliche Skalierungen können mit Hilfe von Transformationen und Normalisierungen kompensiert werden. Mit ihnen können die Verteilungen ihren Annahmen genähert und konsistent gehalten werden.

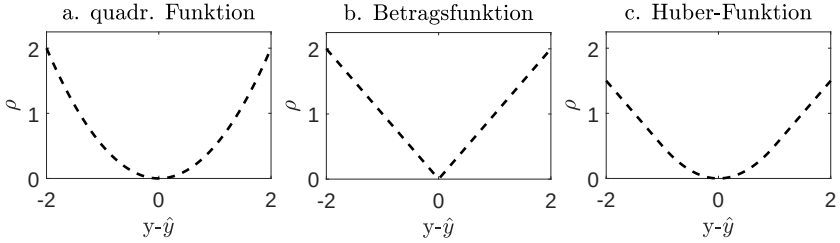

Abbildung 1.7: a: quadratische Lossfunktion b: Betrag als Lossfunktion c: Huber-Lossfunktion

Die *robuste Statistik* beschäftigt sich mit der Kompensation von Ausreißern. Der Entwurf entsprechender Regressionsmodelle heißt *robuste Regression* [101–105]. Es besteht z.B. die Möglichkeit die Lossfunktion $\rho(r)$ anzupassen, um beschränkte Abweichungen des Modells von den Daten zu tolerieren oder um den Einfluss von vertikalen Ausreißern zu mindern. Abbildung 1.7 zeigt Beispiele für Lossfunktionen. Die Methode der kleinsten Fehlerquadrate verwendet eine quadratische Lossfunktion. Mit einer Betragsfunktion oder der Huber-Lossfunktion wird der Einfluss von vertikalen Ausreißern gemindert. Die Wahl von ρ hat allerdings Einfluss auf die Eigenschaften des Optimierungsproblems der Parameterschätzung. So kann beispielsweise die Möglichkeit einer analytischen Lösung oder die Konvexität des Problems verloren gehen.

Ein Verfahren zur Kompensation von Multikollinearität sind Regularisierungen. Die Parameterschätzung mit Hilfe einer *Tikhonov*-Regularisierung wird Ridge-Regression (RR) [106] genannt und lässt sich wie folgt formulieren:

$$\hat{\boldsymbol{\theta}}_{\mathrm{RR}} = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \sum_{\mathrm{n}\,=\,1}^{N} \left(y_n - f(\boldsymbol{x}_n, \boldsymbol{\theta})\right)^2 + \lambda_{\mathrm{RR}} \sum_{\mathrm{j}\,=\,1}^{d} \theta_j^2 \tag{1.20}$$

Die Wahl eines geeigneten Regularisierungsparameters λ_{RR} kann beispielsweise mit Hilfe einer Kreuzvalidierung erfolgen. Bei ANNs wird das Anlernen unter Kollinearität zwar nicht unmöglich, die Netze neigen allerdings zu Überanpassung. Regularisierungen bei ANNs [107, 108] führen ähnlich wie in der Ridge-Regression (1.20) einen Strafterm für die Quadrate der Parameter, bzw. Gewichte in das Optimierungsproblem ein.

Kollinearität bedeutet, dass die Streuung der Daten durch weniger Komponenten erklärt, bzw. größtenteils erklärt werden kann, als Eingangsgrößen im Datensatz vorhanden sind. Hauptkomponentenanalyse (engl. *principal component analysis*, PCA) [109] und partielle Regressionsverfahren (engl. *partial least squares* PLS) [110] bieten eine Möglichkeit, die wichtigsten Eingangsgrößen, bzw. Linearkombinationen aus ihnen zu identifizieren. Dabei berücksichtigt die PCA ausschließlich die Eingangsgrößen, während PLS-Verfahren auch die Streuung der Zielgröße berücksichtigen. Für nichtlineare Zusammenhänge zwischen den Eingangsgrößen existiert z.B. die Kernel-PCA [111].

Bei fehlerbehafteten Eingangsgrößen kann die Parameterschätzung angepasst werden, dass die Residuen nicht mehr univariat in Richtung der Zielgröße berechnet werden, sondern multivariat unter Berücksichtigung aller fehlerbehafteter Größen. Entsprechende Ansätze finden sich unter orthogonale Regression (engl. *total least squares* TLS) [112–114].

Der Einfluss eines Datentupels im Optimierungsproblem hängt von der Lossfunktion ab. Mit einer gewichteten Regression (engl. *weighted least squares*, WLS) [63] können für die Datentupel individuelle Wichtungen eingeführt werden, um beispielsweise Heteroskedastizität, eine ungleichmäßige Verteilung von Punkten im Eingangsraum oder Ausreißer zu kompensieren. Dadurch erweitert sich die analytische Lösung des LS-Problems mit der Wichtungsmatrix $\boldsymbol{W}$ zu

$$\hat{\boldsymbol{\theta}}_{\mathrm{WLS}} = (\boldsymbol{X}^T\boldsymbol{W}\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{W}\boldsymbol{y},\ \boldsymbol{W} = \begin{pmatrix} w_1 & 0 & \dots & 0 \\ 0 & w_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & w_N \end{pmatrix}. \quad (1.21)$$

1.2.6 Hybride Modelle

Mit hybriden Modellen wird in der Robotik [115–117] sowohl Vorwissen genutzt, als auch eine geringe Datenabdeckung kompensiert. Dabei wird der Ansatz des Verstärkens verwendet, bei denen die Modellfehler eines analytischen Modells von einem datengetriebenen Modell abgebildet werden. Durch den Ansatz werden höhere Modellgüten erreicht, sowohl gegenüber datengetriebenen Modellen, die den gesamten Zusammenhang abbilden, als

auch gegenüber dem analytischen Modell. Im Gegensatz zu [118] werden datengetriebene Modelle mit lokalen Basisfunktionen ohne konstanten Term empfohlen, damit die zu erwartenden hohen Fehler des datengetriebenen Modells für Extrapolationen das hybride Modell nicht korrumpieren. Die Prädiktionen des datengetriebenen Modells streben damit für Extrapolationen gegen 0, wodurch die Prädiktion des hybriden Modells ausschließlich durch das analytische Modell bestimmt wird.

In [119] werden ein modellprädiktiver Regler und ein neuronaler adaptiver Regler in einer hybriden Struktur kombiniert. Der modellprädiktive Regler basiert auf einem ANN und ist sehr genau, verliert aber für Extrapolationen seine Zuverlässigkeit. Der neuronale adaptive Regler kann für die Extrapolationen robuste Werte liefern und die Regelung stabilisieren. Die Wichtung zwischen den beiden Reglern erfolgt durch die Parzen-Fenster-Methode.

1.3 Offene Probleme

Die datengetriebene Modellbildung zeigt eine Vielfalt an Modellfamilien sowie Kompensationsverfahren für Datensätze mit Einschränkungen in der Datenqualität. Es zeigen sich aber auch folgende offene Probleme:

1. *Fehlende Quantifizierung von Datenqualität:*
   Es existiert keine einheitliche Taxonomie und Kategorisierung für Datenqualität in Regressionsproblemen. Weiterhin mangelt es an interpretierbaren Bewertungskriterien für Datenqualität.

2. *Unübersichtliche Visualisierung von Datensätzen:*
   Die Visualisierung hochdimensionaler Datensätze ist häufig unübersichtlich. Es fehlen Visualisierungsroutinen, die dem Anwender ausschließlich die relevanten Eingangsgrößen darstellen.

3. *Unzuverlässige Quantifizierung von Modellqualität:*
   Reale Anwendungen haben gezeigt, dass bekannte Verfahren zur Modellbewertung die Modellgüte u.U. überschätzen. Das hängt vor allem mit Besonderheiten in der Datenverteilung zusammen. Es fehlen konservative Abschätzungen der Modellgüte als Ergänzung zu gängigen Verfahren.

4. *Fehlende Systematik zur Erfragung und Integration von Vorwissen:* Theoretische Ansätze zur Integration von Vorwissen in die Modellbildung existieren. Es fehlt allerdings ein systematischer Ansatz, um Vorwissen über Modellverläufe von Domänenexperten zu erfragen und in datengetriebene Modelle zu integrieren.

5. *Fehlendes Entwurfsverfahren einer allgemeinen hybriden Modellstruktur:* Es ist bekannt, dass komplexe datengetriebene Modelle in Extrapolationsbereichen ihre Prädiktionsfähigkeit verlieren können, weshalb in kritischen Anwendungen auf einfache, aber robuste, Modelle zurückgegriffen wird. Die hohe Genauigkeit komplexer Modelle kann damit häufig nicht genutzt werden.

1.4 Ziele und Aufgaben

Das Ziel dieser Arbeit ist die Entwicklung neuer Methoden, um den Entwurfsprozess von Regressionsmodellen bei Einschränkungen in der Datenqualität zu optimieren. Die Anwendung auf Benchmark-Datensätze sowie auf Echtwelt-Probleme aus dem Turbomaschinenbau, der Energieinformatik und der Medizintechnik dient der Untersuchung der neuen Methoden und dem Vergleich mit dem herkömmlichen Entwurfsprozess. Folgende Teilziele sind daher zu erfüllen:

1. Aspekte von Datenqualität, die den Entwurf von Regressionsmodellen betreffen, sind zu identifizieren und in bestehende Taxonomien einzuordnen,

2. bestehende Kenngrößen sind zu modifizieren und neue Bewertungskriterien zu entwickeln, um eine eingeschränkte Datenabdeckung zu quantifizieren,

3. Visualisierungsroutinen zur übersichtlichen Darstellung der Datenqualität sind zu implementieren,

4. neue Bewertungskriterien zur Modellbewertung bei eingeschränkter Datenabdeckung sind zu entwickeln,

5. eine systematische Vorgehensweise zur Erfragung und Integration von Vorwissen in nichtlineare Regressionsprobleme ist zu erarbeiten,

6. ein modifizierter automatisierter Entwurfsprozess ist zu entwickeln, um höhere Prädiktionsgüten, besonders bei einer eingeschränkten Datenabdeckung, zu erreichen,
7. Benchmark-Datensätze sind zu simulieren, um die Phänomene einer eingeschränkten Datenabdeckung ausführlich abzubilden,
8. die modifizierten und neu entwickelten Verfahren sind mit Hilfe der simulierten Benchmark-Datensätze und bekannter Benchmark-Datensätze zu validieren und
9. ihre Vorteile sind anhand von realen Anwendungsfällen aus dem Turbomaschinenbau, der Energieinformatik und der Medizintechnik zu diskutieren.

In Kapitel 2 wird zunächst eine ausführliche Taxonomie von Datenqualität in Regressionsproblemen eingeführt. Zur Veranschaulichung und zur Methodenevaluation werden Benchmark-Datensätze vorgestellt, welche die unterschiedlichen Aspekte der Datenqualität abbilden. Anschließend werden bekannte und neue Kenngrößen zu Gütekriterien hinsichtlich Datenqualität aggregiert. In Kapitel 3 werden neue Bewertungsstrategien für Regressionsmodelle bei beschränkter Datenqualität vorgeschlagen.

In Kapitel 4 werden dann neue Strategien zur Modellbildung vorgestellt, um zuverlässigere Modelle bei beschränkter Datenqualität zu bilden. Bei der Modellbildung findet die Kombination einfacher und komplexer Modelle in einer hybriden Modellstruktur sowie die Integration von Vorwissen besondere Beachtung.

Eine Dokumentation der Implementierung der vorgestellten Methoden findet sich in Kapitel 5. In Kapitel 6 werden die Verfahren auf Problemstellungen aus dem Turbomaschinenbau, der Energieinformatik und der Medizintechnik angewandt. Eine Zusammenfassung der Arbeit findet sich in Kapitel 7.

2 Neue Bewertungskriterien für Datenqualität

2.1 Taxonomie beschränkter Datenqualität

Der Entwurf von Regressionsmodellen ist eine Aufgabenstellung im *Data Mining*, wodurch sich gängige Schemata des Data-Mining-Prozesses auf Regressionsmodelle übertragen lassen. Der Data-Mining-Prozess besteht nach [120] aus Vorbereitung, Vorverarbeitung, Analyse und Nachbereitung. Die Bewertung der Datenqualität ist in der Vorverarbeitung einzuordnen, die Bewertung der Modellqualität in der Nachbereitung. Die Vorverarbeitung ist als wichtiger Bestandteil im Data Mining ausgewiesen und es finden sich für diverse Anwendungsfälle vorverarbeitende Schritte, vor allem im Bereich der Datenlager (engl. *Data Warehousing*) [121] und der Unternehmensdatenanalyse (engl. *Business Intelligence*) [122], um beispielsweise Datensätze nicht-numerischer Merkmale zu bereinigen [123]. In der Bereinigung werden unter anderem Duplikate entfernt, Widersprüchlichkeiten aufgelöst oder fehlende Werte imputiert. Durch das ständige Erschließen neuer Anwendungsfelder für Data-Mining-Aufgaben und das Entstehen immer größerer Datensammlungen und Datenbanken, rückt die Untersuchung von Datenqualität in den letzten Jahren in den Fokus von Forschung und Industrie [124–126].

Eine systematische Quantifizierung von Datenqualität in Datensätzen für den Entwurf von Regressionsmodellen fehlt, obwohl es Vorschläge zur Vorverarbeitung gibt: Normalisierungen, der Ausgleich von Schiefe in univariaten Verteilungen oder statistische Ausreißerdetektionen versuchen Einschränkungen in der Datenqualität zu kompensieren. Außerdem gibt es Ansätze für multivariate Ausreißerdetektionen sowie Ein-Klassen-Klassifikatoren zum Erkennen von nicht oder spärlich erfassten Bereichen.

Durch die Zugänglichkeit zu Tools und Programmiersprachen wie MATLAB oder R werden Regressionsmodelle von Experten unterschiedlicher Domänen entworfen. Dadurch werden häufig Verfahren angewandt, deren Voraussetzungen in den Datensätzen verletzt werden. Damit grenzt sich der Data-Mining-Prozess von der klassischen Statistik ab, da aus der Sicht des Data-Minings die Gültigkeit eines datengetriebenen Modells weniger von statistischen Verteilungsannahmen als vielmehr durch eine in der Modellvalidierung festgestellte Güte bestimmt wird. Für die Validierung und zum einheitlichen Methodenvergleich werden Kreuzvalidierungen und ähnliche auf Residuen basierende Verfahren oder informationstheoretische Maße verwendet. Regressionsmodelle und ihre Bewertungen verlieren aber ihre Gültigkeit, wenn die folgenden generellen Annahmen verletzt werden:

- Die Eingangsvektoren in der Anwendung entstammen der gleichen Verteilung wie die Eingangsvektoren der Lerndaten und
- die Zielgröße in der Anwendung wird durch den gleichen Prozess / Zusammenhang generiert wie die Zielgrößen der Lerndaten.

Außerdem verlieren gängige Distanzmetriken wie die euklidische Distanz in hochdimensionalen Datensätzen aufgrund einer geringen Datendichte ihre Aussagekraft [83, 127].

Eine automatisierte, quantitative und gegenüber der Datendichte robuste Analyse der Datenqualität ist demnach interessant für die Wahl und Anpassung der Modellstruktur, sie beschleunigt die visuelle Inspektion der Daten und lässt Rückschlüsse auf die Datenerhebung und ihre mögliche Verbesserung zu.

Allgemeine Taxonomien für Datenqualität wurden bereits vorgestellt [128, 129], wobei sich als Konsens die Aspekte Korrektheit (engl. *accuracy*), Vollständigkeit (engl. *completeness*), Konsistenz (engl. *consistency*), Verfügbarkeit (engl. *accessibility*) und Aktualität (engl. *timeliness*) ergeben haben.

In den folgenden Abschnitten wird die generelle Taxonomie der Datenqualität hinsichtlich des Entwurfs von Regressionsmodellen interpretiert und erweitert. Datenqualitätsprobleme aus realen Anwendungsfällen dienen als Anregung zur Definition von Subklassen der Taxonomie und entsprechenden Bewertungskriterien. Bei den Bewertungskriterien wird weitestgehend

auf bestehende Maße aus der Statistik, der angewandten Mathematik und der explorativen Datenanalyse zurückgegriffen. Für einige Bewertungskriterien werden neue Maße entwickelt.

Um die Quantifizierung von Datenqualität als vorverarbeitenden Schritt im Modellentwurf für ein breites Feld von Anwendungsbereichen und Nutzergruppen zu etablieren, ergeben sich folgende Anforderungen für die verwendeten Algorithmen:

- schnelle Berechnung,
- einfache Parametrierung und
- hohe Interpretierbarkeit.

Es ist zu beachten, dass eine hohe Datenqualität nichts über die Eignung von Daten aussagt, ihren zugrundeliegenden Zusammenhang mit Hilfe modellbildender Verfahren abbilden zu können. Falls ungeeignete Eingangsgrößen in einem Datensatz erfasst sind, können modellbildende Verfahren keine relevanten Zusammenhänge abbilden, auch wenn die Daten den vorhandenen Eingangsraum gleichmäßig abdecken.

Abbildung 2.1 ordnet die Einschränkungen der Datenqualität für den Entwurf von Regressionsmodellen aus Abschnitt 1.2.4 in die gängige Taxonomie für Datenqualität ein.

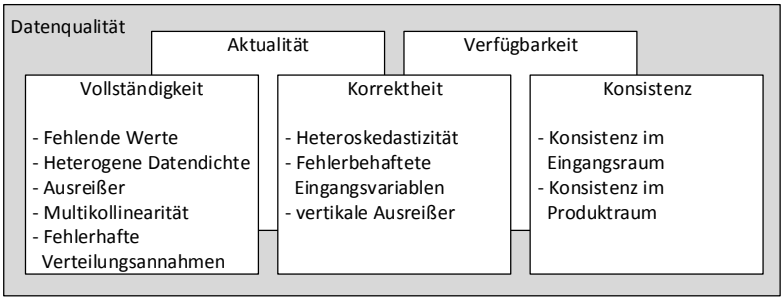

Abbildung 2.1: Neue systematische Einordnung regressionsspezifischer Einschränkungen der Datenqualität in gängige Taxonomie

Da beim Entwurf von Regressionsmodellen davon ausgegangen wird, dass ein Lerndatensatz bereits vorliegt und in dieser Arbeit ausschließlich statische Modelle betrachtet werden, sind die Aspekte Verfügbarkeit und Aktualität zu vernachlässigen.

In den folgenden Abschnitten wird besonders die Vollständigkeit von Datensätzen untersucht. Neben fehlenden Werten, die durch Übertragungsfehler o.ä. auftreten können und deren Behandlung bereits ausgiebig in der Literatur behandelt wird, werden der Vollständigkeit auch heterogene Datendichte, Ausreißer, Multikollinearität und fehlerhafte Verteilungsannahmen zugeordnet. Anschaulich beschreiben die Einschränkungen, wie die Daten im Eingangsraum verteilt sind, bzw. wie der Eingangsraum mit Daten abgedeckt ist.

Im Folgenden wird ein neuer Teilaspekt der Vollständigkeit, die Datenabdeckung, eingeführt.

Die Datenabdeckung beschreibt, wie gleichmäßig die Eingangsvektoren in einem Datensatz verteilt sind. Regressionen werden zunehmend auf Datensätzen angewandt, deren Eingangsvektoren nicht durch eine statistische Versuchsplanung [130] festgelegt wurden[1]. Stattdessen werden die Daten beispielsweise durch die Überwachung technischer Systeme automatisiert gesammelt. Damit bilden bereits die Eingangsdaten von Regressionen Phänomene des Systems ab und widersprechen statistischen Verteilungsannahmen. Die Verteilung der Eingangsdaten hat Einfluss auf die Zuverlässigkeit eines Regressionsmodells. Durch die visuelle Analyse verschiedener Datensätze konnten einige Phänomene der Datenabdeckung identifiziert werden, die im Folgenden als

- Korrelationen,
- Cluster,
- Konfigurationen,
- Ausreißer und
- Orthogonalität

[1] Die Versuchsplanung stellt eine gleichmäßige Verteilung der Eingangsdaten sicher, um alle Zustände eines betrachteten Systems zu erfassen.

bezeichnet werden und als Erweiterung der heterogenen Datendichte und der fehlerhaften Verteilungsannahmen verstanden werden.

2.2 Benchmark-Datensätze

In dieser Arbeit wird zur Validierung von Methoden häufig auf bekannte Benchmark-Datensätze für Regressionsprobleme zurückgegriffen [131–135]. Die Datensätze weisen zum Teil die Phänomene Korrelationen, Cluster, Konfigurationen, Ausreißer und Orthogonalität auf, wie in Tabelle 2.1 gezeigt wird[2]. Dennoch sind sie nur bedingt geeignet, um systematisch zu analysieren, ob durch entsprechende Bewertungskriterien die Phänomene erkannt werden und ob gegenseitige Beeinflussungen der Bewertungskriterien auftreten.

Datensatz	Korr	Cluster	Konfig	Aus	Ortho
Abalone	ja	ja[3]	ja	ja	nein
Airfoil Self Noise	nein	ja[3]	ja	nein	ja
Boston Housing	ja	ja	ja	nein	ja
California Housing	ja	nein	ja	ja	nein
Computeractivity CPU	ja	ja	nein	ja	ja
Concrete	nein	ja	ja	ja	ja
Delta Ailerons	nein	ja	nein	ja	nein
Delta Elevators	nein	ja[3]	ja	ja	nein
Red Wine Quality	nein	nein	nein	ja	nein
White Wine Quality	nein	nein	nein	ja	nein

Tabelle 2.1: Übersicht über bekannte Benchmark-Datensätze und eine Einschätzung auftretender Phänomene

Deshalb wird ein Datensatz D_{Sim} simuliert, dessen Eingangsgrößen die genannten Phänomene aufweisen. Der Datensatz D_{Sim} besteht aus vier Teildatensätzen, die in Tabelle 2.2 dargestellt sind.

Die Teildatensätze beinhalten zehn Eingangsgrößen und drei mögliche Zielgrößen und umfassen jeweils 200 Datentupel. Die Eingangsgrößen der

[2] Die Übersicht entspricht der subjektiven Wahrnehmung des Autors.
[3] Cluster, die durch Konfigurationen entstehen.

Datensatz	Datenmatrix	Zielgrößenvektoren
$D_{\text{Sim,Lern},1}$	$\boldsymbol{X}^{200\times 10}_{\text{Sim,Lern},1}$	$\boldsymbol{y}^{200\times 1}_{\text{i,Sim,Lern},1}$ $;i=1,2,3$
$D_{\text{Sim,Lern},2}$	$\boldsymbol{X}^{200\times 10}_{\text{Sim,Lern},2}$	$\boldsymbol{y}^{200\times 1}_{\text{i,Sim,Lern},2}$ $;i=1,2,3$
$D_{\text{Sim,Lern},3}$	$\boldsymbol{X}^{200\times 10}_{\text{Sim,Lern},3}$	$\boldsymbol{y}^{200\times 1}_{\text{i,Sim,Lern},3}$ $;i=1,2,3$
$D_{\text{Sim,Test}}$	$\boldsymbol{X}^{200\times 10}_{\text{Sim,Test}}$	$\boldsymbol{y}^{200\times 1}_{\text{i,Sim,Test}}$ $;i=1,2,3$

Tabelle 2.2: Bezeichner der Teildatensätze von D_{Sim}

Teildatensätze unterscheiden sich ausschließlich in der Ausprägung der Phänomene Korrelation (x_1, x_2), Cluster (x_3, x_4), Konfigurationen (x_6), Ausreißer (x_7, x_8) und Orthogonalität (x_9, x_{10}). Die Eingangsgrößen des Testdatensatzes weisen die Phänomene nicht auf, sondern folgen einer multivariaten Gleichverteilung. Die Zielgrößen folgen in allen Lern- und Testdatensätzen den Zusammenhängen

$$y_1 = \sin(2\pi x_3) + \sin(2\pi x_4) + 0.2x_6, \tag{2.1}$$

$$y_2 = \sin(2\pi x_5) + 0.2 \cdot (x_6 + x_1 + x_2) \text{ und} \tag{2.2}$$

$$y_3 = \sin(2\pi x_5) + \sin(2\pi x_6 x_1 x_2). \tag{2.3}$$

Bei den Lerndatensätzen sind die Zielgrößen mit einem homoskedastischen Rauschen überlagert. Tabelle 2.3 gibt eine Übersicht über die Besonderheiten der Teildatensätze.

Datensatz	Ausprägung der Phänomene	Rauschen auf Zielgröße
$D_{\text{Sim,Lern},1}$	hoch	$\mathcal{N}(0, 0.01)$
$D_{\text{Sim,Lern},2}$	mittel	$\mathcal{N}(0, 0.01)$
$D_{\text{Sim,Lern},3}$	niedrig	$\mathcal{N}(0, 0.01)$
$D_{\text{Sim,Test}}$	ohne	ohne

Tabelle 2.3: Details der Teildatensätze von D_{Sim}

Die Phänomene werden im Folgenden anhand von Eingangsgrößen der Datenmatrix $X_{\text{Sim,Lern},1}$ sowie der bekannten Benchmark-Datensätze erläutert.

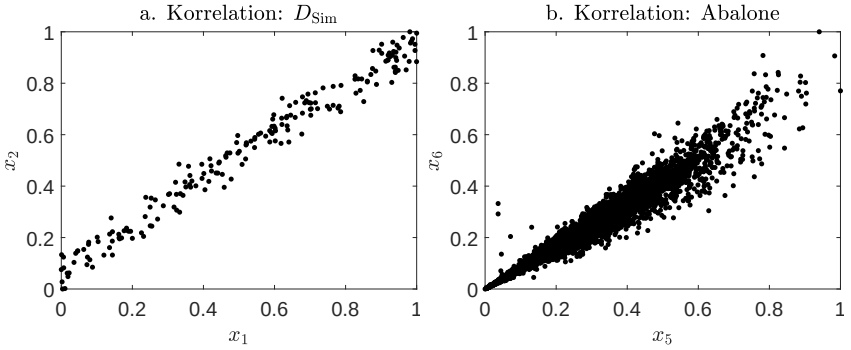

Abbildung 2.2: Korrelationen im a. Datensatz D_{Sim} b. Datensatz „Abalone“

Bei starker Korrelation ist nur ein geringer Teil des zweidimensionalen Eingangsraums mit Daten abgedeckt, obwohl Histogramme beider Eingangsgrößen eine ganzheitliche Abdeckung vermuten lassen.

Abbildung 2.2a zeigt korrelierende Eingangsgrößen im Datensatz $D_{\text{Sim,Lern,1}}$. Abbildung 2.2b zeigt korrelierende Eingangsgrößen im Datensatz „Abalone“ [131]. Ist eine Korrelation systembedingt, sind die Eingangsgrößen redundant und können für die Modellbildung selektiert oder transformiert und reduziert werden.

Liegen die Daten in Clustern vor, bietet sich das Bilden von lokalen Teilmodellen an. Abbildung 2.3a zeigt zwei Eingangsgrößen des Datensatzes $D_{\text{Sim,Lern,1}}$ mit Clustern.

Abbildung 2.3b zeigt zwei Eingangsgrößen des Datensatzes „Concrete“ [131, 134] mit Clustern. Die Bewertung, ob und wie viele Cluster in einem Datensatz vorliegen, ist ein nichttriviales Problem und ist neben der Regression und der Klassifikation eine der drei formalisierten Problemstellungen im Data-Mining.

Konfigurationen liegen vor, wenn für eine Eingangsgröße im Verhältnis zu den anderen Eingangsgrößen nur wenige unterschiedliche Ausprägungen existieren. Dadurch entstehen Cluster. Konfigurationen können auf ordinal- oder nominalskalierte Eingangsgrößen hinweisen. Abbildung 2.4a zeigt zwei

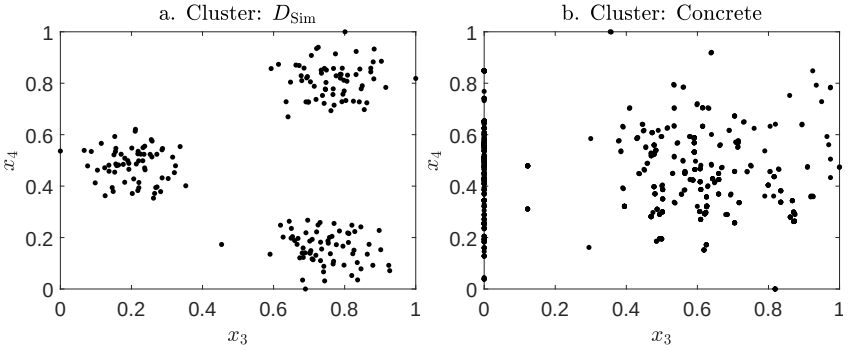

Abbildung 2.3: Cluster im a. Datensatz D_{Sim} b. Datensatz „Concrete“

Eingangsgrößen des Datensatzes $D_{\text{Sim,Lern,1}}$. Bei einer der Eingangsgrößen liegen Konfigurationen vor. Abbildung 2.4b zeigt zwei Eingangsgrößen des Datensatzes „Abalone“, von denen eine Konfigurationen aufweist.

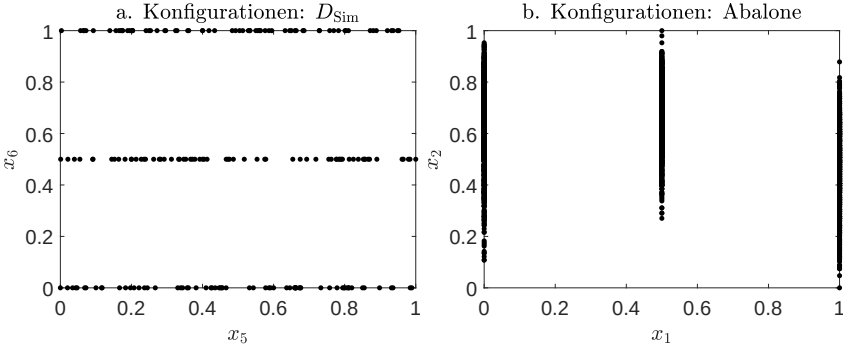

Abbildung 2.4: Konfigurationen im a. Datensatz D_{Sim} b. Datensatz „Abalone“

Ausreißer werden Datentupel genannt, die aufgrund ihrer Lage im Eingangsraum einen großen Einfluss auf die Modellbildung ausüben können. Generell werden Ausreißer als Datentupel beschrieben, die sich vom Großteil der Datentupel eines Datensatzes deutlich unterscheiden. Abbildung 2.5a zeigt zwei Eingangsgrößen des Datensatzes $D_{\text{Sim,Lern,1}}$ mit

Ausreißern. Abbildung 2.5b zeigt zwei Eingangsgrößen des Datensatzes „California Housing“ [133] mit Ausreißern.

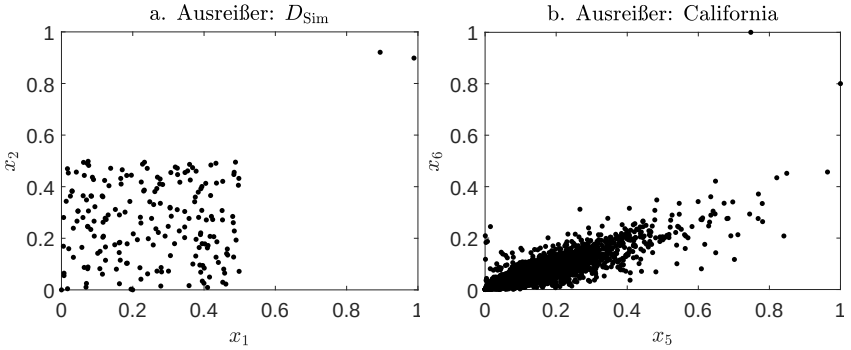

Abbildung 2.5: Ausreißer im a. Datensatz D_{Sim} b. Datensatz „California“

Die Detektion von Ausreißern ist abhängig von der jeweiligen Anwendung. In [82, 83] findet sich eine Übersicht über gängige Ansätze. Bei Ausreißerdetektionen stellt sich anwendungsspezifisch die Frage, ab wann ein Datentupel ein Ausreißer ist, und ob Gruppen von Datentupeln, die entsprechend weit entfernt vom Großteil der Daten liegen, eine Gruppe von Ausreißern darstellt oder bereits ein Cluster, das nicht von der Modellbildung auszuschließen ist.

Orthogonalität beschreibt nichtkorrelierte Eingangsgrößen mit vielen Datentupeln für einen bestimmten Wert. Dadurch liegen für Regressionen keine Daten vor, die Wechselwirkungen zweier Eingangsgrößen auf die Zielgröße beschreiben. Abbildung 2.6a zeigt zwei Eingangsgrößen des Datensatzes $D_{\text{Sim,Lern,1}}$ mit Orthogonalität. Abbildung 2.6b zeigt zwei Eingangsgrößen des Datensatzes „Boston Housing“ mit Orthogonalität. Bei starker Orthogonalität ist nur ein geringer Teil des zweidimensionalen Eingangsraums mit Daten abgedeckt, obwohl Histogramme beider Eingangsgrößen eine ganzheitliche Abdeckung vermuten lassen.

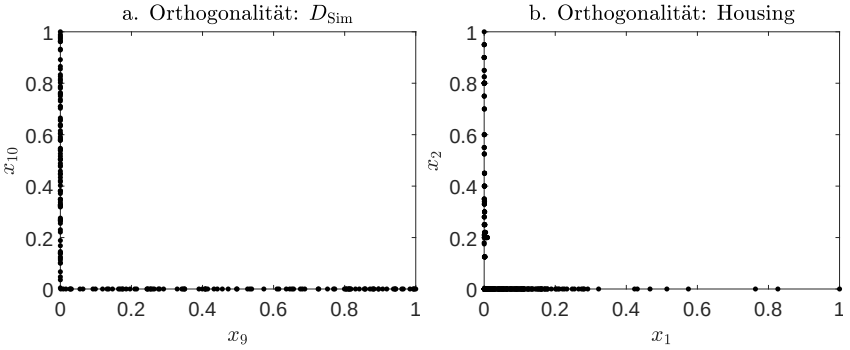

Abbildung 2.6: Orthogonalität im a. Datensatz D_{Sim} b. Datensatz „Boston Housing“

2.3 Datenabdeckung

2.3.1 Übersicht

Im folgenden Abschnitt werden neue Bewertungskriterien, die zum Teil auf bekannten Bewertungskriterien aufbauen, zu einem neuen modularen Kriterium für Datenabdeckung kombiniert. Die Kriterien beziehen sich ausschließlich auf die uni- und bivariaten Verteilungen der Eingangsvektoren und berücksichtigen nicht die Zielgröße. Sie sind daher als Ergänzung zur herkömmlichen Merkmalsbewertung für den Entwurf von Regressionsmodellen zu sehen. Auf multivariate Verfahren mit mehr als zwei Eingangsgrößen wird aufgrund des Fluchs der Dimensionalität und zu Gunsten der Interpretierbarkeit der Kriterien verzichtet. Stattdessen werden alle bivariaten Projektionen eines hochdimensionalen Datensatzes bewertet und die Ergebnisse beispielsweise mit Hilfe des Minimal- oder Mittelwerts aggregiert. Die quantifizierten Einschränkungen der Datenabdeckung sind die in Abschnitt 2.2 vorgestellten Phänomene. Anders als die in [90] vorgestellten Scagnostics werden in den hier vorgestellten Bewertungskriterien keine Maße aus der Graphentheorie verwendet, um eine einfache Implementierung zu ermöglichen. Außerdem liegt der Schwerpunkt auf Phänomenen, die beim Entwurf von Regressionsmodellen relevant sind. Alle Bewertungskriterien liegen im Intervall $[0, 1]$, um eine hohe Interpretierbarkeit zu

gewährleisten. Ein niedriger Wert für ein Bewertungskriterium deutet auf das Vorliegen des entsprechenden Phänomens und damit auf eine geringe Datenqualität hin. Um eine aussagekräftige Bewertung durch die Kriterien zu ermöglichen und Effekte unterschiedlicher Skalierungen zu vermeiden, werden alle Eingangsgrößen auf das Einheitsintervall normiert. Die Bewertungskriterien wurden entwickelt, um die in den Benchmark-Datensätzen in unterschiedlicher Ausprägung realisierten Einschränkungen der Datenqualität abzubilden. Eine kurze Vorstellung der Kriterien findet sich bereits in [136].

2.3.2 Korrelationen

Als Hilfsgröße, um Datenqualität bezüglich Korrelationen zu quantifizieren, wird der empirische Korrelationskoeffizient r_{x_j,x_l} verwendet. Daraus berechnet sich das Bewertungskriterium

$$q_{\text{Korr},x_j,x_l} = 1 - |r_{x_j,x_l}|, q_{\text{Korr},x_j,x_l} \in [0,1]. \tag{2.4}$$

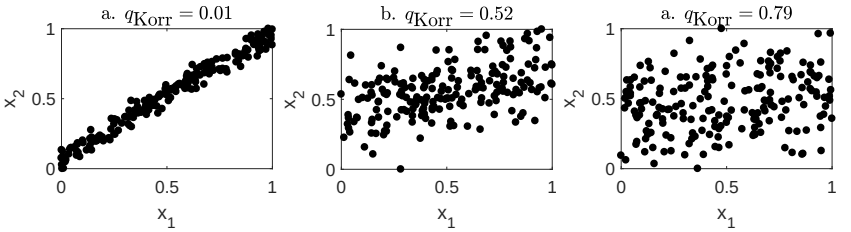

Abbildung 2.7: Korrelierende Eingangsgrößen und die entsprechende Bewertung für a. $D_{\text{Sim,Lern},1}$ b. $D_{\text{Sim,Lern},2}$ c. $D_{\text{Sim,Lern},3}$

Abbildung 2.7 zeigt das Verhalten von q_{Korr} für die korrelierenden Eingangsgrößen für die Teildatensätze im Datensatz D_{Sim}.

Das Kriterium kann lineare Zusammenhänge finden, wobei das Vorzeichen des Zusammenhangs keine Rolle spielt. Auf das Finden von komplexen, nichtlinearen Zusammenhängen wird verzichtet, da vor allem die linearen Korrelationen dazu führen, dass große zusammenhängende Bereiche des betrachteten Eingangsraums nicht mit Daten abgedeckt sind. Um

einen Datensatz mit mehr als zwei Eingangsgrößen zu bewerten, wird der Mittelwert

$$\overline{Q}_{\text{Korr}} = \frac{2}{p(p-1)} \sum_{j=1}^{p-1} \sum_{l=j+1}^{p} q_{\text{Korr},x_j,x_l} \tag{2.5}$$

oder der Minimalwert

$$Q_{\text{Korr,min}} = \min_{\substack{j=1,\ldots,(p-1) \\ l=j+1,\ldots,p}} q_{\text{Korr},x_j,x_l} \tag{2.6}$$

verwendet. Während der Mittelwert einen allgemeinen Einblick hinsichtlich Korrelationen gibt, liefert der Minimalwert eine Aussage darüber, ob überhaupt eine Eingangsgröße mit einer anderen korreliert. Die Aggregation wird auch für die weiteren Kriterien übernommen.

2.3.3 Cluster

In [137] wird die Bimodalität der Häufigkeitsverteilung der paarweisen Distanzen zwischen den Datentupeln als visuelles Kriterium für Cluster verwendet. Um Bimodalität zu quantifizieren, kann der *Hartigans Dip-Test of Unimodality* verwendet werden [138, 139]. Der Dip-Test liefert den sogenannten Dip-Index v_{Dip}. Zwischen v_{Dip} und der Bimodalität wird ein sigmoidaler Zusammenhang vermutet, bei dem ein größerer Dip-Index mit einer ausgeprägteren Bimodalität korreliert. Bei der Verwendung der paarweisen Distanzen entspricht die Bimodalität dem Vorliegen von Clustern.

Das Bewertungskriterium ergibt sich mit dem Parameter τ_{Cluster} parametrierbar zu

$$q_{\text{Cluster},x_j,x_l} = 1 - \frac{1}{1 + \exp\left(\frac{-2\ln|99|}{\tau_{\text{Cluster}}}\left(v_{\text{Dip}} - \frac{\tau_{\text{Cluster}}}{2}\right)\right)}, q_{\text{Cluster},x_j,x_l} \in [0,1]. \tag{2.7}$$

Je kleiner $\tau_{\text{Cluster}} > 0$ gewählt wird, desto empfindlicher ist das Bewertungskriterium. Einen angemessenen Kompromiss zwischen Empfindlichkeit und Robustheit stellt $\tau_{\text{Cluster}} = 0.06$ dar.

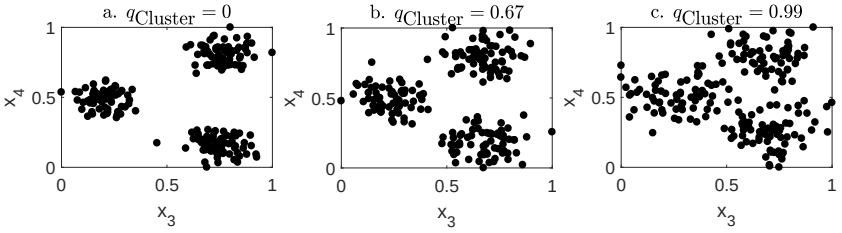

Abbildung 2.8: Eingangsgrößen mit Cluster und die entsprechende Bewertung für a. $D_{\text{Sim,Lern},1}$ b. $D_{\text{Sim,Lern},2}$ c. $D_{\text{Sim,Lern},3}$

Abbildung 2.8 zeigt das Verhalten von q_{Cluster} für die Eingangsgrößen mit Cluster für die Teildatensätze im Datensatz D_{Sim}.

Um einen Datensatz mit mehr als zwei Eingangsgrößen zu bewerten, wird der Mittelwert

$$\overline{Q}_{\text{Cluster}} = \frac{2}{p(p-1)} \sum_{j=1}^{p-1} \sum_{l=j+1}^{p} q_{\text{Cluster},x_j,x_l} \tag{2.8}$$

oder der Minimalwert

$$Q_{\text{Cluster,min}} = \min_{\substack{j=1,\ldots,(p-1) \\ l=j+1,\ldots,p}} q_{\text{Cluster},x_j,x_l} \tag{2.9}$$

verwendet.

2.3.4 Konfigurationen

Sei c_j die Anzahl unterschiedlicher Ausprägungen von x_j. Dann ergibt sich als Bewertungskriterium

$$q_{\text{Konfig},x_j} = \frac{c_j}{\max_l c_l},\ l = 1, \ldots, p. \tag{2.10}$$

Gleichmäßig wenige Ausprägungen aller Eingangsgrößen lassen auf eine statistische Versuchsplanung schließen und sind nicht als Einschränkung in der Datenqualität zu bewerten. Daher wird das Bewertungskriterium

in Abhängigkeit zum Maximalwert univariater Ausprägungen ($\max_l c_l$) berechnet. Abbildung 2.9 zeigt das Verhalten von q_{Konfig} für die Eingangsgröße mit Konfigurationen im Vergleich zu einer Eingangsgröße mit vielen Ausprägungen für die Teildatensätze im Datensatz D_{Sim}.

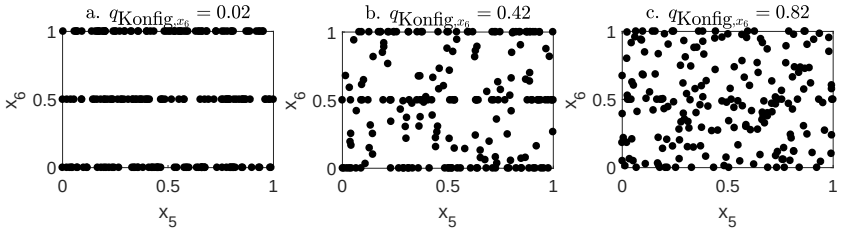

Abbildung 2.9: Eingangsgrößen mit Konfigurationen und die entsprechende Bewertung für a. $D_{\text{Sim,Lern},1}$ b. $D_{\text{Sim,Lern},2}$ c. $D_{\text{Sim,Lern},3}$

Um einen Datensatz mit mehr als einer Eingangsgröße zu bewerten, wird der Mittelwert

$$\overline{Q}_{\text{Konfig}} = \frac{1}{p} \sum_{j=1}^{p} q_{\text{Konfig},x_j} \tag{2.11}$$

oder der Minimalwert

$$Q_{\text{Konfig,min}} = \min_{j=1,\ldots,p} q_{\text{Konfig},x_j} \tag{2.12}$$

verwendet.

2.3.5 Ausreißer

Das Finden von Ausreißern ist daten- und anwendungsabhängig. Es wird daher ein intuitiver, verständlich parametrierbarer Ansatz gewählt. Da bereits ein einziger Ausreißer Annahmen über Verteilungen und damit Modellentwurf sowie -Validierung korrumpieren kann, ist ein Bewertungskriterium zu finden, dass nichts über die Anzahl an Ausreißern aussagt, sondern wie sehr sich ein möglicher Ausreißer von den übrigen Daten unterscheidet. Dazu werden die Distanzen aller Eingangsvektoren zu ihrem k-ten

nächsten Nachbarn verwendet. Ein Wert $k > 1$ erlaubt das Erkennen eines Ausreißers, auch wenn es sich um eine Gruppe von k Eingangsvektoren handelt.

$\boldsymbol{d}^{N\times 1}_{\text{k-NN},x_j,x_l}$ beinhaltet die euklidische Distanz jedes Datentupels zu seinem k-ten nächsten Nachbarn unter Berücksichtigung der Eingangsgrößen x_j und x_l. Zur Bewertung von Ausreißern wird zum einen die maximale Distanz

$$d_{\text{k-NN},x_j,x_l,\max} = \max_{\substack{j=1,\ldots,(p-1)\\ l=j+1,\ldots,p}} d_{\text{k-NN},x_j,x_l} \tag{2.13}$$

eines Datentupels zu seinem k-ten nächsten Nachbarn unter Berücksichtigung der Eingangsgrößen x_j und x_l verwendet, was den potentiellen Ausreißer charakterisiert. Zum anderen ein Referenzwert, der die normalen Punkte charakterisiert. Dazu wird $d_{\text{k-NN},x_j,x_l,0.95}$ verwendet, das 0.95-Quantil von $\boldsymbol{d}^{N\times 1}_{\text{k-NN},x_j,x_l}$[4]. Maximalwert und Referenzwert bilden den Indikator

$$q_{\text{Aus,Ind},x_j,x_l} = \begin{cases} 1, & \text{falls } d_{\text{k-NN},x_j,x_l,\max} = 0 \\ \frac{d_{\text{k-NN},x_j,x_l,0.95}}{d_{\text{k-NN},x_j,x_l,\max}} & \text{sonst.} \end{cases} \tag{2.14}$$

Je kleiner $q_{\text{Aus,Ind}}$, desto eher liegen Ausreißer bzw. liegt ein Ausreißer vor. Damit der Anwender die Empfindlichkeit parametrieren kann, wird eine Sigmoidalfunktion verwendet und das Bewertungskriterium ergibt sich mit den Randbedingungen

$$q_{\text{Aus},x_j,x_l}(q_{\text{Aus,Ind},x_j,x_l} = \tau_{\text{Aus},1}) \stackrel{!}{=} 0.01 \tag{2.15a}$$

$$q_{\text{Aus},x_j,x_l}(q_{\text{Aus,Ind},x_j,x_l} = \tau_{\text{Aus},2}) \stackrel{!}{=} 0.99 \tag{2.15b}$$

zu

$$q_{\text{Aus},x_j,x_l} = \frac{1}{1+\exp\left(-\ln|\frac{0.01}{98.01}|\frac{q_{\text{Aus,Ind},x_j,x_l}}{\tau_{\text{Aus},2}-\tau_{\text{Aus},3}} + \ln|\frac{0.01}{0.99}|\frac{\tau_{\text{Aus},2}}{\tau_{\text{Aus},2}-\tau_{\text{Aus},3}}\right)} \tag{2.16}$$

Abbildung 2.10 zeigt das Verhalten von q_{Aus} für die Eingangsgrößen mit Ausreißern für die Teildatensätze im Datensatz D_{Sim} mit den empirisch gewählten Parametern $k = 2$, $\tau_{\text{Aus},2} = 0.05$ und $\tau_{\text{Aus},3} = 0.3$.

[4] Auf ein variables Quantil wird zu Gunsten des Parametrierungsaufwands verzichtet.

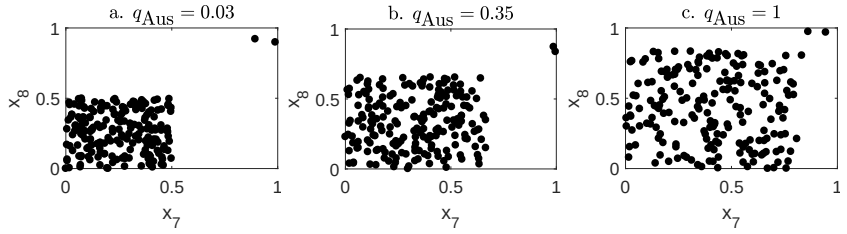

Abbildung 2.10: Eingangsgrößen mit Ausreißern und die entsprechende Bewertung für a. $D_{\mathrm{Sim,Lern},1}$ b. $D_{\mathrm{Sim,Lern},2}$ c. $D_{\mathrm{Sim,Lern},3}$ mit $k = 2$

Um einen Datensatz mit mehr als zwei Eingangsgrößen zu bewerten, wird der Mittelwert

$$\overline{Q}_{\mathrm{Aus}} = \frac{2}{p(p-1)} \sum_{j=1}^{p-1} \sum_{l=j+1}^{p} q_{\mathrm{Aus},x_j,x_l} \tag{2.17}$$

oder der Minimalwert

$$Q_{\mathrm{Aus,min}} = \min_{\substack{j=1,\ldots,(p-1) \\ l=j+1,\ldots,p}} q_{\mathrm{Aus},x_j,x_l} \tag{2.18}$$

verwendet.

2.3.6 Orthogonalität

Da bisher keine Kenngrößen Orthogonalität zuverlässig erkennen können, wird im folgenden Abschnitt ein Bewertungskriterium vorgestellt, um mit einigen Hilfsgrößen ein Maß für Orthogonalität bereitzustellen.

Mit den Indexmengen

$$\begin{aligned} &\mathcal{I}_{\boldsymbol{X}} = \{1, \ldots, N\}, \\ &\mathcal{I}_{\mathrm{In},j} = \{n \in \mathcal{I}_{\boldsymbol{X}} | x_{\mathrm{n,j}} \in [c - \tau_{\mathrm{Ortho}}, c + \tau_{\mathrm{Ortho}}]\} \text{ und} \\ &\mathcal{I}_{\mathrm{Out},j} = \mathcal{I}_{\boldsymbol{X}} \setminus \mathcal{I}_{\mathrm{In},j} \end{aligned} \tag{2.19}$$

ergeben sich die mittleren absoluten Abweichungen

$$e_{l,\mathrm{Out},j} = \sqrt{\frac{1}{|\mathcal{I}_{\mathrm{Out},j}|} \sum_{n \in \mathcal{I}_{\mathrm{Out},j}} \left(x_{n,l} - \frac{1}{|\mathcal{I}_{\mathrm{Out},j}|} \sum_{z \in \mathcal{I}_{\mathrm{Out},j}} x_{z,l} \right)} \text{ und} \quad (2.20)$$

$$e_{l,\mathrm{In},j} = \sqrt{\frac{1}{|\mathcal{I}_{\mathrm{In},j}|} \sum_{n \in \mathcal{I}_{\mathrm{In},j}} \left(x_{n,l} - \frac{1}{|\mathcal{I}_{\mathrm{In},j}|} \sum_{z \in \mathcal{I}_{\mathrm{In},j}} x_{z,j} \right)}$$

und das Bewertungskriterium

$$q_{\mathrm{Ortho},x_j,x_l} = \min \left\{ \min_c \frac{e_{j,\mathrm{Out},l}}{e_{j,\mathrm{In},l}}, \min_c \frac{e_{l,\mathrm{Out},j}}{e_{l,\mathrm{In},j}} \right\}. \quad (2.21)$$

τ_{ortho} ist ein empirisch zu wählender Parameter, der die Empfindlichkeit des Bewertungskriteriums bestimmt. c bestimmt für welche Bereiche die Eingangsgrößen hinsichtlich Orthogonalität untersucht werden. Abbildung 2.11 veranschaulicht die Parameter und Kenngrößen $c, \tau_{\mathrm{Ortho}}, e_{l,\mathrm{Out},j}$ und $e_{l,\mathrm{In},j}$. $e_{l,\mathrm{In},j}$ beschreibt die Streuung in x_l-Richtung der Datentupel, die innerhalb eines kleinen Bereichs von x_j liegen. $e_{j,\mathrm{Out},j}$ beschreibt die Streuung in x_l-Richtung der Datentupel, die außerhalb des kleinen Bereichs von x_j liegen. Je kleiner der Quotient $\frac{e_{l,\mathrm{Out},j}}{e_{l,\mathrm{In},j}}$, desto stärker liegt Orthogonalität vor.

Abbildung 2.12 zeigt das Verhalten von q_{Ortho} für die Eingangsgrößen mit Orthogonalität für die Teildatensätze im Datensatz D_{Sim}.

Um einen Datensatz mit mehr als zwei Eingangsgrößen zu bewerten, wird der Mittelwert

$$\overline{Q}_{\mathrm{Ortho}} = \frac{2}{p(p-1)} \sum_{j=1}^{p-1} \sum_{l=j+1}^{p} q_{\mathrm{Ortho},x_j,x_l} \quad (2.22)$$

oder der Minimalwert

$$Q_{\mathrm{Ortho,min}} = \min_{\substack{j=1,\ldots,(p-1) \\ l=j+1,\ldots,p}} q_{\mathrm{Ortho},x_j,x_l} \quad (2.23)$$

verwendet.

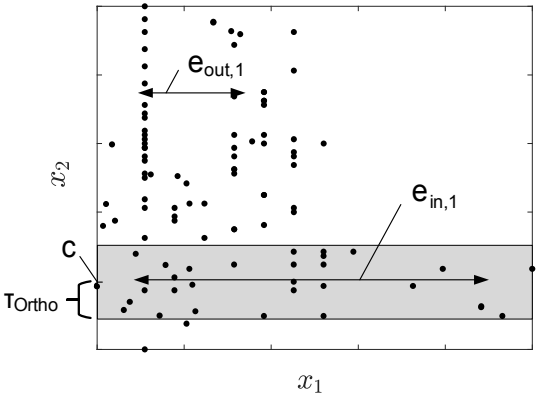

Abbildung 2.11: Kenngrößen zur Berechnung des Bewertungskriteriums für Orthogonalität

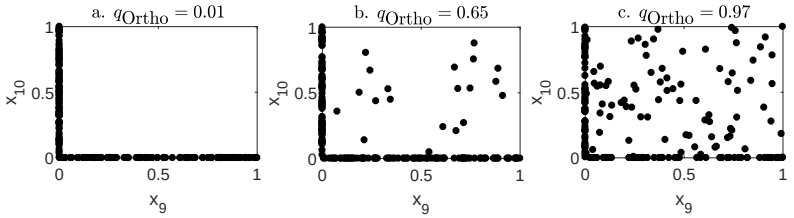

Abbildung 2.12: Eingangsgrößen mit Ausreißern und die entsprechende Bewertung für a. $D_{\text{Sim,Lern},1}$ b. $D_{\text{Sim,Lern},2}$ c. $D_{\text{Sim,Lern},3}$

2.3.7 Aggregierte Bewertung

Um eine aggregierte Bewertung für die Datenabdeckung eines Datensatzes zu erhalten, kann die gewichtete Summe

$$\tilde{Q} = \left(w_{\text{Korr}}\bar{Q}_{\text{Korr}} + \ldots + w_{\text{Ortho}}\bar{Q}_{\text{Ortho}}\right) \frac{1}{w_{\text{Korr}} + \ldots + w_{\text{Ortho}}} \qquad (2.24)$$

verwendet werden. Die Wichtungen sind individuell vom Anwender zu wählen. Alternativ können anstatt der mittleren Bewertungen der Phäno-

mene $\bar{Q}$ auch die minimalen Bewertungen $Q_{\min}$ zur Aggregation verwendet werden. Für eine ausführliche Untersuchung empfiehlt es sich beide Aggregationen zu verwenden. Die mittleren Bewertungen liefern eine Aussage darüber, welche Phänomene in einem Datensatz häufig auftreten. Die minimalen Bewertungen liefern eine Aussage darüber, wie sehr die Phänomene im Einzelfall ausgeprägt sind.

2.4 Validierung

Für die Quantifizierung der Datenabdeckung ist nicht nur wichtig, dass die Bewertungskriterien die entsprechenden Phänomene erkennen, sondern dass sie sich robust gegenüber den anderen Phänomenen verhalten. Dazu werden alle Bewertungskriterien auf die Eingangsgrößen des Datensatzes D_{Sim} verwendet.

Die grau hinterlegten Felder der Tabellen 2.5, 2.6 und 2.7 entsprechen der Eignung der Bewertungskriterien, die zu ihnen gehörende Phänomene in der jeweiligen Ausprägung zu erkennen. Die übrigen Felder entsprechen der Robustheit der Bewertungskriterien gegenüber anderen Phänomenen.

Es wird davon ausgegangen, dass eine starke Ausprägung eines Phänomens zur Bewertung nahe 0 durch das entsprechende Kriterium führt, eine mittlere Ausprägung zur Bewertung nahe 0.5 und eine leichte Ausprägung zur Bewertung nahe 1 führt. Tabelle 2.4 zeigt für welche Wertebereiche die Bewertungskriterien gut (+), mäßig (o) und schlecht (-) geeignet sind.

Zielbewertung	$q \leq 0.25$	$0.25 < q < 0.75$	$0.75 \leq q$
1	-	o	+
0.5	-	+	-
0	+	o	-

Tabelle 2.4: Einteilung der Eignung für die Bewertungskriterien

Die quantitativen Ergebnisse der Bewertungskriterien finden sich in Anhang A.1.

Kriterium	Korr	Cluster	Konfig	Aus	Ortho
q_{Korr}	+	+	+	+	o
q_{Cluster}	+	+	-	+	+
q_{Konfig}	+	+	+	+	+
q_{Aus}	+	+	+	+	+
q_{Ortho}	+	o	+	+	+

Tabelle 2.5: Gute (+), mäßige (o) und schlechte (-) Eignung der Bewertungskriterien bei starker Ausprägung der Phänomene.

Kriterium	Korr	Cluster	Konfig	Aus	Ortho
q_{Korr}	+	+	+	+	o
q_{Cluster}	+	+	+	+	+
q_{Konfig}	+	+	+	+	+
q_{Aus}	+	+	+	+	+
q_{Ortho}	+	o	+	+	+

Tabelle 2.6: Gute (+), mäßige (o) und schlechte (-) Eignung der Bewertungskriterien bei mittlerer Ausprägung der Phänomene.

Kriterium	Korr	Cluster	Konfig	Aus	Ortho
q_{Korr}	+	+	+	+	+
q_{Cluster}	+	+	+	+	+
q_{Konfig}	+	+	+	+	+
q_{Aus}	+	+	+	+	+
q_{Ortho}	+	o	+	+	+

Tabelle 2.7: Gute (+), mäßige (o) und schlechte (-) Eignung der Bewertungskriterien bei schwacher Ausprägung der Phänomene.

Die unterschiedlichen Phänomene werden generell getrennt voneinander identifiziert. Bei einigen Kriterien kann es aber zu Fehlinterpretationen kommen.

Korrelationen werden bei Orthogonalität erkannt. Allerdings kann durch q_{Ortho} erkannt werden, ob es sich tatsächlich um Korrelationen oder um Orthogonalität handelt, da sich Korrelationen nicht auf q_{Ortho} auswirken.

Orthogonalität wird u.U. bei Clustern erkannt. Allerdings kann durch q_{Cluster} erkannt werden, ob es sich tatsächlich um Orthogonalität oder um Cluster handelt, da sich Orthogonalität nicht auf q_{Cluster} auswirkt.

Cluster werden bei Konfigurationen erkannt. Das Verhalten ist zu erwarten, da Konfigurationen eine Sonderform von Clustern sind.

Weiterhin werden die bekannten Benchmark-Datensätze, für die in Tabelle 2.1 bereits eine manuelle Einschätzung bezüglich der Phänomene erfolgt ist, mit Hilfe der Kriterien bewertet. Die Ergebnisse und Abweichungen von der manuellen Einschätzung finden sich in Tabelle 2.8.

Eine Bewertung wird markiert, wenn das Phänomen gemäß manueller Einschätzung auftritt, für die minimale Bewertung durch das entsprechende Kriterium allerdings $>= 0.5$ gilt oder wenn das Phänomen gemäß manueller Einschätzung nicht auftritt, für die minimale Bewertung durch das entsprechende Kriterium allerdings < 0.5 gilt.

Datensatz	Korr	Cluster	Konfig	Aus	Ortho
Abalone	ja	ja	ja	ja	nein
Airfoil Self Noise	ja	ja	ja	ja	ja
Boston Housing	ja	ja	ja	nein	ja
California Housing	ja	ja	ja	ja	ja
Computeractivity CPU	ja	ja	ja	ja	ja
Concrete	ja	ja	ja	ja	ja
Delta Ailerons	ja	nein	ja	ja	ja
Delta Elevators	nein	ja	ja	ja	ja
Red Wine Quality	ja	nein	ja	ja	ja
White Wine Quality	ja	nein	ja	ja	ja

Tabelle 2.8: Ergebnisse der Untersuchung der Benchmark-Datensätze mit Hilfe der vorgestellten Kriterien. Widersprüche zwischen manueller Bewertung und der Kriterien sind rot markiert.

Die Bewertung durch die Kriterien entspricht zum Großteil den manuellen Einschätzungen. Die auftretenden Fehleinschätzungen sind einerseits in den vorhandenen Ausreißern begründet, welche die Kriterien beeinflussen, andererseits hat die aktuelle Parametrierung der Kriterien keinen Anspruch an Allgemeingültigkeit.

Um die Anwendbarkeit der Bewertungskriterien auch bei großen Datensätzen festzustellen, wird untersucht wie die Implementierung der Kriterien skaliert. Tabelle 2.9 zeigt die benötigte Zeit der Bewertungskriterien für eine zunehmende Anzahl an Datentupeln.

N	q_{Korr}	q_{Cluster}	q_{Konfig}	q_{Aus}	q_{Ortho}
10	0.080	0.090	0.080	0.110	0.320
50	0.080	0.070	0.080	0.070	0.290
100	0.070	0.080	0.070	0.070	0.290
500	0.070	0.290	0.070	0.070	0.340
1000	0.080	1.160	0.080	0.090	0.430
5000	0.080	28.150	0.090	0.090	0.630
10000	0.080	111.680	0.110	0.120	0.860

Tabelle 2.9: Rechenzeiten der Bewertungskriterien in Sekunden

Für q_{Ortho} ist ein Anstieg der benötigten Zeit ab 1000 Datentupeln erkennbar. Einen starken Anstieg der benötigten Zeit zeigt q_{Cluster}. Grund dafür ist die Betrachtung der paarweisen Distanzen, die zeit- und rechenintensiv bestimmt werden. Um eine Skalierung der Kriterien für große Datensätze zu ermöglichen, werden bei den beiden kritischen Bewertungskriterien Stichproben der Größe 1000 der berücksichtigten Datentupel gezogen, sobald die gesamte Anzahl an Datentupeln größer wird als 1000. Die Bewertung von Datensätzen skaliert außerdem näherungsweise quadratisch mit der Anzahl an Eingangsgrößen, da alle zweidimensionalen Projektionen bewertet werden. Durch die Nutzung der Stichproben ergeben sich die Rechenzeiten aus Tabelle 2.10. Skalierungseffekte sind nicht mehr erkennbar. Es wird davon ausgegangen, dass 1000 Datentupel genügen, um in einem Streuwolkendiagramm die zu untersuchenden Effekte abzubilden. Eine Ausnahme stellt die Berechnung von q_{Aus} dar, bei der jedes einzelne Datentupel berücksichtigt wird, weshalb bei großen Datensätzen weiterhin Skalierungseffekte in Rechenzeit und vor allem Speicheraufwand existieren.

N	q_{Korr}	q_{Cluster}	q_{Konfig}	q_{Aus}	q_{Ortho}
10	0.080	0.070	0.080	0.080	0.300
100	0.070	0.080	0.070	0.070	0.300
1000	0.070	0.950	0.070	0.070	0.360
10000	0.070	1.170	0.080	0.110	0.430
100000	0.090	1.190	0.090	0.390	0.450

Tabelle 2.10: Rechenzeiten der Bewertungskriterien in Sekunden mit beschleunigter Erkennung von Cluster und Orthogonalität

2.5 Zusammenfassung

Einschränkungen der Datenqualität beim Entwurf von Regressionsmodellen wurden in eine allgemeine Taxonomie integriert. Zusätzlich wurde die Datenabdeckung eingeführt, die bekannte und neue Einschränkungen zusammenfasst. Für die zur Datenabdeckung gehörenden Einschränkungen wurden Bewertungskriterien vorgestellt, die eine einheitliche Bewertung von Datensätzen ermöglichen. Die vorgestellten Bewertungskriterien quantifizieren Phänomene eingeschränkter Datenabdeckung für ein- bzw. zweidimensionale Projektionen. Zusätzlich werden Möglichkeiten vorgestellt, die Bewertungskriterien für höherdimensionale Datensätze zu aggregieren. Für Datensätze mit vielen Eingangsgrößen bieten die Bewertungskriterien eine Möglichkeit, die visuelle Inspektion der Daten für den Anwender zu vereinfachen und zu beschleunigen. Besonders interessante Projektionen können dann automatisiert ausgewählt und dem Anwender präsentiert werden. Abseits vom Entwurf von Regressionsmodellen können die Bewertungskriterien auch generell zur Datenanalyse verwendet werden. So wird in [140] ein Tabletsystem vorgestellt, das versucht, durch individualisierte Inhalte die Betreuung von Menschen mit Demenz zu verbessern. Dazu werden alle Interaktionen mit dem Tablet protokolliert und in Datensätze umgewandelt [141]. Die Zusammenhänge zwischen aufgerufenen Inhalten oder entsprechender Themen mit automatisiert erkannten Emotionen oder manuellen Bewertungen können anhand der vorgestellten Kriterien in den jeweiligen Datensätzen quantifiziert werden.

3 Neue Bewertungskriterien für Modellqualität

3.1 Übersicht

Im vorangegangenen Kapitel wurden Bewertungskriterien eingeführt, um eine ungleichmäßige Datenabdeckung im Eingangsraum zu erkennen und zu quantifizieren. Bei einer herkömmlichen Modellbewertung kann eine ungleichmäßige Datenabdeckung zur Überschätzung der Modellqualität führen. Als Beispiel werden auf Basis der ungleichmäßig verteilten Eingangsdaten von $D_{\mathrm{Sim,Lern},1}$ zwei unterschiedlich komplexe MLP-Netze angelernt sowie jeweils eine 10-fache Kreuzvalidierung mit der Modellstruktur durchgeführt. Während das einfachere Modell aus Abbildung 3.1a den Zusammenhang gut abbildet, ist beim komplexeren Modell aus Abbildung 3.1b Überanpassung erkennbar.

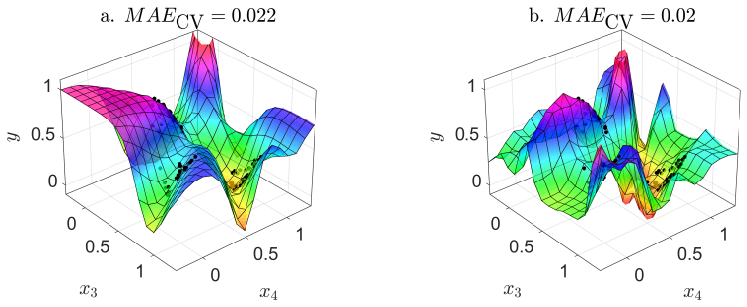

Abbildung 3.1: a. Einfaches und b. komplexes Modell bei niedriger Datenqualität sowie Kreuzvalidierungsergebnisse

Das Modell hat eine niedrige Generalisierungsfähigkeit. Der Kreuzvalidierungsfehler deutet allerdings auf die Überlegenheit des komplexeren Modells hin.

In einem automatisierten Entwurf wird in solchen Fällen ein überangepasstes Modell gewählt. Grund ist, dass Validierungsverfahren, die auf der randomisierten Entnahme von Datentupeln und der Bewertung von Residuen basieren, bei einer ungleichmäßigen Datenabdeckung u.U. keine Aussage über die Generalisierungsfähigkeit treffen können. Eine Möglichkeit konservativere Abschätzungen der Modellgüte, das heißt unter besonderer Berücksichtigung der Extrapolationsgüte, zu erreichen, ist die gezielte Entnahme ganzer Datencluster.

Im folgenden Kapitel wird die sogenannte Verlaufsvalidierung (VV) vorgestellt, um Modelle zu bewerten. Die entsprechenden Bewertungskriterien basieren nicht auf Residuen, sondern ausschließlich auf dem Funktionsverlauf von Modellen. Dadurch ist die Verlaufsvalidierung besonders für Datensätze niedriger Datenqualität geeignet sowie für nicht abgedeckte Bereiche, in denen aufgrund von nicht vorhandenen Daten keine Residuen bestimmt werden können. Sie bewertet damit die Extrapolationsfähigkeit eines Modells im Sinne von Abschnitt 1.2.1.

3.2 Bewertungskriterien

Die Verlaufsvalidierung entspricht einer automatisierten visuellen Bewertung von Modellen. Sie beruht auf der Annahme, dass die geschätzte Zielgröße guter Modelle in Bereichen zwischen zwei benachbarten Eingangsvektoren der Lerndaten $\mathbf{x}_a$ und $\mathbf{x}_b$ kein überschwingendes Verhalten zeigt, sondern näherungsweise linear bzw. monoton verläuft.

Ein deutlich nichtlineares Verhalten zwischen den Eingangsvektoren deutet auf Überanpassung und damit auf eine geringe Generalisierungs- und Prädiktionsfähigkeit im untersuchten Bereich hin. Die Verlaufsvalidierung untersucht nicht die Residuen, weshalb keine quantitativen Aussagen über die zu erwartende Prognosegenauigkeit möglich sind. Abbildung 3.2 vergleicht Beispiele von lokalen Funktionsverläufen. Als ideales Modell wird ein linearer Verlauf zwischen den Zielgrößen angesehen. Ebenso ist

jeder lineare Funktionsverlauf gemäß der Verlaufsvalidierung akzeptabel, unabhängig von der Distanz zum abzubildenden Verlauf[1].

Die Verlaufsvalidierung bewertet einen nichtlinearen Funktionsverlauf im Allgemeinen als ungeeignet, weil zwischen benachbarten Eingangsvektoren keine Lerndaten existieren, und der einfachste Verlauf zwischen den beiden Eingangsvektoren gemäß des Prinzips von Ockhams Rasiermesser [142] am geeignetsten ist. Ausnahmen, z.B. in Bereichen von Extremwerten, sind möglich, aber stellen nicht den Regelfall dar.

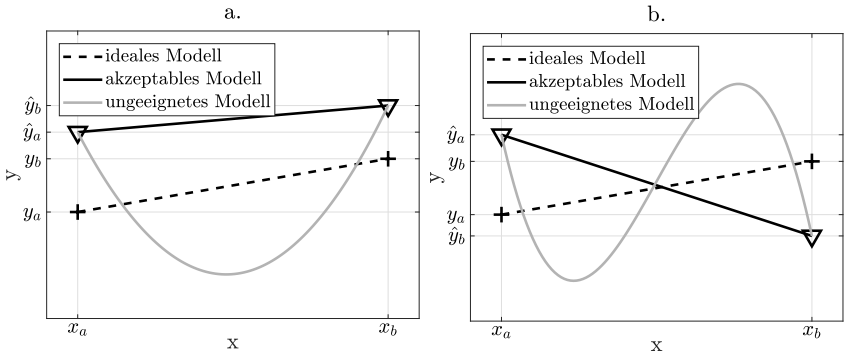

Abbildung 3.2: Lokale Bewertung von Modellen im Sinne der Verlaufsvalidierung

Es wird die Zielgröße eines Modells an ν Stellen $\hat{y}_{\mathrm{a,j}}, j = 1, ..., \nu$ betrachtet, die gleichmäßig verteilt auf der Gerade zwischen $\mathbf{x}_\mathrm{a}$ und $\mathbf{x}_\mathrm{b}$ im Eingangsraum liegen. $\hat{y}_{\mathrm{x}_a} = \hat{y}_{\mathrm{a},1}$ ist die vom Modell geschätzte Zielgröße am Punkt $\mathbf{x}_\mathrm{a}$. $\hat{y}_{\mathrm{x}_b} = \hat{y}_{a,\nu}$ ist die vom Modell geschätzte Zielgröße am Punkt $\mathbf{x}_\mathrm{b}$. Außerdem gilt

$$\hat{y}_{\max} = \max_{j=1,\ldots,\nu} \hat{y}_{\mathrm{a,j}} \text{ und} \tag{3.1}$$

$$\hat{y}_{\min} = \min_{j=1,\ldots,\nu} \hat{y}_{\mathrm{a,j}}. \tag{3.2}$$

[1] Abweichungen vom abzubildenden Verlauf werden bereits in Kreuzvalidierungen berücksichtigt.

Maximum und Minimum der Zielgröße im gesamten Lerndatensatz sind $y_{\max}$ und $y_{\min}$. Für sie gilt

$$y_{\max} \neq y_{\min}, \tag{3.3}$$

da sonst kein Regressionsproblem vorliegt.

In Anlehnung an [143, 144] werden die Bewertungskriterien $Q_{\mathrm{VV},1} \in [0,1]$, $Q_{\mathrm{VV},2} \in [0,1]$ und $Q_{\mathrm{VV},3} \in [0,1]$ berechnet, die den Modellverlauf lokal untersuchen, um eine Aussage über Überanpassung zu treffen. Dazu quantifizieren die Kriterien auf drei unterschiedliche Arten Nichtlinearität. Je kleiner die Werte der Bewertungskriterien ausfallen, desto eher liegt für das untersuchte Modell nichtlineares Verhalten, respektive Überanpassung im betrachteten Bereich des Eingangsraums vor.

Die einzelnen Bewertungskriterien berechnen sich folgendermaßen:

$$Q_{\mathrm{VV},1} = \begin{cases} 1, & \text{falls } \hat{y}_{\max} = \hat{y}_{\min} \\ \frac{|\hat{y}_{\mathbf{x}_a} - \hat{y}_{\mathbf{x}_b}|}{|\hat{y}_{\max} - \hat{y}_{\min}|} & \text{sonst.} \end{cases} \tag{3.4}$$

$Q_{\mathrm{VV},1}$ untersucht, ob im Funktionsverlauf zwischen $\mathbf{x}_{\mathrm{a}}$ und $\mathbf{x}_{\mathrm{b}}$ lokale Minima oder Maxima existieren. Dabei gilt

$$|\hat{y}_{\max} - \hat{y}_{\min}| \geq |\hat{y}_{\mathbf{x}_b} - \hat{y}_{\mathbf{x}_a}|. \tag{3.5}$$

Ein weiteres Kriterium bewertet anhand der maximalen lokalen Steigung

$$v_{\max,\mathrm{a,b}} = \max_{j=1,\ldots,(\nu-1)} (|\hat{y}_{\mathrm{a,j+1}} - \hat{y}_{\mathrm{a,j}}|) \tag{3.6}$$

den Funktionsverlaufs zwischen $\mathbf{x}_a$ und $\mathbf{x}_b$:

$$Q_{\mathrm{VV},2} = \begin{cases} 1, & \text{falls } v_{\max,\mathrm{a,b}} = 0 \\ \left(\frac{|\hat{y}_{\mathbf{x}_b} - \hat{y}_{\mathbf{x}_a}|}{\nu \cdot v_{\max,\mathrm{a,b}}} \right)^{\frac{1}{q_{\mathrm{VV},2}}}, & \text{sonst.} \end{cases} \tag{3.7}$$

Mit $q_{\mathrm{VV},2} > 1$ wird reguliert, wie empfindlich $Q_{\mathrm{VV},2}$ große Steigungen bestraft. Je größer $q_{\mathrm{VV},2}$ gewählt wird, desto toleranter ist $Q_{\mathrm{VV},2}$.

Die Abweichungen des Funktionsverlaufs und einer linearen Interpolation von $\hat{y}$ zwischen $\mathbf{x}_a$ und $\mathbf{x}_b$ werden von folgendem Kriterium bewertet:

$$Q_{\mathrm{VV},3} = \max \left(0, 1 - \frac{\max\limits_{j=1,\ldots,\nu} (|\hat{y}_{\mathrm{lin,j}} - \hat{y}_{\mathrm{j}}|)}{|\hat{y}_{\mathbf{x}_b} - \hat{y}_{\mathbf{x}_a}|} \right). \tag{3.8}$$

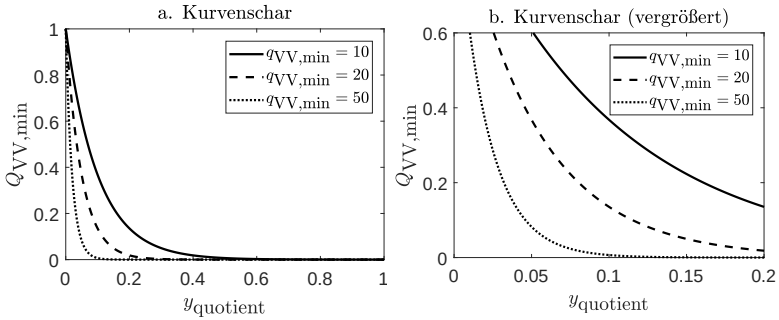

Abbildung 3.3: Kurvenschar und vergrößerter Bildausschnitt zur Findung eines geeigneten $q_{\mathrm{VV,min}}$

Die lineare Interpolation entspricht

$$\hat{y}_{\mathrm{lin,j}} = \frac{\hat{y}_{\mathbf{x}_b} - \hat{y}_{\mathbf{x}_a}}{\nu - 1} \cdot (j-1), j = 1, \ldots, \nu. \tag{3.9}$$

Da ein Modell bei minimalen Schwankungen des Funktionsverlaufs selbst mit lokalen Extremwerten zwischen benachbarten Eingangsvektoren die Generalisierungsfähigkeit nicht verliert, verhindert $Q_{\mathrm{VV,min}}$ eine ungerechtfertigte schlechte Bewertung. Das heißt bei einem größeren Quotienten

$$y_{\mathrm{quotient}} = \frac{\hat{y}_{\max} - \hat{y}_{\min}}{y_{\max} - y_{\min}} \tag{3.10}$$

werden schlechtere Bewertungen durch die Bewertungskriterien zugelassen:

$$Q_{\mathrm{VV,min}} = \exp(-q_{\mathrm{VV,min}} \cdot y_{\mathrm{quotient}}). \tag{3.11}$$

Um ein geeignetes $q_{\mathrm{VV,min}}$ zu finden, wird die Kurvenschar aus Abbildung 3.3 betrachtet.

Aufgrund des Kurvenverlaufs wird $q_{\mathrm{VV,min}} = 20$ gewählt. Damit ergibt sich ein angemessener Kompromiss zwischen einer hohen Fehlertoleranz bei geringen Schwankungen der geschätzten Zielgröße und einer geringen Fehlertoleranz bei großen Schwankungen der geschätzten Zielgröße.

Anschließend werden die drei Bewertungskriterien zu

$$Q_{\mathrm{VV}} = \max\left\{\min\left\{Q_{\mathrm{VV},1}, Q_{\mathrm{VV},2}, Q_{\mathrm{VV},3}\right\}, Q_{\mathrm{VV,min}}\right\} \in [0,1] \tag{3.12}$$

aggregiert.

Die Verlaufsvalidierung gibt keine globale Abschätzung der Modellqualität und erkennt keine Überanpassung an Messrauschen. Sie ermöglicht stattdessen die Bewertung der lokalen Zuverlässigkeit eines Modells ohne Daten im betrachteten Bereich zu benötigen. Das ist insbesondere bei künstlichen neuronalen Netzen von Vorteil, da ein wiederholtes Anlernen der gleichen Modellstruktur bei gleichen Lerndaten aufgrund zufällig initialisierter Startwerte und Multimodalität des Optimierungsproblems zu unterschiedlichen Modellen führen kann. Herkömmliche, fehlerbasierte Validierungsverfahren bieten zwar die Möglichkeit eine generelle globale Modellgüte zu bestimmen, die lokalen Prädiktionsgüten können allerdings schwanken. In solchen Fällen kann die Verlaufsvalidierung verwendet werden, um eine Abschätzung der lokalen Prädiktionsgüte zu erhalten. Weiterhin kann die Verlaufsvalidierung für N_{VV} unterschiedliche Bereiche des Eingangsraums aggregiert werden, um eine globale Aussage über das Modell zu ermöglichen.

3.3 Finden benachbarter Eingangsvektoren

Die Verlaufsvalidierung stellt eine Ergänzung zu fehlerbasierten Bewertungsmaßen dar und bringt deshalb vor allem in Bereichen weniger oder gar keiner Lerndaten einen Mehrwert für die Modellbewertung. Die Identifikation solcher Bereiche kann visuell oder mit Hilfe von Dichteschätzern o.ä. erfolgen. Da der abzubildende Zusammenhang für den Bereich jedoch nicht bekannt ist, werden Paare von Datentupeln der Lerndaten benötigt, deren Eingangsvektoren den zu untersuchenden Bereich eingrenzen.

Sei $\boldsymbol{x}_{\mathrm{p}} \notin \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ der Eingangsvektor, in dessen Umgebung ein Modell hinsichtlich Überanpassung untersucht werden soll, dann werden für

jede der p Eingangsgrößen zwei Datentupel $\boldsymbol{x}_{\mathrm{n}_1,j}$ und $\boldsymbol{x}_{\mathrm{n}_2,j}, j = 1, \ldots, p$ ausgewählt, für die

$$\mathrm{n}_1 = \underset{\substack{n=1,\ldots,N \\ x_{n,j} > x_{\mathrm{p},j}}}{\operatorname{argmin}} |\boldsymbol{x}_n - \boldsymbol{x}_{\mathrm{p}}| \text{ und} \tag{3.13}$$

$$\mathrm{n}_2 = \underset{\substack{n=1,\ldots,N \\ x_{n,j} < x_{\mathrm{p},j}}}{\operatorname{argmin}} |\boldsymbol{x}_n - \boldsymbol{x}_{\mathrm{p}}| \tag{3.14}$$

gilt. Durch die individuelle Betrachtung der Eingangsgrößen zur Findung geeigneter Paare von Datentupeln wird sichergestellt, dass die Verlaufsvalidierung tatsächlich den zu untersuchenden Bereich - nämlich diesseits und jenseits von $\boldsymbol{x}_p$ - berücksichtigt.

3.4 Validierung

Im folgenden Abschnitt wird die Verlaufsvalidierung anhand von D_{Sim} validiert. Vier MLP-Netze unterschiedlicher Struktur (4, 8, 15 und 30 Neuronen in der verdeckten Schicht) werden mit Hilfe von $D_{\mathrm{Sim,Lern,1}}$ angelernt, um y_1 abzubilden. Für drei Bereiche (A, B und C) des unvollständig abgedeckten Eingangsraums wird die Verlaufsvalidierung durchgeführt und mit den Abweichungen vom tatsächlichen Zusammenhang verglichen[2].

Abbildung 3.4 zeigt verschiedene Ansichten des Eingangsraums, die drei Stellen, für die lokale Prädiktionsgüten bestimmt werden, sowie drei Geradenabschnitte, die entsprechende Eingangsvektoren der Lerndaten verbinden und entlang derer der Verlauf der Zielgröße berücksichtigt wird[3].

Tabelle 3.1 vergleicht die Ergebnisse. Bei einer herkömmlichen Modellselektion wird beispielsweise das Modell gewählt, dass den geringsten Fehler über Validierungsdaten oder Kreuzvalidierungen erzielt. Gemäß Tabelle 3.1 zeigen die Modelle / Modellstrukturen 3 und 4 die besten Ergebnisse.

[2] Der reale Zusammenhang ist nur bekannt, da es sich um einen simulierten Benchmark-Datensatz handelt.

[3] Für jeden der Bereiche existieren gemäß (3.13) und (3.14) jeweils mehrere Geradenabschnitte, aus Gründen der Übersichtlichkeit wird allerdings nur jeweils ein Geradenabschnitt dargestellt und in der Validierung berücksichtigt.

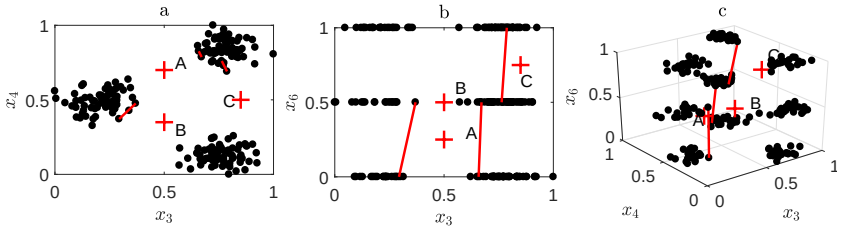

Abbildung 3.4: a. Projektion auf x_3 und x_4 b. Projektion auf x_3 und x_6 c. Gesamtdarstellung des dreidimensionalen Eingangsraums von D_{Sim} und die drei zu untersuchenden Bereiche.

Da es sich um einen simulierten Benchmark-Datensatz handelt, können die Prädiktionen in den ausgewählten Bereichen des Eingangsraums mit dem wahren Zusammenhang verglichen werden. Anhand der mittleren absoluten Fehler zeigt sich, dass die Modelle 3 und 4 in den untersuchten Bereichen deutlich stärker vom realen Zusammenhang abweichen als die anderen Modelle. Die Ergebnisse der Kreuzvalidierungen zeigen aber eine höhere Prädiktionsgüte als für die Modelle 1 und 2 an. Die Ergebnisse der Kreuzvalidierungen sind demnach für die untersuchten Bereiche nicht repräsentativ. Die Verlaufsvalidierung erkennt die Überanpassung in allen drei Bereichen und lässt sich damit als ergänzendes Maß in der Modellvalidierung verwenden.

Modell	MAE_{CV}	MAE_{A}	MAE_{B}	MAE_{C}	Q_{VV}:A	Q_{VV}:B	Q_{VV}:C
1–	0.036	0.122	0.043	0.052	0.747	0.574	0.769
2–	0.010	0.000	0.003	0.001	0.767	0.856	0.890
3–	0.004	0.028	0.071	0.033	0.248	0.026	0.098
4–	0.007	0.210	0.115	0.031	0.000	0.001	0.112

Tabelle 3.1: Vergleich der Ergebnisse der Verlaufsvalidierung für drei Bereiche des Eingangsraums mit den mittleren absoluten Fehlern vom tatsächlichen Zusammenhang für vier Modelle sowie den Ergebnissen von Kreuzvalidierungen.

Zur Veranschaulichung zeigt Abbildung 3.5 den Verlauf der Zielgröße der Modelle in den Bereichen A, B und C sowie den realen Zusammenhang von y_1.

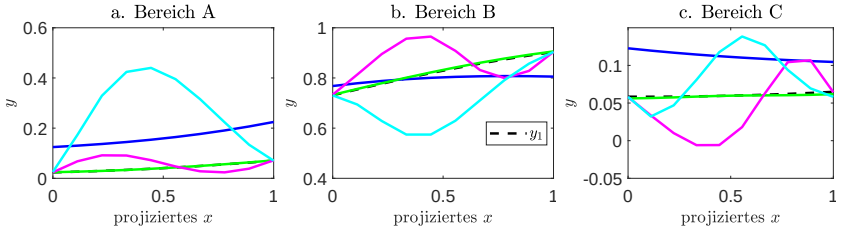

Abbildung 3.5: Verlauf der Modelle 1-4 für die Bereiche A, B und C sowie der für y_1 von D_{Sim} bekannte tatsächliche Zusammenhang. Die Legende für die Modelle findet sich in Tabelle 3.1.

Die hohen Modellfehler in den untersuchten Bereichen korrelieren mit deutlichen Nichtlinearitäten, wie sie besonders für die Modelle 3 und 4 auftreten und damit die Ergebnisse der Verlaufsvalidierung bestätigen.

Die Verlaufsvalidierung eignet sich demnach zur Abschätzung der lokalen Prädiktionsgüte bzw. der Erkennung von Überanpassung einzelner Modelle in Bereichen, für die keine Lerndaten vorliegen. Sie kann als Ergänzung herkömmlicher Validierungen verwendet werden, um beispielsweise die Plausibilität von Modellen in Extrapolationsbereichen zu überprüfen. Die Verlaufsvalidierung kann bereits im Modellentwurf erfolgen oder in der Modellanwendung. Bei einer Verlaufsvalidierung in der Modellanwendung sind die Lerndaten bei der Implementierung des Modells zu hinterlegen, was zu einem erhöhten Speicherbedarf führt.

Weiterhin wird die Eignung zur globalen Bewertung eines Modells überprüft. Die Verlaufsvalidierung kann in der Modellselektion überprüfen, ob das Modell mit dem geringsten Fehler über Validierungsdaten im Verhältnis zu einem einfacheren Modell Überanpassung aufweist.

Um den Einfluss der Datenqualität zu berücksichtigen, werden alle drei Lerndatensätze $D_{\text{Sim,Lern},1}$, $D_{\text{Sim,Lern},2}$ und $D_{\text{Sim,Lern},3}$ verwendet. Die tatsächliche Generalisierungsfähigkeit wird anhand der Testdaten bewertet, die gleichverteilt vorliegen und damit eine globale Prädiktionsgüte abbilden. Außerdem werden für die Modelle Verlaufsvalidierungen für zufällig verteilte Eingangsvektoren durchgeführt. Die Ergebnisse finden sich in Abbildung 3.6.

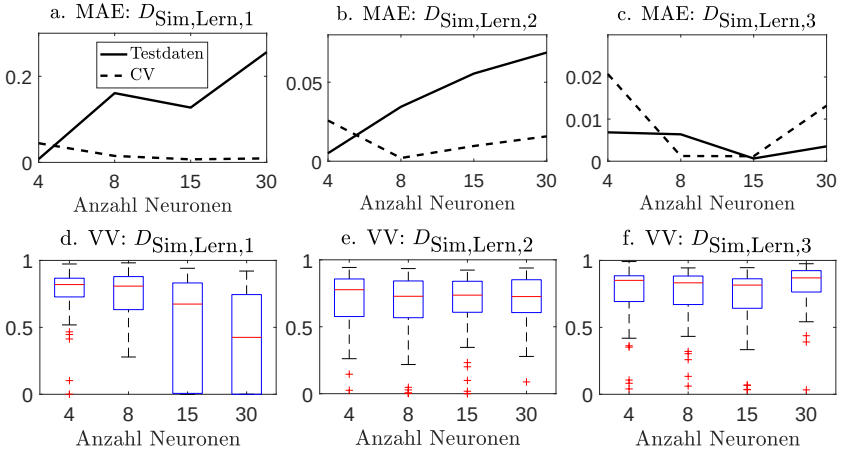

Abbildung 3.6: a.-c. zeigen die mittleren absoluten Fehler über der steigenden Modellkomplexität bei einer 10-fachen Kreuzvalidierung für $D_{\text{Sim,Lern,1}}$, $D_{\text{Sim,Lern,2}}$ und $D_{\text{Sim,Lern,3}}$ sowie die mittleren absoluten Fehler für die Testdaten ($D_{\text{Sim,Test}}$) bei Modellentwurf mit $D_{\text{Sim,Lern,1}}$, $D_{\text{Sim,Lern,2}}$ und $D_{\text{Sim,Lern,3}}$. d.-f. zeigen die Bewertungen durch die Verlaufsvalidierung über der steigenden Modellkomplexität bei Modellentwurf mit $D_{\text{Sim,Lern,1}}$, $D_{\text{Sim,Lern,2}}$ und $D_{\text{Sim,Lern,3}}$.

Für $D_{\text{Sim,Lern,2}}$ und $D_{\text{Sim,Lern,3}}$ lässt sich anhand der Testdaten keine Überanpassung erkennen. Der mittlere absolute Fehler nimmt für $D_{\text{Sim,Lern,2}}$ zwar mit steigender Komplexität des Modells zu, ist allerdings für alle Modelle geringer als 10%. Auch die Verlaufsvalidierung liefert konstant hohe Werte, was auf eine angemessene Modellkomplexität schließen lässt. Für $D_{\text{Sim,Lern,1}}$ nimmt der Kreuzvalidierungsfehler mit steigender Komplexität immer weiter ab. Für die Modelle 3 und 4 nimmt allerdings auch die Bewertung durch die Verlaufsvalidierung deutlich ab, was auf Überanpassung schließen lässt. Das wird durch den mittleren absoluten Fehler über Testdaten bestätigt, der mit steigender Komplexität deutlich zunimmt und bis etwa 30% steigt. Die deutlich überschätzte Generalisierungsfähigkeit durch die Kreuzvalidierungen ist auf die geringe Datenqualität des Datensatzes zurückzuführen. Allerdings liegt die Überanpassung nur lokal in den Bereichen der geringen Datendichte vor, für Bereiche hoher Datendichte

wird eine hohe Modellkomplexität eventuell benötigt. Unter Umständen ist die Komplexität auch in den Bereichen der geringen Datenqualität geeignet, es fehlen aber die zur Abbildung der Zusammenhänge erforderlichen Informationen. Eine Möglichkeit, Modelle mit lokal unterschiedlicher Modellkomplexität zu entwerfen, wird in Abschnitt 4.2 diskutiert.

Es zeigt sich, dass die Verlaufsvalidierung auch für eine globale Modellbewertung durch die Untersuchung an zufälligen Bereichen des Eingangsraums geeignet ist. Sie stellt allerdings keine hinreichende Bewertung eines Modells dar, sondern generiert zusätzliche Informationen zu herkömmlichen Bewertungen wie z.B. Kreuzvalidierungen.

3.5 Zusammenfassung

Es wurde mit der Verlaufsvalidierung eine Modellbewertung vorgestellt, die als Ergänzung von herkömmlichen Modellbewertungen zu verstehen ist. Da sie nicht auf Residuen basiert, kann die Verlaufsvalidierung keine Aussage über die Quantität zu erwartender Modellfehler leisten. Stattdessen wird der Funktionsverlauf von Modellen hinsichtlich ihrer Plausibilität untersucht, womit beispielsweise ein unerwartetes Versagen in Extrapolationsbereichen vorhergesagt werden kann.

4 Neue Verfahren zur Hypothesengenerierung und Parameterschätzung

4.1 Integration von Vorwissen

4.1.1 Übersicht

Bisher wurden neue Möglichkeiten aufgezeigt, im Entwurf von Regressionsmodellen sowohl die Daten- als auch die Modellqualität zu quantifizieren. In den folgenden Abschnitten werden neue Verfahren der Hypothesengenerierung und der Parameterschätzung (siehe Abschnitt 1.2.1) vorgestellt.

Um Modelle zu entwerfen, die z.B. auch in Bereichen ohne Daten robuste Prädiktionen generieren, kann Vorwissen über den abzubildenden Funktionsverlauf integriert werden. Es wird gezeigt, wie Vorwissen systematisch von einem Domänenexperten erfragt und in mathematische Formulierungen überführt sowie in den Modellentwurf integriert werden kann. Für eine Subgruppe von Vorwissen wird exemplarisch ein Verfahren zur Integration des Vorwissens in SVRs vorgestellt.

4.1.2 Systematische Erfassung von Vorwissen

Vorwissen über den Modellverlauf oder über eine der Ableitungen des Modellverlaufs kann sowohl in der Hypothesengenerierung als auch in der Parameterschätzung berücksichtigt werden. Vorwissen bleibt im Modellentwurf ungenutzt, wenn es nicht vom Domänenexperten erfragt wird. Das führt zum Verlust von de facto vorhandenen Informationen. Es wurde ein

Fragebogen [145] entworfen[1], um Vorwissen für Regressionsprobleme zu erfassen. Tabelle 4.1 zeigt einen Auszug des Fragebogens.

Vorwissen über	Nummer
Verhalten für $x \to \infty$	3

□ Liegt ein monotoner Verlauf mit gesättigtem Anstieg vor (a)?

□ Flacht die Kurve ab (a)?

□ Erhöht sich die Steigung bis zu einem konstanten Wert (b)?

□ Wird die Steigung unendlich groß (c)?

□ Liegt eine Polstelle mit/ohne Vorzeichenwechsel vor (c)?

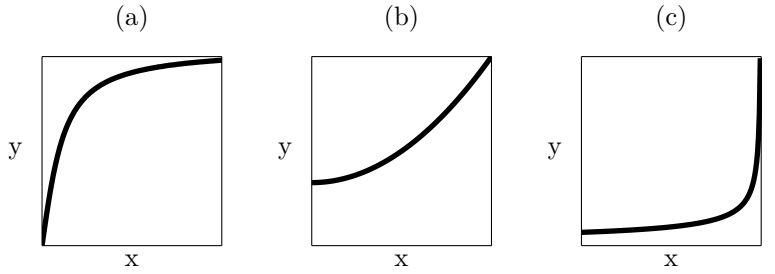

Tabelle 4.1: Auszug aus dem Fragebogen zum systematischen Erfassen von Vorwissen in Regressionsproblemen

Die Fragen sind absichtlich stellenweise redundant und mit beispielhaften Abbildungen verknüpft, um einen verständlichen Zugang zu bieten und möglichst viele Informationen vom Domänenexperten zu gewinnen und auf Konsistenz zu prüfen. Die Erfassung bezieht sich auf die folgenden Eigenschaften: Kenntnis der Modellstruktur, Symmetrie, Grenzwertverhalten, Monotonie, Umkehr der Monotonie, Stetigkeit und Differenzierbarkeit, Definitions- und Wertebereich, Positivität und Negativität, Unimodalität und Konvexität, Glattheit, Integralrestriktionen sowie der Datenqualität (Ausreißer, Multikollinearitäten, heterogene Verteilungen).

[1] Download auf www.researchgate.net/profile/wolfgang_doneit/publications

Kritisch ist zunächst die Überführung des erfragten Vorwissens, das als Beschreibung vorliegt, in seine mathematische Bedeutung.

Beschreibung	Bedeutung
zuverlässige Datentupel	$f(x_{\text{zuv}}) \stackrel{!}{=} y_{\text{zuv}}$
Wertebereich	$y_{\min} \stackrel{!}{\leq} f(x) \stackrel{!}{\leq} y_{\max}, \forall x \in \mathcal{D}$
Sättigung	$\lvert f(x) - y_{\text{Sätt}} \rvert \stackrel{!}{\leq} \varepsilon, \forall x > x_{\text{Sätt}}$
Verlauf steigend	$\frac{\partial f(x)}{\partial x} \stackrel{!}{\geq} 0, \forall x \in \mathcal{D}$
Verlauf fallend	$\frac{\partial f(x)}{\partial x} \stackrel{!}{\leq} 0, \forall x \in \mathcal{D}$
Symmetrie zur Y-Achse	$f(x) \stackrel{!}{=} f(-x), \forall x \in \mathcal{D}$
Symmetrie zum Ursprung	$f(x) \stackrel{!}{=} -f(-x), \forall x \in \mathcal{D}$
Symmetrisch zu bel. Punkt x_{Sym}	$f(x_{\text{Sym}} + \Delta x) \stackrel{!}{=} 2f(x_{\text{Sym}}) - f(x_{\text{Sym}} - \Delta x), \forall (x \pm \Delta x) \in \mathcal{D}$
Monotonie mit Sättigung ab $x_{\text{Sätt}}$	$\frac{\partial f(x)}{\partial x} \stackrel{!}{\geq} 0, \forall x \in \mathcal{D}$ und $\frac{\partial f(x)}{\partial x} \stackrel{!}{\leq} \varepsilon, \varepsilon \geq 0, \forall x > x_{\text{Sätt}}$
Steigung wird unendlich groß	$\frac{\partial^2 f(x)}{\partial x \partial x} \stackrel{!}{\geq} 0, \forall x \in \mathcal{D}$
Monotonieumkehr bei x_{Mon}	$\frac{\partial f(x)}{\partial x} \stackrel{!}{\geq} 0, \forall x < x_{\text{Mon}}$ und $\frac{\partial f(x)}{\partial x} \stackrel{!}{\leq} 0, \forall x > x_{\text{Mon}}$

Tabelle 4.2: Überführung von erfassten Anforderungen in mathematische Bedeutungen im univariaten Fall

Beispiele von Beschreibungen und der jeweiligen Bedeutung finden sich in Tabelle 4.2. Die Beispiele sind univariat, sind aber analog auf den multivariaten Fall übertragbar. Anschließend ist das Vorwissen mit Hilfe eines der drei Zugänge

- Strukturansätze,
- Restriktionen oder

- Strafterme

in den Modellentwurf zu integrieren. Aufgrund der vielen Möglichkeiten, das Vorwissen in Abhängigkeit der bevorzugten Modellfamilie, der Metaparametrierung oder des verwendeten Lösungsalgorithmus bzw. existierender Implementierungen zu integrieren, kann für diesen Schritt keine ausführliche Anleitung erfolgen. Tabelle 4.3 bewertet die Eignung einiger Modellfamilien zur Integration von Vorwissen.

	Polynom	MLP-Netze	SVR
Strukturwahl	✓	✗	✗
Restriktionen	✓	✗	✓
Strafterme	✓	✓	✓

Tabelle 4.3: Praktikable Zugänge zur Integration von Vorwissen unterschiedlicher Modellfamilien

Polynommodelle ermöglichen aufgrund der hohen Interpretierbarkeit eine gezielte Strukturwahl zur Integration von Vorwissen. Da die Parameterschätzung ein lineares Problem darstellt, können Restriktionen mit bereits implementierten Softwarelösungen integriert werden. Strafterme erfordern die Modifikation der Zielfunktion, die aber häufig von bereits bestehenden Lösern beherrschbar bleibt.

Bei MLP-Netzen erfolgt i.d.R. keine explizite Wahl der Modellstruktur, sondern eine implizite Wahl durch einen Komplexitätsgrad (Anzahl an Neuronen in verdeckten Schichten, Zustandsfunktionen,...). Die analytische Beschreibung des Regressionsmodells bleibt dem Anwender unbekannt. Eine Integration von Vorwissen im Sinne von Eigenschaften des Modellverlaufs erfolgt daher selten über eine manuelle Strukturwahl. In [146] werden allerdings zwei Strategien vorgestellt, um Vorwissen einerseits in der Modellstruktur und andererseits als Restriktionen für die Parameter innerhalb der Struktur zu integrieren. Für die Modellierung dynamischer Systeme werden in [147] Konzepte zur Integration von Vorwissen vorgestellt. Ein Verfahren zur restringierten Parameterschätzung bei ANNs wird auch in [148] vorgestellt. Die Verwendung von Straftermen ist denkbar, eine detaillierte Beschreibung kann aufgrund der unterschiedlichen Lernverfahren der Parameterschätzung von MLP-Netzen jedoch nicht erfolgen.

SVRs können beliebig nichtlineare Zusammenhänge abbilden und stellen im Falle eines Gaußkernels eine Superposition von Gaußfunktionen dar. Die Modellstruktur ergibt sich nach einer Metaparametrierung implizit in der Parameterschätzung, weshalb eine Strukturwahl zur Integration von Vorwissen nicht praktikabel scheint. Trotz der hohen Modellkomplexität resultiert die Parameterschätzung in einem quadratischen Problem, wodurch Restriktionen ähnlich wie bei Polynommodellen verwendet werden können [24, 149, 150]. Die Parameterschätzung erfolgt bei SVRs nicht durch Minimierung der Fehlerquadrate, sondern durch Minimierung des ε-insensitiven Fehlers. Dadurch kann ein Sonderfall von Straftermen bei Verwendung gängiger Löser integriert werden: Zur Integration von zuverlässigen Datentupeln, Wertebereichen oder Sättigung lassen sich punktweise Anforderungen der Art

$$f(\boldsymbol{x}_{\mathrm{R},i}) \overset{!}{=} y_{\mathrm{R},i} \text{ oder } y_{\mathrm{R},i} - \Delta y_{\mathrm{R},i} \overset{!}{\leq} f(\boldsymbol{x}_{\mathrm{R},i}) \overset{!}{\leq} y_{\mathrm{R},i} + \Delta y_{\mathrm{R},i}, \quad i = 1, \ldots, N_{\mathrm{R}} \tag{4.1}$$

formulieren. Beim Zugang durch Strafterme entsprechen die Anforderungen künstlich erzeugten Datentupeln, sogenannten virtuellen Datentupeln, mit $\varepsilon = \Delta y_{\mathrm{R}}$. Der Ansatz der virtuellen Datentupel (engl. *virtual samples method*, VSM) wurde im Bereich der Bildklassifikationen entwickelt [151]. Dazu wurden die vorhandenen Bilder manuell rotiert oder skaliert und zusätzlich zu den aus den ursprünglichen Bildern erzeugten Datentupeln in den Lerndatensatz eingefügt. Bislang fehlt allerdings eine systematische Vorgehensweise, um virtuelle Datentupel zur Integration von Vorwissen in SVRs zu verwenden.

4.1.3 Vorwissen in Stützvektor-Regressionen mit virtuellen Datentupeln

Übersicht

Eine SVR minimiert die Residuen über den Lerndaten, die größer als ein definierter Wert ε sind. Dabei handelt es sich um einen Metaparameter, der Überanpassung und den daraus folgenden Verlust der Generalisierungsfähigkeit des Modells verhindern kann. Weiterhin wird ein zweiter Metaparameter C verwendet, um die Modellkomplexität zu regularisieren

und ein möglichst einfaches Modell zu erhalten. Der zweite Metaparameter stellt in der Parameterschätzung einen Kompromiss aus Genauigkeit und Modellkomplexität ein, wobei die zu erreichende Genauigkeit im Sinne einer Approximation der Lerndaten durch den ersten Metaparameter bestimmt wird. Die tatsächliche Modellgüte, d.h. die Prädiktionsgüte und Generalisierungsfähigkeit, ist von beiden Metaparametern abhängig, weshalb die Metaparametrierung einer SVR ein Mehrvariablenproblem ist. Hinzu kommt außerdem die Metaparametrierung einer geeigneten Kernelfunktion, um die generelle Flexibilität eines SVR-Modells festzulegen.

Die Verfahren zur Parameterschätzung einer herkömmlichen SVR wurden für die ausschließliche Verwendung der beiden genannten Metaparameter ε und C entwickelt. Es existieren allerdings Erweiterungen zur Verwendung individueller Metaparameter ε_n [152], bzw. C_n [153]. Dabei bestimmt ε_n die zu erzielende Genauigkeit des Modells für $\boldsymbol{x}_n$ und C_n wichtet den Strafterm für ein Residuum größer als ε für $\boldsymbol{x}_n$ im Optimierungsproblem der Parameterschätzung. Während die Bestimmung geeigneter Metaparameter für herkömmliche Lerndatensätze z.B. mit Hilfe von Kreuzvalidierungen zur Minimierung des Modellfehlers erfolgt, repräsentiert bei der Verwendung von virtuellen Datentupeln zur Integration von Vorwissen ε_n die Lage und Genauigkeit von Vorwissen und C_n die Priorität, d.h. den Kompromiss zwischen Abbildung der herkömmlichen Daten und dem Vorwissen. Es ergeben sich vier Ansätze, um die Metaparameter einer SVR zu wählen:

- Ansatz 1: Einheitliches ε, einheitliches C,
- Ansatz 2: Einheitliches ε, individuelle C_n,
- Ansatz 3: Individuelle ε_n, einheitliches C und
- Ansatz 4: Individuelle ε_n, individuelle C_n.

Eine Untersuchung der Vor- und Nachteile der Ansätze hinsichtlich unterschiedlicher Gütekriterien fehlt bislang. Zudem wurde Ansatz 4 in der Literatur bislang gar nicht erwähnt. Im folgenden Abschnitt werden die Ansätze hinsichtlich ihrer Eignung zur Integration von Vorwissen anhand unterschiedlicher Gütekriterien (Modellgenauigkeit über Testdaten, benötigte Zeit zur Parameterschätzung, Aufwand zur Auswahl der Metaparameter) verglichen. Ansatz 4 scheint besonders für die Integration von Straftermen für Vorwissen gemäß (4.1) geeignet. Die Genauigkeit der virtuellen Datentupel kann unabhängig von den regulären Datentupeln

gewählt werden, was zu einer hohen Genauigkeit für das Vorwissen bei gleichzeitigem Erhalt der Generalisierungsfähigkeit des Modells führen kann. Individuell gewichtete virtuelle Datentupel können das Einhalten der gewählten Genauigkeit fordern.

Quadratische Löser haben häufig einen hohen Rechen- und Speicheraufwand. Deshalb wurden effiziente Lösungsalgorithmen entwickelt, z.B. der SMO-Algorithmus. Der SMO-Algorithmus ist schneller und weniger speicherintensiv als herkömmliche quadratische Löser oder der Chunking-Algorithmus [154]. Allerdings existiert kein SMO-Algorithmus für Ansatz 4. Deshalb wird eine einfach zu implementierende Erweiterung des SMO-Algorithmus für Ansatz 4 vorgestellt, die generalisierte sequentielle minimale Optimierung (engl. *generalized sequential minimal optimization*, GSMO).

Stützvektor-Regressionen

Modelle von SVRs haben die Form

$$f(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{\phi}(\boldsymbol{x}) + b \text{ mit } \boldsymbol{w} \in \mathcal{F}, b \in \mathcal{R}, \tag{4.2}$$

in der $\boldsymbol{\phi}(\cdot)$ eine implizit definierte Transformation $\mathcal{X} \rightarrow \mathcal{F}$ in Abhängigkeit einer Kernelfunktion k ist. Häufige Verwendung findet z.B. der gaußsche Kernel

$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left(-\frac{1}{2}(\boldsymbol{x}_i - \boldsymbol{x}_j)^T \boldsymbol{\Sigma}^{-\frac{1}{2}} (\boldsymbol{x}_i - \boldsymbol{x}_j)\right) \tag{4.3}$$

bzw. vereinfacht mit einheitlicher Varianz

$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left(-\frac{1}{2}\frac{||\boldsymbol{x}_i - \boldsymbol{x}_j||^2}{\sigma^2}\right) \tag{4.4}$$

mit

$$\boldsymbol{\phi}^T(\boldsymbol{x}_i) \cdot \boldsymbol{\phi}(\boldsymbol{x}_j) = k(\boldsymbol{x}_i, \boldsymbol{x}_j) = k_{ij}. \tag{4.5}$$

Eine äquivalenter Beschreibung der Modelle ist

$$f(\boldsymbol{x}) = \sum_{n=1}^{N} \theta_n k(\boldsymbol{x}, \boldsymbol{x}_n) + b, \tag{4.6}$$

deren Zusammenhang mit (4.2) durch die primäre und duale Problemformulierung gegeben ist, die später beschrieben wird. Damit werden zur Anwendung des Modells ausschließlich die Eingangsvektoren $\boldsymbol{x}_n$ benötigt, für deren zu schätzenden Parameter $\theta_n \neq 0$ gilt. Solche Eingangsvektoren werden Stützvektoren genannt.

Ziel der SVR ist das Finden einer Funktion $f(\boldsymbol{x})$, die für die Lerndaten bei $\boldsymbol{x}_n$ maximal um ε von y_n abweicht und dabei eine möglichst geringe Modellkomplexität im Merkmalsraum $\mathcal{F}$ aufweist. Um unlösbare Problemstellungen zu vermeiden, werden Abweichungen größer als ε erlaubt, aber in der Zielfunktion der Parameterschätzung bestraft. Der Metaparameter ε beschreibt daher den insensitiven Bereich einer Lossfunktion.

In der Modellbeschreibung (4.6) werden alle Eingangsvektoren $\boldsymbol{x}_n$ mit $\theta_n \neq 0$ Stützvektoren genannt, für die außerdem $|f(\boldsymbol{x}_n) - y_n| \geq \varepsilon$ gilt. Ausschließlich die Stützvektoren bestimmen den Funktionsverlauf von $f(\boldsymbol{x})$, d.h. je weniger Stützvektoren eine SVR beinhaltet, desto weniger Zeit und Speicherplatz bedarf es, um das Modell zu speichern oder anzuwenden.

Entsprechend der Modellbeschreibungen (4.2) und (4.6) existieren zwei Formulierungen für die Zielfunktionen: Es gibt zum einen die primäre Formulierung [152, 153]

$$\left\{\hat{\boldsymbol{w}}, \hat{b}\right\} = \underset{\boldsymbol{w},b,\boldsymbol{\xi},\boldsymbol{\xi}^*}{\operatorname{argmin}} \quad \underbrace{\frac{1}{2}||\boldsymbol{w}||^2}_{\text{Modellkomplexität}} + C \underbrace{\sum_{n=1}^{N}(\xi_n + \xi_n^*)}_{\text{Genauigkeit}} \tag{4.7a}$$

$$\text{s.t.} \quad \begin{aligned} y_n - \boldsymbol{w}^T\boldsymbol{\phi}(\boldsymbol{x}_n) - b &\leq \varepsilon + \xi_n \\ \boldsymbol{w}^T\boldsymbol{\phi}(\boldsymbol{x}_n) + b - y_n &\leq \varepsilon + \xi_n^* \\ \xi_n, \xi_n^* &\geq 0, \end{aligned} \tag{4.7b}$$

wobei der Regularisierungsparameter C den Kompromiss zwischen Modellkomplexität und Genauigkeit der Approximation der Lerndaten bestimmt.

Zum anderen gibt es die duale Formulierung

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \; \varepsilon \sum_{n=1}^{N} |\theta_n| - \sum_{n=1}^{N} \theta_n y_n + \frac{1}{2} \sum_{n=1}^{N} \sum_{j=1}^{N} \theta_n \theta_j k_{nj} \tag{4.8a}$$

$$\text{s.t.} \; \sum_{n=1}^{N} \theta_n = 0 \text{ und } \theta_n \in [-C, C] \,, \tag{4.8b}$$

in die (4.7) mit Hilfe von Lagrange-Multiplikatoren und (4.5) überführt werden kann. In der dualen Formulierung wird die Berechnung von b nach der Schätzung von $\boldsymbol{\theta}$ durchgeführt, in dem die Karush-Kuhn-Tucker-Bedingungen verwendet werden [153, 155]. Beide Formulierungen beschreiben konvexe Probleme. Das folgt für die zweite Formulierung zum einen aus der Mercer-Bedingung, die für die gegebene Kernelfunktion erfüllt ist und eine positiv semidefinite Matrix $\boldsymbol{K} = [k_{nj}]_{1,1}^{N,N}$ impliziert, zum anderen aus der Bedingung, dass die Summe zweier konvexer Funktionen ebenfalls konvex ist. Aufgrund der ausschließlich linearen Randbedingungen in (4.8) kann das strenge Dualitätstheorem angewendet werden und es existiert keine Dualitätslücke. Daraus folgt, dass die Minima beider Formulierungen identisch sind.

Sequentielle minimale Optimierung

Die sequentielle minimale Optimierung (engl. *sequential minimal optimization*, SMO) als Verfahren zur Parameterschätzung wurde ursprünglich für die binäre Klassifikation mit Hilfe von Stützvektor-Maschinen entwickelt und für Stützvektor-Regressionen erweitert. Eine detaillierte Beschreibung des SMO-Algorithmus findet sich in [156]. Entsprechende Varianten, um individuelle Metaparameter C_n für Ansatz 2) oder ε_n für Ansatz 3) verwenden zu können, werden in [153] und [157] vorgestellt.

Generell handelt es sich bei SMO um einen Zwei-Koordinaten-Relaxationsalgorithmus [158], der auf (4.8) angewendet wird. Er besteht aus drei Schritten, die solange wiederholt werden, bis der Algorithmus konvergiert:

1. Wahl von zwei geeigneten Parametern,

2. analytisches Lösen des Subproblems und
3. Anpassung des Parameters b.

Die meisten Erweiterungen des SMO betreffen den ersten Schritt, d.h. die Heuristiken zur Wahl geeigneter Parameter eines Optimierungsschritts, da er ausschlaggebend für die Konvergenzgeschwindigkeit ist. Eine Modifikation des zweiten Schritts, um die Realisierung von Ansatz 4 zu ermöglichen, fehlt bislang. Das bedeutet, dass bisherige SMO-Algorithmen nicht auf SVRs mit individuellen C_n und ε_n angewandt werden können.

Realisierung von virtuellen Datentupeln in den Ansätzen 1-4

Im Folgenden wird beschrieben, wie virtuelle Datentupel des Typs $f(\boldsymbol{x}_\mathrm{R}) = y_\mathrm{R}$ mit einer geforderten Genauigkeit $\varepsilon_{\mathrm{R,gef}}$ verwendet werden. Zunächst werden für die regulären Datentupel geeignete Metaparameter ε_reg, C_reg und σ z.B. mit Hilfe einer Gittersuche (engl. *grid search*) identifiziert. In den Ansätzen 3 und 4 werden die geforderten Genauigkeiten $\varepsilon_{\mathrm{R},i,\mathrm{gef}}, i = 1,\ldots,N_\mathrm{R}$ für die virtuellen Datentupel verwendet. In den Ansätzen 1 und 2 wird ein ε_R für das Vorwissen als das Minimum aller geforderten Genauigkeiten $\varepsilon_{\mathrm{R},i,\mathrm{gef}}$ gewählt. ε_R ist häufig kleiner als ε_reg. Das heißt, für das einheitliche ε muss $\varepsilon = \min\{\varepsilon_\mathrm{R}, \varepsilon_\mathrm{reg}\}$ gelten. Danach werden C_reg und σ für das festgelegte ε neu bestimmt.

In den Ansätzen 1 und 3 wird für das einheitliche C der entsprechende Wert C_reg gewählt und die virtuellen Datentupel werden dupliziert, um die Einhaltung des Vorwissens zu erzwingen. In den Ansätzen 2 und 4 wird die Einhaltung durch individuelle Werte $C_{\mathrm{R},i}$ der virtuellen Datentupel erreicht, deren Verhältnis zu C_reg der jeweiligen Anzahl an Duplikaten in den Ansätzen 1 und 3 entspricht.

Je größer die Anzahl an Duplikaten für ein virtuelles Datentupel ist, desto höher ist seine Wichtung. Nachteil ist die unbekannte Anzahl an Duplikaten für die virtuellen Datentupel, die benötigt werden, um die Randbedingungen zu erfüllen. Deshalb wird, beginnend mit der stärksten Verletzung der Randbedingungen, die Anzahl der Duplikate sukzessive erhöht, bis alle Randbedingungen erfüllt sind. Ein weiterer Nachteil ist der hohe Aufwand bezüglich Speicher und Rechenzeit, der mit einer großen Anzahl an zusätzlichen virtuellen Datentupeln einhergeht. Im Gegensatz

dazu führen individuelle C_n in den Ansätzen 2 und 4 zu keinem solchen gesteigerten Aufwand. Eine gängige Vorgehensweise zur Wahl geeigneter C_n ist die Wahl eines hohen Werts für virtuelle Datentupel im Verhältnis zu C_{reg}. Daraus folgt, dass Ansatz 4 den besten Kompromiss darstellt zwischen Flexibilität und Parametrierungs-, Rechen- und Speicheraufwand. Ob die Flexibilität in ingenieurwissenschaftlichen Anwendungen notwendig ist, wird später anhand von Modellgenauigkeit und Modellkomplexität untersucht.

Generalisierte sequentielle minimale Optimierung

Die generalisierte sequentielle minimale Optimierung (engl.: *generalized sequential minimal optimization*, GSMO) stellt eine Modifikation des SMO dar, um Ansatz 4 zu lösen. Vom gewöhnlichen SMO-Algorithmus unterscheidet er sich zunächst nur in Schritt 2). Damit wird im Folgenden die analytische Lösung eines Zwei-Variablen-Problems (θ_u,θ_v) gesucht, wobei u und v so gewählt sind, dass $\boldsymbol{x}_u \neq \boldsymbol{x}_v$ gilt und alle anderen Parameter $\theta_n | n = \{1, \ldots, N\} \setminus \{u, v\}$ konstant sind.

Dazu wird die duale Formulierung (4.8) angepasst zu

$$\left\{\hat{\theta}_u, \hat{\theta}_v\right\} = \underset{\theta_u, \theta_v}{\operatorname{argmin}} \sum_{n=1}^{N} \varepsilon_n |\theta_n| - \sum_{n=1}^{N} \theta_n y_n + \frac{1}{2} \sum_{n=1}^{N} \sum_{j=1}^{N} \theta_n \theta_j k_{nj} \tag{4.9a}$$

$$\sum_{n=1}^{N} \theta_n = 0, \theta_u \in [-C_u, C_u], \theta_v \in [-C_v, C_v]. \tag{4.9b}$$

Mit Hilfe der Gleichheitsbedingung wird θ_u eliminiert, indem die parametrische Darstellung

$$\boldsymbol{\theta}(\theta_v) = \boldsymbol{\theta}^* + (\boldsymbol{e}_v - \boldsymbol{e}_u)(\theta_v - \theta_v^*), \tag{4.10}$$

gewählt wird, wobei $\boldsymbol{e}_k$ den Einheitsvektor (k-te Spalte der Identitätsmatrix) darstellt. $\boldsymbol{\theta}^*$ ist ein beliebiger Vektor, der die Bedingungen erfüllt. Im Algorithmus wird der Parametervektor des letzten Optimierungsschritts als $\boldsymbol{\theta}^*$ angenommen. Der zweite Summand in (4.10) beschreibt den Anstieg zum neuen Parameterwert θ_v und den für die Randbedingung notwendigen gleichzeitigen Abstieg zum neuen Parameterwert θ_u. Durch (4.10) und

$-C_u \leq \theta_u \leq C_u$ aus (4.9b) wird das zulässige Intervall für θ_v beschränkt auf

$$\theta_v \in [-C_v, C_v] \cap [-C_u + \theta_u^* + \theta_v^*, C_u + \theta_u^* + \theta_v^*] \tag{4.11}$$

bzw. in kompakter Darstellung mit $s = \theta_u^* + \theta_v^*$ auf

$$\underbrace{\max\{-C_v, -C_u + s\}}_{=\underline{\theta}_v} \leq \theta_v \leq \underbrace{\min\{C_v, C_u + s\}}_{=\overline{\theta}_v}. \tag{4.12}$$

Das Einsetzen von (4.10) in (4.9) resultiert im Optimierungsproblem

$$\hat{\theta}_v = \underset{\theta_v}{\operatorname{argmin}} \underbrace{\boldsymbol{\varepsilon}^T |\boldsymbol{\theta}(\theta_v)| - \boldsymbol{y}^T \boldsymbol{\theta}(\theta_v) + \frac{1}{2} \boldsymbol{\theta}^T(\theta_v) \boldsymbol{K} \boldsymbol{\theta}(\theta_v)}_{W(\theta_v)}. \tag{4.13}$$

Um die analytische Lösung von (4.13) zu finden, wird die symmetrische Ableitung von W in Abhängigkeit von θ_v untersucht. Sie ist gegeben durch

$$W'(\theta_v) = \varepsilon_v \operatorname{sgn}(\theta_v) - \varepsilon_u \operatorname{sgn}(\theta_u^* + \theta_v^* - \theta_v) - y_v + y_u + (\boldsymbol{\theta}')^T \boldsymbol{K} \boldsymbol{\theta}. \tag{4.14}$$

Mit $\boldsymbol{\theta}'(\theta_v) = \boldsymbol{e}_v - \boldsymbol{e}_u$ und (4.10) folgt

$$\begin{aligned} W'(\theta_v) &= \underbrace{\varepsilon_v \operatorname{sgn}(\theta_v) + \varepsilon_u \operatorname{sgn}(\theta_v - s) - y_v + y_u + (\boldsymbol{e}_v - \boldsymbol{e}_u)^T \boldsymbol{K} \boldsymbol{\theta}^*}_{\tilde{W}'(\theta_v)} \\ &\quad + \underbrace{(\boldsymbol{e}_v - \boldsymbol{e}_u)^T \boldsymbol{K} (\boldsymbol{e}_v - \boldsymbol{e}_u)}_{=k_{uu}+k_{vv}-2k_{uv}=:\eta} (\theta_v - \theta_v^*) \\ &= \tilde{W}'(\theta_v) + \eta(\theta_v - \theta_v^*) \quad \text{für } \theta_v \neq 0 \text{ und } \theta_v \neq s. \end{aligned} \tag{4.15}$$

Die Ableitung ist abschnittsweise linear und weist zwei Unstetigkeiten im möglichen Intervall auf, die durch die Signum-Funktionen entstehen. Die linearen Teilabschnitte haben eine einheitliche Steigung $\eta > 0$ (Mercer-Bedingung und $\boldsymbol{x}_u \neq \boldsymbol{x}_v$). Unter Berücksichtigung von $\theta_v \in [\underline{\theta}_v, \overline{\theta}_v]$ lässt sich (4.13) mit Hilfe einer Kurvendiskussion der Ableitung lösen. Dafür werden die linken und rechten Grenzwerte der Unstetigkeiten benötigt:

$$W'(0^{\pm}) = \begin{cases} W'(0) \pm \varepsilon_v & s \neq 0 \\ W'(0) \pm (\varepsilon_v + \varepsilon_u) & s = 0 \end{cases} \tag{4.16a}$$

$$W'(s^{\pm}) = \begin{cases} W'(s) \pm \varepsilon_u & s \neq 0 \\ W'(s) \pm (\varepsilon_v + \varepsilon_u) & s = 0. \end{cases} \tag{4.16b}$$

Weiterhin wird das Vorzeichenwechsel-Kriterium anstatt der Gradientenbedingung verwendet, um den die Zielfunktion minimierenden Wert für θ_v in einer der Unstetigkeiten zu beschreiben.

Es wird untersucht, in welchem der linearen Teilabschnitte bzw. in welcher der Unstetigkeiten der Vorzeichenwechsel der Ableitung geschieht. Gemäß der Untersuchung wird eine allgemeine analytische Lösung parametriert. In Tabelle 4.4 beschreiben die Fälle 1-3 den Vorzeichenwechsel in einem der linearen Teilabschnitte und die Fälle 4 und 5 den Vorzeichenwechsel in einer der Unstetigkeiten.

Fall	$W'(\theta_v) = \tilde{W}'(a) + \eta(\theta_v - \theta_v^*)$
$1: 0 < \min\left\{W'(0^-), W'(s^-)\right\}$	$a = \min\{0, s\} - 1$
$2: \min\left\{W'(0^+), W'(s^+)\right\} < 0$ $< \max\left\{W'(0^-), W'(s^-)\right\}$	$a = \frac{1}{2}s$
$3: \max\left\{W'(0^+), W'(s^+)\right\} < 0$	$a = \max\{0, s\} + 1$
Fall	**Parametrierung**
$4: W'(0^-) \leq 0 \leq W'(0^+)$	$\tilde{W}'(a) := \eta\theta_v^*$
$5: W'(s^-) \leq 0 \leq W'(s^+)$	$\tilde{W}'(a) := -\eta\theta_u^*$

Tabelle 4.4: Fälle des Vorzeichenwechsels in linearen Teilabschnitten und Unstetigkeiten sowie zugehörige Parametrierungen

Aus der Gradientenbedingung $W'(\hat{\theta}_v) = 0$ folgt für die Fälle 1-3 (lineare Teilabschnitte) $\hat{\theta}_v = \theta_v^* - \frac{1}{\eta}\tilde{W}'(a)$ und aus dem Vorzeichenwechsel-Kriterium folgt für Fall 4 $\hat{\theta}_v = 0$ und für Fall 5 $\hat{\theta}_v = s$, falls die Lösung im zulässigen Bereich liegt. Andernfalls wird sie auf die zugehörige Grenze $\underline{\theta}_v$ bzw. $\overline{\theta}_v$ projiziert. Die allgemeine analytische Lösung des Optimierungsproblems bezüglich (4.13) unter der Bedingung (4.12) zur Lösung von (4.9) ergibt sich zu

$$\hat{\theta}_v = \min\left\{\max\left\{\theta_v^* - \frac{1}{\eta}\tilde{W}'(a), \underline{\theta}_v\right\}, \overline{\theta}_v\right\} \tag{4.17a}$$

$$\hat{\theta}_u = s - \hat{\theta}_v \tag{4.17b}$$

für alle Fälle aus Tabelle 4.4 unter Verwendung der zugehörigen Werte von a bzw. $\tilde{W}'(a)$.

Validierung

Zunächst werden die Ansätze 1-4 hinsichtlich der Realisierung von virtuellen Datentupeln zur Integration von Vorwissen verglichen. Im Folgenden wird der Vergleich auf einen Teildatensatz des simulierten Benchmark-Datensatzes D_{Sim} erweitert. Außerdem wird untersucht, inwiefern sich der Rauschterm der regulären Datentupel auf die Ansätze auswirkt. Es wird dabei die Modellgenauigkeit, die Modellkomplexität und die benötigte Zeit zur Parameterschätzung für die jeweiligen SVR-Modelle betrachtet.

Die Modellgenauigkeit wird mit dem mittleren absoluten Fehler (MAE) über Testdaten ($D_{\mathrm{Sim,Test}}$) angegeben. Die Anzahl an Stützvektoren (SV) entspricht der Modellkomplexität. Die benötigte Zeit zur Parameterschätzung wird für MATLABs *quadprog* gemessen. Als anschauliches Beispiel werden die Eingangsgrößen x_3 und x_4 von $D_{\mathrm{Sim,Lern},2}$ mit der Zielgröße y_2 als Lerndatensatz verwendet. Die nicht abgedeckten Bereiche des Eingangsraums werden mit punktweisem Vorwissen

$$f(\boldsymbol{x}_{\mathrm{R},1}) = f(x_3 = 0, x_4 = 0) \stackrel{!}{=} 0.1 \text{ mit } \varepsilon_{\mathrm{R},1,\mathrm{gef}} = 0.001 \text{ und} \tag{4.18a}$$

$$f(\boldsymbol{x}_{\mathrm{R},2}) = f(x_3 = 0, x_4 = 1) \stackrel{!}{=} 1.1 \text{ mit } \varepsilon_{\mathrm{R},2,\mathrm{gef}} = 0.01 \tag{4.18b}$$

über den realen Zusammenhang kompensiert. Als erster Schritt werden geeignete Metaparameter für die regulären Datentupel festgelegt. Entsprechende Heuristiken und Ansätze finden sich in [159]. Im Folgenden wird eine Gittersuche zur Minimierung der mittleren absoluten Fehler über Kreuzvalidierungen verwendet. Die Ergebnisse der Kreuzvalidierungen sind in Abbildung 4.1 dargestellt.

Für die unterschiedlichen Ausprägungen des Rauschens auf der Zielgröße ergeben sich die Metaparameter aus Tabelle 4.5.

Für mittleres und starkes Rauschen ändert sich ausschließlich $\varepsilon_{\mathrm{reg}}$.

Gemäß der vorgestellten Realisierung von virtuellen Datentupeln gilt $\varepsilon_{\mathrm{R}} = \min\{\varepsilon_{\mathrm{R},1,\mathrm{gef}}, \varepsilon_{\mathrm{R},2,\mathrm{gef}}\} = 0.001$ und in den Ansätzen 1 und 2 wird das einheitliche ε auf $\min\{\varepsilon_{\mathrm{R}}, \varepsilon_{\mathrm{reg}}\} = 0.001$ festgelegt und C_{reg} und σ neu bestimmt, während in den Ansätzen 3 und 4 für die regulären Datentupel $\varepsilon_{\mathrm{reg}}$ und für die virtuellen Datentupel die geforderten Genauigkeiten $\varepsilon_{\mathrm{R},1,\mathrm{gef}}$ und $\varepsilon_{\mathrm{R},2,\mathrm{gef}}$ gewählt werden.

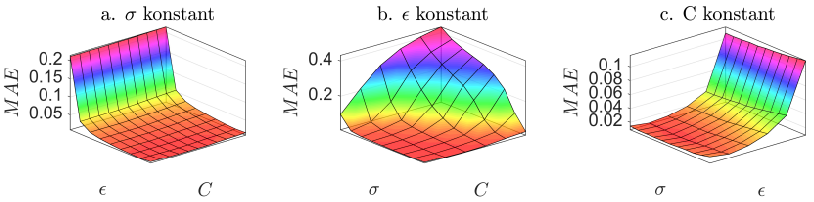

Abbildung 4.1: Gittersuche zur Auswahl geeigneter Metaparameter für die regulären Datentupel mit Hilfe von Kreuzvalidierungen bei leichtem Rauschen.

Rauschen	ε_{reg}	C_{reg}	σ
leicht	0.015	10	0.3
mittel	0.025	10	0.3
stark	0.05	10	0.3

Tabelle 4.5: Metaparameter bei unterschiedlichem Rauschen auf der Zielgröße. Die Optimierung der Metaparameter erfolgt mit Hilfe einer Gittersuche und Kreuzvalidierungen.

In den Ansätzen 1 und 3 wird das einheitliche C auf C_{reg} festgelegt und die Einhaltung der Randbedingungen durch eine geeignete Anzahl an Duplikaten der virtuellen Datentupel erreicht. In den Ansätzen 2 und 4 wird für die regulären Datentupel C_{reg} verwendet und für die beiden virtuellen Datentupel wird C proportional zur geeigneten Anzahl an Duplikaten aus den Ansätzen 1 und 3 gewählt. Die Parameterschätzungen werden für jeden Ansatz zehnmal für zufällig gezogene Stichproben (90% der regulären Datentupel) wiederholt, um zufällige Effekte auszuschließen. Abbildung 4.2 zeigt den Funktionsverlauf von einem Modell ohne Vorwissen und einem Modell mit Vorwissen (Ansatz 4).

Da die Verwendung von Duplikaten den gleichen Einfluss auf die Modelle hat wie die Wichtung der duplizierten Datentupel, haben die Modelle der Ansätze 1 und 2 sowie 3 und 4 den gleichen Funktionsverlauf. Als Referenz dient ein herkömmlicher Modellentwurf mit SVRs ohne die Integration von Vorwissen (Ansatz 0). Für eine quantitative Bewertung der Ansätze

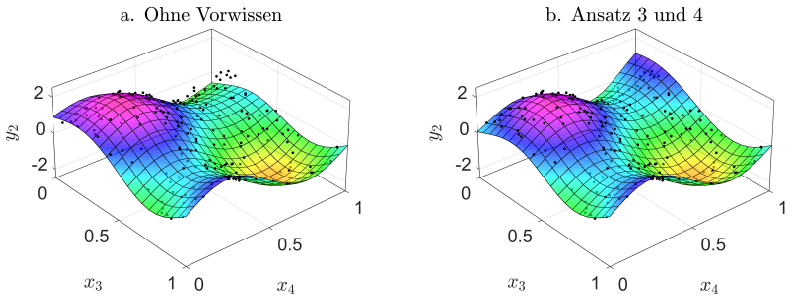

Abbildung 4.2: Testdaten und Beispiel eines SVR-Modells a. ohne Vorwissen b. mit Vorwissen (Ansatz 4)

der Modelle zeigt Abbildung 4.3 einen Vergleich der Ansätze anhand der oben genannten Kriterien.

Zusammenfassend ist Ansatz 4 den übrigen Ansätzen hinsichtlich der Integration von Vorwissen mit virtuellen Datentupeln überlegen. Mit Ansatz 4 kann ein geeignetes ε_{reg} für die regulären Datentupel ausgewählt werden, unabhängig davon, welche Genauigkeit für die virtuellen Datentupel des Vorwissens gefordert ist. Außerdem kann eine Wichtung der Datentupel anstelle von Duplikaten verwendet werden, was zu einem geringeren Speicherbedarf führt. Das beschleunigt ebenfalls die Metaparametrierung: Anstatt die Anzahl der Duplikate iterativ zu erhöhen, kann direkt ein hoher Wert für die individuellen Wichtungen der virtuellen Datentupel gewählt werden. Eine initial hoch gewählte Anzahl an Duplikaten führt dagegen zu einem enormen Speicher- und Zeitbedarf.

Für Ansatz 4 wurde der GSMO-Algorithmus vorgestellt. Im Folgenden werden die Ergebnisse des GSMO für drei Variationen des oben eingeführten Regressionsproblems mit den Ergebnissen des *quadprog* Lösers von MATLAB verglichen, um die Funktionsfähigkeit des GSMO-Algorithmus zu überprüfen. Die Variationen beziehen sich ausschließlich auf die Randbedingungen, d.h. y_{V1} und y_{V2} werden gemäß Tabelle 4.6 verändert.

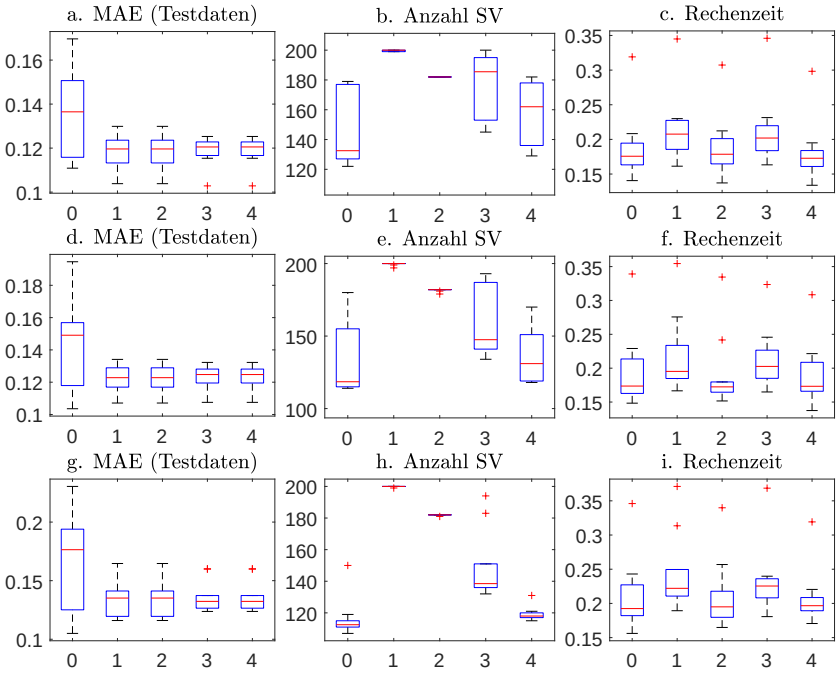

Abbildung 4.3: Quantitativer Vergleich der Ansätze bei a.-c. leichtem Rauschen d.-f. mittlerem Rauschen g.-i. starkem Rauschen.

Der Vergleich erfolgt anhand des erreichten Gütewerts des gesamten Optimierungsproblems

$$Q_{\text{Opt}} = \min_{\boldsymbol{\theta}} \sum_{n=1}^{N} \varepsilon_n |\theta_n| - \sum_{n=1}^{N} \theta_n y_n + \frac{1}{2} \sum_{n=1}^{N} \sum_{j=1}^{N} \theta_n \theta_j k_{nj} \tag{4.19a}$$

$$\text{s.t.} \sum_{n=1}^{N} \theta_n = 0 \text{ and } \theta_n \in [-C_n, C_n]\,. \tag{4.19b}$$

sowie der mittleren absoluten Abweichung (MAD) der entstehenden SVR-Modelle über den Testdaten.

Variation	y_{V1}	y_{V2}	$Q(QP)$	$Q(GSMO)$	MAD
1	0	1	-10.9459	-10.9459	7.2E-07
2	1	0	-9.3466	-9.3466	2.4E-06
3	-0.5	0.5	-10.3042	-10.3042	1.5E-06

Tabelle 4.6: Vergleich von Modellen mit individuellen ε_i und C_i nach Parameterschätzungen durch einen herkömmlichen Lösers eines quadratischen Programms (QP) und GSMO.

Die Unterschiede zwischen den Lösungen sind gering und können aufgrund ihrer Größenordnung mit numerischen Effekten erklärt werden.

Es ist anzumerken, dass sich die Verwendung des GSMO-Algorithmus hinsichtlich der Rechenzeit erst lohnt, wenn effektive Heuristiken zur Auswahl geeigneter Variablen für den analytischen Schritt verwendet werden. Zudem ist der Algorithmus wie z.B. die beliebte SVM-Bibliothek LIBSVM in C/C++ zu implementieren, um die Rechenzeit weiter zu reduzieren. Ohne eine solche Implementierung erfolgt die Integration von Vorwissen in SVRs besser mit Hilfe eines quadratischen Lösers. Dadurch ist die vorgestellte Methodik allerdings aufgrund des hohen Speicher- und Rechenaufwands beschränkt. Abbildung 4.4 zeigt eine quantitative Untersuchung der Rechenzeit des *quadprog*-Lösers von MATLAB für SVR-Probleme am Beispiel von $D_{\text{Sim,Lern},1}$, bei denen die Datentupel sukzessive vervielfacht werden. Die Untersuchung berücksichtigt außerdem unterschiedliche Metaparametrierungen.

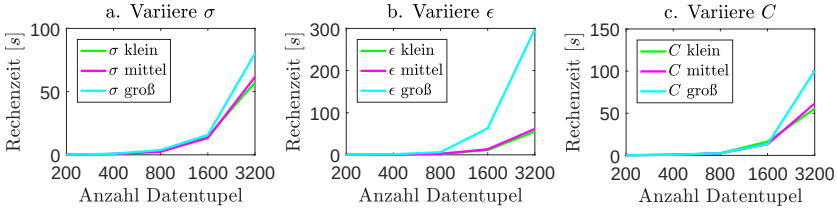

Abbildung 4.4: Rechenzeiten des *quadprog*-Lösers für steigende Anzahl an Datentupel und unterschiedliche Metaparametrierungen

Die Untersuchung zeigt, dass die Rechenzeit quadratisch mit der Anzahl der Datentupel skaliert. Außerdem ist zu sehen, dass die Rechenzeiten auch mit zunehmenden Werten für die Metaparameter steigen. Für Datensätze mit mehr als 5000 Datentupeln scheint *quadprog* bei SVRs daher nicht geeignet - vor allem da für die Gittersuche zur Optimierung der Metaparametrierung eine Vielzahl an Parameterschätzungen nötig sind. Die Ansätze 2 und 4 haben den Vorteil, dass generell weniger Datentupel verwendet werden, als bei den anderen Ansätzen, da keine Duplikate der virtuellen Datentupel erstellt werden, sondern die Wichtung variiert wird. Allerdings ergeben sich die diskutierten Vorteile eher unter der Verwendung von individuell höheren Werten für die Metaparameter einzelner Datentupel, was ebenfalls zu erhöhten Rechenzeiten führen kann.

4.1.4 Zusammenfassung

Es wurde eine Methodik vorgestellt, Vorwissen mit Hilfe eines Fragebogens von Domänenexperten zu erfragen, um es in Gleichungen und Ungleichungen zu überführen. Weiterhin wurden unterschiedliche Ansätze untersucht, Vorwissen mit Hilfe von virtuellen Datentupeln in Stützvektor-Regressionen zu integrieren. Für den am besten geeigneten Ansatz fehlte bislang eine Modifikation des SMO-Algorithmus, der als Stand-der-Technik-Löser für das SVR-Problem gilt. Eine entsprechende Modifikation wurde vorgestellt und anhand des Benchmark-Datensatzes und künstlich generiertem Vorwissen validiert. Nächste Schritte sind die Entwicklung neuer Heuristiken, um den ersten Schritt des SMO-Algorithmus für den GSMO-Algorithmus anzupassen und den gesamten Algorithmus beispielsweise in C++ zu implementieren. Damit kann die Rechenzeit deutlich verkürzt werden.

Virtuelle Datentupel eignen sich außerdem nur für eine geringe Teilmenge möglichen Vorwissens. Bei strukturellem Vorwissen bietet sich eher die Verwendung von interpretierbaren Modellfamilien an.

4.2 Hybride Modelle

4.2.1 Übersicht

Eine eingeschränkte Datenabdeckung wurde als Herausforderung im Entwurf und der Anwendung von Regressionsmodellen erkannt. Generell wird davon ausgegangen, dass vor allem komplexe Modelle in den Bereichen des Eingangsraums versagen, in denen keine Lerndaten zum Modellentwurf vorliegen. Einfachere, robuste Modelle erreichen allerdings nicht die partiell hohe Prädiktionsgüte der komplexen Modelle.

Bisher wurden in den Kapiteln 2 und 3 Bewertungskriterien vorgestellt, um die Datenabdeckung in einem Datensatz gemäß verschiedener Phänomene zu quantifizieren und die Modellqualität im Sinne der Prädiktionsgüte für Bereiche geringer Datenabdeckung abzuschätzen.

Die Güte einzelner Prädiktionen kann während der Anwendung von Regressionsmodellen auch durch die lokale Datendichte abgeschätzt werden, wie in [66] am Beispiel von ANNs gezeigt wird.

Im folgenden Kapitel wird ein neues Entwurfsverfahren für *hybride Modelle* vorgestellt, um zuverlässigere Modelle - auch bei eingeschränkter Datenabdeckung - zu entwerfen. Das Entwurfsverfahren bietet zunächst die Möglichkeit die hybriden Modelle automatisiert zu entwerfen. Weiterhin kann Vorwissen in Form von Modellstrukturen in den Entwurf integriert werden, um eine Robustheit des hybriden Modells in den Bereichen geringer Datenabdeckung zu ermöglichen.

Die hybriden Modelle bestehen aus zwei Teilmodellen, dem *Interpolationsmodell* $f_{\mathrm{I}}(\boldsymbol{x})$ und dem *Extrapolationsmodell* $f_{\mathrm{E}}(\boldsymbol{x})$, die den gleichen Zusammenhang i.A. mit unterschiedlicher Komplexität abbilden. Mit Hilfe eines unscharfen Ein-Klassen-Klassifikators, Dichteschätzers o.ä. als Koordinator $g(\boldsymbol{x}) \in [0,1]$ werden die Modelle stetig ineinander überführt:

$$f_{\mathrm{hybrid}} = f_{\mathrm{I}}(\boldsymbol{x})g(\boldsymbol{x}) + f_{\mathrm{E}}(\boldsymbol{x})(1 - g(\boldsymbol{x})). \tag{4.20}$$

$g(\boldsymbol{x})$ gibt eine Aussage darüber, ob ein beliebiger Eingangsvektor in einem Bereich liegt, der vom Lerndatensatz abgedeckt ist (Interpolationen, $g(\boldsymbol{x}) \approx 1$), oder nicht (Extrapolationen, $g(\boldsymbol{x}) \approx 0$) [72]. Damit wird erreicht, dass der Ausgang des hybriden Modells in Bereichen hoher Datenabdeckung

maßgeblich vom Interpolationsmodell bestimmt wird und in Bereichen niedriger Datenabdeckung vom Extrapolationsmodell.

Als Teilmodelle können sowohl analytische Modelle, als auch datengetriebene Modelle beliebiger Modellfamilien verwendet werden. Bei Kenntnis allgemeiner Zusammenhänge kann die Verwendung eines analytischen Modells aufgrund der Robustheit und Interpretierbarkeit für Extrapolationen von Vorteil sein.

4.2.2 Abschätzung der lokalen Datenabdeckung

Zur Abschätzung der lokalen Datenabdeckung werden 1K-SVMs verwendet. Exemplarisch wurde der Verlauf von entsprechenden Klassengrenzen bereits in Abbildung 1.6 gezeigt. Je größer σ gewählt wird, desto größer wird der Bereich des Eingangsraums, der den Lerndaten zugewiesen wird.

Für die hybriden Modelle wird allerdings keine binäre Klassifikation benötigt, sondern eine stetige Funktion zwischen 0 und 1, um die Teilmodelle ineinander zu überführen und Sprünge zu vermeiden. Gemäß (4.2) und (4.6) kann die Entscheidungsfunktion der 1K-SVM in

$$g_{\text{1K-SVM}}(\boldsymbol{x}) = \operatorname{sgn}\left(\boldsymbol{w} \cdot \boldsymbol{\phi}(\boldsymbol{x}) - b\right), \boldsymbol{w} \cdot \boldsymbol{\phi}(\boldsymbol{x}) \geq 0 \tag{4.21}$$

überführt werden, wobei $\boldsymbol{\phi}$ eine Transformation in einen Merkmalsraum und $\boldsymbol{w}$ den Parametervektor einer Ebene im Merkmalsraum darstellt. Anschließend wird die Entscheidungsfunktion für eine kontinuierliche Aussage angepasst:

$$\tilde{g}_{\text{1K-SVM}}(\boldsymbol{x}) = \begin{cases} 1 & \text{, wenn } (\boldsymbol{w} \cdot \phi(\boldsymbol{x})) - b \geq 0 \\ 1 - \frac{(\boldsymbol{w} \cdot \phi(\boldsymbol{x})) - b}{-b} & \text{, wenn } (\boldsymbol{w} \cdot \phi(\boldsymbol{x})) - b < 0. \end{cases} \tag{4.22}$$

Die entsprechenden Verläufe von $\tilde{g}_{\text{1K-SVM}}$ für zwei Eingangsgrößen von $D_{\text{Sim,Lern,1}}$ sind in Abbildung 4.5 dargestellt.

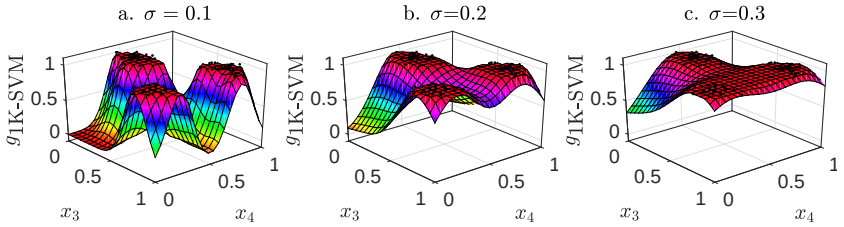

Abbildung 4.5: Graduelle Zuweisung zur Klasse der Lerndaten durch 1K-SVMs mit unterschiedlicher Metaparametrierung.

4.2.3 Entwurfsmethodik

In Anlehnung an Abschnitt 1.2.1 zeigt Abbildung 4.6a das Schema für den Entwurf von Regressionsmodellen. Auf Basis von Lerndaten wird mit Hypothesengenerierung und Parameterschätzung ein Modellpool an möglicherweise geeigneten Modellen erstellt. Restriktionen an Modellstrukturen und Modellfamilien resultieren aus eingebrachtem Vorwissen oder zur Verfügung stehender Implementierungen. Anschließend wird auf Basis von Validierungsdaten in der Modellselektion das Modell aus dem Modellpool gewählt, das gemäß eines Gütekriteriums (z.B. mittlerer absoluter Fehler, Bestimmtheitsmaß o.ä.) am besten ist:

$$i_{\text{Selektion}} = \underset{i}{\operatorname{argmin}} \frac{1}{N_{\text{Val}}} \sum_{n=1}^{N_{\text{Val}}} |f_i(\boldsymbol{x}_n) - y_n|. \tag{4.23}$$

Die Entwurfsmethodik für hybride Modelle beinhaltet ebenfalls die Erstellung eines Modellpools. Weiterhin wird ein Pool an unterschiedlich metaparametrierten 1K-SVMs erstellt, die als potentielle Koordinatoren bzw. Ein-Klassen-Klassifikatoren im hybriden Modell dienen. Anstatt einer einfachen Modellselektion (4.23) findet eine Optimierung der hybriden Struktur statt:

$$\{i_{\text{Opt}}, j_{\text{Opt}}, l_{\text{Opt}}\} = \underset{i,j,l}{\operatorname{argmin}} \frac{1}{N_{\text{Val}}} \sum_{n=1}^{N_{\text{Val}}} |f_{\text{I},i}(\boldsymbol{x}_n) g_l(\boldsymbol{x}_n) + f_{\text{E},j}(\boldsymbol{x}_n)(1 - g_l(\boldsymbol{x}_n)) - y_n|. \tag{4.24}$$

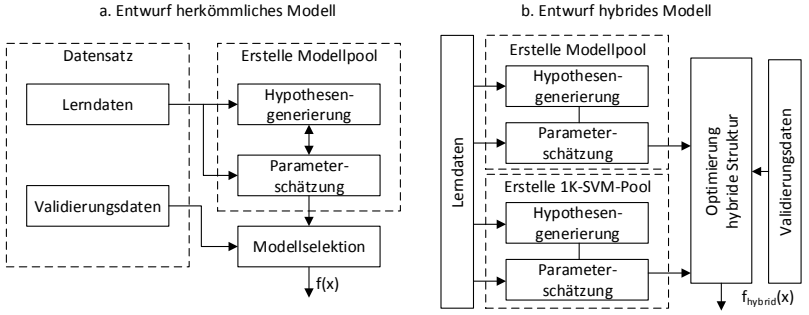

Abbildung 4.6: Entwurfsmethodik für a. herkömmliches Modell b. hybrides Modell

4.2.4 Bewertung von hybriden Modellen

Zur Bewertung von hybriden Modellen werden Datensätze gemäß Abbildung 4.8 aufgeteilt. Zunächst werden randomisiert 10% der Daten als erster Testdatensatz entnommen. Es ist davon auszugehen, dass ein randomisiert entnommener Datensatz gegebenenfalls keine ausreichende Aussage über die Generalisierungsfähigkeit der Modelle zulässt. Deshalb wird ein weiterer, zweiter Testdatensatz entnommen. Dazu wird zunächst randomisiert ein Datentupel $(\boldsymbol{x}_\mathrm{a}, y_\mathrm{a})$ sowie zwei Eingangsgrößen x_u, x_v ausgewählt. Anschließend werden alle Datentupel, für die

$$\boldsymbol{x}_i \;\mid\; \max\{|x_{\mathrm{i,u}} - x_{\mathrm{a,u}}|, |x_{\mathrm{i,v}} - x_{\mathrm{a,v}}|\} \leq 0.05 \tag{4.25}$$

gilt, als zweiter Testdatensatz entnommen[2]. Abbildung 4.7 zeigt am Beispiel der Eingangsgrößen *frequency* und *angle of attack* des Datensatzes *Airfoil Self Noise* die Entnahme eines zweiten Testdatensatzes.

Mit dem zweiten Testdatensatz wird die Prädiktionsgüte für einen Bereich des Eingangsraums überprüft, der vollständig aus dem Datensatz entfernt wurde. Die Modelle sind dadurch gezwungen zu extrapolieren.

[2]es handelt sich dabei um die Chebyshev-Distanz unter der Berücksichtigung der u-ten und v-ten Eingangsgröße

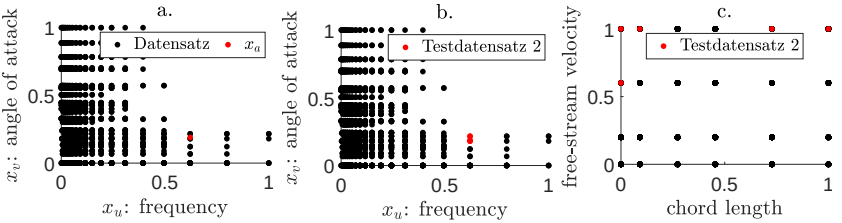

Abbildung 4.7: Cluster-Entnahme eines zweiten Testdatensatzes durch a. randomisierte Auswahl eines Datentupels und b. Auswahl gemäß (4.25). c. zeigt eine weitere Ansicht, bei der die Datentupel des zweiten Testdatensatzes beliebig verteilt sein können.

Es kann zwar kein Modell in einem solchen Bereich Prädiktionsgüten wie in Interpolationsbereichen erzielen, es kann aber überprüft werden, ob ein Modell in Extrapolationsbereichen noch akzeptable Prädiktionen liefert. Der übrige Datensatz wird randomisiert in Lern- und Validierungsdaten aufgeteilt. Um eine ausführliche Untersuchung des Entwurfsverfahrens von hybriden Modellen sicherzustellen, wird der Anteil der Validierungsdaten variiert.

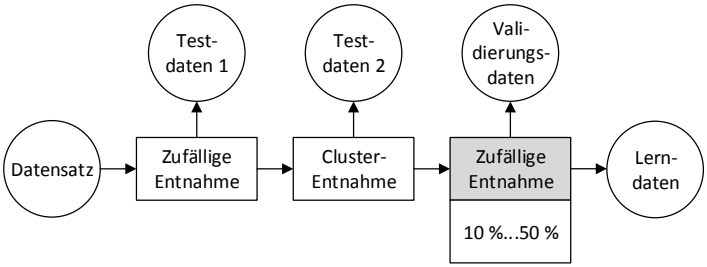

Abbildung 4.8: Schema zur Erstellung der Teildatensätze für den Modellvergleich. Die Entnahme der Validierungsdaten wird variiert, um eine allgemeine Gültigkeit des Modellvergleichs sicherzustellen.

4.2.5 Validierung

Die Validierung der hybriden Modelle untersucht zwei Fragestellungen für einen vollständig datengetriebenen Entwurf, d.h. ohne Verwendung von analytischen Teilmodellen:

1. Erzielen hybride Modelle höhere Prädiktionsgüten als herkömmliche Modelle aus dem gleichen Modellpool?
2. Ist das gewählte Interpolationsmodell für Interpolationen signifikant besser als das Extrapolationsmodell, bzw. ist das Extrapolationsmodell für Extrapolationen signifikant besser als das Interpolationsmodell?

Dazu werden die in Tabelle 4.7 aufgeführten Benchmark-Datensätze sowie die in Abschnitt 2.2 vorgestellten simulierten Datensätze verwendet[3].

Name	Kurzname	Datentupel	Eingangsgrößen
Abalone	Abalone	4177	8
Airfoil Self Noise	Airfoil	1503	5
Boston Housing	Boston	506	13
California Housing	California	20640	8
Computeractivity CPU	Computer	8192	21
Concrete	Concrete	1030	8
Delta Ailerons	Ailerons	7129	5
Delta Elevators	Elevators	9517	6
Red Wine Quality	Redwine	1599	11
White Wine Quality	Whitewine	4898	11

Tabelle 4.7: Übersicht über die verwendeten Datensätze. Die Anzahl der Datentupel bezieht sich auf den gesamten Datensatz.

Die Datensätze werden gemäß Abschnitt 4.2.4 aufgeteilt. Die Parameterschätzung aller betrachteten Modellstrukturen erfolgt auf Basis der Lerndaten. Zur Modellselektion des besten herkömmlichen Modells (f_{Ref}) und zur Optimierung der hybriden Struktur (f_{Hybrid}) dienen die Validierungsdaten und (4.23) bzw. (4.24). Anschließend werden f_{Ref} und f_{Hybrid}

[3]Die Datensätze sind über das *UCI Repository* [131], bzw. unter http://www.cs.toronto.edu/~delve/data/datasets.html verfügbar

anhand des mittleren absoluten Fehlers für die Testdaten verglichen. Abbildung 4.9 zeigt eine schematische Darstellung der Validierung.

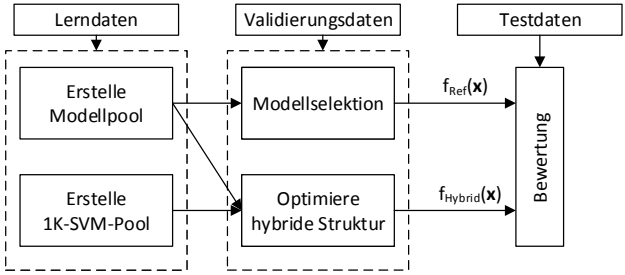

Abbildung 4.9: Schema zum Vergleich herkömmlicher Modelle mit hybriden Modellen

Zusätzlich wird das hybride Modell auf Sinnhaftigkeit geprüft. Dazu wird die Differenz der graduellen Fehler

$$e_{\mathrm{g,I}}(\boldsymbol{x}) = g(\boldsymbol{x})(|f_I(\boldsymbol{x}) - y| - |f_E(\boldsymbol{x}) - y_i|) \tag{4.26}$$

bzw.

$$e_{\mathrm{g,E}}(\boldsymbol{x}) = (1 - g(\boldsymbol{x}))\,(|f_I(\boldsymbol{x}) - y| - |f_E(\boldsymbol{x}) - y_i|) \tag{4.27}$$

verwendet. Mit T-Tests werden die Nullhypothesen und T-Tests die Nullhypothesen

1. „Das Interpolationsmodell weist für Interpolationen im Mittel keinen geringeren Fehler als das Extrapolationsmodell auf."
2. „Das Extrapolationsmodell weist für Extrapolationen im Mittel keinen geringeren Fehler als das Interpolationsmodell auf."

überprüft, bzw. es wird bestimmt, mit welcher statistischen Signifikanz sie verworfen werden können.

Die verwendeten Modellstrukturen für den Modellpool sind

- MLP-Netze ($N_{\mathrm{Neuronen}} = 3, \ldots, 17, 30, 50$),

- SVR ($\sigma = 0.1, 0.2, 0.3, \ldots, 1, 1.2, 1.5, 2$)
- MARS und
- LMN (LOLIMOT).

Der 1K-SVM-Pool besteht aus unterschiedlich metaparametrierten 1K-SVMs ($\sigma = 0.01, 0.025, 0.05, 0.1, 0.2, 0.3, \ldots, 1, 1.5, 10$).

Zunächst wird anhand des Datensatzes Airfoil der Einfluss des Anteils entnommener Validierungsdaten auf die Prädiktionsgüte untersucht. Abbildung 4.10 zeigt die mittleren absoluten Fehler für zufällig entnommene Testdaten und für Testdaten durch Cluster-Entnahme in Abhängigkeit von der Aufteilung in Lern- und Validierungsdaten.

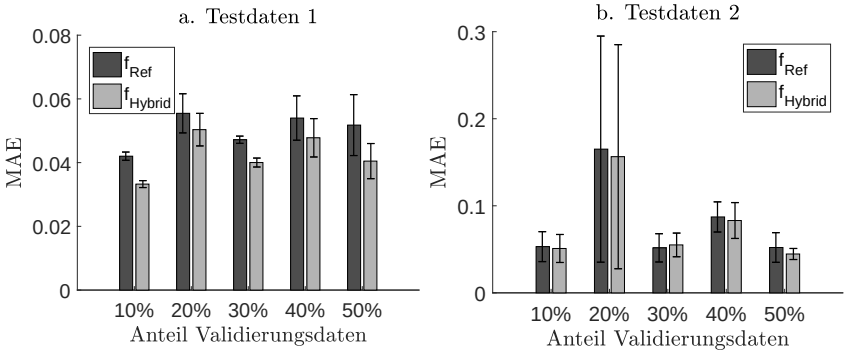

Abbildung 4.10: Ergebnisse über die Testdatensätze bei unterschiedlicher Aufteilung in Lern- und Validierungsdaten für den Datensatz Airfoil.

Bei den zufällig entnommenen Testdatensätzen (Testdaten 1) zeigt sich eine deutliche Überlegenheit der hybriden Modelle unabhängig von der Aufteilung in Lern- und Validierungsdaten. Bei den Testdatensätzen durch Cluster-Entnahme (Testdaten 2) sind keine signifikanten Unterschiede zwischen hybriden und herkömmlichen Modellen zu erkennen. Die Modelle weisen allerdings eine deutlich reduzierte Prädiktionsgüte auf. Das lässt darauf schließen, dass in der Optimierung der hybriden Modelle durch zufällig entnommene Validierungsdaten zumindest bei der vorliegenden Datenabdeckung kein wesentlich robusteres Extrapolationsmodell als das Interpolationsmodell gewählt wird. Dennoch werden durch hybride Modelle

höhere Prädiktionsgüten erreicht als bei herkömmlichen Modellen, wie anhand der zufällig entnommenen Testdaten gezeigt wurde.

Für die weitere Validierung wird ein Anteil von 10% Validierungsdaten bei der Aufteilung der Datensätze verwendet. Die mittleren absoluten Fehler für Testdaten von f_{Ref} und f_{Hybrid} werden quantitativ angegeben. Die Überprüfung der Sinnhaftigkeit der hybriden Modelle wird qualitativ dargestellt. Tabelle 4.8 zeigt die Ergebnisse für zufällig entnommene Testdaten und Tabelle 4.9 für Testdaten durch Cluster-Entnahme bei den bekannten Benchmark-Datensätzen. Tabelle 4.10 zeigt die Ergebnisse für die simulierten Benchmark-Datensätze. Da die Testdaten der simulierten Benchmark-Datensätze bereits die Generalisierungsfähigkeit überprüfen, wird eine Entnahme gemäß Abbildung 4.9 nicht benötigt und es wird nur ein Testdatensatz betrachtet, der sowohl ausreichend Interpolationen als auch Extrapolationen überprüft. Die Validierungsdaten werden aus den ursprünglichen Lerndaten der simulierten Benchmark-Datensätze entnommen.

Für die Mehrheit der verwendeten Datensätze weisen die automatisiert entworfenen hybriden Modelle bei beiden Testdatensätzen eine höhere Prädiktionsgüte als herkömmliche Modelle auf. Weiterhin kann zumindest für die zufällig entnommenen Testdaten für neun von zehn verwendeten bekannten Benchmark-Datensätzen gezeigt werden, dass nach der Optimierung des hybriden Modells das Interpolationsmodell in den Interpolationsbereichen signifikant besser ist, als das Extrapolationsmodell und vice versa. Dass die Signifikanzen für die Testdaten aus der Cluster-Entnahme für manche Datensätze nicht nachgewiesen werden können, kann an zufälligen Effekten der Testdaten liegen. Der mittlere Wert des Koordinators $\bar{g}$ ist z.B. für den Datensatz „Whitewine“ sehr gering, d.h. es liegen u.U. gar nicht genügend Daten vor, um interpretierbare Aussagen treffen zu können. Die Definition einer Interpolation bzw. Extrapolation wird jeweils durch die in der Optimierung ausgewählten 1K-SVM bestimmt und ist damit von den zufällig ausgewählten Validierungsdaten abhängig. Daher impliziert $\bar{g}$ Informationen über die Datenqualität.

4.2.6 Zusammenfassung

Es wurde ein neues Entwurfsverfahren für hybride Modelle vorgestellt. Die Wichtung der Teilmodelle der hybriden Modellstruktur erfolgt durch eine Ein-Klassen-Stützvektor-Maschine. Mit Hilfe bekannter Benchmark-Datensätze wurden hybride Modelle, die entsprechend der Methodik entworfen werden, mit herkömmlichen Modellen verglichen. Bei fast allen betrachteten Datensätzen konnte eine höhere Prädiktionsgüte bei den hybriden Modellen nachgewiesen werden. Die Struktur ermöglicht weiterhin, ein analytisches Modell mit einem datengetriebenen Modell zu kombinieren. Durch die Robustheit des analytischen Modells kann die mangelhafte Extrapolationsfähigkeit komplexer datengetriebener Modelle kompensiert werden. Zukünftig kann untersucht werden, ob andere Auswahlstrategien für die Validierungsdaten, z.B. eine Cluster-Entnahme, besser für die Optimierung von hybriden Modellen geeignet sind und weitere Verbesserungen der Prädiktionsgüte nach sich ziehen.

Datensatz	$\bar{g}$	f_{Ref}	f_{Hybrid}		f_{Int}	f_{Ext}
Abalone	0.981	0.053	**0.052**	Interpolation	$+^{***}$	$-$
				Extrapolation	$-$	$+$
Concrete	0.740	0.051	**0.046**	Interpolation	$+$	$-$
				Extrapolation	$-$	$+^{***}$
Boston	0.393	**0.055**	0.059	Interpolation	$+$	$-$
				Extrapolation	$-$	$+$
Airfoil	0.595	0.050	**0.041**	Interpolation	$+^{***}$	$-$
				Extrapolation	$-$	$+^{***}$
California	0.514	0.073	**0.071**	Interpolation	$+^{***}$	$-$
				Extrapolation	$-$	$+^{***}$
Computer	0.950	0.021	**0.019**	Interpolation	$+^{***}$	$-$
				Extrapolation	$-$	$+^{*}$
Ailerons	0.850	0.027	**0.027**	Interpolation	$+^{***}$	$-$
				Extrapolation	$-$	$+^{*}$
Elevators	0.656	**0.039**	0.040	Interpolation	$-$	$+$
				Extrapolation	$-$	$+^{***}$
Redwine	0.302	0.101	**0.087**	Interpolation	$+^{***}$	$-$
				Extrapolation	$-$	$+^{***}$
Whitewine	0.410	0.089	**0.076**	Interpolation	$+^{***}$	$-$
				Extrapolation	$-$	$+^{***}$

Tabelle 4.8: Vergleich herkömmlicher Modelle mit hybriden Modellen aus dem gleichen Modellpool anhand des mittleren absoluten Fehlers für die zufällig extrahierten Testdaten. $\bar{g}$ gibt die mittlere Wichtung der Modelle durch den Koordinator an. Je höher $\bar{g}$ desto eher handelt es sich bei den Testdaten um Interpolationen gemäß des gewählten Koordinators. Das Modell mit dem graduell geringeren mittleren absoluten Fehler für Inter- bzw. Extrapolationen ist mit einem + gekennzeichnet, das mit dem Höheren mit einem $-$. Die Markierungen $*$ ($p = 0.05$), $**$ ($p = 0.01$) und $***$ ($p = 0.001$) indizieren unterschiedliche statistische Signifikanzen.

Datensatz	$\bar{g}$	f_{Ref}	f_{Hybrid}		f_{Int}	f_{Ext}
Abalone	0.984	0.053	**0.052**	Interpolation	$+^{***}$	$-$
				Extrapolation	$-$	$+$
Concrete	0.589	0.086	**0.078**	Interpolation	$+^{*}$	$-$
				Extrapolation	$-$	$+$
Boston	0.198	**0.016**	0.028	Interpolation	$-$	$+$
				Extrapolation	$-$	$+^{*}$
Airfoil	0.000	**0.080**	0.158	Interpolation	$-$	$+$
				Extrapolation	$-$	$+^{***}$
California	0.151	0.073	**0.070**	Interpolation	$-$	$+$
				Extrapolation	$-$	$+^{***}$
Computer	0.953	0.021	**0.019**	Interpolation	$+^{***}$	$-$
				Extrapolation	$+$	$-$
Ailerons	0.557	**0.041**	0.042	Interpolation	$-$	$+$
				Extrapolation	$+$	$-$
Elevators	0.463	0.040	**0.039**	Interpolation	$-$	$+$
				Extrapolation	$-$	$+^{*}$
Redwine	0.052	**0.104**	0.107	Interpolation	$-$	$+$
				Extrapolation	$-$	$+^{***}$
Whitewine	0.204	0.110	**0.089**	Interpolation	$-$	$+$
				Extrapolation	$-$	$+^{***}$

Tabelle 4.9: Vergleich herkömmlicher Modelle mit hybriden Modellen aus dem gleichen Modellpool anhand des mittleren absoluten Fehlers für die Testdaten durch Cluster-Entnahme.

Datensatz	$\bar{g}$	f_{Ref}	f_{Hybrid}		f_{Int}	f_{Ext}
$D_{\text{Sim,Lern,1}}$	0.951	**0.0118**	0.0151	Interpolation	$+^{***}$	$-$
				Extrapolation	$+$	$-$
$D_{\text{Sim,Lern,2}}$	0.486	0.0229	**0.0084**	Interpolation	$-$	$+$
				Extrapolation	$-$	$+^{***}$
$D_{\text{Sim,Lern,3}}$	0.024	0.0143	**0.0141**	Interpolation	$-$	$+$
				Extrapolation	$-$	$+^{***}$

Tabelle 4.10: Vergleich herkömmlicher Modelle mit hybriden Modellen aus dem gleichen Modellpool anhand des mittleren absoluten Fehlers für die Testdaten bei den simulierten Benchmark-Datensätzen.

4.3 Zusammenfassung

Es wurden zwei neue Ansätze der Hypothesengenerierung und Parameterschätzung vorgestellt, um die Prädiktionsgüte automatisiert zu entwerfender Regressionsmodelle vor allem bei einer geringen Datenabdeckung zu erhöhen. Zum einen wurde gezeigt, wie systematisch Vorwissen erfasst und in SVRs integriert werden kann. Dadurch kann z.B. ein Modellversagen in von regulären Datentupeln nicht abgedeckten Bereichen reduziert werden. Allerdings eignet sich das Verfahren aufgrund der aktuell zur Verfügung stehenden Algorithmen lediglich für Datensätze mit einer Anzahl an Datentupel im unteren vierstelligen Bereich. Zum anderen wurde eine neue Struktur und eine entsprechende Entwurfsmethodik für hybride Modelle vorgestellt. Bei den hybriden Modellen werden zwei Modelle unterschiedlicher Struktur kombiniert und in Abhängigkeit der lokalen Datendichte gewichtet. Das führt zur Gültigkeit eines einfacheren robusten Modells für nicht abgedeckte Bereiche und zur Gültigkeit und damit hohen Genauigkeit eines komplexen Modells in der Nähe der Lerndaten. Da für hybride Modelle beliebige Modellstrukturen verwendet werden können, bietet sich eine direkte Integration von punktweisem Vorwissen nicht an. Allerdings kann Vorwissen über analytische Zusammenhänge verwendet werde, um eine entsprechende Modellstruktur für das Extrapolationsmodell, das in nicht abgedeckten Bereichen gültig ist, zu definieren. Tabelle 4.11 zeigt eine Übersicht, unter welchen Bedingungen die beiden Ansätze von Vorteil sind.

	SVR mit Vorwissen	Hybride Modelle
Vorwissen	✓	(✓)
Hohe Anzahl DT	✗	✓
geringe Datenabdeckung	✓	✓

Tabelle 4.11: Einsatzbedingungen der Ansätze

5 Implementierung

5.1 Übersicht

Die Mehrzahl der vorgestellten Verfahren oder Bewertungskriterien wurden implementiert und in die MATLAB-Toolbox SciXMiner integriert, um die Methoden quelloffen zur Verfügung zu stellen. Die Implementierungen werden in den folgenden Abschnitten vorgestellt. Tabelle 5.1 zeigt eine Übersicht aller umgesetzten Funktionalitäten.

Funktion	Theorie	Implementierung
Korrelationen	2.3.2	5.2.2
Cluster	2.3.3	5.2.2
Konfigurationen	2.3.4	5.2.2
Ausreißer	2.3.5	5.2.2
Orthogonalität	2.3.6	5.2.2
Verlaufsvalidierung	3	5.2.3
GSMO	4.1.3	5.3
Hybride Modelle	4.2.3	5.4

Tabelle 5.1: Übersicht der Implementierungen und Nummern der zugehörigen Abschnitte

Die Berechnung von Korrelationen, Clustern, Konfigurationen, Ausreißern und Orthogonalität ist bereits in die aktuelle Version der SciXMiner-Toolbox DaMoQ integriert[1]. Die Verlaufsvalidierung ist ebenfalls integriert und wird mit der nächsten Release zur Verfügung gestellt. Die Toolboxen für hybride Modelle und zur Verwendung des GSMO-Algorithmus sind noch nicht finalisiert. Nächster Schritt ist die Verbesserung der Benutzerfreundlichkeit im Rahmen studentischer Arbeiten zur Integration von

[1] Download unter https://sourceforge.net/projects/scixminer/

GSMO und hybrider Modelle in eine Release-Version von SciXMiner. Eine kurze Beschreibung der bisherigen Implementierungen finden sich in den nächsten Abschnitten.

5.2 SciXMiner-Toolbox „DaMoQ“

5.2.1 Allgemein

DaMoQ ist eine Erweiterung der MATLAB-Toolbox SciXMiner zur Analyse von Daten- und Modellqualität im Entwurfsprozess von Regressionsmodellen [160].

DaMoQ umfasst bisher die Bewertung der Datenqualität gemäß der in Kapitel 2 vorgestellten Bewertungskriterien und die Bewertung der Modellqualität gemäß der in Kapitel 3 vorgestellten Verlaufsvalidierung.

5.2.2 Bewertung Datenqualität

In Abschnitt 2.3 werden Bewertungskriterien vorgestellt, um Korrelationen, Cluster, Konfigurationen, Ausreißer und Orthogonalität in Histogrammen und Streuwolkendiagrammen zu quantifizieren.

Tabelle 5.2 zeigt, welche für die Bewertungskriterien benötigten Parameter über DaMoQ angepasst werden können. Die Anpassung erfolgt über die Ansicht *Datenqualität*. Die Analyse der Datenqualität sowie die entsprechenden Visualisierungen werden über die Menüelemente von DaMoQ aufgerufen.

Abbildung 5.1 zeigt die entsprechende Ansicht sowie unterschiedliche Visualisierungen. Eine ausführliche Beschreibung der Visualisierungen wurde in [160] publiziert.

GUI	Parameter	Gleichung
Parameter cluster 1	τ_{Cluster}	(2.7)
Parameter outlier 1	k	(2.13)
Parameter outlier 2	$\tau_{\text{Aus},1}$	(2.15a)
Parameter outlier 3	$\tau_{\text{Aus},2}$	(2.15b)
Parameter orthogonality	τ_{Ortho}	(2.19)
Weight corr	w_{Korr}	(2.24)
Weight cluster	w_{Cluster}	(2.24)
Weight config	w_{Konfig}	(2.24)
Weight outlier	w_{Aus}	(2.24)
Weight orthogonality	w_{Ortho}	(2.24)

Tabelle 5.2: Zuordnung der Metaparameter zur GUI (siehe Abbildung 5.1)

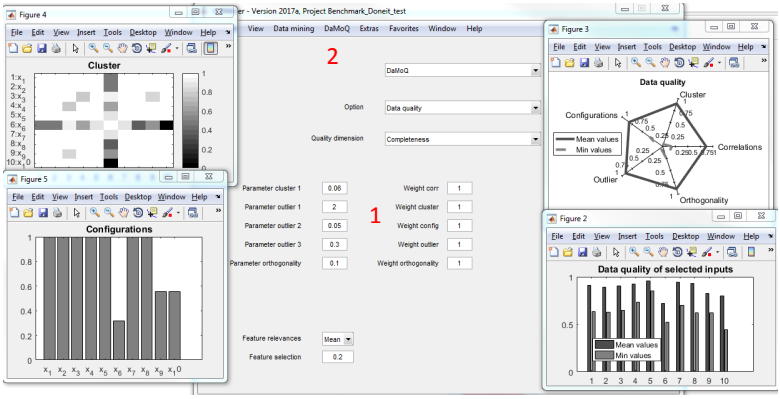

Abbildung 5.1: Übersicht von DaMoQ für die Bewertung der Datenqualität. Die Kontrollelemente erlauben die Einstellung der in Tabelle 5.2 aufgeführten Parameter (1). Über die Menüelemente wird die Bewertung und die Visualisierung der Datenqualität aufgerufen (2).

5.2.3 Bewertung Modellqualität

In Kapitel 3 wurde die Verlaufsvalidierung vorgestellt. Sie kann entweder zur lokalen Bewertung eines Modells verwendet werden oder durch die

lokale Bewertung an unterschiedlichen Stellen zur globalen Bewertung eines Modells.

In der Ansicht *Modellqualität* von DaMoQ kann für die globale Bewertung eine Anzahl an Stellen gewählt werde, an denen das bereits entworfene und angewandte Modell zu untersuchen ist. Die Stellen im Eingangsraum werden randomisiert ausgewählt. Alternativ können Eingangsvektoren hinterlegt werden, um eine lokale Bewertung zu ermöglichen.

Die Ergebnisse der Modellbewertung werden gespeichert und optional wie in Abbildung 5.2 visualisiert.

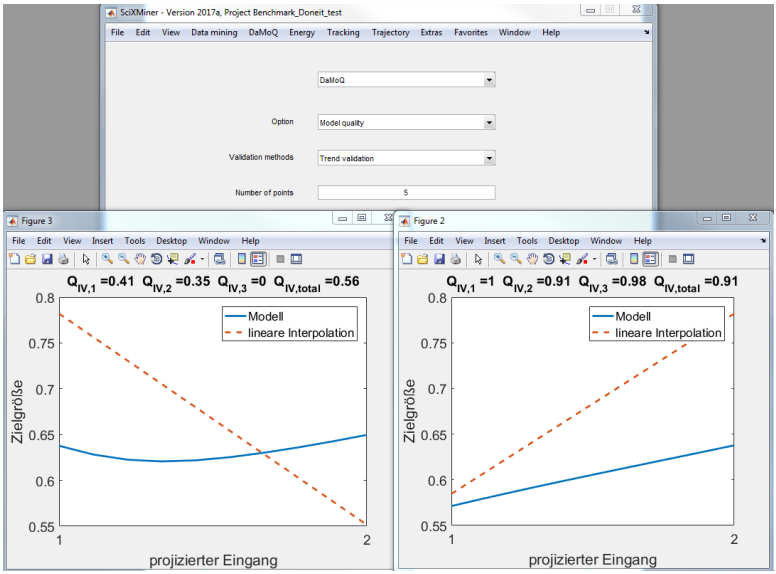

Abbildung 5.2: Übersicht von DaMoQ für die Bewertung der Modellqualität sowie Visualisierungen der Verlaufsvalidierung

5.3 Generalisierte sequentielle minimale Optimierung

Der GSMO-Algorithmus wurde in MATLAB implementiert. Die Implementierung des analytischen Teils der GSMO findet sich in Anhang A.2. Um den GSMO-Algorithmus auf Echt-Welt-Probleme anwenden zu können, ist zunächst eine Auswahl oder Optimierung der Heuristik zur Auswahl des betrachteten Indexpaares (u, v) notwendig. Weiterhin kann eine temporäre Berechnung der ausschließlich im aktuellen analytischen Schritt benötigten Einträge der Kernelmatrix den Speicheraufwand reduzieren. Eine Implementierung in C/C++ kann gegenüber einer reinen MATLAB-Implementierung weitere Geschwindigkeitsvorteile erreichen.

5.4 SciXMiner-Toolbox „Hybrid Models“

Die MATLAB-Toolbox SciXMiner ermöglicht nach Auswahl eines Lerndatensatzes und einer Modellstruktur die Parameterschätzung und die anschließende Anwendung eines Regressionsmodells. Weiterhin stehen Validierungsverfahren (z.B. Kreuzvalidierungen) zur Verfügung.

Im Entwurfsprozess eines geeigneten Regressionsmodells werden häufig viele Modellstrukturen hinsichtlich ihrer Eignung untersucht - das bedeutet der Anwender stellt eine neue Modellstruktur ein, validiert sie und vergleicht die Ergebnisse mit anderen Modellstrukturen. Mit Hilfe sogenannter Makro-Dateien können die händischen Einstellungen automatisiert werden.

Für die Implementierung eines automatisierten Entwurfs hybrider Modelle wurde eine GUI entwickelt, die den Entwurf von Regressionsmodellen vereinfacht und automatisiert.

Abbildung 5.3 zeigt die grafische Benutzerschnittstelle (engl. *graphical user interface*, GUI) der SciXMiner-Toolbox Hybrid Models. Nach dem Einstellen der Eingangsgrößen und der Zielgröße in SciXMiner können mit der neuen Benutzeroberfläche die Modellstrukturen für den Modell- und 1K-SVM-Pool eingestellt werden.

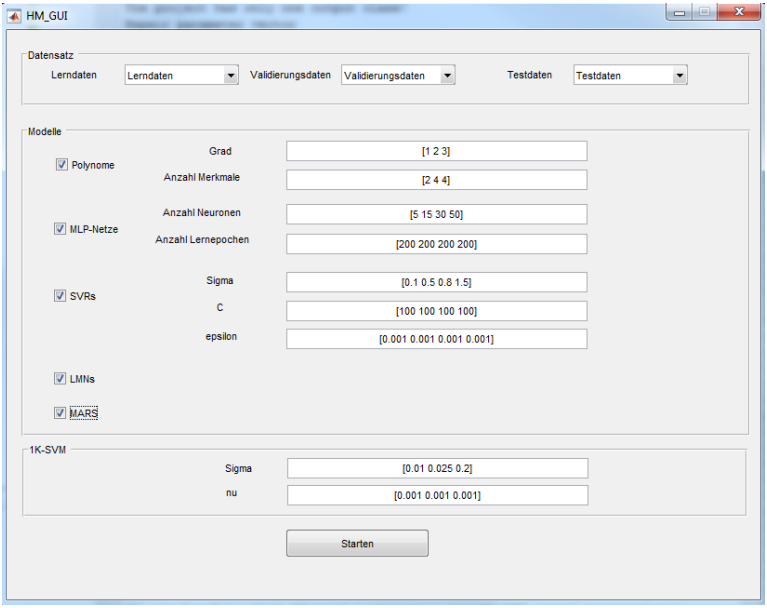

Abbildung 5.3: GUI zur Konfiguration aller zu untersuchenden Modellstrukturen und der Metaparametrierungen der 1K-SVMs

Weiterhin erfolgt mit Hilfe der in SciXMiner hinterlegten linguistischen Termen der Ausgangsgrößen die Einteilung in Lern-, Validierungs- und Testdaten.

Der automatisierte Entwurf optimiert das hybride Modell und selektiert das beste herkömmliche Modell gemäß Abschnitt 4.2.3. Anschließend werden beide Modelle anhand der Testdaten verglichen. Die Ergebnisse und die (Teil-)Modelle werden hinterlegt und die Ergebnisse visualisiert. Damit wird der der Entwurfsaufwand für den Anwender minimiert.

6 Anwendung

6.1 Labyrinthdichtungen

6.1.1 Problemstellung

Turbomaschinen wandeln entweder als Kraftmaschine die Energie eines strömenden Fluids in mechanische Energie um (Turbinen) oder verrichten als Arbeitsmaschine an einem Fluid Arbeit (z.B. Pumpen). In beiden Fällen wird die Effizienz einer Turbomaschine u.a. durch die in den Einzelkomponenten verwendeten Labyrinthdichtungen beeinflusst [161]. Für ihre Auslegung und Optimierung wird ein grundlegendes Verständnis des Durchflussverhaltens und der Wärmeübergangscharakteristik benötigt. Die Zusammenhänge sind allerdings nicht vollständig bekannt. In [1] werden deshalb Forschungsarbeiten im Bereich des Durchflussverhaltens von Labyrinthdichtungen zur statistischen Untersuchung und datengetriebenen Modellierung in einem Datensatz ($N = 16225$) zusammengefasst. Die darin enthaltenen Datentupel ordnen Geometriekonfigurationen sowie strömungsmechanische und thermodynamische Randbedingungen von unterschiedlichen Labyrinthdichtungen bzw. entsprechender Messungen als Eingangsgrößen einen Durchflussbeiwert C_d als Zielgröße zu.

Darauf aufbauend werden in [161–163] experimentelle und numerische Untersuchungen durch Multiple Lineare Regressionen und künstliche neuronale Netze unterstützt. Da die datengetriebenen Modelle in bisherigen Validierungen den Durchflussbeiwert unbekannter (d.h. nicht in den Lerndaten verfügbarer) Eingangsgrößen sehr genau prädizieren konnten, werden sie als kostengünstiger und schnell zu berechnender Ersatz für Simulationen der numerischen Strömungsmechanik (engl. *computational fluid dynamics*, CFD) oder zusätzliche Messungen gesehen. Allerdings kommt

es bei der Anwendung der Modelle vereinzelt zu einem unerwarteten Modellversagen, d.h. in manchen Fällen weichen die Prädiktionen deutlich von plausiblen Wertebereichen des Durchflussbeiwerts ab.

Es wird vermutet, dass solch lokales Modellversagen mit einer eingeschränkten Datenabdeckung zusammenhängt. Bislang wurden dazu keine Untersuchungen durchgeführt.

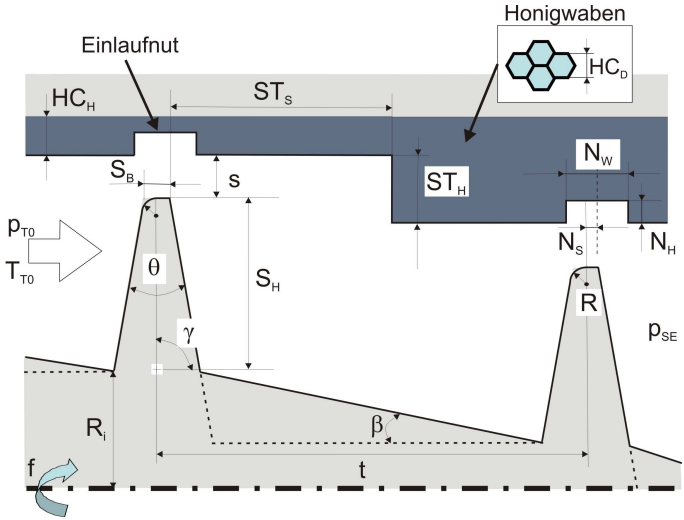

Abbildung 6.1: Schematische Darstellung einer Labyrinthdichtung mit verschiedenen Kenngrößen nach [1]

In Abbildung 6.1 sind einige geometrische Parameter sowie thermodynamische Einflussgrößen zur Abbildung des Durchflussbeiwerts C_d von Labyrinthdichtungen veranschaulicht. In Tabelle 6.1 werden die Bezeichner der im Datensatz erfassten Kenngrößen ihren Bedeutungen zugeordnet. Zur datengetriebenen Modellierung wird der modifizierte (mit der Teilung t in dimensionslose geometrische Verhältnisse überführte) Datensatz auf Einheitsintervalle normiert. Es ergeben sich damit zehn Eingangsgrößen. Zur Veranschaulichung der eingeschränkten Datenabdeckung werden einige der Eingangsgrößen mit Hilfe von Streuwolkendiagrammen in Abbildung 6.2 visualisiert.

Bezeichner	Einheit	Beschreibung
ST_{H}	$[mm]$	Stufenhöhe
ST_{S}	$[mm]$	Stufenshift
HC_{d}	$[mm]$	Honigwabendurchmesser
N_{W}	$[mm]$	Nutweite
N_{S}	$[mm]$	Nutversatz
n	$-$	Spitzenanzahl
S_{H}	$[mm]$	Spitzenhöhe
S_{B}	$[mm]$	Spitzenbreite
s	$[mm]$	Spaltweite
π	$-$	Druckverhältnis zwischen Eintrittsdruck p_{T0} und Austrittsdruck p_{SE}
t	$[mm]$	Teilung

Tabelle 6.1: Potentielle Einflussgrößen von Labyrinthdichtungen auf den Durchflussbeiwert C_{d}

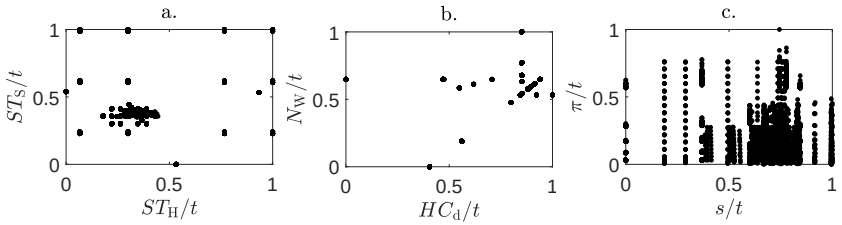

Abbildung 6.2: Streuwolkendiagramme verschiedener Eingangsgrößen der Labyrinthdichtungen

6.1.2 Bewertung der Datenqualität

Mit Hilfe der in Abschnitt 5.2 vorgestellten SciXMiner-Toolbox DaMoQ wird der gesamte Datensatz gemäß der in Kapitel 2 vorgestellten Bewertungsmaße für Datenqualität untersucht. Die Übersichten über die Datenqualität des Datensatzes bzw. der einzelnen Eingangsgrößen sind in Abbildung 6.3 zu sehen. Die niedrigen mittleren Bewertungen für fast alle Phänomene bedeuten, dass nicht nur eine Eingangsgröße oder eine bivaria-

te Projektion eine besonders ungleichmäßige Verteilung aufweist, sondern die Mehrheit der Eingangsgrößen. Die beste Bewertung erreicht der Datensatz hinsichtlich Korrelationen, d.h. Merkmalsreduktionsverfahren wie z.B. eine Hauptkomponentenanalyse werden nur geringfügig Vorteile im Modellentwurf bringen. Die schlechteste Bewertung erreicht der Datensatz hinsichtlich Ausreißern.

Ein Entfernen der Ausreißer kann zu einem kompakteren und gleichmäßiger abgedeckten Eingangsraum führen, für den Extrapolationen ohne aufwändige statistische Verfahren erkannt werden können. Weiterhin können anstatt eines globalen Modells auch lokale Modelle für einzelne Konfigurationen oder Cluster erstellt werden, wie bereits in [1, 162] geschehen. Cluster sind beispielsweise in Abbildung 6.2a deutlich zu erkennen.

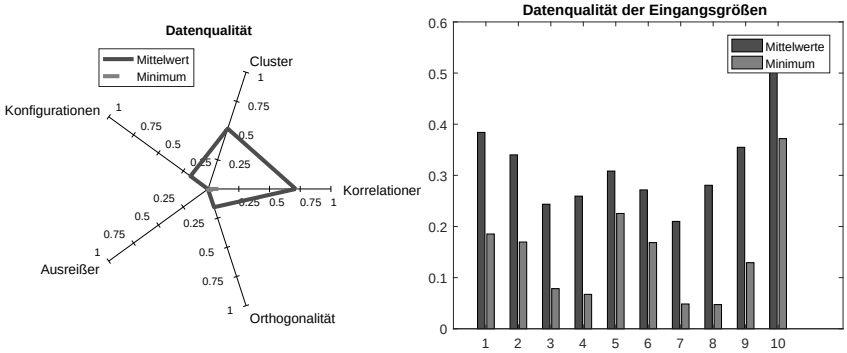

Abbildung 6.3: Übersicht über die Datenqualität gemäß DaMoQ-Toolbox als Übersicht (links) und für die Eingangsgrößen $\frac{ST_H}{t}$ (1), $\frac{ST_S}{t}$ (2), $\frac{HC_d}{t}$ (3), $\frac{N_W}{t}$ (4), $\frac{N_S}{t}$ (5), n (6), $\frac{S_H}{t}$ (7), $\frac{S_B}{t}$ (8), $\frac{s}{t}$ (9) und π (10) (rechts)

6.1.3 Bewertung der Modellqualität

Um die bisherigen Untersuchungen und Probleme zu reproduzieren und zu verdeutlichen, werden auf dem gesamten Datensatz Kreuzvalidierungen für unterschiedlich komplexe MLP-Netze durchgeführt. Dazu wird die Anzahl an Neuronen in der verdeckten Schicht von 5 bis zu 40 Neuronen variiert.

Weiterhin wird die in Kapitel 3 vorgestellte Verlaufsvalidierung verwendet, um lokale Überanpassung für die MLP-Netze zu erkennen.

Abbildung 6.4 zeigt die Ergebnisse der Kreuzvalidierungen und Abbildung 6.5 die Ergebnisse der Verlaufsvalidierung. Der mittlere absolute Fehler nimmt bei den Kreuzvalidierungen mit zunehmender Komplexität der MLP-Netze ab.

Allerdings fallen auch die Bewertungskriterien der Verlaufsvalidierung bereits bei der Verwendung von 10 Neuronen in der verdeckten Schicht stark ab. Das deutet auf Überanpassung hin, die von Kreuzvalidierungen nicht erkannt wird.

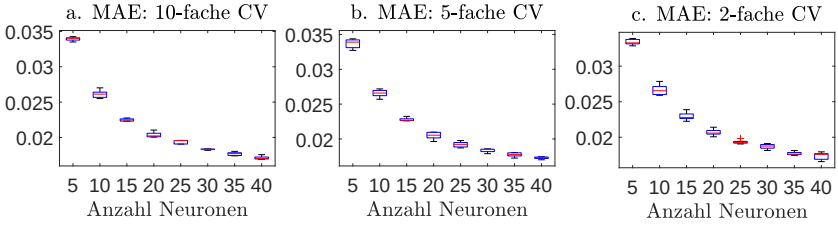

Abbildung 6.4: Ergebnisse von jeweils 5 Kreuzvalidierungen für MLP-Netze mit zunehmender Komplexität

Dass die Fehler für Kreuzvalidierungen bei zunehmender Modellkomplexität abnehmen, die Verlaufsvalidierung aber durch die niedrigen Werte der Bewertungskriterien ausgeprägte Nichtlinearitäten zwischen den Eingangsvektoren der Lerndaten indizieren, deutet darauf hin, dass die Lerndaten näherungsweise Duplikate der Eingangsvektoren enthalten und Kreuzvalidierungen ausschließlich in der Nähe der Lerndaten validieren und keine Generalisierungsfähigkeit überprüfen.

Es ist daher anzunehmen, dass Modelle, die auf Grundlage von Kreuzvalidierungen selektiert werden, nur in unmittelbarer Umgebung der Lerndaten hohe Prädiktionsgüten erreichen.

Im Folgenden wird deshalb untersucht, ob das automatisierte Entwurfsverfahren für hybride Modelle eine geeignete Strategie darstellt, Modelle zur Abbildung des Durchflussverhaltens von Labyrinthdichtungen zu erstellen.

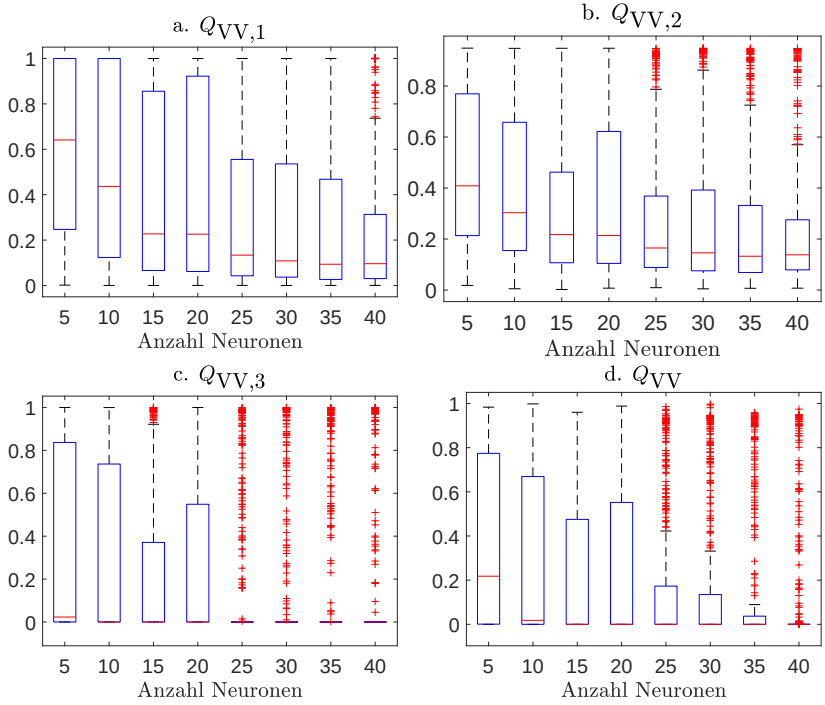

Abbildung 6.5: Ergebnisse der Verlaufsvalidierung für MLP-Netze mit zunehmender Komplexität

Da in Interpolationsbereichen bereits herkömmliche Modelle akzeptable Ergebnisse liefern und die abzubildenden Zusammenhänge in den Extrapolationsbereichen unbekannt sind, werden SVRs mit Vorwissen gemäß Tabelle 4.11 nicht untersucht. Weiterhin kann die Größe des Datensatzes zu einem hohen Rechen- und Speicheraufwand führen. Vor allem die Anzahl der virtuellen Datentupel zur Integration des Vorwissens wird bei 10 Eingangsgrößen hoch sein.

6.1.4 Hybride Modelle

Nach der Aufteilung des Datensatzes wird gemäß Abschnitt 4.2.3 bzw. dem in Abbildung 4.9 gezeigten Schema sowohl ein Modell f_{Ref} durch herkömmliche Modellselektion als auch ein hybrides Modell f_{Hybrid} mit Hilfe der Optimierung hybrider Modelle mit den Validierungsdaten bestimmt und anschließend mit Hilfe der Testdatensätze verglichen. Als Bewertungskriterium dient jeweils der mittlere absolute Fehler. Als mögliche Modelle werden Polynome (1. Grades, 10 Merkmale), MLP-Netze (5,10,...,40 Neuronen), SVRs ($\sigma = 0.2, 0.6, \ldots, 1.4$), MARS und LMNs (Partitionierung mit LOLIMOT) verwendet. Da sich die Ergebnisse vor allem in Abhängigkeit des zweiten Testdatensatzes deutlich unterscheiden können, wird die Einteilung des Datensatzes und die sich anschließende Modellbildung mehrfach wiederholt.

6.1.5 Ergebnisse

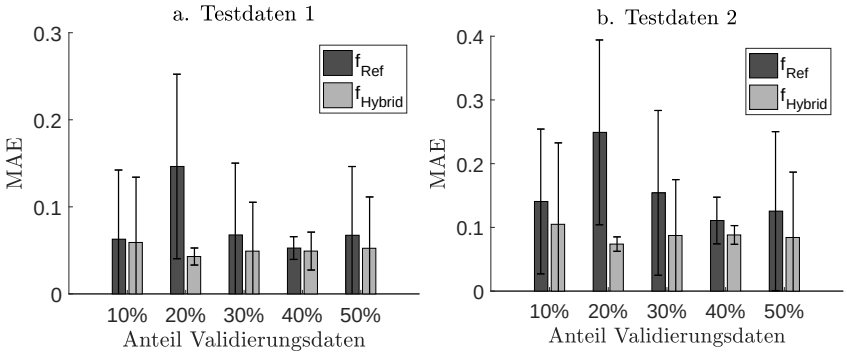

Abbildung 6.6: Ergebnisse über die Testdatensätze bei unterschiedlicher Aufteilung in Lern- und Validierungsdaten.

Abbildung 6.6 zeigt Mittelwert und Standardabweichung des MAE über 10 Wiederholungen des Validierungsschemas bei unterschiedlicher Aufteilung in Lern- und Validierungsdaten. Für beide Testdatensätze und bei allen untersuchten Aufteilungen in Lern- und Validierungsdaten zeigen die hybriden Modelle geringere Modellfehler als die herkömmlichen Modelle.

Zu einer deutlichen Verbesserung durch die Verwendung hybrider Modelle kommt es tendenziell bei einem höheren Anteil an Validierungsdaten. Das ist darauf zurückzuführen, dass bei einem zu geringen Anteil an Validierungsdaten eher Interpolationen geprüft werden und zur Optimierung des hybriden Modells nicht ausreichend Informationen vorhanden sind.

Weiterhin lässt sich mit Hilfe der 1K-SVM die erwartete Zuverlässigkeit des Modells vorhersagen.

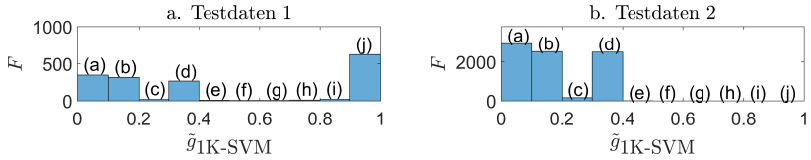

Abbildung 6.7: Häufigkeiten F der diskretisierten Ergebnisse der Ein-Klassen-Klassifikation für die Testdaten eines Validierungsdurchlaufs

Abbildung 6.7 zeigt die Ergebnisse des Koordinators $\tilde{g}_{\text{1K-SVM}}(\boldsymbol{x})$ für einen Durchlauf der Validierung.

Es ist zu sehen, dass für den ersten Testdatensatz etwa die Hälfte der Prädiktionen als Interpolationen ($\tilde{g}_{\text{1K-SVM}}(\boldsymbol{x}) \approx 1$) klassifiziert werden und die andere Hälfte als unterschiedlich deutliche Extrapolationen ($\tilde{g}_{\text{1K-SVM}}(\boldsymbol{x}) < 0.5$). Für den zweiten Testdatensatz werden dagegen ausschließlich Extrapolationen erkannt.

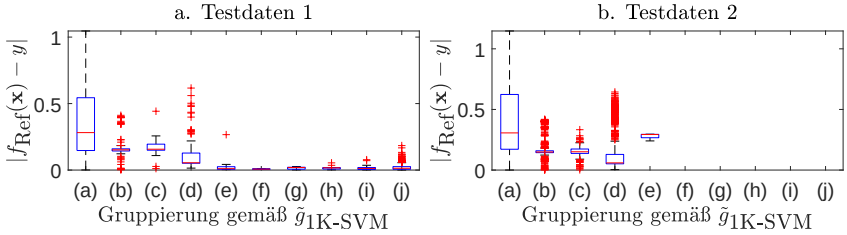

Abbildung 6.8: Absolute Fehler von f_{Ref} für die Testdaten

In Abbildung 6.8 werden die absoluten Fehler des herkömmlichen Modells in Abhängigkeit des Inter- bzw. Extrapolationsgrades dargestellt.

Es ist der vermutete Trend, dass die Prädiktionsgüte mit dem Extrapolationsgrad sinkt, zu beobachten. Für Eingangsvektoren mit $\tilde{g}_{\text{1K-SVM}}(\boldsymbol{x}) \approx 0$ sind deutlich höhere Modellfehler zu erkennen als für Interpolationen.

Für den zweiten Testdatensatz zeigt sich zwar ein leicht erhöhter Modellfehler für die Gruppe, die am ehesten als Interpolationen zu verstehen sind (Gruppe (e)), dabei handelt es sich aber eher um ein statistisches Problem, da für die Gruppe nur wenige Eingangsvektoren vorliegen und der Effekt vermutlich nicht signifikant ist.

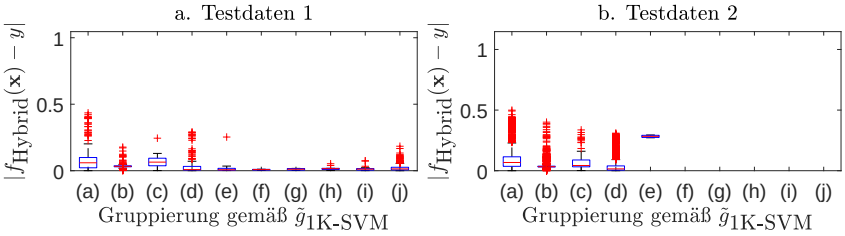

Abbildung 6.9: Absolute Fehler von f_{Hybrid} für die Testdaten

Abbildung 6.9 zeigt die Ergebnisse für das hybride Modell.

Auch beim hybriden Modell sinkt die Prädiktionsgüte mit steigendem Extrapolationsgrad, sie ist jedoch besonders für deutliche Extrapolationen höher als die des herkömmlichen Modells.

Die Tabellen 6.2 und 6.3 zeigen für jeweils einen Validierungsdurchlauf zum einen die mittleren absoluten Fehler und zum anderen die Signifikanzen, mit denen das Interpolationsmodell in Interpolationsbereichen dem Extrapolationsmodell überlegen ist und vice versa.

6.1.6 Zusammenfassung

Es kommt aufgrund einer ungleichmäßigen Datenabdeckung des Eingangsraums zur Selektion überangepasster Modelle, die versagen, wenn Prädiktionen abseits der Lerndaten erfolgen.

Anteil Val.-Daten	f_{Ref}	f_{Hybrid}		f_{Int}	f_{Ext}
10%	0.016	0.015	Interpolation	$+^{***}$	$-$
			Extrapolation	$-$	$+^{**}$
20%	0.258	0.054	Interpolation	$-$	$+$
			Extrapolation	$-$	$+^{***}$
30%	0.163	0.114	Interpolation	$+^{***}$	$-$
			Extrapolation	$-$	$+^{***}$
40%	0.041	0.028	Interpolation	$-$	$+$
			Extrapolation	$-$	$+^{***}$
50%	0.018	0.016	Interpolation	$+^{***}$	$-$
			Extrapolation	$-$	$+^{***}$

Tabelle 6.2: Vergleich herkömmlicher Modelle mit hybriden Modellen über den ersten Testdatensatz für jeweils einen Validierungsdurchlauf der unterschiedlichen Aufteilungen in Lern- und Validierungsdaten

Anteil Val.-Daten	f_{Ref}	f_{Hybrid}		f_{Int}	f_{Ext}
10%	0.096	0.026	Interpolation	$-$	$+$
			Extrapolation	$-$	$+^{***}$
20%	0.404	0.077	Interpolation	$-$	$+$
			Extrapolation	$-$	$+^{***}$
30%	0.296	0.188	Interpolation	$+^{***}$	$-$
			Extrapolation	$-$	$+^{***}$
40%	0.152	0.082	Interpolation	$-$	$+$
			Extrapolation	$-$	$+^{***}$
50%	0.062	0.017	Interpolation	$-$	$+$
			Extrapolation	$-$	$+^{***}$

Tabelle 6.3: Vergleich herkömmlicher Modelle mit hybriden Modellen über den zweiten Testdatensatz für jeweils einen Validierungsdurchlauf der unterschiedlichen Aufteilungen in Lern- und Validierungsdaten

Es wurde gezeigt, dass sich die automatisierte Entwurfsmethodik für hybride Modelle dazu eignet, Modelle zu erzeugen, die in der Nähe der Lerndaten die Genauigkeit herkömmlicher, unter Umständen überangepasster Modelle, erreichen und in Extrapolationsbereichen robuste Prädiktionen liefern.

Weiterhin ermöglicht die Struktur der hybriden Modelle bei ihrer Anwendung eine Abschätzung der zu erwartenden Zuverlässigkeit.

6.2 Kalibrierung biomedizinischer Mensch-Maschine-Schnittstellen

6.2.1 Problemstellung

In der Medizintechnik werden häufig myoelektrische Signale (EMG-Signale), die sich durch die willentliche Aktivierung eines Muskels beeinflussen lassen, zur Steuerung von beliebigen Endeffektoren (z.B. Prothesen oder Rollstühlen [164–166]) verwendet. Weiterhin finden myoelektrische Signale Anwendung in der medizinischen Rehabilitation [167–169]. Für eine hohe Nutzerakzeptanz erfolgt die Messung der Signale nicht-invasiv über Oberflächenelektroden. Im Gegensatz zu invasiven Messungen ist die Signalqualität bei Oberflächenelektroden geringer. Es werden nicht nur die gewünschten Signale gemessen, sondern es kommt zur ungewollten Aufzeichnung von anderen Signalquellen in der Umgebung der Elektrode [170, 171]. Ein weiteres Problem stellen Koaktivierungen dar. Dabei kommt es bei der willentlichen Aktivierung eines Muskels zur simultanen ungewollten Aktivierung eines anderen Muskels. Wenn die gemessene Aktivierung beider Muskeln als Signalkanäle verwendet wird, führen solche Koaktivierungen zu Abhängigkeiten zwischen den Kanälen.

Der Betriebsmodus einer MMS, das heißt die gesamte Signalverarbeitung von der Bewegungsabsicht des Anwenders bis zur tatsächlichen Bewegung des Endeffektors, lässt sich schematisch wie in Abbildung 6.10 darstellen.

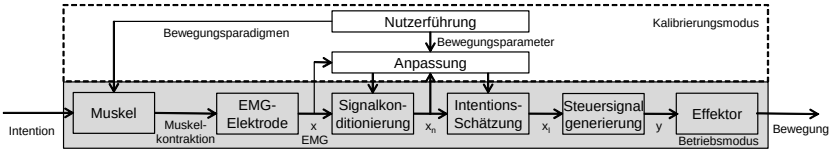

Abbildung 6.10: Signalfluss in Mensch-Maschine-Schnittstellen [3]

Eine Bewegungsabsicht, die Intention, führt zur Kontraktion eines Muskels. EMG-Elektroden messen die Muskelkontraktion und führen sie in digitaler Form einer Recheneinheit (z. B. Mikrocontroller) zu. Dort werden die Signale konditioniert und der Einfluss anwenderabhängiger Kontraktionsstärken oder Ausreißer minimiert. Aus den konditionierten Signalen wird die Intention des Anwenders geschätzt, um ein Steuersignal für einen Effektor (z. B. Rollstuhl) zu generieren. Die manuelle Anpassung durch einen Experten oder eine automatisierte Anpassung individualisiert die Schnittstelle durch Adaption der Parameter in Signalkonditionierung und Intentionsgenerierung. Hierfür sind Lerndaten notwendig, die eine Nutzerführung durch Vorgabe von Bewegungstrajektorien und den zugehörigen Bewegungsparametern (Sollwerte) automatisiert generieren kann.

In [172–174] wird eine neuartige MMS vorgestellt, mit der hochgradig querschnittgelähmte Menschen mit Hilfe der äußeren Ohrmuskulatur einen Rollstuhl steuern können. Dazu werden zwei unabhängige EMG-Signale (am linken und rechten Ohr, $i = 1, 2$) mit einer Abtastfrequenz von 200 Hz gemessen und in ein digitales Signal $x_i[k]$ überführt. Das Signal wird in Echtzeit durch drei aufeinanderfolgende Operatoren (mittelwertfreie Gleichrichtung, Tiefpassfilterung, Normalisierung) verarbeitet [175]. Abbildung 6.11 visualisiert die Effekte der Operatoren auf Rohsignale.

Die Gleichrichtung korrigiert das Signal um seinen statischen und hardwarebedingten Mittelwert-Gleichanteil $x_{\mathrm{mean},i}$ und bildet einen Betrag

$$x_{\mathrm{r},i}[k] = 2 \cdot \left| x_i[k] - x_{\mathrm{mean},i} \right|. \tag{6.1}$$

Die Tiefpassfilterung nutzt einen *root-mean-square*-Filter (Parameter $m = 20$) in Kombination mit einem *infinite-impulse-response*-Filter (Parameter $a = 0.9$) und liefert so eine Information über den Energieinhalt des Signals

$$x_{\mathrm{f},i}[k] = a \cdot x_{\mathrm{f},i}[k-1] + (1-a) \cdot \sqrt{\frac{1}{m+1} \sum_{j=0}^{m} x_{\mathrm{r},i}^2[k-j]}. \tag{6.2}$$

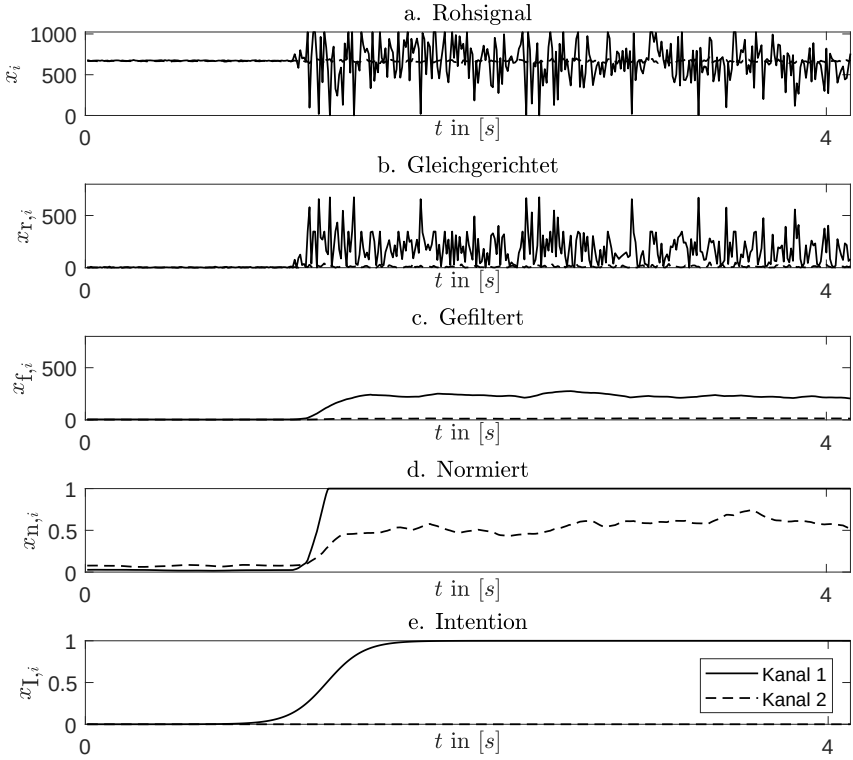

Abbildung 6.11: Rohsignal sowie durch signalverarbeitende Operatoren transformierte Signale von zwei Kanälen

Die Normalisierung skaliert das gefilterte Signal in Abhängigkeit von den Extremwerten $x_{\mathrm{f,max},i}$ und $x_{\mathrm{f,min},i}$ auf einen Bereich von 0 bis 1

$$x_{\mathrm{n},i}[k] = \begin{cases} 0 & \text{wenn } x_{\mathrm{f},i}[k] < x_{\mathrm{f,min},i} \\ \frac{x_{\mathrm{f},i}[k] - x_{\mathrm{f,min},i}}{x_{\mathrm{f,max},i} - x_{\mathrm{f,min},i}} & \text{wenn } x_{\mathrm{f,min},i} \leq x_{\mathrm{f},i}[k] \leq x_{\mathrm{f,max},i} \\ 1 & \text{wenn } x_{\mathrm{f},i}[k] > x_{\mathrm{f,max},i}. \end{cases} \tag{6.3}$$

Je nachdem, wie $x_{\mathrm{f,max},i}$ und $x_{\mathrm{f,min},i}$ gewählt werden, kann durch (6.3) eine Ausreißerbereinigung stattfinden. Üblich ist das Ausschließen der unteren und oberen 5% des Wertebereichs von $x_{\mathrm{f},i}[k]$. $x_{\mathrm{f,min},i}$ ist folglich ein

Schwellwert, der zur Unterscheidung zwischen Relaxation und willentlicher Kontraktion eines Muskels dient. Die Erlangung der Parameter $x_{\mathrm{f,max},i}$ und $x_{\mathrm{f,min},i}$ kann als unilaterale Kalibrierung bezeichnet und $x_{\mathrm{n},i}[k]$ als Muskelaktivität interpretiert werden, wobei große Werte für eine starke Muskelaktivität stehen [175].

Liegen in den Signalen Koaktivierungen vor, sind in einem nächsten Schritt die Schätzungen der intentionierten Signale des Anwenders zu bestimmen und die Signalanteile der Koaktivierungen zu entfernen. Hierzu sind i. A. nichtlineare Transformationen $f_i(\cdot)$ zu verwenden

$$\hat{x}_{\mathrm{I},i}[k] = f_i(x_{\mathrm{n},1}[k], x_{\mathrm{n},2}[k]). \tag{6.4}$$

Ziel ist es, z.B. mit Hilfe von Regressionsmodellen, die vom Anwender intentionierten Signale $x_{\mathrm{I},i}[k]$ zu berechnen. Abbildung 6.11e zeigt z.B. die Intention des Nutzers, die aufgrund der Vorgabe von Sollwerten an den Anwender bekannt sind. Die normierten Signale aus Abbildung 6.11d widersprechen den Intentionen allerdings. Ziel der nichtlinearen Transformationen ist in diesem Beispiel die Eliminierung der (Ko-)Aktivierung von $x_{\mathrm{n},2}$.

Daraus lassen sich die Zielgrößen Translationsgeschwindigkeit $v_{\mathrm{trans}}[k]$ und Rotationsgeschwindigkeit $v_{\mathrm{rot}}[k]$ eines Rollstuhls berechnen. Mit $\mathbf{x}[k] = (x_{\mathrm{I},1}[k], x_{\mathrm{I},2}[k])^T$ und $\mathbf{y} = (v_{\mathrm{trans}}[k], v_{\mathrm{rot}}[k])^T$ ist es das Ziel, einen Zusammenhang der Form

$$\mathbf{y} = g(\mathbf{x}) \tag{6.5}$$

zu formulieren[1].

6.2.2 Entwicklung einer bilateralen Kalibrierung

Es wird eine grafische Benutzeroberfläche vorgestellt, die dem Anwender die intuitive Kalibrierung einer zweikanaligen MMS ermöglicht. Details finden

[1] Im idealen Fall kann beispielsweise

$$\mathbf{y} = \mathbf{A}\mathbf{x} \text{ mit } \mathbf{A} = \begin{bmatrix} 0.5 & 0.5 \\ 1 & -1 \end{bmatrix}$$

verwendet werden. Auf diese Weise ergibt sich die Translationsgeschwindigkeit aus der Summe zweier gemessener Signale, die Rotationsgeschwindigkeit aus deren Differenz.

sich in [3]. Die Kalibrierung erfolgt zunächst univariat zur Bestimmung von $x_{\mathrm{f,min},i}$ und $x_{\mathrm{f,max},i}$. Anschließend erfolgt die sogenannte bilaterale Kalibrierung zur Bestimmung der Regressionsmodelle $f_i(\cdot)$. Dazu wird der Nutzer durch eine Datenerhebung geleitet, um Lerndaten für den Modellentwurf zu erheben.

Abbildung 6.12 zeigt die GUI während der Kalibrierungsroutine.

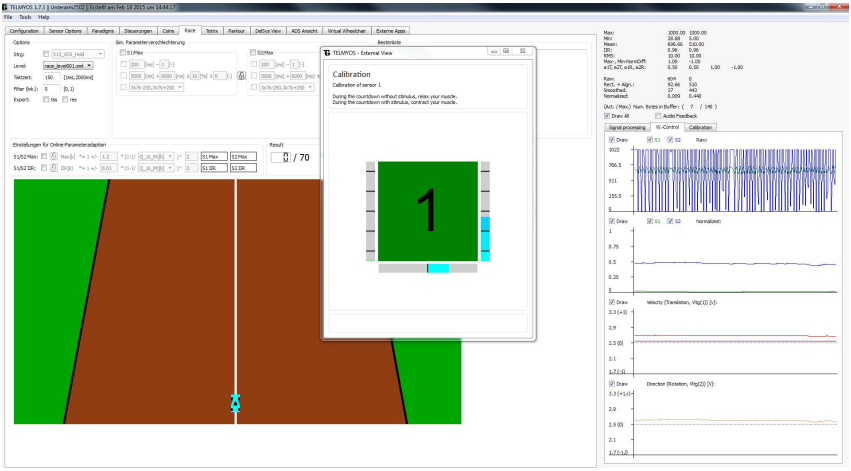

Abbildung 6.12: GUI während der Kalibrierungsroutine mit Anweisungen an den Anwender zur Erhebung der Lerndaten [3]

Die Regressionsmodelle schätzen auf Basis der gemessenen und durch Koaktivierung belasteten Signale die vom Nutzer beabsichtigten Signale, d.h. seine Intention.

Die Anzahl der Kalibrierungsschritte in der Datenerhebung stellt einen Kompromiss aus benötigten Informationen, um eine zuverlässige Intentionsschätzung zu entwerfen, und aus einer möglichst geringen Belastung des Anwenders dar. Zudem sind nur bestimmte Intentionen robust ansteuerbar: Eine Viertelkontraktion oder ein Vielfaches der Kontraktion eines Muskels im Vergleich zu einem anderen können die Anwender nicht leisten.

Beschreibung	Sollwerte		Istwerte	
max. Kontr. $x_{\mathrm{n},1}$	$x_{\mathrm{I},1,1} = 1$	$x_{\mathrm{I},2,1} = 0$	$x_{\mathrm{n,med},1,1}$	$x_{\mathrm{n,med},2,1}$ +
halbe Kontr. $x_{\mathrm{n},1}$	$x_{\mathrm{I},1,2} = 0.5$	$x_{\mathrm{I},2,2} = 0$	$x_{\mathrm{n,med},1,2}$	$x_{\mathrm{n,med},2,2}$ +
max. Kontr. $x_{\mathrm{n},2}$	$x_{\mathrm{I},1,3} = 0$	$x_{\mathrm{I},2,3} = 1$	$x_{\mathrm{n,med},1,3}$	$x_{\mathrm{n,med},2,3}$ +
halbe Kontr. $x_{\mathrm{n},2}$	$x_{\mathrm{I},1,4} = 0$	$x_{\mathrm{I},2,4} = 0.5$	$x_{\mathrm{n,med},1,4}$	$x_{\mathrm{n,med},2,4}$ +
max. Kokontr.	$x_{\mathrm{I},1,5} = 1$	$x_{\mathrm{I},2,5} = 1$	$x_{\mathrm{n,med},1,5}$	$x_{\mathrm{n,med},2,5}$ +
halbe Kokontr.	$x_{\mathrm{I},1,6} = 0.5$	$x_{\mathrm{I},2,6} = 0.5$	$x_{\mathrm{n,med},1,6}$	$x_{\mathrm{n,med},2,6}$ +

Tabelle 6.4: Übersicht der Kalibrierungsschritte zur bilateralen Kalibrierung: Sollwerte, die von der Nutzerführung anzufordern sind; Istwerte, die aus den aufgenommenen Zeitreihen extrahiert werden. Die Laufindizes zählen kontrahierten Muskel (Signalkanal) und Nummer des Kalibrierungsschritts der bilateralen Kalibrierung.

Deshalb wird sich zunächst darauf beschränkt, sechs einfach zu reproduzierende Kombinationen aus Relaxation, maximaler und halbierter Kontraktion beider Muskeln anzufordern.

Tabelle 6.4 zeigt die Sollwerte für die Kalibrierungsschritte. Die aufgezeichneten Signale werden als $x_{\mathrm{n,bikalib},i,j}[k]$ bezeichnet[2].

Die Abbildungen 6.13a und 6.13b zeigen aufgezeichnete Zeitreihen für zwei Kalibrierungsschritte.

Da die Zeitreihen unbeabsichtigte Schwankungen aufweisen und die geforderten Kontraktionen häufig erst mit Verzögerung erfolgen, wird zum Entwurf der Regressionsmodelle aus jeder aufgezeichneten Zeitreihe $x_{\mathrm{n,bikalib},i,j}[k]$ der Median $x_{\mathrm{n,med},i,j}$ als robuster Mittelwert der normierten Sensorwerte für einen Kalibrierungsschritt extrahiert. Die extrahierten Werte entsprechen den Istwerten der bilateralen Kalibrierung und werden in Tabelle 6.4 den Sollwerten zugeordnet. Die Datentupel $(\boldsymbol{x}_{\mathrm{n,med},j}, x_{\mathrm{I},i,j})$ gehen als Lerndaten in den Entwurf der Regressionsmodelle der Intentionsschätzung ein, d.h. $\boldsymbol{x}_{\mathrm{n,med},j}$ entsprechen Eingangsvektoren und die Sollwerte $x_{\mathrm{I},i,j}$ den Zielgrößen für die beiden Regressionsmodelle $f_i(x_{\mathrm{n},1}, x_{\mathrm{n},2})$. Abbildung 6.13 zeigt die Überführung der Zeitreihen in den Eingangsraum sowie die Extraktion der Medianwerte, die eine Zeitreihe aggregieren und repräsentieren.

[2] i zählt den Kanal und j den Kalibrierungsschritt

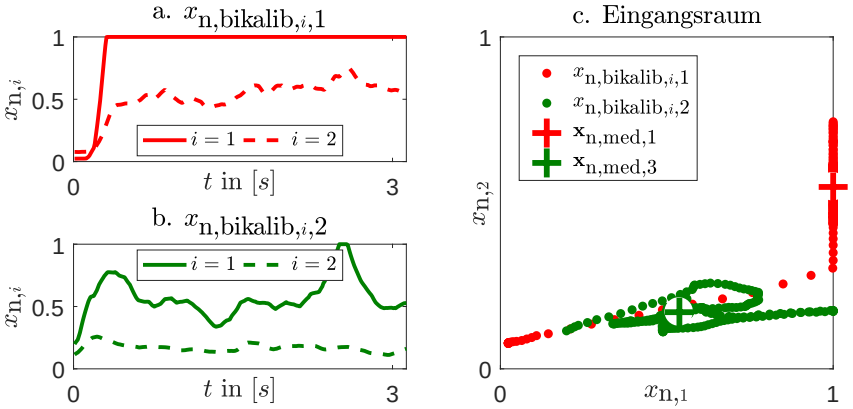

Abbildung 6.13: Überführung der Zeitreihen in den Eingangsraum der Regressionsmodelle

Weiterhin zeigt Abbildung 6.14 die Lerndatensätze von f_1 und f_2 für den idealen Fall, d.h. die Eingangsvektoren entsprechen den Sollwerten beider Signalkanäle.

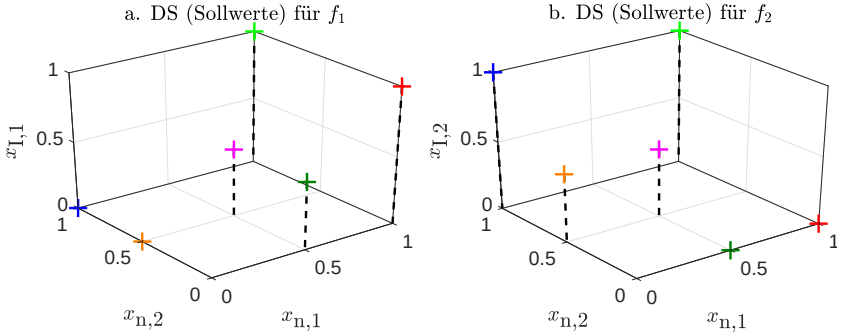

Abbildung 6.14: Idealer Fall der Datensätze zur bilateralen Kalibrierung

Zur Evaluierung der bilateralen Kalibrierung wurden sowohl künstliche Datensätze simuliert, als auch Messungen in Laborversuchen durchgeführt. Insgesamt liegen 8 Datensätze (DS) vor. Für Datensatz 5 liegen die

aufgenommenen Zeitreihen aller sechs Kalibrierungsschritte vor. Für die Datensätze 6-8 wurden keine Zeitreihen für die halbe Kokontraktion aufgenommen und die künstlichen Datensätze 1-4 simulieren direkt die zu extrahierenden Medianwerte. Es liegen daher keine simulierten Zeitreihen vor. Um die Datensätze 6-8 trotzdem zur Methodenentwicklung verwenden zu können, werden die Medianwerte der halben Kokontraktionen imputiert. Dazu werden die halbierten Medianwerte der maximalen Kokontraktion verwendet. Abbildung 6.15 zeigt die Eingangsräume der Datensätze.

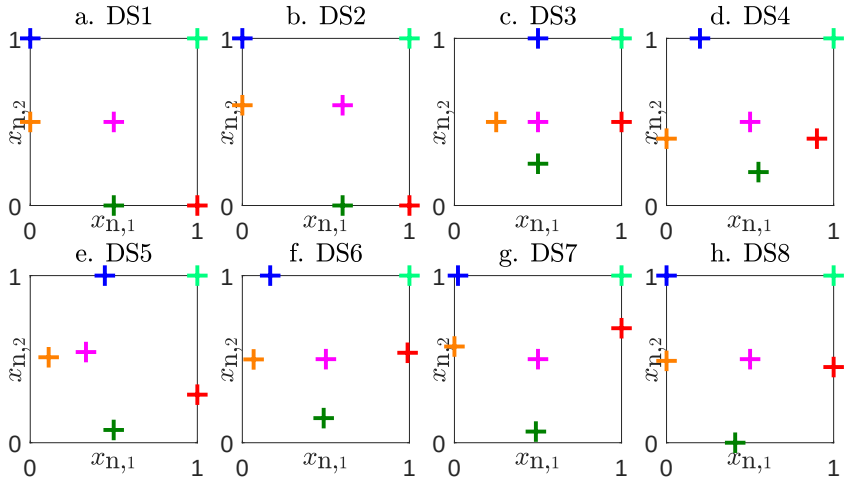

Abbildung 6.15: Datensätze zur Methodenentwicklung. Die Markierungen zeigen die Lage der Medianwerte im Eingangsraum der zu entwerfenden Regressionsmodelle zur Intentionsschätzung. Die zugehörigen Werte der Zielgrößen sind Tabelle 6.4 zu entnehmen.

Komplexe Modellstrukturen sind gegebenenfalls in der Lage die Streuungen der Zeitreihen abzubilden, um die Benutzerfreundlichkeit der MMS zu erhöhen. Damit die Lerndaten entsprechende Informationen bereitstellen, können beispielsweise für jeden Kalibrierungsschritt nicht nur die Medianwerte, sondern noch weitere Quantilwerte extrahiert werden. Für die Datensätze 5-8 werden erweiterte Lerndatensätze $DS5^*$ bis $DS8^*$ erstellt, die für jeden Kalibrierungsschritt neun Datentupel extrahieren. Dabei han-

delt es sich um sämtliche Kombinationen der 0.25-, 0.5- und 0.75-Quantile der Zeitreihen.

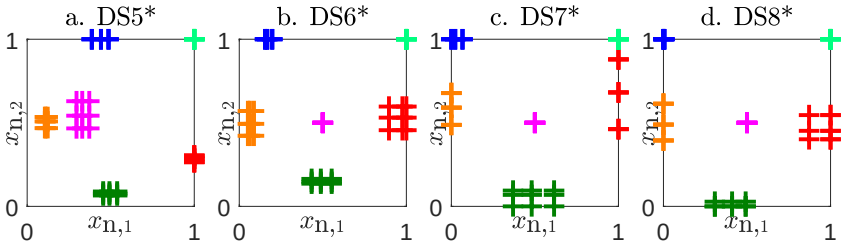

Abbildung 6.16: Erweiterte Datensätze zur Methodenentwicklung. Die Markierungen zeigen die Lage der betrachteten Quantile im Eingangsraum der zu entwerfenden Regressionsmodelle zur Intentionsschätzung. Die zugehörigen Werte der Zielgrößen sind Tabelle 6.4 zu entnehmen.

Abbildung 6.16 zeigt die erweiterten Datensätze. Je weiter die neun Datentupel eines Kalibrierungsschritts voneinander entfernt liegen, desto weniger ist der Benutzer fähig die Sollwerte willentlich über einen Zeitraum zu generieren. Ein gutes Modell bildet die Streuung ab, um dennoch die intentionierten Signale erzeugen zu können.

6.2.3 Bewertung der Datenqualität

Abbildung 6.17 zeigt die Ergebnisse einer Untersuchung der Datenqualität durch DaMoQ für $DS5$ und $DS5^*$. Obwohl $DS5$ optisch keine der Phänomene aus Abschnitt 2.3 aufzuweisen scheint, indizieren die Bewertungskriterien das Vorliegen von Clustern. Ähnliche Ergebnisse sind für die übrigen herkömmlichen Datensätze $DS1$ bis $DS8$ zu beobachten, was auf die sehr geringe Anzahl an Datentupeln ($N = 6$) zurückzuführen ist. Für Datensätze mit $N < 10$ scheinen die Bewertungskriterien aus Abschnitt 2.3 nicht geeignet. Bei den erweiterten Datensätzen $DS5^*$ bis $DS8^*$ sind tatsächlich Cluster vorhanden, die den sechs Kalibrierungsschritten entsprechen (vgl. Abbildung 6.13c). In der bilateralen Kalibrierung bedeuten die Cluster, dass der Anwender für die Kalibrierungsschritte unterschiedliche, aber gleichmäßige Zeitreihen generieren kann. Der Anwender weist in solchen Fällen eine hohe Eignung zur Steuerung der MMS auf, da

er unterschiedliche Aktivierungsniveaus der Signalkanäle willentlich und robust ansteuern kann. In [176] werden deshalb andere Bewertungskriterien der Datenqualität untersucht, um einen angemessenen Komplexitätsgrad der Kalibrierungsfunktionen zu identifizieren.

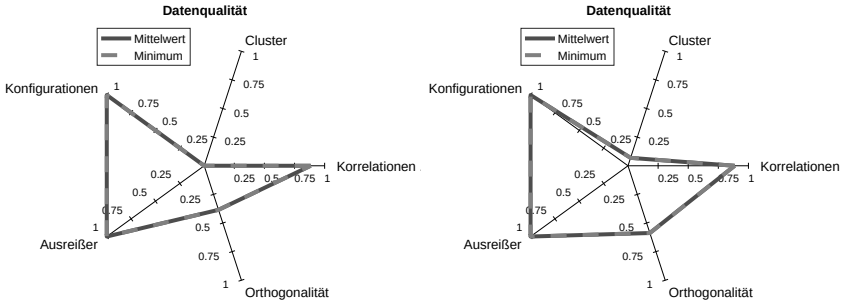

Abbildung 6.17: Übersicht der Datenqualität von a. $DS5$ b. $DS5^*$

6.2.4 Modellentwurf mit Polynomen

Die polynomiale Modellstruktur

$$\hat{x}_{\mathrm{I},i}[k] = f_i(x_{\mathrm{n},1}[k], x_{\mathrm{n},2}[k]) = \begin{pmatrix} \theta_{1,i} \\ \theta_{2,i} \\ \theta_{3,i} \\ \theta_{4,i} \\ \theta_{5,i} \end{pmatrix}^T \cdot \begin{pmatrix} x_{\mathrm{n},1}[k] \\ x_{\mathrm{n},2}[k] \\ x_{\mathrm{n},1}[k]x_{\mathrm{n},2}[k] \\ (x_{\mathrm{n},1}[k])^2 \\ (x_{\mathrm{n},2}[k])^2 \end{pmatrix} \tag{6.6}$$

scheint nach einem Vergleich basierend auf den Datensätzen zur Abbildung der nichtlinearen Koaktivierungen geeignet. Bei gemessener Relaxation beider Muskel kann auf eine intentionierte Relaxation geschlossen werden. Das Vorwissen ist in die Modellstruktur integriert, indem auf einen skalaren Offset verzichtet wird. Um den zulässigen Wertebereich nicht zu verletzen, wird $\hat{x}_{\mathrm{I},i}$ in der Modellanwendung auf das Intervall $[0, 1]$ beschränkt. Die Anpassung der 10 Parameter $\boldsymbol{\theta}_i = (\theta_{1,i}, \ldots, \theta_{5,i}), i = 1, 2$ erfolgt durch

$$\hat{\boldsymbol{\theta}}_i = \underset{\boldsymbol{\theta}_i}{\operatorname{argmin}} \sum_{j=1}^{6} \left(f_i \left(x_{\mathrm{n,med},1,j}, x_{\mathrm{n,med},2,j}; \boldsymbol{\theta}_i \right) - x_{\mathrm{I},i,j} \right)^2 . \tag{6.7}$$

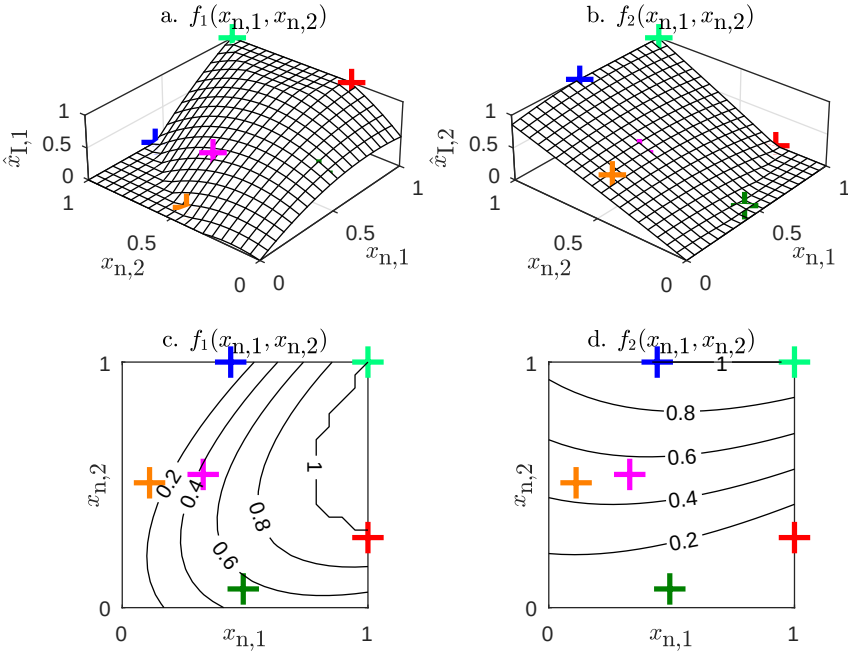

Abbildung 6.18: Regressionsmodelle basierend auf $DS5$ in drei- und zweidimensionaler Darstellung

Die uni- und bilaterale Kalibrierung findet automatisiert statt, das heißt sie kann zwar von medizinischem Fachpersonal betreut werden, manuelle Einstellungen sind aber nicht vorzunehmen. Abbildung 6.18 zeigt die für $DS5$ entworfenen Polynommodelle zur Intentionsschätzung. Abbildung 6.19 zeigt für $DS5$ die Sollwerte der Kalibrierungsschritte, die aufgenommenen Zeitreihen sowie die aufgenommenen Zeitreihen nach Anwendung der Intentionsschätzung. Die Zeitreihen entsprechen nach der Intentionsschätzung deutlich mehr den Sollwerten, d.h. den eigentlich intentionierten Aktivierungen.

In der praktischen Anwendung wird die Kalibrierung häufig vom medizinischen Fachpersonal nachjustiert, um die Bedienbarkeit der MMS zu erhöhen und Frustration beim Anwender zu vermeiden. Da die manuelle

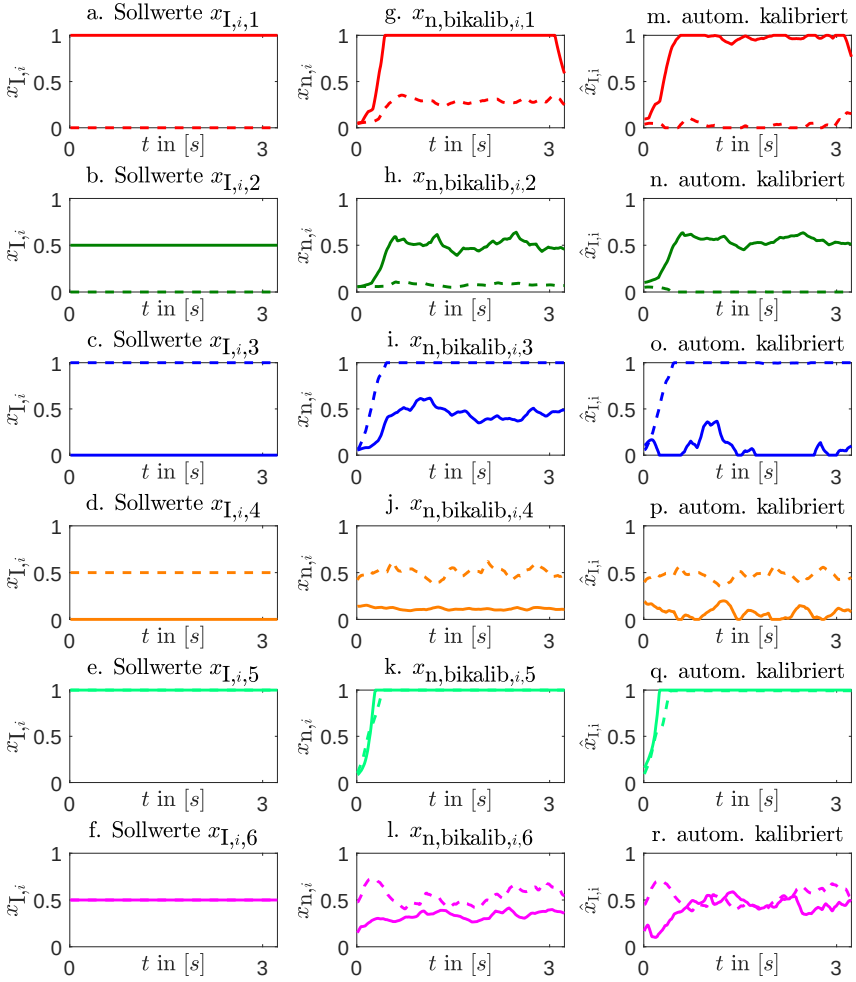

Abbildung 6.19: Verlauf der a.-f. Sollwerte g.-l. Kalibrierungszeitreihen m.-r. geschätzten intentionierten Signale für die Zeitreihen aus Datensatz 5

Anpassung von Parametern der Regressionsmodelle wenig intuitiv ist, findet die Nachjustierung lediglich durch Anpassung von $x_{\mathrm{f,min},i}$ und $x_{\mathrm{f,max},i}$ statt. Dadurch lässt sich die Erkennung von maximalen Kontraktionen robuster gestalten, allerdings wird der Wertebereich der normierten Signale u.U. stark eingeschränkt. Dadurch erhöht sich der Einfluss von geringfügigen Schwankungen, wodurch das Ansteuern von Betriebspunkten abseits der Kalibrierungspunkte erschwert wird.

Zusätzlich kann die Parameterschätzung der Regressionsmodelle der bilateralen Kalibrierung automatisch wiederholt werden, da sich die aufgenommenen Zeitreihen $x_{\mathrm{n,bikalib},i,j}[k]$ durch die nachjustierten Normierungsparameter geändert haben. Dabei treten die durch die Nachjustierung eliminierten Koaktivierung in den Kalibrierungszeitreihen teilweise wieder auf. In Abbildung A.5 im Anhang A.4 werden die Auswirkungen auf die Intentionsschätzung durch die manuelle Anpassung und die wiederholte Parameterschätzung dargestellt.

Die Parameter der polynomialen Modelle sind interpretierbar sowie einfach und schnell zu schätzen. Für eine genauere Abbildung der Koaktivierungen, z.B. bei Verwendung der erweiterten Datensätze $DS5^*$ - $DS8^*$, ist die Modellstruktur u.U. nicht flexibel genug. Außerdem kann die Annahme, dass bei gemessener Relaxation beider Muskel auf eine intentionierte Relaxation beider Muskel geschlossen werden kann, erweitert werden: So liegt bei einer gemessenen Relaxation des i-ten Muskels eine intentionierte Relaxation des i-ten Muskels unabhängig vom anderen Muskel vor. Weiterhin lässt sich von einer erhöhten gemessenen Aktivierung des i-ten Muskels auf eine erhöhte intentionierte Aktivierung des i-ten Muskels schließen. Beides kann als punktweise Anforderungen in den Modellentwurf integriert werden. Als Alternative zu den vorgestellten Polynommodellen werden im folgenden Abschnitt Stützvektor-Regressionen mit virtuellen Datentupeln zur Abbildung der Koaktivierungen untersucht. Das erweiterte Vorwissen hinsichtlich Relaxation und Monotonie wird bislang ausschließlich in die SVRs integriert. Gemäß Tabelle 4.11 wird auf die Verwendung von hybriden Modellen verzichtet, da keine geringe Datenabdeckung vorliegt und z.B. Monotonie als Vorwissen mit dem vorgestellten Entwurfsverfahren für hybride Modelle nicht sicherzustellen ist.

6.2.5 Modellentwurf mit Stützvektor-Regressionen

Im automatisierten Entwurf der Kalibrierungsfunktionen mit Hilfe von SVRs ist zu beachten, dass

- der Entwurf maximal einige Sekunden dauert, um Frustration zu vermeiden,
- f_i monoton mit x_i steigt und
- $f_i(x_i = 0) = 0$ gilt.

Zur Reduktion des Entwurfsaufwands wird $C_{\mathrm{reg}} = 10$ und $\epsilon_{\mathrm{reg}} = 0.001$ empirisch gewählt. Im Entwurf der Kalibrierungsfunktionen wird ausschließlich σ bzw., um eine größere Flexibilität zu ermöglichen, $\mathbf{\Sigma}$ (vgl. (4.3)) gemäß

$$\hat{\mathbf{\Sigma}} = \underset{\mathbf{\Sigma}}{\operatorname{argmin}} \left(\sum_{j=1}^{6} f_i(x_{\mathrm{n,med},1,j}, x_{\mathrm{n,med},2,j}; \mathbf{\Sigma}) - x_{\mathrm{I},i,j} \right) \tag{6.8}$$

mit einer Gittersuche optimiert. Die Monotonieanforderung wird als Restriktion und die punktweisen Randbedingungen als virtuelle Datentupel gemäß Ansatz 4 aus Abschnitt 4.1.3 in die Parameterschätzung integriert. Für die virtuellen Datentupel zur Einhaltung der punktweisen Randbedingungen werden individuelle Werte C_i benötigt. Weiterhin kann unter Berücksichtigung der nachträglichen Beschränkung des Modellverlaufs auf das zulässige Intervall $[0, 1]$ den Modellen eine erhöhte Flexibilität ermöglicht werden, indem die punktweisen Restriktionen zu

$$f_i(x_i = 0) \overset{!}{\leq} 0 \tag{6.9}$$

relaxiert werden. Daher werden virtuelle Datentupel der Art $f_i(x_i = 0) = -0.1$ mit $\varepsilon_{\mathrm{R}} = 0.1$ integriert. Da für die gewöhnlichen Datentupel ein deutlich kleinerer Wert für ε von Vorteil ist, werden individuelle Werte ε_i benötigt.

6.2.6 Ergebnisse

Die Polynommodelle und SVRs werden anhand der summierten Abweichungen

$$SSE = \sum_{i=1}^{2} \left(\sum_{j=1}^{6} (\hat{x}_{\mathrm{I},i}(x_{\mathrm{n,med},1,j}, x_{\mathrm{n,med},2,j}) - x_{\mathrm{I},i,j})^2 \right) \tag{6.10}$$

über die Lerndaten sowie die Rechenzeiten der Parameterschätzung[3] verglichen.

Für den Vergleich werden zusätzlich SVRs ohne Vorwissen entworfen (SVR*). Ihre Strukturwahl erfolgt über Kreuzvalidierung. Für die herkömmlichen Datensätze umfassen die Validierungsdatensätze der Kreuzvalidierung jeweils nur ein Datentupel.

Für die erweiterten Datensätze werden jeweils alle Datentupel, die zu einem Kalibrierungsschritt gehören, als Validierungsdaten verwendet. Tabelle 6.5 zeigt die Ergebnisse für die herkömmlichen Datensätze $DS1$ - $DS8$ und Tabelle 6.6 zeigt die Ergebnisse für die erweiterten Datensätze $DS5^*$ - $DS8^*$.

	SSE			Zeit [s]		
Datensatz	Polynom	SVR	SVR*	Polynom	SVR	SVR*
DS1	0.0000	0.0000	0.0000	0.0004	3.4052	2.0036
DS2	0.0000	0.0000	0.0000	0.0040	3.2486	1.9899
DS3	0.0066	0.0660	0.0000	0.0009	3.3254	1.9894
DS4	0.0000	0.0000	0.0000	0.0008	3.1893	1.9503
DS5	0.0095	0.0363	0.0229	0.0008	3.2475	1.9362
DS6	0.0010	0.0246	0.0000	0.0008	3.1135	1.9519
DS7	0.0138	0.0055	0.0402	0.0008	3.1278	1.9202
DS8	0.0053	0.0180	0.0000	0.0008	3.2122	1.9369

Tabelle 6.5: Vergleich des Polynomansatzes und SVRs mit (SVR) bzw. ohne (SVR*) Vorwissen anhand von SSE und Rechenzeit.

Für die herkömmlichen Datensätze weisen die Polynommodelle den geringeren Fehler als die SVRs mit Vorwissen auf.

[3] bei den SVRs inklusive der Optimierung der Metaparameter

	SSE			Zeit [s]		
Datensatz	Polynom	SVR	SVR*	Polynom	SVR	SVR*
DS5*	0.2839	0.0473	0.0190	0.0004	12.5550	3.6049
DS6*	0.2433	0.0524	0.0011	0.0008	12.4381	3.4784
DS7*	1.5608	0.8536	0.0752	0.0003	11.2637	3.2861
DS8*	0.3621	0.0158	0.0418	0.0002	11.7225	3.3840

Tabelle 6.6: Vergleich des Polynomansatzes und SVRs mit (SVR) bzw. ohne (SVR*) Vorwissen anhand von SSE und Rechenzeit für die erweiterten Datensätze mit Informationen über die Streuung.

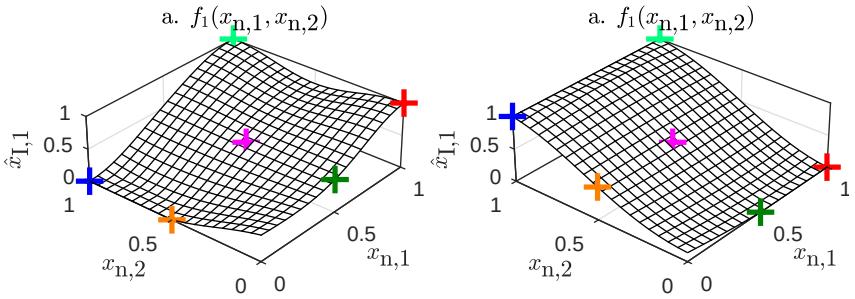

Abbildung 6.20: SVRs ohne Vorwissen für $DS1$. Bei gemessener Relaxation beider Signalkanäle wird keine intentionierte Relaxation der Signalkanäle geschätzt.

Das liegt zum einen daran, dass ein parameterlineares Modell mit 5 Parametern und 6 Datentupeln bei der Minimierung der Fehlerquadrate näherungsweise interpoliert, zum anderen daran, dass die Parameterschätzung der SVRs einen Kompromiss aus Näherung an die Kalibrierungsdaten und Beachtung der virtuellen Datentupel unter Berücksichtigung der Restriktionen optimiert.

Bei den erweiterten Datensätzen weisen die SVRs mit Vorwissen den geringeren Fehler als die Polynommodelle auf, da sie eine wesentlich höhere Flexibilität als die Polynommodelle aufweisen. Trotzdem erfüllen sie die Randbedingungen und sind demnach als Kalibrierungsfunktionen geeignet. Abbildung 6.21 zeigt sowohl die Polynommodelle als auch die SVRs für $DS7$ und $DS7^*$. Die SVRs ohne Vorwissen weisen für die herkömmli-

chen Datensätze einen mit den Polynommodellen vergleichbaren Fehler auf und für die erweiterten Datensätze den geringsten Fehler. Abbildung 6.20 zeigt anhand von $DS1$ den Nachteil der SVRs ohne Vorwissen: Die Lerndaten werden zwar sehr genau approximiert, die Randbedingungen zur Gewährleistung der Benutzerfreundlichkeit (z.B. geschätzte intentionierte Relaxation bei gemessener Relaxation) werden allerdings verletzt. SVRs ohne Vorwissen sind daher als Kalibrierungsfunktionen ungeeignet.

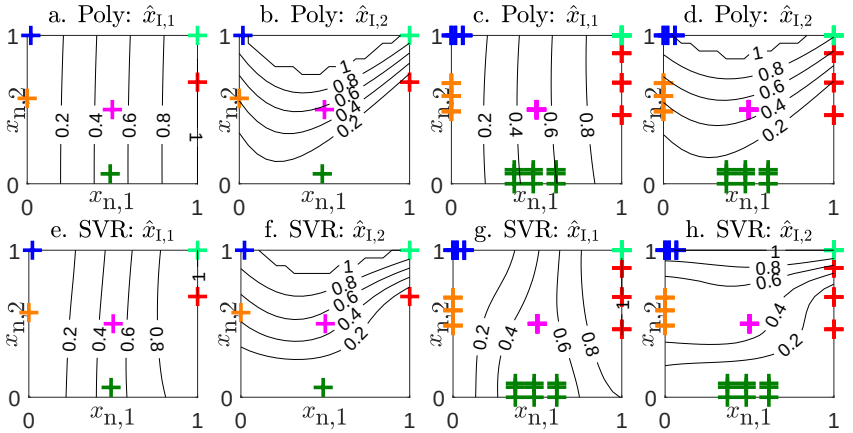

Abbildung 6.21: Modellverläufe der automatisiert entworfenen SVRs mit Vorwissen für die Datensätze 1-8

Die SVRs bilden um die Daten konstanter Zielgrößen Plateaus, was es dem Anwender vereinfacht, Aktivierungslevels zu halten, d.h. die MMS gleichmäßig anzusteuern. Die Darstellung aller Modelle findet sich in Anhang A.3, Abbildungen A.1, A.2, A.3 und A.4. Der Nachteil der SVRs liegt in der Rechenzeit, die deutlich höher als bei den Polynommodellen, aufgrund der absoluten Rechenzeiten aber noch akzeptabel für den Anwendungsfall ist.

6.2.7 Zusammenfassung

Es wird eine Kalibrierungsroutine für zweikanalige Mensch-Maschine-Schnittstellen vorgestellt. Mit Hilfe einer GUI werden dem Anwender visu-

ell Sollwerte vorgegeben und die gemessenen Signale werden aufgezeichnet. Anschließend werden aus den aufgezeichneten Zeitreihen automatisiert Lerndaten extrahiert und Regressionsmodelle entworfen, die Koaktivierungen zwischen den Signalkanälen abbilden und die gemessenen Signale in die ursprünglich intentionierten Signale des Anwenders überführen.

Als Regressionsmodelle wurden Polynome zweiten Grades sowie SVRs verwendet. Um bei den u.U. sehr komplexen SVRs eine für MMS geforderte Robustheit zu gewährleisten, wird Vorwissen in Form von virtuellen Datentupeln und Restriktionen im Modellentwurf integriert. Beide Modellstrukturen sind generell geeignet, Koaktivierungen zwischen zwei Signalkanälen abzubilden. Anhand simulierter und gemessener Datensätze werden die Modelle untersucht. Es zeigt sich, dass SVRs bei genügend Lerndaten auch komplexe Zusammenhänge abbilden und eine präzise Steuerung einer MMS ermöglichen können, dabei allerdings deutlich länger für den Modellentwurf als die Polynommodelle benötigen, was in der realen Anwendung die Nutzerakzeptanz verschlechtern kann.

Ein aussagekräftiger Vergleich ist erst nach Implementierung der Ansätze in ein entsprechendes Kalibrierungs- bzw. Steuerungssystem einer MMS möglich. Bislang wurde die automatisierte Kalibrierung ausschließlich unter der Verwendung der Polynommodelle erfolgreich implementiert [3]. In einer Probandenstudie können nach einer vollständigen Implementierung die Modelle anhand von Anwendungsdaten verglichen werden, die unabhängig von den Kalibrierungsdaten unter realen Anwendungsbedingungen erhoben werden. Weiterhin kann die Eignung der Modelle auch abseits von quantitativen Vergleichen durch Befragungen der Probanden und des medizinischen Fachpersonals bewertet werden. Außerdem kann zukünftig die Möglichkeit untersucht werden, inkrementelles Lernen zu verwenden, um die Modelle sich ändernden Umgebungsbedingungen anzupassen. Dazu gehört auch der Lerneffekt bzw. die Anpassung des Anwenders an die MMS [177]. Weiterhin kann die Kalibrierungsroutine individuell auf den Anwender angepasst werden. In [178] wird eine solche adaptive Kalibrierungsroutine entworfen. Es werden für geeignete Anwender u.U. weitere Kalibrierungsschritte unternommen, um die Steuerungsfähigkeit der MMS zu verbessern.

6.3 Zeitreihenprädiktion von Energiedaten

6.3.1 Problemstellung

Die Umwandlung von Sonnenlicht in elektrische Energie, Photovoltaik (PV), bietet die Möglichkeit einer Energieversorgung, bei der die begrenzt vorhandenen fossilen Brennstoffe nicht aufgebraucht werden und leistet einen Beitrag zur Ausstoßreduktion von Schadstoffen [179]. PV-Anlagen erzeugen sogenannte erneuerbare Energie, die eine wichtige Rolle in der von der Bundesregierung angestoßenen Energiewende spielt: Für das Jahr 2050 ist in Deutschland ein Anteil von mehr als 80% erneuerbare Primärenergie vorgesehen [180].

Der Nachteil von PV-Anlagen, die aktuell etwa ein Fünftel der erneuerbaren Energie in Deutschland liefern, liegt in der schwankenden Energieerzeugung in Abhängigkeit des Wetters [181]. Zum Erhalt der Netzstabilität bei der zunehmenden Integration von PV-Anlagen in das Stromnetz ist das Vorhersagen der zur erwartenden Stromerzeugung einer Anlage von großer Bedeutung. Daher sind entsprechende Algorithmen Bestandteil der Forschungsinfrastruktur *Energy Lab 2.0*, in der neue Ansätze zur Stabilisierung von Energienetzen untersucht werden [182].

Leistungsprognosen erfolgen entweder durch die möglichst detaillierte Modellierung des physikalischen Zusammenhangs der Anlage und aktueller Wetterprognosen oder durch datengetriebene Modelle [183]. Es existieren auch Ansätze von Kombinationen beider Varianten [184].

Die datengetriebenen Modelle werden anhand der verwendeten Daten unterschieden. Je nach Verfügbarkeit und Ressourcen werden ausschließlich historische Daten der vorherzusagenden Leistungszeitreihe verwendet oder zusätzlich aktuelle und historische Daten exogener Zeitreihen, z.B. Wetterdaten oder Wetterprognosen. Beispiele für datengetriebene Leistungsprognosen ohne Wetterdaten finden sich in [185]. Die Differenz zwischen dem Zeitpunkt des Modellentwurfs und dem Zeitpunkt, für den die Leistung prognostiziert wird, heißt Prognosehorizont.

Die Zeitreihen werden in Eingangsgrößen und Zielgröße eines Regressionsproblems transformiert.

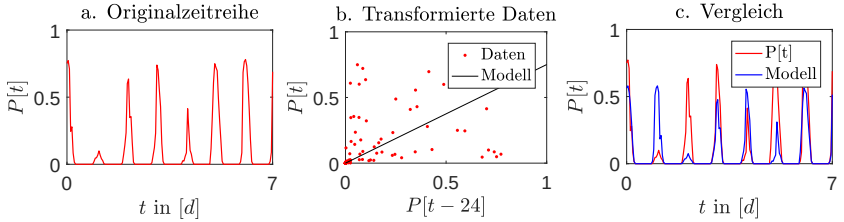

Abbildung 6.22: Beispiel einer Leistungszeitreihe, eines transformierten Datensatzes sowie eines linearen Modells zur Abbildung und Prognose der Zeitreihe

Abbildung 6.22 zeigt eine kurze Zeitreihe, den transformierten Datensatz als Eingangs- und Zielgröße sowie ein lineares datengetriebenes Modell in der für Regressionsmodelle typischen Darstellung und als Zeitreihe im Vergleich mit der Originalzeitreihe. Es handelt sich um ein Modell mit einem Prognosehorizont von 24 Stunden und der allgemeinen Modellstruktur

$$\hat{P}[t] = \theta P[t - 24h]. \tag{6.11}$$

Um genauere Prognosen zu ermöglichen, werden mehr Eingangsgrößen und komplexere Modellstrukturen verwendet. Mit genaueren Prognosen kann die Wirtschaftlichkeit des Netzbetriebs erhöht werden. Allerdings bedarf es dafür des Know-Hows und der Ressourcen zum Entwurf komplexer Modelle.

Beispiele für Leistungsprognosen mit Hilfe komplexer Regressionsmodelle finden sich in [186–188].Weiterhin haben einfache, lineare Ansätze den Vorteil der Interpretierbarkeit und Robustheit. Das ist für Netzbetreiber von Bedeutung, da sich die Leistungsprognose direkt auf die Netzstabilität auswirken kann und damit auch auf die Gewährleistung von kritischer Infrastruktur wie z.B. Krankenhäusern.

Im Folgenden wird gezeigt, dass sich der automatisierte Entwurfsprozess hybrider Modelle aus Abschnitt 4.2.3 für Leistungsprognosen von PV-Anlagen eignet und die hybriden Modelle eine höhere Prädiktionsgüte als herkömmliche Modelle aufweisen. Hybride Modelle können die Genauigkeit nichtlinearer Modelle für Interpolationen und die Robustheit linearer Modelle für Extrapolationen vereinen.

Sogenannte probabilistische Vorhersagen, die statt einer herkömmlichen Prädiktion eine Wahrscheinlichkeitsverteilung vorhersagen, werden nicht behandelt. In [189] wird aber am Beispiel von PV-Systemen ein neues Verfahren zum Entwurf solcher probabilistischer Regressionsmodelle vorgestellt.

6.3.2 Datenvorverarbeitung

Beim Wettbewerb „Global Energy Forecasting Competition 2014“ war u.a. die Vorhersage der Leistung in Megawatt (MW) von drei PV-Anlagen gefordert. Dazu wurden an den Anlagen Daten von über zwei Jahren erhoben. Die Daten, das heißt die Leistungsdaten der Anlage sowie standortgenaue Wetterprognosen, liegen in einer stündlichen Auflösung vor. Die Wetterprognosen bestehen aus 12 Zeitreihen der Wettervariablen

- Flüssigwassergehalt (engl. *total column liquid water*, tclw) in $\frac{kg}{m^2}$,
- Eiswassergehalt (engl. *total column ice water*, tciw) in $\frac{kg}{m^2}$,
- Luftdruck (engl. *surface pressure*, SP) in Pa,
- relative Feuchtigkeit (engl. *relative humidity*, r) in %,
- Bewölkung (engl. *total cloud cover*, TCC) von 0 bis 1,
- zonale Windgeschwindigkeit (engl. *10-metre U wind component*, 10u) in $\frac{m}{s}$,
- meridionale Windgeschwindigkeit (engl. *10-metre V wind component*, 10v) in $\frac{m}{s}$,
- Lufttemperatur (engl. *2-metre temperature*, 2T) in K,
- Solarstrahlung an der Oberfläche (engl. *surface solar rad down*, SSRD) in $\frac{J}{m^2}$,
- thermische Strahlung an der Oberfläche (engl. *surface thermal rad down*, STRD) in $\frac{J}{m^2}$,
- Solarstrahlung oberhalb der Atmosphäre (engl. *top net solar rad*, TSR) in $\frac{J}{m^2}$ und

- Niederschlag (engl. *total precipitation*, TP) in m.

Die ursprünglich als zonaler und meridionaler Anteil vorliegenden Prognosen der Windgeschwindigkeit sind in eine absolute Windgeschwindigkeit (10p) fusioniert.

Für den Entwurf von Regressionsmodellen werden die Zeitreihen in Einzelmerkmale umgewandelt. Die Einzelmerkmale entsprechen den Werten der Zeitreihen an vergangenen Abtastzeitpunkten. Wird beispielsweise das erste Jahr der Aufzeichnungen stündlich abgetasteter Zeitreihen als Lerndaten verwendet, entsteht ein Datensatz mit $24 \cdot 365 = 8760$ Datentupeln. Es wird ein Prognosehorizont von 24 Stunden angenommen. Daher werden zur Prognose von $P[t+24h]$ von der Leistungszeitreihe die Einzelmerkmale $P[t], P[t-1h], \ldots$ betrachtet. Zur Simulation von Wetterprognosen, werden zum Modellentwurf trotz des Prognosehorizonts die zukünftigen Daten der Solarstrahlung $TSR[t+24h], TSR[t+23h], \ldots$ verwendet. Bei den Regressionsmodellen werden zeitvariante Änderungen der Zusammenhänge vernachlässigt.

Zeitreihenwerte, die mehr als 30 Tage in die Vergangenheit reichen, sind für die Leistungsprognosen i.d.R. nicht relevant [190] und werden deshalb nicht berücksichtigt. Alle potentiellen Eingangsgrößen und die Zielgröße werden auf das Einheitsintervall normiert. Anschließend findet eine Reduktion der potentiellen Eingangsgrößen durch eine univariate Merkmalsbewertung anhand der Korrelationskoeffizienten mit der Zielgröße statt.

Es entsteht ein Pool möglicher Eingangsgrößen bestehend aus den als Prognosen angenommenen 13 Wettervariablen für die Zeitpunkte $t+24h, t+23h, t+22h, t+21h, t+20h, t+19h, t+18h, t, t-24h, t-48h, \ldots, t-696h$ sowie den Werten der Leistungszeitreihe für die Zeitpunkte $t, t-1h, t-2h, t-3h, t-4h, t-5h, t-6h, t-24h, t-48h, \ldots, t-696h$. Es liegen demnach $13 \cdot (7+30) + 1 \cdot (6+30) = 517$ potentielle Eingangsgrößen vor.

Für die Untersuchung von hybriden Modellen für Leistungsprognosen und einen Vergleich mit herkömmlichen Modellen werden die Daten wie folgt in Lern-, Validierungs- und Testdaten eingeteilt:

- Zufällige Entnahme von zehn Prozent der Daten als Testdaten (Testdatensatz 1). Anschließend zufällige Entnahme von zwanzig Prozent als Validierungsdaten,

- Entnahme der Datentupel des zweiten Jahres als Testdaten (Testdatensatz 2). Anschließend zufällige Entnahme von 20 Prozent als Validierungsdaten und
- Entnahme des zweiten Jahres sowie der zweiten Hälfte des ersten Jahres als Testdaten (Testdatensatz 3). Anschließend zufällige Entnahme von zwanzig Prozent als Validierungsdaten.

Vor dem jeweiligen Modellentwurf findet zur Vermeidung von Überanpassung eine weitere Merkmalsreduktion durch eine univariate Merkmalsselektion statt. Für lineare Modelle werden die Eingangsgrößen gewählt, die den größten Korrelationskoeffizienten mit der Zielgröße aufweisen. Für nichtlineare Modelle erfolgt die Berechnung des Korrelationskoeffizienten auf Basis eines MLP-Netzes mit 2 Neuronen in der verdeckten Schicht. Dadurch werden Eingangsgrößen gewählt, bei denen ein nichtlinearer Zusammenhang mit der Zielgröße besteht. Um eine ausführliche Untersuchung zu gewährleisten, erfolgt eine Merkmalsreduktion auf 2, 5, 10, 15 und 20 Merkmale.

Da die Leistung zwischen Sonnenuntergang und Sonnenaufgang garantiert Null ist, werden zum Modellentwurf und zur Bewertung ausschließlich die Tagwerte betrachtet. Generell kann unter Berücksichtigung der Uhrzeit als Eingangsgröße punktweises Vorwissen genutzt werden, um einen Modellausgang von Null zu erreichen. Aufgrund der großen Anzahl an Datentupeln werden solche Ansätze gemäß Tabelle 4.11 nicht untersucht.

6.3.3 Datenqualität

Zunächst wird die Datenqualität mit Hilfe von DaMoQ bestimmt. Da sich durch die unterschiedlichen Aufteilungen in Teildatensätze sowie die Merkmalsselektionen eine Vielzahl an Datensätzen ergeben, die für den Modellentwurf von hybriden Modelle verwendet werden, wird die Datenqualität nur anhand eines Beispiels analysiert (Erste PV-Anlage, Lerndatensatz 1, 20 Merkmale).

Abbildung 6.23 zeigt die Qualitätsübersicht von DaMoQ für den Datensatz und ein beispielhaftes Streuwolkendiagramm. Das Streuwolkendiagramm bestätigt die Bewertung von DaMoQ, dass vor allem Korrelationen und Ausreißer auftreten. Die Korrelationen entstehen durch die Auswahl von

mehreren Abtastzeitpunkten einer Zeitreihe, die entweder eine sehr geringe zeitliche Differenz aufweisen oder deren Differenzen in etwa der Wellenlänge zyklischer Effekte (Tage, Wochen) entsprechen. Abbildung 6.23 zeigt allerdings auch, dass die unterschiedlichen Zeitreihen der Wetterdaten korrelieren. Zur Kompensation der Einschränkungen in der Datenqualität können Merkmalsreduktionsverfahren wie PCA oder PLS verwendet werden. In [191] werden Verfahren der Merkmalsreduktion und der Parameterschätzung für lineare Modelle für Leistungsprognosen anhand dieses Datensatzes verglichen. Eine Merkmalsreduktion durch PCA bietet weiterhin den Vorteil, dass Ausreißer anschließend häufig durch einfache univariate Verfahren erkannt und entfernt werden können.

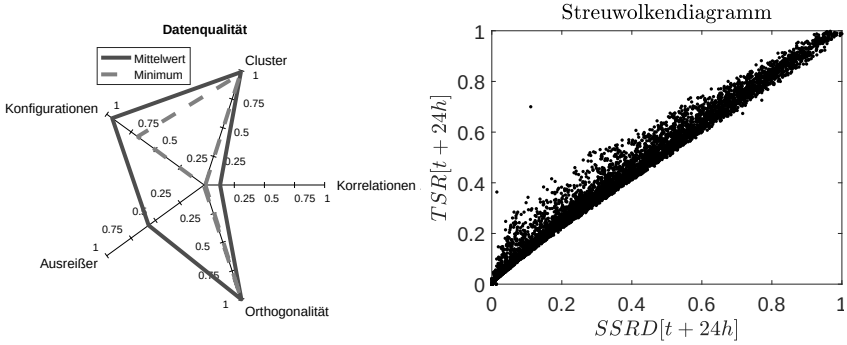

Abbildung 6.23: a. Ergebnis der Untersuchung der Datenqualität durch DaMoQ b. Beispielhaftes Streuwolkendiagramm zweier Merkmale mit Korrelationen und Ausreißern

6.3.4 Hybride Modelle

Auf Basis der verschiedenen Teildatensätze wird ein Vergleich zwischen herkömmlicher Modellselektion aus einem Modellpool und der Optimierung eines hybriden Modells gemäß Abschnitt 4.2.3 durchgeführt. Zur Vereinfachung werden ausschließlich MLP-Netze sowie lineare Polynommodelle verwendet. Weiterhin wird die Optimierung der hybriden Modelle zur Erhöhung der Interpretierbarkeit der hybriden Modelle so restringiert, dass zwingend ein lineares Modell als Extrapolationsmodell verwendet wird.

Dadurch kann zwar gegebenenfalls nicht das volle Potential hybrider Modellstrukturen hinsichtlich der Prädiktionsfähigkeit ausgenutzt werden, die für den domänenspezifischen Anwender wichtige Interpretationsfähigkeit des Modells wird aber erhöht.

6.3.5 Ergebnisse

Abbildung 6.24 zeigt die mittleren absoluten Fehler eines durch Modellselektion gewählten Modells und eines durch Optimierung gewonnenen hybriden Modells mit einem linearen Modell als Extrapolationsmodell für die verschiedenen Aufteilungen des GEFCom-Datensatzes der unterschiedlichen PV-Anlagen und für die verschiedenen Anzahlen an selektierten Merkmalen.

Weiterhin sind die mittleren absoluten Fehler in den Tabellen 6.7, 6.8 und 6.9 dargestellt.

	Testdaten 1		Testdaten 2		Testdaten 3	
Merkmale	f_{Ref}	f_{Hybrid}	f_{Ref}	f_{Hybrid}	f_{Ref}	f_{Hybrid}
2	0.096	0.096	0.094	0.094	0.101	0.101
5	0.074	0.074	0.072	0.072	0.082	**0.079**
10	0.070	0.070	0.073	0.073	0.088	**0.082**
15	0.071	0.071	0.073	0.073	0.087	**0.081**
20	**0.069**	0.070	0.075	**0.074**	0.087	0.087

Tabelle 6.7: Mittlere absolute Fehler für Anlage 1

	Testdaten 1		Testdaten 2		Testdaten 3	
Merkmale	f_{Ref}	f_{Hybrid}	f_{Ref}	f_{Hybrid}	f_{Ref}	f_{Hybrid}
2	0.086	0.086	0.088	0.088	**0.093**	0.094
5	0.070	0.070	0.072	0.072	0.074	0.074
10	0.071	0.071	0.075	0.075	0.079	0.079
15	0.072	0.072	0.079	**0.078**	0.086	**0.082**
20	0.072	0.072	0.078	0.078	0.090	**0.086**

Tabelle 6.8: Mittlere absolute Fehler für Anlage 2

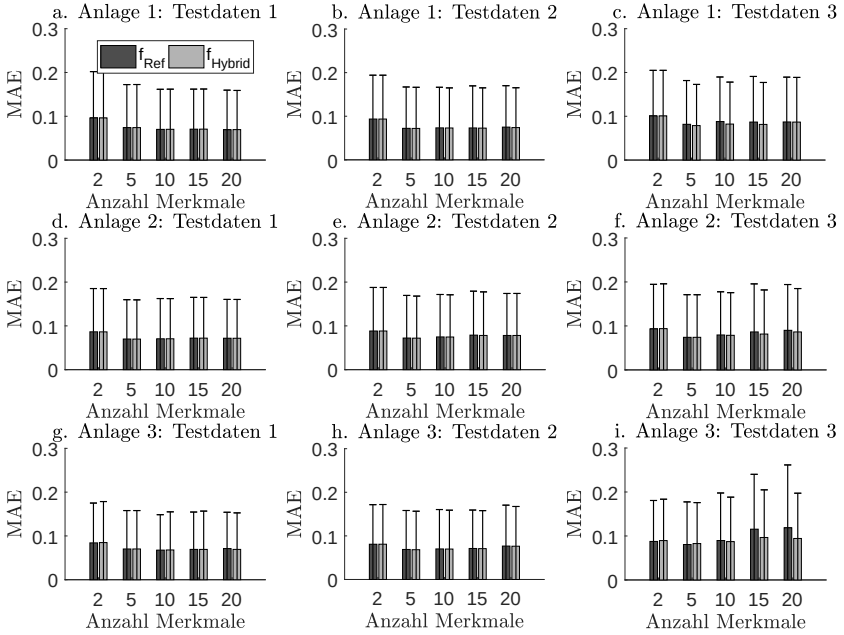

Abbildung 6.24: Ergebnisse der Modelle für die im GEFCom-Datensatz erfassten PV-Anlagen

Die hybriden Modelle zeigen vergleichbare Ergebnisse wie die herkömmlichen Modelle. Für die dritten Testdatensätze erzielen die hybriden Modelle tendenziell bessere Ergebnisse, aber nicht statistisch signifikant.

Weiterhin zeigt sich, dass die hybriden Modelle im Verhältnis zu herkömmlichen Modelle besser abschneiden, wenn die Anzahl der selektierten Merkmale erhöht wird.

Bei den dritten Testdatensätzen handelt es sich am ehesten um Extrapolationen, da große zusammenhängenden Zeiträume der Zeitreihen vorenthalten werden. Auch eine höhere Anzahl an selektierten Merkmalen führt aufgrund des Fluchs der Dimensionalität zu mehr Extrapolationen.

Merkmale	Testdaten 1		Testdaten 2		Testdaten 3	
	f_{Ref}	f_{Hybrid}	f_{Ref}	f_{Hybrid}	f_{Ref}	f_{Hybrid}
2	**0.084**	0.085	0.081	0.081	**0.088**	0.090
5	0.070	0.070	0.069	**0.068**	**0.081**	0.083
10	0.068	0.068	0.070	0.070	0.090	**0.087**
15	0.069	0.069	0.071	0.071	0.115	**0.097**
20	0.071	**0.069**	0.077	**0.076**	0.119	**0.094**

Tabelle 6.9: Mittlere absolute Fehler für Anlage 3

Anlage	Testdatensatz	f_{Ref}	f_{Int}	f_{Ext}
1	1	10 Neuronen	12 Neuronen	linear
1	2	10 Neuronen	10 Neuronen	linear
1	3	8 Neuronen	8 Neuronen	linear
2	1	4 Neuronen	4 Neuronen	linear
2	2	5 Neuronen	5 Neuronen	linear
2	3	10 Neuronen	10 Neuronen	linear
3	1	4 Neuronen	15 Neuronen	linear
3	2	15 Neuronen	12 Neuronen	linear
3	3	15 Neuronen	15 Neuronen	linear

Tabelle 6.10: Durch herkömmliche Modellselektion gewählte Modelle und durch Optimierung hybrider Modelle gewählte Interpolations- bzw. Extrapolationsmodelle

Tabelle 6.10 ist zu entnehmen, dass bei der Optimierung der hybriden Modelle für das Interpolationsmodell ein ähnlich komplexes Modell gewählt wird, wie das durch herkömmliche Modellselektion gewählte Modell.

Das bedeutet, dass die Validierungsdaten sehr nahe an den Lerndaten liegen. Das lässt sich mit den starken Korrelationen im Datensatz erklären.

In Abbildung 6.25 ist dargestellt, wie groß der Anteil an erkannten Interpolationen für die unterschiedlichen Testdatensätze gemäß der individuell optimierten 1K-SVM ist.

Es zeigt sich, dass vor allem in den Fällen, in denen das hybride Modell dem herkömmlichen Modell überlegen ist, Extrapolationen erkannt wurden und das lineare Extrapolationsmodell verwendet wurde.

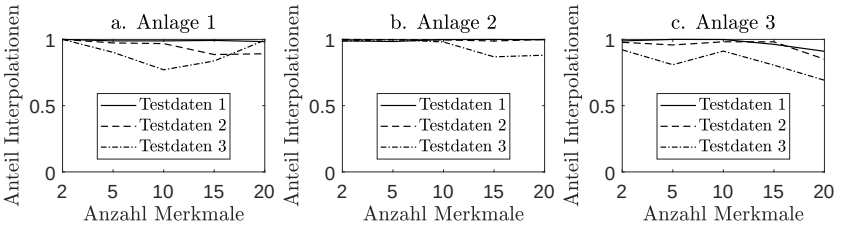

Abbildung 6.25: Anteile der Interpolationen für die Testdaten bei den hybriden Modellen

Weiterhin bestätigt die Abbildung, dass bei den dritten Testdatensätzen und bei mehr selektierten Merkmalen mehr Extrapolationen auftreten.

6.3.6 Zusammenfassung

Anhand einer Untersuchung auf Basis des GEFCom-Datensatzes aus dem Jahr 2014 wurde gezeigt, dass hybride Modelle und der vorgestellte Entwurfsprozess gemäß Abschnitt 4.2.3 für datengetriebene Leistungsprognosen geeignet sind. Sie verbinden die Genauigkeit von komplexen nichtlinearen Modellen mit der Robustheit linearer Modelle und bieten damit in der praktischen Anwendung einen deutlichen Vorteil gegenüber der Verwendung von einem herkömmlichen Modell mit nur einer Modellstruktur, bei dem sich der Entwickler für Genauigkeit oder Robustheit entscheiden muss. Als nächster Schritt kann der Entwurf von hybriden Modellen als eigenständiger Dienst in der serviceorientierten Softwarearchitektur im Rahmen des Energy-Lab 2.0 [192] implementiert werden.

7 Zusammenfassung

Die gestiegene preiswert verfügbare Rechenleistung und die Akzeptanz komplexer datengetriebener Modelle führt inzwischen häufig zu einer Substitution regelbasierter Expertensysteme durch automatisiert entworfene, nicht interpretierbare Black-Box-Modelle. Beim Entwurf solcher Modelle bestimmt die Datenqualität maßgeblich die Anwendbarkeit der Modelle. Ein Aspekt der Datenqualität ist die Datenabdeckung. Sie beschreibt, ob die zum Modellentwurf verfügbare Datenbasis alle potentiellen Anwendungsfälle des zu bildenden Modells abdeckt, d.h. ob ein solches Modell in der Anwendung zuverlässig ist und nicht unvorhergesehen versagt. In dieser Arbeit wurde der automatisierte Entwurf von Regressionsmodellen unter der besonderen Berücksichtigung eingeschränkter Datenabdeckung untersucht. Dazu wurden

1. hybride Modelle, deren Teilmodelle durch eine Ein-Klassen-Stützvektor-Maschine gewichtet werden, und eine neue Strategie zum Modellentwurf der hybriden Modelle vorgestellt, um vor allem in Extrapolationsbereichen höhere Prädiktionsgüten als herkömmliche Modelle zu erreichen,

2. die hybriden Modelle und ihr automatisierter Entwurf anhand von Benchmark-Datensätzen validiert und ihre Überlegenheit gegenüber herkömmlichen Modellen anhand von realen Anwendungsszenarien des Turbomaschinenbaus, der Energieinformatik und im Bereich von biomedizinischen Mensch-Maschine-Schnittstellen nachgewiesen.

3. Ansätze zur Integration von Vorwissen in Stützvektor-Regressionen verglichen sowie ein erweiterter Algorithmus für das Optimierungsproblem des vielversprechendsten Ansatzes entwickelt und implementiert,

4. Fragen zu Gewinnung von domänenspezifischem Vorwissen vorgestellt,

5. Bewertungskriterien modifiziert und validiert, um die Modellqualität besonders in Bereichen unzureichender Datenabdeckung abzuschätzen und Überanpassung von Modellen zu erkennen,
6. Benchmark-Datensätze erstellt, um Einschränkungen der Datenabdeckung systematisch abzubilden,
7. bestehende Kenngrößen modifiziert und neue Bewertungskriterien entwickelt, die verschiedene Phänomene einer eingeschränkten Datenabdeckung quantifizieren,
8. die Bewertungskriterien anhand der simulierten Benchmark-Datensätze sowie bekannter Benchmark-Datensätze von Regressionsproblemen validiert,
9. verständliche und übersichtliche Visualisierungsroutinen implementiert, um die Datenqualität von Datensätzen darzustellen und
10. unterschiedliche Aspekte der Datenqualität hinsichtlich Regressionsproblemen in eine einheitliche Taxonomie überführt.

Die Bewertungskriterien für Daten- und Modellqualität sowie die entsprechenden Visualisierungsroutinen wurden in einer Erweiterung der MATLAB-Toolbox SciXMiner implementiert und quelloffen zur Verfügung gestellt.

Der Algorithmus für das modifizierte Optimierungsproblem zur Integration von Vorwissen in Stützvektor-Regressionen wurde in MATLAB implementiert. Als nächster Schritt ist der Algorithmus durch Auswahl oder Entwicklung einer geeigneten Heuristik zu beschleunigen und in C/C++ zu implementieren.

Eine neue automatisierte Entwurfsmethodik von hybriden Modellen sieht die Kombination von zwei unterschiedlich komplexen Modellen vor, deren lokale Wichtung durch eine Ein-Klassen-Stützvektor-Maschine erfolgt. Die hybriden Modelle sind sowohl bei der Mehrzahl der untersuchten Benchmark-Datensätze, als auch bei den untersuchten Echt-Welt-Problemen herkömmlichen Modellen überlegen. Die Implementierung einer grafischen Benutzeroberfläche zur Konfiguration der Rahmenbedingungen im automatisierten Entwurf der hybriden Modelle ermöglicht die transparente Untersuchung und Weiterentwicklung des Entwurfsprozesses sowie

den Vergleich mit herkömmlichen Modellen. Sie ist Bestandteil einer Erweiterung der MATLAB-Toolbox SciXMiner.

Die Entwurfsmethodik kann zukünftig im Energy-Lab 2.0 eingesetzt werden, um datengetriebene Leistungsprognosen mit bisherigen Expertensystemen zu kombinieren und zuverlässigere sowie robuste Prognosen zu ermöglichen. Es ist ein entsprechender Dienst in der serviceorientierten Softwarearchitektur zu implementieren, der ausschließlich den Entwurf des Pools der Ein-Klassen-Klassifikatoren sowie die Optimierung der hybriden Modelle beinhaltet. Für den Modellpool kann auf bestehende Implementierungen zurückgegriffen werden.

Weiterhin ist zu untersuchen, ob die Verwendung von mehr als zwei Modellstrukturen in hybriden Modellen eine weitere Erhöhung der Prädiktionsgüte ermöglicht.

A Anhang

A.1 Bewertungskriterien für Datenabdeckung

Kriterium	Korr	Cluster	Konfig	Aus	Ortho
q_{Korr}	0.010	0.980	0.980	0.850	0.380
q_{Cluster}	0.990	0.000	0.060	0.990	0.980
q_{Konfig}	1.000	1.000	0.020	1.000	1.000
q_{Aus}	1.000	1.000	1.000	0.030	1.000
q_{Ortho}	1.000	0.350	0.910	0.960	0.010

Tabelle A.1: Bewertungskriterien für zweidimensionale Projektionen von $D_{\text{Sim,Lern},1}$ (hohe Ausprägung der Phänomene). Korr: $x1$ u. $x2$, Cluster: $x3$ u. $x4$, Konfig: $x5$ u. $x6$, Aus: $x7$ u. $x8$, Ortho: $x9$ u. $x10$.

Kriterium	Korr	Cluster	Konfig	Aus	Ortho
q_{Korr}	0.520	0.970	0.950	0.860	0.540
q_{Cluster}	0.990	0.670	0.950	0.990	0.980
q_{Konfig}	1.000	1.000	0.420	1.000	1.000
q_{Aus}	1.000	1.000	1.000	0.350	1.000
q_{Ortho}	0.930	0.650	0.960	0.990	0.650

Tabelle A.2: Bewertungskriterien für zweidimensionale Projektionen von $D_{\text{Sim,Lern},2}$ (mittlere Ausprägung der Phänomene). Korr: $x1$ u. $x2$, Cluster: $x3$ u. $x4$, Konfig: $x5$ u. $x6$, Aus: $x7$ u. $x8$, Ortho: $x9$ u. $x10$.

Kriterium	Korr	Cluster	Konfig	Aus	Ortho
q_{Korr}	0.790	0.960	1.000	0.950	0.790
q_{Cluster}	0.990	0.990	0.990	0.990	0.980
q_{Konfig}	1.000	1.000	0.820	1.000	1.000
q_{Aus}	1.000	1.000	1.000	1.000	1.000
q_{Ortho}	0.940	0.710	0.950	0.840	0.970

Tabelle A.3: Bewertungskriterien für zweidimensionale Projektionen von $D_{\text{Sim,Lern},3}$ (geringe Ausprägung der Phänomene). Korr: $x1$ u. $x2$, Cluster: $x3$ u. $x4$, Konfig: $x5$ u. $x6$, Aus: $x7$ u. $x8$, Ortho: $x9$ u. $x10$.

A.2 Implementierung des GSMO-Algorithmus

Listing A.1 zeigt die MATLAB-Implementierung des in Abschnitt 4.1.3 vorgestellten analytischen Teils des GSMO-Algorithmus.

```
%Parametervektor nach letztem Optimierungsschritt: beta
beta_alt=beta;
beta_su = beta(u);
beta_sv = beta(v);
s_s = beta_su+beta_sv;
eta = K(u,u)+K(v,v)-2*K(u,v);
%Wähle u und v, dass x_u ungleich x_v
if abs(eta-xu)>0
    %Berechne linke und rechte Grenzwerte der Unstetigkeiten
    W0=epsilon(u)*sign(-s_s)-y(v)+y(u)+(einheitsvektor(N,v)-
    einheitsvektor(N,u))'*K*beta_alt+eta*(-beta_alt(v));
    Ws=epsilon(v)*sign(s_s)-y(v)+y(u)+(einheitsvektor(N,v)-
    einheitsvektor(N,u))'*K*beta_alt+eta*(s_s-beta_alt(v));
    Wm0=W0-epsilon(v);
    Wp0=W0+epsilon(v);
    Wms=Ws-epsilon(u);
    Wps=Ws+epsilon(u);

    %Falls die Unstetigkeiten an der gleichen Stelle auftreten
    if s_s==0
        Wm0=Wm0-epsilon(u);
        Wp0=Wp0+epsilon(u);
        Wms=Wms-epsilon(v);
        Wps=Wps+epsilon(v);
    end

    %Prüfe die Bedingungen der unterschiedlichen Fälle des
```

```
    %Vorzeichenwechsels

    %Fall 1:
    if 0<min([Wm0 Wms])
        a=min([0 s_s])-1;
        Wa=epsilon(v)*sign(a)+epsilon(u)*sign(a-s_s)-y(v)+y(u)
    +(einheitsvektor(N,v)-einheitsvektor(N,u))'*K*beta_alt;
        test=beta_alt(v)-Wa/eta;
    %Fall 2:
    elseif 0>min([Wp0 Wps])&&0<max([Wm0 Wms])
        a=0.5*s_s;
        Wa=epsilon(v)*sign(a)+epsilon(u)*sign(a-s_s)-y(v)+y(u)
    +(einheitsvektor(N,v)-einheitsvektor(N,u))'*K*beta_alt;
        test=beta_alt(v)-Wa/eta;
    %Fall 3:
    elseif 0>max([Wp0 Wps])
        a=max([0 s_s])+1;
        Wa=epsilon(v)*sign(a)+epsilon(u)*sign(a-s_s)-y(v)+y(u)
    +(einheitsvektor(N,v)-einheitsvektor(N,u))'*K*beta_alt;
        test=beta_alt(v)-Wa/eta;
    %Fall 4:
    elseif 0>=Wm0 && 0 <=Wp0
        a=0;
        Wa=0;
        test=0;
    %Fall 5:
    elseif 0 >=Wms && 0 <=Wps
        a=s_s;
        Wa=0;
        test=s_s;
    %Fehlerbehandlung
    else
        keyboard;
    end

    %Bestimme untere und obere Grenze
    untere_Grenze=max([-C(v) -C(u)+s_s]);
    obere_Grenze=min([C(v) C(u)+s_s]);

    %Berechne analytische Lösung:
    beta_j_new_gr=min([max([test untere_Grenze]) obere_Grenze
    ]);

end
```

Listing A.1: Implementierung des analytischen Schritts der GSMO

A.3 Regressionsmodelle zur Intentionsschätzung

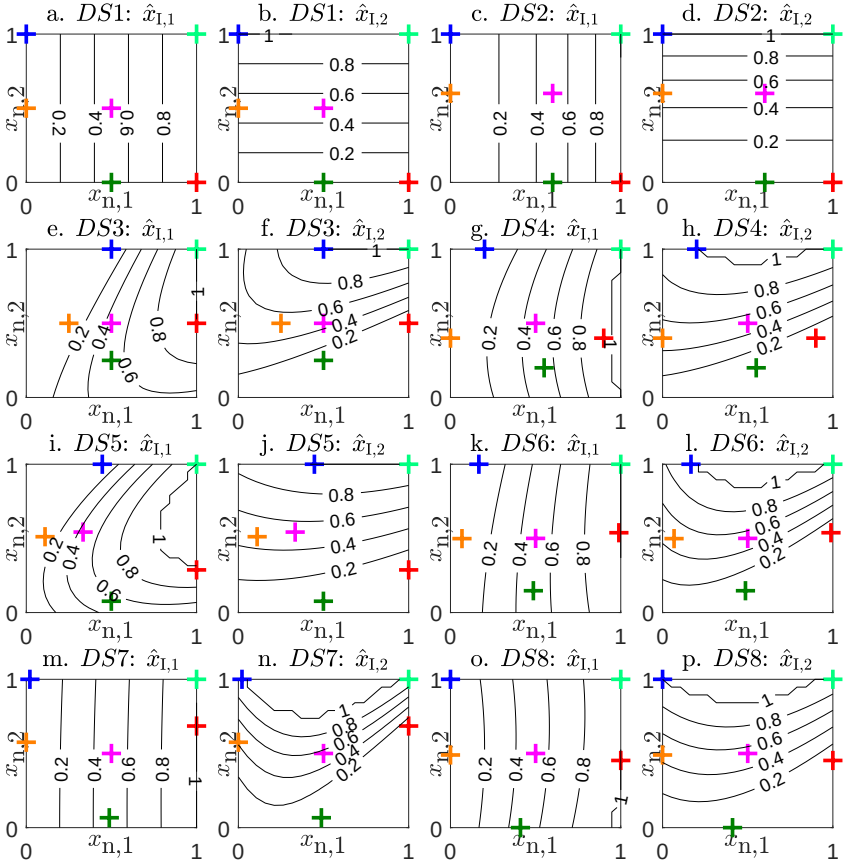

Abbildung A.1: Polynommodelle zur Abbildung der herkömmlichen Datensätze

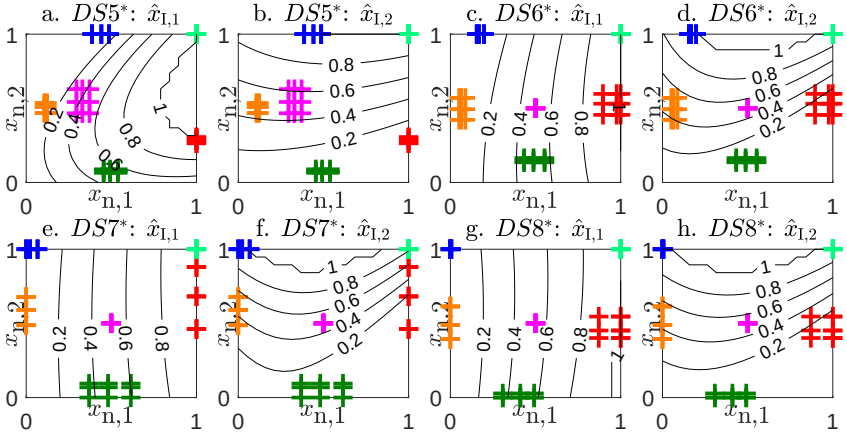

Abbildung A.2: Polynommodelle zur Abbildung der erweiterten Datensätze

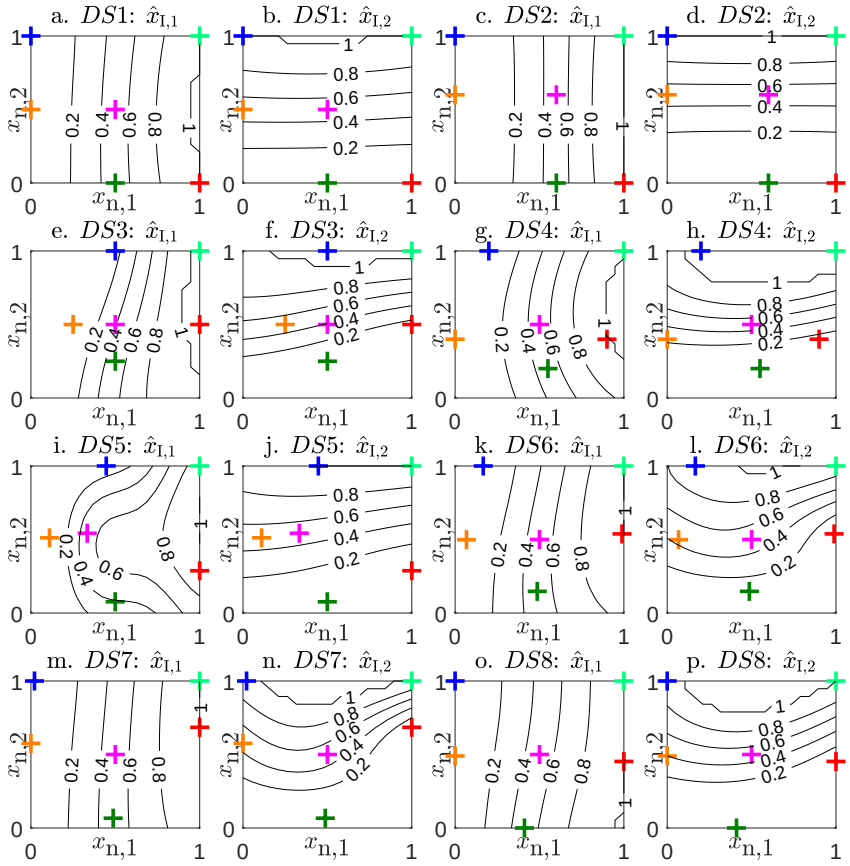

Abbildung A.3: SVR-Modelle zur Abbildung der herkömmlichen Datensätze

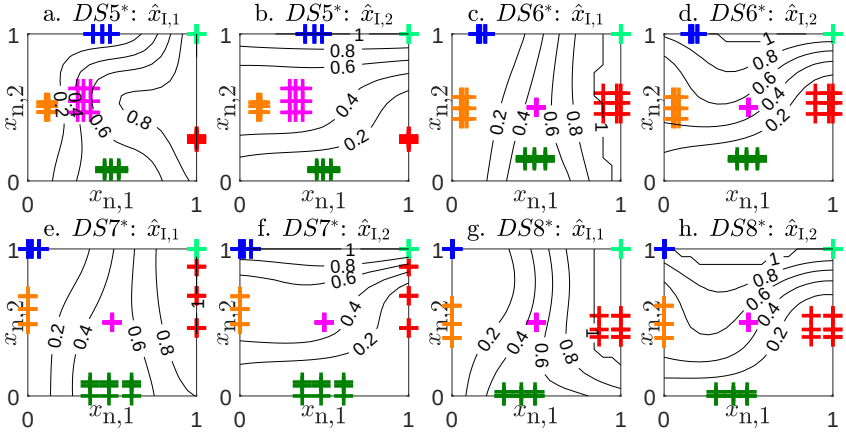

Abbildung A.4: SVR-Modelle zur Abbildung der erweiterten Datensätze

A.4 Nachjustierung von Kalibrierungsfunktionen

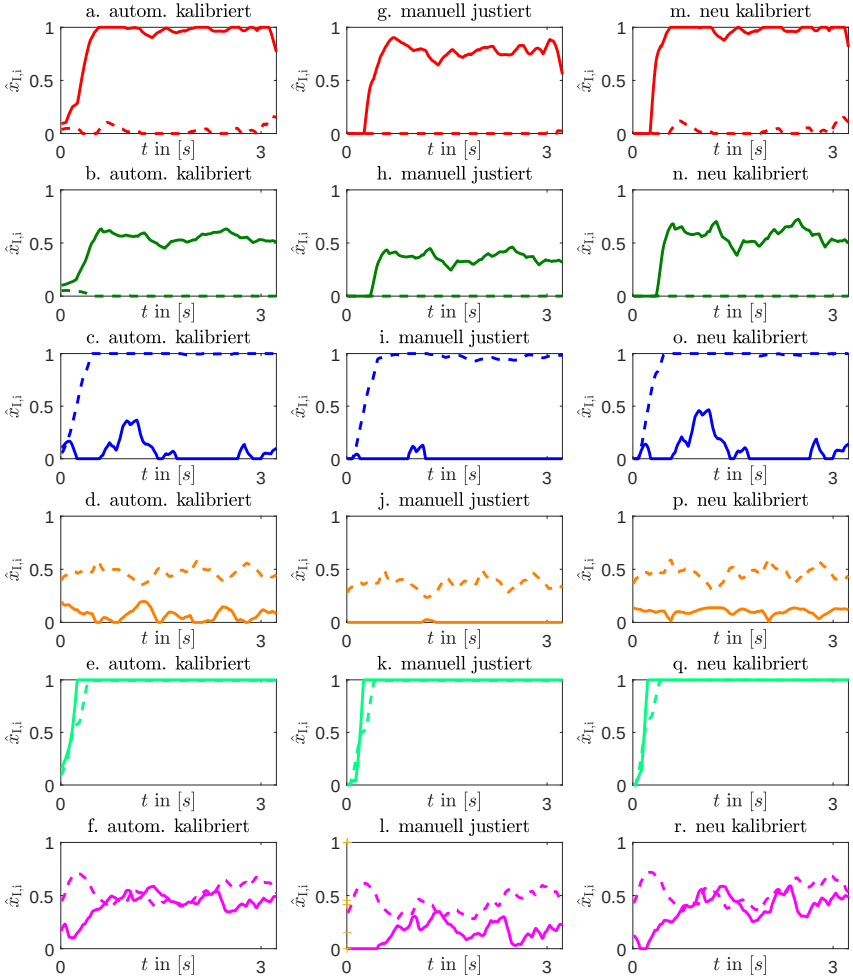

Abbildung A.5: Effekte der manuellen Nachjustierung und erneuten Parameterschätzung der Kalibrierung

Literaturverzeichnis

[1] Pychynski, T.: *Anwendung von Data Mining Methoden zur Analyse von Turbomaschinenkomponenten am Beispiel des Durchflussverhaltens von Labyrinthdichtungen*. Diplomarbeit, Karlsruher Institut für Technologie (KIT). 2009.

[2] Voigtländer, K.: *Ein Beitrag zur Modellierung und Regelung nichtlinearer dynamischer Systeme mittels neuronaler Strukturen*. Dissertation, Technische Universität Dresden, Shaker. 2000.

[3] Doneit, W.; Tuga, M. R.; Mikut, R.; Liebetanz, D.; Rupp, R.; Reischl, M.: Kalibrierungs- und Trainingsstrategien zur individuellen Signalgenerierung für die myoelektrische Steuerung technischer Hilfsmittel. *Technisches Messen* 82(9) (2015), S. 411–421.

[4] Gröll, L.: *Methodik zur Integration von Vorwissen in die Modellbildung*, Bd. 52. Karlsruhe: KIT Scientific Publishing. 2015.

[5] Box, G. E.; Jenkins, G. M.; Reinsel, G. C.: *Time Series Analysis: Forecasting and Control*. Hoboken, NJ, USA: John Wiley & Sons. 2011.

[6] Dannecker, L.: *Energy Time Series Forecasting : Efficient and Accurate Forecasting of Evolving Time Series from the Energy Domain*. Wiesbaden: Springer Vieweg. 2015.

[7] De Gooijer, J. G.; Hyndman, R. J.: 25 Years of Time Series Forecasting. *International Journal of Forecasting* 22(3) (2006), S. 443–473.

[8] Mikut, R.: *Data Mining in der Medizin und Medizintechnik*. Karlsruhe: Universitätsverlag Karlsruhe. 2008.

[9] Bishop, C. M.: *Neural Networks for Pattern Recognition*. Oxford: Oxford University Press. 1995.

[10] Haykin, S.: *Neural Networks: A Comprehensive Foundation*. Upper Saddle River, NJ, USA: Prentice Hall. 1994.

[11] Cortez, P.; Embrechts, M. J.: Opening Black Box Data Mining Models Using Sensitivity Analysis. In: *Proc., IEEE Symposium on Computational Intelligence and Data Mining, Paris, France*, S. 341–348. Piscataway, NJ, USA: IEEE. 2011.

[12] Tzeng, F.-Y.; Ma, K.-L.: Opening the Black Box-Data Driven Visualization of Neural Networks. In: *Proc., IEEE Visualization, Minneapolis, MN*, S. 383–390. Piscataway, NJ, USA: IEEE. 2005.

[13] Huang, G.-B.; Zhu, Q.-Y.; Siew, C.-K.: Extreme Learning Machine: Theory and Applications. *Neurocomputing* 70(1) (2006), S. 489–501.

[14] Huang, G.-B.; Zhou, H.; Ding, X.; Zhang, R.: Extreme Learning Machine for Regression and Multiclass Classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 42(2) (2012), S. 513–529.

[15] Nerrand, O.; Roussel-Ragot, P.; Urbani, D.; Personnanz, L.; Dreyfus, G.: Training Recurrent Neural Networks: Why and How? A Illustration in Dynamical Process Modeling. *IEEE Transactions on Neural Networks* 5(2) (1994), S. 178–184.

[16] Connor, J. T.; Martin, R. D.; Atlas, L. E.: Recurrent Neural Networks and Robust Time Series Prediction. *IEEE Transactions on Neural Networks* 5(2) (1994), S. 240–254.

[17] Schmidhuber, J.: Deep Learning in Neural Networks: An Overview. *Neural Networks* 61 (2015), S. 85–117.

[18] LeCun, Y.; Bengio, Y.; Hinton, G.: Deep Learning. *Nature* 521(7553) (2015), S. 436–444.

[19] Yang, J.; Nguyen, M. N.; San, P. P.; Li, X. L.; Krishnaswamy, S.: Deep Convolutional Neural Networks on Multichannel Time Series for Human Activity Recognition. In: *Proc., 24th International Joint Conference on Artificial Intelligence, Buenos Aires, Argentina*. Palo Alto, CA, USA: AAAI Press. 2015.

[20] Zheng, Y.; Liu, Q.; Chen, E.; Ge, Y.; Zhao, J. L.: Time Series Classification Using Multi-Channels Deep Convolutional Neural Networks. In: *Proc., International Conference on Web-Age Information Management, Macau, China*, S. 298–310. Cham, Switzerland: Springer. 2014.

[21] Du, J.; Xu, Y.: Hierarchical Deep Neural Network for Multivariate Regression. *Pattern Recognition* 63 (2017), S. 149–157.

[22] LeCun, Y.; Kavukcuoglu, K.; Farabet, C.: Convolutional Networks and Applications in Vision. In: *Proc., IEEE International Symposium on Circuits and Systems (ISCAS), Paris, France*, S. 253–256. Piscataway, NJ, USA: IEEE. 2010.

[23] LeCun, Y.; Bengio, Y.: Convolutional Networks for Images, Speech, and Time Series. *The Handbook of Brain Theory and Neural Networks* 3361(10) (1995), S. 1995.

[24] Lauer, F.; Bloch, G.: Incorporating Prior Knowledge in Support Vector Regression. *Machine Learning* 70(1) (2008), S. 89–118.

[25] Chang, C.-C.; Lin, C.-J.: Training ν-Support Vector Regression: Theory and Algorithms. *Neural Computation* 14(8) (2002), S. 1959–1978.

[26] Gu, B.; Sheng, V. S.; Wang, Z.; Ho, D.; Osman, S.; Li, S.: Incremental Learning for ν-Support Vector Regression. *Neural Networks* 67 (2015), S. 140–150.

[27] Qiu, S.; Lane, T.: Multiple Kernel Learning for Support Vector Regression. Techn. Ber. 2005.

[28] Ruppert, D.; Wand, M. P.: Multivariate Locally Weighted Least Squares Regression. *The Annals of Statistics* (1994), S. 1346–1370.

[29] Bänfer, O.; Nelles, O.: Polynomial Model Tree (POLYMOT) - A New Training Algorithm for Local Model Networks with Higher Degree Polynomials. In: *Proc., IEEE International Conference on Control and Automation, Christchurch, New Zealand*, S. 1571–1576. Piscataway, NJ, USA: IEEE. 2009.

[30] Breiman, L.; Friedman, J. H.; Olshen, R. A.; Stone, C. J.: *Classification and Regression Trees*. Belmont, CA, USA: Wadsworth. 1984.

[31] Loh, W.-Y.: Classification and Regression Trees. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 1(1) (2011), S. 14–23.

[32] Babuška, R.: *Fuzzy Modeling for Control*. Boston: Kluwer. 1998.

[33] Murray-Smith, R.: Local Model Networks and Local Learning. In: *Proc., Fuzzy Duisburg, Duisburg*, S. 24–35. Duisburg. 1994.

[34] Murray-Smith, R.; Johansen, T.: *Multiple Model Approaches to Nonlinear Modelling and Control*. Boca Raton, FL, USA: CRC Press. 1997.

[35] Nauck, D.; Klawonn, F.; Kruse, R.: *Foundations of Neuro-Fuzzy Systems*. New York, NY, USA: John Wiley & Sons. 1997.

[36] Takagi, T.; Sugeno, M.: Fuzzy Identification of Systems and Its Applications to Modeling and Control. *IEEE Transactions on Systems, Man and Cybernetics* 15(1) (1985), S. 116–132.

[37] Kroll, A.: Zur regelungsorientierten Ableitung von Takagi-Sugeno-Modellen. *at-Automatisierungstechnik Methoden und Anwendungen der Steuerungs-, Regelungs-und Informationstechnik* 59(12) (2011), S. 705–720.

[38] Nelles, O.; Fischer, M.: Local Linear Model Trees (LOLIMOT) for Nonlinear System Identification of a Cooling Blast. In: *Proc., European Congress on Intelligent Techniques and Soft Computing (EUFIT), Aachen*, S. 1187–1191. Aachen: Günter Mainz. 1996.

[39] Nelles, O.; Hecker, O.; Isermann, R.: Automatische Strukturselektion für Fuzzy-Modelle zur Identifikation nichtlinearer, dynamischer Prozesse. *at-Automatisierungstechnik* 46(6) (1998), S. 302–311.

[40] Kroll, A.: *Computational Intelligence*. De Gruyter Oldenbourg. 2016.

[41] Kruse, R.; Borgelt, C.; Braune, C.; Mostaghim, S.; Steinbrecher, M.: *Computational Intelligence: A Methodological Introduction*. London, UK: Springer. 2016.

[42] Koskela, T.; Varsta, M.; Heikkonen, J.; Kaski, K.: Time Series Prediction Using Recurrent SOM with Local Linear Models. *International Journal of Knowledge-Based Intelligent Engineering Systems* 2 (1998) 1, S. 60–68.

[43] Nelles, O.: *Nonlinear System Identification: From Classical Approaches to Neural Networks and Fuzzy Models*. Berlin Heidelberg: Springer. 2013.

[44] Schulte, H.: *Approximative Modellierung, Systemidentifikation und Reglerentwurf mittels gewichteter Kombinationen lokaler Zustandsraummodelle am Beispiel fluidischer Antriebe*. Dissertation, Universität Kassel. 2005.

[45] Hartmann, B.; Nelles, O.: Identifikation mit achsenschrägen, lokal polynomialen Modellnetzen. *at-Automatisierungstechnik* 62(6) (2014), S. 394–407.

[46] Bishop, C. M.: *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Secaucus, NJ, USA: Springer. 2006.

[47] Waterhouse, S. R.: *Classification and Regression Using Mixtures of Experts*. Dissertation, University of Cambridge. 1998.

[48] Yuksel, S. E.; Wilson, J. N.; Gader, P. D.: Twenty Years of Mixture of Experts. *IEEE Transactions on Neural Networks and Learning Systems* 23 (2012) 8, S. 1177–1193.

[49] Breiman, L.: Bagging Predictors. *Machine Learning* 24(2) (1996), S. 123–140.

[50] Schapire, R.; Freund, Y.; Bartlett, P.; Lee, W.: Boosting the Margin: A New Explanation for the Effectiveness of Voting Methods. *The Annals of Statistics* 26(5) (1998), S. 1651–1686.

[51] Avnimelech, R.; Intrator, N.: Boosting Regression Estimators. *Neural Computation* 11(2) (1999), S. 499–520.

[52] Aha, D. W.; Kibler, D.; Albert, M. K.: Instance-Based Learning Algorithms. *Machine Learning* 6(1) (1991), S. 37–66.

[53] Moore, A. W.: Efficient Memory-Based Learning for Robot Control. Techn. Ber., University of Cambridge. 1990.

[54] Schaal, S.; Atkeson, C. G.: Robot Juggling: Implementation of Memory-Based Learning. *IEEE Control Systems* 14(1) (1994), S. 57–71.

[55] Bontempi, G.; Birattari, M.; Bersini, H.: Lazy Learning for Local Modelling and Control Design. *International Journal of Control* 72(7-8) (1999), S. 643–658.

[56] Zapranis, A.; Livanis, E.: Prediction Intervals for Neural Network Models. In: *Proc., 9th WSEAS International Conference on Computers, Genova, Italy*, S. 76. World Scientific and Engineering Academy and Society (WSEAS), Stevens Point, WI, USA: WSEAS Press. 2005.

[57] Shrestha, D. L.; Solomatine, D. P.: Machine Learning Approaches for Estimation of Prediction Interval for the Model Output. *Neural Networks* 19(2) (2006), S. 225–235.

[58] Rasmussen, C. E.; Williams, C. K.: Gaussian Processes in Machine Learning. In: *Advanced Lectures on Machine Learning*, S. 63–71. Wiesbaden: Springer. 2004.

[59] Moré, J. J.: The Levenberg-Marquardt Algorithm: Implementation and Theory. In: *Numerical Analysis* (Watson, G. A., Hg.), S. 105–116. Berlin: Springer. 1977.

[60] Platt, J.: Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines. Techn. Ber. MSR-TR-98-14, Microsoft. 1998.

[61] Akaike, H.: Information Theory and an Extension of the Maximum Likelihood Principle. In: *Proc., 2nd International Symposium on Information Theory, Tsahkadsor, Armenia*, S. 267–281. Budapest, Hungary: Akademie. 1972.

[62] Mallows, C. L.: Some Comments on Cp. *Technometrics* 15(4) (1973), S. 661–675.

[63] Fahrmeir, L.; Kneib, T.; Lang, S.: *Regression*. Berlin Heidelberg: Springer. 2009.

[64] Bosnić, Z.; Kononenko, I.: Comparison of Approaches for Estimating Reliability of Individual Regression Predictions. *Data & Knowledge Engineering* 67(3) (2008), S. 504–516.

[65] Briesemeister, S.; Rahnenführer, J.; Kohlbacher, O.: No Longer Confidential: Estimating the Confidence of Individual Regression Predictions. *PLoS ONE* 7(11) (2012).

[66] Bishop, C. M.: Novelty Detection and Neural Network Validation. *IEE Proceedings - Vision, Image and Signal Processing* 141(4) (1994), S. 217–222.

[67] Parzen, E.: On Estimation of a Probability Density Function and Mode. *The Annals of Mathematical Statistics* (1962), S. 1065–1076.

[68] Tagscherer, D.-I. M.; Lewandowski, A.: Abschätzung der Vertrauenswürdigkeit von Neuronalen Netzprognosen bei der Prozessoptimierung. *VDI Berichte* 1526 (2000), S. 335–340.

[69] Leonard, J.; Kramer, M. A.; Ungar, L.: A Neural Network Architecture that Computes its Own Reliability. *Computers & Chemical Engineering* 16(9) (1992), S. 819–835.

[70] Leonard, J. A.; Kramer, M. A.; Ungar, L. H.: Using Radial Basis Functions to Approximate a Function and its Error Bounds. *IEEE Transactions on Neural Networks* 3(4) (1992), S. 624–627.

[71] de Ridder, D.; Tax, D.; Duin, R.: An Experimental Comparison of One-Class Classification Methods. In: *Proc., 4th Annual Conference of the Advanced School for Computing and Imaging, Delft, Netherlands*, S. 213–218. 1998.

[72] Tax, D. M. J.: *One-Class Classification*. Dissertation, Technische Universität Delft. 2001.

[73] Shin, H. J.; Eom, D.-H.; Kim, S.-S.: One-class Support Vector Machines - an Application in Machine Fault Detection and Classification. *Computers & Industrial Engineering* 48(2) (2005), S. 395–408.

[74] Pimentel, M. A.; Clifton, D. A.; Clifton, L.; Tarassenko, L.: A Review of Novelty Detection. *Signal Processing* 99 (2014), S. 215–249.

[75] Vapnik, V.; Golowich, S. E.; Smola, A.: Support Vector Method for Function Approximation, Regression Estimation, and Signal Processing. In: *Proc., Advances in Neural Information Processing Systems 9*. Cambridge, MA, USA: MIT Press. 1996.

[76] Schölkopf, B.; Platt, J. C.; Shawe-Taylor, J.; Smola, A. J.; Williamson, R. C.: Estimating the Support of a High-Dimensional Distribution. *Neural Computation* 13(7) (2001), S. 1443–1471.

[77] Schafer, J. L.; Graham, J. W.: Missing Data: Our View of the State of the Art. *Psychological Methods* 7(2) (2002), S. 147.

[78] Musil, C. M.; Warner, C. B.; Yobas, P. K.; Jones, S. L.: A Comparison of Imputation Techniques for Handling Missing Data. *Western Journal of Nursing Research* 24(7) (2002), S. 815–829.

[79] Bandemer, H.; Bellmann, A.: *Statistische Versuchsplanung*. Stuttgart: Teubner. 1994.

[80] Scott, D. W.: *Multivariate Density Estimation: Theory, Practice, and Visualization*. Hoboken, NJ, USA: John Wiley & Sons. 2015.

[81] Breunig, M. M.; Kriegel, H.-P.; Ng, R. T.; Sander, J.: LOF: Identifying Density-Based Local Outliers. In: *ACM SIGMOD record*, Bd. 29(2), S. 93–104. New York, NY, USA: ACM. 2000.

[82] Rousseeuw, P. J.; Van Zomeren, B. C.: Unmasking Multivariate Outliers and Leverage Points. *Journal of the American Statistical Association* 85 (1990) 411, S. 633–639.

[83] Aggarwal, C. C.; Yu, P. S.: Outlier Detection for High Dimensional Data. In: *Proc., ACM SIGMOD International Conference on Management of Data, Santa Barbara, CA, USA*, S. 37–46. New York, NY, USA: ACM. 2001.

[84] Ben-Gal, I.: Outlier Detection. In: *Data Mining and Knowledge Discovery Handbook*, S. 131–146. New York, NY, USA: Springer. 2005.

[85] Blatná, D.: Outliers in Regression. *Trutnov* 30 (2006).

[86] Hodge, V. J.; Austin, J.: A Survey of Outlier Detection Methodologies. *Artificial Intelligence Review* 22(2) (2004), S. 85–126.

[87] Koenker, R.; Bassett, G.: Robust Tests for Heteroscedasticity Based on Regression Quantiles. *Econometrica* 50(1) (1982), S. 43–61.

[88] Breusch, T. S.; Pagan, A. R.: A Simple Test for Heteroscedasticity and Random Coefficient Variation. *Econometrica* 47(5) (1979), S. 1287–1294.

[89] Tukey, J. W.; Tukey, P. A.: Computer Graphics and Exploratory Data Analysis: An Introduction. *The Collected Works of John W. Tukey: Graphics: 1965-1985* 5 (1988), S. 419.

[90] Wilkinson, L.; Anand, A.; Grossman, R. L.: Graph-Theoretic Scagnostics. In: *Proc., INFOVIS, Minneapolis, MN, USA*, Bd. 5, S. 21. IEEE, Piscataway, NJ, USA: IEEE. 2005.

[91] Wilkinson, L.; Wills, G.: Scagnostics Distributions. *Journal of Computational and Graphical Statistics* 17(2) (2008), S. 473–491.

[92] Cherkassky, V.; Ma, Y.: Multiple Model Regression Estimation. *IEEE Transactions on Neural Networks* 16(4) (2005), S. 785–798.

[93] García-Escudero, L. A.; Gordaliza, A.; Mayo-Iscar, A.; San Martín, R.: Robust Clusterwise Linear Regression Through Trimming. *Computational Statistics & Data Analysis* 54(12) (2010), S. 3057–3069.

[94] Bauer, C.: *Entwicklung einer Methode zur automatisierten Bewertung der Konsistenz von Datenquellen mit Data Mining*. Bachelorarbeit, Karlsruher Institut für Technologie. 2012.

[95] Quandt, R. E.; Ramsey, J. B.: Estimating Mixtures of Normal Distributions and Switching Regressions. *Journal of the American Statistical Association* 73(364) (1978), S. 730–738.

[96] Aitkin, M.; Wilson, G. T.: Mixture Models, Outliers, and the EM Algorithm. *Technometrics* 22(3) (1980), S. 325–331.

[97] McLachlan, G.; Peel, D.: *Finite Mixture Models*. Hoboken, NJ, USA: John Wiley & Sons. 2004.

[98] He, H.; Chen, S.; Li, K.; Xu, X.: Incremental Learning from Stream Data. *IEEE Transactions on Neural Networks* 22(12) (2011), S. 1901–1914.

[99] Tsymbal, A.: The Problem of Concept Drift: Definitions and Related Work. *Computer Science Department, Trinity College Dublin* 106(2) (2004).

[100] Vijayakumar, S.; Schaal, S.: Local Adaptive Subspace Regression. *Neural Processing Letters* 7(3) (1998), S. 139–149.

[101] Rousseeuw, P. J.; Leroy, A. M.: *Robust Regression and Outlier Detection*, Bd. 589. New York, NY, USA: John Wiley & Sons. 2005.

[102] Meer, P.; Mintz, D.; Rosenfeld, A.; Kim, D. Y.: Robust Regression Methods for Computer Vision: A Review. *International Journal of Computer Vision* 6(1) (1991), S. 59–70.

[103] Gabrel, V.; Murat, C.; Thiele, A.: Recent Advances in Robust Optimization: An Overview. *European Journal of Operational Research* 235(3) (2014), S. 471–483.

[104] Alfons, A.; Croux, C.; Gelper, S.; et al.: Sparse Least Trimmed Squares Regression for Analyzing High-Dimensional Large Data Sets. *The Annals of Applied Statistics* 7(1) (2013), S. 226–248.

[105] Yin, S.; Wang, G.; Yang, X.: Robust PLS Approach for KPI-related Prediction and Diagnosis against Outliers and Missing Data. *International Journal of Systems Science* 45(7) (2014), S. 1375–1382.

[106] Hoerl, A. E.; Kennard, R. W.: Ridge Regression: Biased Estimation for Nonorthogonal Problems. *Technometrics* 12(1) (1970), S. 55–67.

[107] Larsen, J.; Hansen, L. K.: Generalization Performance of Regularized Neural Network Models. In: *Proc., 4th IEEE Workshop of Neural Networks for Signal Processing, Ermioni, Greece*, S. 42–51. New York, NY, USA: IEEE. 1994.

[108] Girosi, F.; Jones, M.; Poggio, T.: Regularization Theory and Neural Networks Architectures. *Neural Computation* 7(2) (1995), S. 219–269.

[109] Jolliffe, I.: *Principal Component Analysis*. New York, NY, USA: Springer. 2002.

[110] Geladi, P.; Kowalski, B. R.: Partial Least-Squares Regression: A Tutorial. *Analytica Chimica Acta* 185 (1986), S. 1–17.

[111] Hoffmann, H.: Kernel PCA for Novelty Detection. *Pattern Recognition* 40(3) (2007), S. 863–874.

[112] Van Huffel, S.; Vandewalle, J.: *The Total Least Squares Problem: Computational Aspects and Analysis*, Bd. 9. Philadelphia, PA, USA: Siam. 1991.

[113] Golub, G. H.; Van Loan, C.: Total Least Squares. In: *Proc., Smoothing Techniques for Curve Estimation, Heidelberg*, S. 69–76. Berlin Heidelberg: Springer. 1979.

[114] Markovsky, I.; Van Huffel, S.: Overview of Total Least-Squares Methods. *Signal Processing* 87(10) (2007), S. 2283–2302.

[115] Reinhart, R. F.; Shareef, Z.; Steil, J. J.: Hybrid Analytical and Data-Driven Modeling for Feed-Forward Robot Control. *Sensors* 17(2) (2017), S. 311.

[116] Caluwaerts, K.; Steil, J. J.: Independent Joint Learning in Practice: Local Error Estimates to Improve Inverse Dynamics Control. In: *Proc., 15th International Conference on Humanoid Robots, Seoul, South Korea*, S. 643–650. IEEE, Piscataway, NJ, USA: IEEE. 2015.

[117] Reinhart, R. F.; Steil, J. J.: Hybrid Mechanical and Data-driven Modeling Improves Inverse Kinematic Control of a Soft Robot. *Procedia Technology* 26 (2016), S. 12–19.

[118] Wen, J.: *Hybrid Approach of Neural Networks with Knowledge Based Explicit Models*. Dissertation, ETH Zürich. 1995.

[119] Tsai, P.-F.; Chu, J.-Z.; Jang, S.-S.; Shieh, S.-S.: Developing a Robust Model Predictive Control Architecture Through Regional Knowledge Analysis of Artificial Neural Networks. *Journal of Process Control* 13(5) (2003), S. 423–435.

[120] Runkler, T.: *Data Mining*. Wiesbaden: Springer. 2009.

[121] Günzel, H.; Bauer, A.: *Data-Warehouse-Systeme: Architektur, Entwicklung, Anwendung*. Heidelberg: dpunkt.verlag. 2013.

[122] Kemper, H.-G.; Mehanna, W.; Unger, C.: *Business Intelligence - Grundlagen und praktische Anwendungen*. Wiesbaden: Springer. 2004.

[123] Rahm, E.; Do, H.: Data cleaning: Problems and Current Approaches. *IEEE Data Engineering Bulletin* 23(4) (2000), S. 3–13.

[124] Otto, B.: *Corporate Data Quality*. Wiesbaden: Springer. 2016.

[125] Guillet, F.; Hamilton, H. J.: *Quality Measures in Data Mining*, Bd. 43. Berlin Heidelberg: Springer. 2007.

[126] Sadiq, S.: *Handbook of Data Quality*. Berlin Heidelberg: Springer. 2013.

[127] Parsons, L.; Haque, E.; Liu, H.: Subspace Clustering for High Dimensional Data: A Review. *ACM SIGKDD Explorations Newsletter* 6(1) (2004), S. 90–105.

[128] Oliveira, P.; Rodrigues, F.; Henriques, P.; Galhardas, H.: A Taxonomy of Data Quality Problems. In: *Proc., 2nd International Workshop on Data and Information Quality*. Citeseer. 2005.

[129] Batini, C.; Cappiello, C.; Francalanci, C.; Maurino, A.: Methodologies for Data Quality Assessment and Improvement. *ACM Computing Surveys* 41(3) (2009), S. 16.

[130] Bandemer, H.: *Theorie und Anwendung der optimalen Versuchsplanung*. Berlin: Akademie. 1977.

[131] Lichman, M.: UCI Machine Learning Repository. URL `http://archive.ics.uci.edu/ml`. 2013.

[132] Nash, W. J.; Sellers, T. L.; Talbot, S. R.; Cawthorn, A. J.; Ford, W. B.: The Population Biology of Abalone (Haliotis Species) in Tasmania. I. Blacklip Abalone (h. rubra) from the North Coast and Islands of Bass Strait. Techn. Ber. 48. 1994.

[133] Pace, R. K.; Barry, R.: Sparse Spatial Autoregressions. *Statistics & Probability Letters* 33(3) (1997), S. 291–297.

[134] Yeh, I.-C.: Modeling of Strength of High-Performance Concrete Using Artificial Neural Networks. *Cement and Concrete Research* 28(12) (1998), S. 1797–1808.

[135] Cortez, P.; Cerdeira, A.; Almeida, F.; Matos, T.; Reis, J.: Modeling Wine Preferences by Data Mining from Physicochemical Properties. *Decision Support Systems* 47(4) (2009), S. 547–553.

[136] Doneit, W.; Mikut, R.; Reischl, M.: Datenqualität in Regressionsproblemen. *arXiv:1701.04342* (2016).

[137] Steinbach, M.; Ertöz, L.; Kumar, V.: The Challenges of Clustering High Dimensional Data. In: *New Directions in Statistical Physics*, S. 273–309. Berlin Heidelberg: Springer. 2004.

[138] Hartigan, J. A.; Hartigan, P.: The Dip Test of Unimodality. *The Annals of Statistics* (1985), S. 70–84.

[139] Freeman, J. B.; Dale, R.: Assessing Bimodality to Detect the Presence of a Dual Cognitive Process. *Behavior Research Methods* 45(1) (2013), S. 83–97.

[140] Schultz, T.; Putze, F.; Schulze, T.; Mikut, R.; Doneit, W.; Kruse, A.; Depner, A.; Franz, I.; Engels, M.; Gaerte, P.; Bothe, D.; Ziegler, C.; Maucher, I.; Ricken, M.; Dimitrov, T.; Herzig, J.; Bernardin, K.; Gehrig, T.; Lohse, J.; Adam, M.; Fischer, M.; Volpe, M.; Simon, C.: I-CARE: Individual Activation of People with Dementia. In: *Proc., KogWis 2016: Space for Cognition, Bremen.* URL `http://nbn-resolving.de/urn:nbn:de:gbv:46-00105521-19`. 2016.

[141] Doneit, W.; Lohse, J.; Glesing, K.; Simon, C.; Fischer, M.; Depner, A.; Kruse, A.; Franz, I.; Schultz, T.; Putze, F.; Schulze, T.; Engels, M. A.; Gaerte, P.; Bothe, D.; Ziegler, C.; Maucher, I.; Ricken, M.; Dimitrov, T.; Herzig, J.; Bernardin, K.; Gehrig, T.; Mikut, R.: Data-driven Analysis of Interactions between People with Dementia and a Tablet Device. *Current Directions in Biomedical Engineering* 3(2) (2017), S. 735–738.

[142] Jefferys, W. H.; Berger, J. O.: Ockham's Razor and Bayesian Analysis. *American Scientist* 80(1) (1992), S. 64–72.

[143] Reuter, W.: *Entwicklung einer neuen Methode zur Bewertung von Überanpassung von datenbasierten Modellen mit Lerndaten.* Bachelorarbeit, Karlsruher Institut für Technologie. 2012.

[144] Doneit, W.; Mikut, R.; Pychynski, T.; Reischl, M.: Abstands- und Monotoniemaße für Regressionsmodelle mit heterogenen Lerndaten. In: *Proc., 24. Workshop Computational Intelligence, Dortmund*, S. 1–16. Karlsruhe: KIT Scientific Publishing. 2014.

[145] Doneit, W.; Mikut, R.; Gröll, L.; Reischl, M.: Fragebogen zur Erfassung von Vorwissen in Funktionsapproximationen (Version 1.0). Techn. Ber., Institut für Angewandte Informatik, KIT. 2015.

[146] Joerding, W. H.; Meador, J. L.: Encoding a Priori Information in Feedforward Networks. *Neural Networks* 4(6) (1991), S. 847–856.

[147] Schenker, B.; Agarwal, M.: Using A Priori Information in Networks. In: *Proc., 2nd International Conference on Artificial Neural Networks, Bournemouth, UK*, S. 242–246. London, UK: IET. 1991.

[148] Di Muro, G.; Ferrari, S.: A Constrained-Optimization Approach to Training Neural Networks for Smooth Function Approximation and System Identification. In: *Proc., IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), Hong Kong, China*, S. 2353–2359. Piscataway, NJ, USA: IEEE. 2008.

[149] Mangasarian, O. L.; Shavlik, J. W.; Wild, E. W.: Knowledge-Based Kernel Approximation. *Journal of Machine Learning Research* 5(9) (2004), S. 1127–1141.

[150] Zhou, J.; Huang, J.: Incorporating Priori Knowledge Into Linear Programming Support Vector Regression. In: *Proc., International Conference on Intelligent Computing and Integrated Systems (ICISS), Guilin, China*, S. 591–595. Piscataway, NJ, USA: IEEE. 2010.

[151] Vetter, T.; Poggio, T.; Bulthoff, H.: 3D Object Recognition: Symmetry and Virtual Views. Techn. Ber., Massachusetts Institute of Technology (MIT). 1992.

[152] Vapnik, V.: *The Nature of Statistical Learning Theory*. New York, NY, USA: Springer. 1995.

[153] Smola, A. J.; Scholkopf, B.: A Tutorial on Support Vector Regression: NeuroCOLT2 Technical Report Series. *NC2-TR-1998–030* (1998).

[154] Vapnik, V. N.; Kotz, S.: *Estimation of Dependences Based on Empirical Data*, Bd. 40. New York, NY, USA: Springer. 1982.

[155] Basak, D.; Pal, S.; Patranabis, D. C.: Support Vector Regression. *Neural Information Processing - Letters and Reviews* 11(10) (2007), S. 203–224.

[156] Flake, G. W.; Lawrence, S.: Efficient SVM Regression Training with SMO. *Machine Learning* 46(1-3) (2002), S. 271–290.

[157] Mattera, D.; Palmieri, F.; Haykin, S.: An Explicit Algorithm for Training Support Vector Machines. *IEEE Signal Processing Letters* 6(9) (1999), S. 243–245.

[158] De Leeuw, J.: Block-Relaxation Algorithms in Statistics. In: *Information Systems and Data Analysis*, S. 308–324. Berlin Heidelberg: Springer. 1994.

[159] Cherkassky, V.; Ma, Y.: Selection of Meta-Parameters for Support Vector Regression. In: *Proc., International Conference on Artificial Neural Networks, Madrid, Spain*, S. 687–693. Berlin Heidelberg: Springer. 2002.

[160] Doneit, W.; Mikut, R.; Gröll, L.; Pychynski, T.; Reischl, M.: DaMoQ: Eine Open-Source-MATLAB-Toolbox zur Bewertung von Daten- und Modellqualität in Regressionen. *at-Automatisierungstechnik* 65(3) (2017), S. 207–2018.

[161] Weinberger, T.: *Einfluss geometrischer Labyrinth-und Honigwabenparameter auf das Durchfluss-und Wärmeübergangsverhalten von Labyrinthdichtungen: Experiment, Numerik und Data Mining*. Dissertation, Karlsruher Institut für Technologie (KIT), Logos. 2014.

[162] Pychynski, T.; Blesinger, G.; Mikut, R.; Dullenkopf, K.; Bauer., H.-J.: Modelling the Labyrinth Seal Discharge Coefficient Using Data Mining Methods. In: *Proc., ASME TURBO EXPO, Glasgow, UK*. New York, NY, USA: ASME. 2010.

[163] Plevan, M.: *Application of Data Mining Methods to the Model Identification of Temperature Distributions of Labyrinth Seals*. Diplomarbeit, Karlsruher Institut für Technologie (KIT). 2011.

[164] Choi, K.; Sato, M.; Koike, Y.: A New, Human-Centered Wheelchair System Controlled by the EMG Signal. In: *Proc., International Joint Conference on Neural Networks, Vancouver, BC, Kanada*, S. 4664–4671. Piscataway, NJ, USA: IEEE. 2006.

[165] Silcox, D. H.; Rooks, M. D.; Vogel, R. R.; Fleming, L. L.: Myoelectric Prostheses. *The Journal of Joint and Bone Surgery* 75(12) (1993), S. 1781–1791.

[166] Schulz, S.; Pylatiuk, C.; Reischl, M.; Martin, J.; Mikut, R.; Bretthauer, G.: A Lightweight Multifunctional Prosthetic Hand. *Robotica* 23(3) (2005), S. 293–299.

[167] Ditunno, J.; Scivoletto, G.: Clinical Relevance of Gait Research Applied to Clinical Trials in Spinal Cord Injury. *Brain Research Bulletin* 78(1) (2009), S. 35–42.

[168] Gopura, R.; Bandara, D.; Gunasekara, J.; Jayawardane, T.: Recent Trends in EMG-Based Control Methods for Assistive Robots. In: *Electrodiagnosis in New Frontiers of Clinical Research* (Turker, H., Hg.), S. 237–268. London, UK: InTech. 2013.

[169] Maciejasz, P.; Eschweiler, J.; Gerlach-Hahn, K.; Jansen-Troy, A.; Leonhardt, S.: A Survey on Robotic Devices for Upper Limb Rehabilitation. *Journal of Neuroengineering and Rehabilitation* 11(1) (2014), S. 3.

[170] De Luca, C.; Merletti, R.: Surface Myoelectric Signal Cross-Talk Among Muscles of the Leg. *Electroencephalography Clinical Neurophysiology* 69 (1988), S. 568–575.

[171] Koh, T.; Grabiner, M.: Cross Talk in Surface Electromyograms of Human Hamstring Muscles. *Journal of Orthopaedic Research* 10 (1992), S. 701–709.

[172] Rupp, R.; Schmalfuß, L.; Tuga, M.; Kogut, A.; Hewitt, M.; Meincke, J.; Duttenhöfer, W.; Eck, U.; Mikut, R.; Reischl, M.; Liebetanz, D.: TELMYOS - a Telemetric Wheelchair Control Interface Based on the Bilateral Recording of Myoelectric Signals from Ear Muscles. In: *Proc., Technically Assisted Rehabilitation (TAR) Conference, Berlin*. 2015.

[173] Schmalfuß, L.; Rupp, R.; Tuga, M.; Kogut, A.; Hewitt, M.; Meincke, J.; Klinker, F.; Duttenhoefer, W.; Eck, U.; Mikut, R.; Reischl, M.; Liebetanz, D.: Steer by Ear: Myoelectric Auricular Control of Powered Wheelchairs for Individuals with Spinal Cord Injury. *Restorative Neurology and Neuroscience* 34(1) (2015), S. 79–95.

[174] Tuga, M. R.: *Development and Implementation of an Adaptive Human-Machine Interface Based on Ear Muscle Signals*. Dissertation, Karlsruher Institut für Technologie (KIT). 2016.

[175] Tuga, M. R.; Rupp, R.; Kogut, A.; Liebetanz, D.; Schmalfuß, L.; Mikut, R.; Reischl, M.: Incremental Parameter Adaptation Scheme for Myoelectric-Controlled Human-Machine Interfaces. *Biomedizinische Technik - Biomedical Engineering* 59(S1) (2014), S. S148–S151.

[176] Doneit, W.; Mikut, R.; Liebetanz, D.; Rupp, R.; Reischl, M.: Control Scheme Selection in Human-Machine-Interfaces by Analysis of Activity Signals. *Current Directions in Biomedical Engineering* 2(1) (2016), S. 707–710.

[177] Tuga, M. R.; Rupp, R.; Liebetanz, D.; Schmalfuß, L.; Hübner, E.; Doneit, W.; Mikut, R.; Reischl, M.: Co-Adaptives Lernen: Untersuchungen einer Mensch-Maschine-Schnittstelle mit anpassungsfähigem Systemverhalten. In: *Proc., 23. Workshop Computational Intelligence, Dortmund*, S. 247–264. Karlsruhe: KIT Scientific Publishing. 2013.

[178] Stäb, L.: *Entwicklung einer adaptiven Kalibrierungsroutine für zweikanalige EMG-Messungen*. Bachelorarbeit, Karlsruher Institut für Technologie. 2016.

[179] Wirth, H.; Schneider, K.: Recent Facts about Photovoltaics in Germany. *Report from Fraunhofer Institute for Solar Energy Systems, Germany* (2013).

[180] Scholz, R.; Beckmann, M.; Pieper, C.; Muster, M.; Weber, R.: Considerations on Providing the Energy Needs Using Exclusively Renewable Sources: Energiewende in Germany. *Renewable and Sustainable Energy Reviews* 35 (2014), S. 109–125.

[181] Woyte, A.; Van Thong, V.; Belmans, R.; Nijs, J.: Voltage Fluctuations on Distribution Level Introduced by Photovoltaic Systems. *IEEE Transactions on Energy Conversion* 21(1) (2006), S. 202–209.

[182] Hagenmeyer, V.; Cakmak, H. K.; Düpmeier, C.; Faulwasser, T.; Isele, J.; Keller, H. B.; Kohlhepp, P.; Kühnapfel, U.; Stucky, U.; Waczowicz, S.; Mikut, R.: Information and Communication Technology in Energy Lab 2.0: Smart Energies System Simulation and Control Center with

an Open-Street-Map-Based Power Flow Simulation Example. *Energy Technology* 4 (2016), S. 145–162.

[183] Almonacid, F.; Rus, C.; Pérez-Higueras, P.; Hontoria, L.: Calculation of the Energy Provided by a PV Generator. Comparative Study: Conventional Methods vs. Artificial Neural Networks. *Energy* 36(1) (2011), S. 375–384.

[184] Yang, H.-T.; Huang, C.-M.; Huang, Y.-C.; Pai, Y.-S.: A Weather-Based Hybrid Method for 1 Day Ahead Hourly Forecasting of PV Power Output. *IEEE Transactions on Sustainable Energy* 5(3) (2014), S. 917–926.

[185] González Ordiano, J. Á.; Waczowicz, S.; Reischl, M.; Mikut, R.; Hagenmeyer, V.: Photovoltaic Power Forecasting using Simple Data-driven Models without Weather Data. *Computer Science - Research and Development* 32 (2017), S. 237–246.

[186] Chaouachi, A.; Kamel, R. M.; Nagasaka, K.: Neural Network Ensemble-Based Solar Power Generation Short-Term Forecasting. *JACIII* 14(1) (2010), S. 69–75.

[187] Cococcioni, M.; D'Andrea, E.; Lazzerini, B.: One Day-Ahead Forecasting of Energy Production in Solar Photovoltaic Installations: An Empirical Study. *Intelligent Decision Technologies* 6(3) (2012), S. 197–210.

[188] Tao, C.; Shanxu, D.; Changsong, C.: Forecasting Power Output for Grid-Connected Photovoltaic Power System without Using Solar Radiation Measurement. In: *Proc., 2nd IEEE International Symposium on Power Electronics for Distributed Generation Systems (PEDG), Hefei, China*, S. 773–777. Piscataway, NJ, USA: IEEE. 2010.

[189] González Ordiano, J. Á.; Doneit, W.; Waczowicz, S.; Gröll, L.; Mikut, R.; Hagenmeyer, V.: Nearest-Neighbor Based Non-Parametric Probabilistic Forecasting with Applications in Photovoltaic Systems. In: *Proc., 26. Workshop Computational Intelligence, Dortmund*, S. 9–30. Karlsruhe: KIT Scientific Publishing. 2016.

[190] Almeida, M. P.; Perpiñán, O.; Narvarte, L.: PV Power Forecast Using a Nonparametric PV Model. *Solar Energy* 115 (2015), S. 354–368.

[191] Bruch, R.: *Vergleich von Verfahren zur Merkmalsreduktion und Parameterschätzung linearer Modelle für Leistungsprognosen von Photovoltaikanlagen.* Bachelorarbeit, Karlsruher Institut für Technologie. 2017.

[192] Düpmeier, C.; Stucky, K.-U.; Mikut, R.; Hagenmeyer, V.: A Concept for the Control, Monitoring and Visualization Center in Energy Lab 2.0. In: *Proc., of the 4th D-A-CH Energy Informatics Conference, Karlsruhe*, S. 83–94. Cham, Switzerland: Springer. 2015.