

Ein Beitrag zur Anwendung von
Support-Vektor-Maschinen zur robusten
nichtlinearen Klassifikation komplexer
biologischer Daten

DISSERTATION

zur Erlangung des akademischen Grades
Doktoringenieur
(Dr.-Ing)

vorgelegt der

Fakultät für Informatik und Automatisierung
der Technischen Universität Ilmenau

von

Dipl.-Ing. (FH) Maria Trommer
geboren am 06.10.1984 in Schmalkalden

Gutachter:

1. Prof. Dr.-Ing. habil. Thomas Rauschenbach (TU Ilmenau)
2. Prof. Dr. sc. nat. Dr.-Ing. Rudolf Baumgart-Schmitt (HS Schmalkalden)
3. Prof. Dr.-Ing. habil. Pu Li (TU Ilmenau)

Tag der Einreichung: 01.07.2016

Tag der wissenschaftlichen Aussprache: 03.03.2017

In Gedenken an meine Mutter

Zusammenfassung

Die vorliegende Arbeit untersucht die Eigenschaften der Support-Vektor-Maschine (SVM) für die Anwendung der Klassifikation komplexer Biosignale. Dabei wird eine Methode zur Erzeugung robuster Klassifikatoren für diese Art von Daten mit Hilfe der SVM vorgestellt. Biologische Daten weisen mehrere Faktoren auf, die eine automatische Klassifikation erschweren. Sie sind von Natur aus stark ungleich verteilt und zeigen starke inter- und intraindividuelle Ausprägungen. Weiterhin weichen die als Belehrungsgrundlage verwendeten Expertenbewertungen bedingt durch die Komplexität der Probleme zu einem bestimmten Grad voneinander ab. Als Basis für die Entwicklung und Erprobung der Methoden werden Schlaf-EEG-Daten verwendet.

Die SVM ist eine anerkannte und oft empfohlene Klassifikationsmethode für verschiedene Aufgabenstellungen. So werden im Laufe der Untersuchungen die Vor- und Nachteile dieser Methode beleuchtet. Es finden an allen Schritten der Modellbildung Untersuchungen zu möglichen Optimierungen statt.

So kann diese Arbeit zeigen, dass der SVM-Klassifikator stark von den Eigenschaften der ausgewählten Belehrungsdaten abhängig ist. Sowohl die Komplexität als auch die resultierende Klassifikationsgüte werden hierdurch stark beeinflusst. Es wird anhand künstlicher Probleme anschaulich gezeigt, wie die SVM parameterabhängig auf ungleichverteilte und untypische Daten reagiert. Als Konsequenz dessen wird eine zusätzliche Bearbeitung der Belehrungsdaten zur Modelloptimierung eingeführt. Durch den hierbei entwickelten Algorithmus, dem DiLa-Filter, stellt diese Arbeit eine neuartige und universelle Methode zur Datenfilterung vor und belegt dessen Funktion an realen Daten. Weiterhin wird ein neuer multikriterieller genetischer Algorithmus zur Parameteroptimierung, der SIGA, beschrieben. Dieser kann durch den Einsatz einer neuen genetischen Operation, der Immigration, zur verbesserten Konvergenz der genetischen Algorithmen beitragen.

Liegen statistische Abhängigkeiten zwischen zeitlich aufeinanderfolgenden Klassifikationen vor, so können diese zum einen in einer kontextbasierten Vorhersage angewendet und zum anderen für die Kombination mehrerer SVM-Modelle (Bagging) genutzt werden.

Somit wird eine Methode zur Erzeugung robuster Klassifikatoren mit Hilfe der SVM vorgestellt. Die Einbindung der SVM auf eine ressourcenarme Plattform zeigt die Restriktionen für den Einsatz der SVM für mobile Anwendungen. Die erzielten Resultate werden direkt mit Ergebnissen der Klassifikation durch neuronale Netze verglichen. Dabei schneiden die Netze in fast allen Gütekriterien besser ab. Die These, die SVM sei ein Ersatz für die neuronalen Netze, kann widerlegt werden. Aus den in dieser Arbeit erlangten Erkenntnissen werden weitere Forschungsfragen für zukünftige Arbeiten formuliert.

Abstract

This thesis analyzes the characteristics of the Support Vector Machine (SVM) for the classification of complex biosignals. An approach for the generation of robust classifiers for such kind of data using the SVM is presented. Biological data exhibit several factors that complicate automatic classification processes. They are distributed in a highly imbalanced way and show many inter- and intra-individual variations. Furthermore, the expert scorings as base of supervised learning can differ to a certain extent, caused by the complexity of the problems. In the validation step we focus on sleep EEG data to verify the developed method.

The SVM is a recognized and recommended method for a variety of application tasks. In analyzing the SVM, the advantages and disadvantages of this method are highlighted. Moreover, this work will address optimizations in all stages of the modeling.

This thesis demonstrates, that the SVM classifier strongly depends on the characteristics of chosen training data, influencing both, the complexity and the resulting classification quality. Based on artificial problems it illustrates, how the SVM deals with imbalanced and untypical data depending on the SVM-parameter. As a consequence, training data are modified to optimize the model. With the developed algorithm, the DiLa filter, this thesis presents a novel and universal method for data filtering and proves its performance, using real data. Furthermore a new multi-criteria genetic algorithm for parameter optimization, the SIGA, is integrated. Introducing a new genetic operation, the immigration, this algorithm improves the convergence of genetic algorithms.

When statistical dependencies between consecutive classifications are available, these in turn can be applied for context-based prediction on the one hand and can be used for the combination of several SVM-classifiers (Bagging) on the other hand.

As a result, an approach for the generation of robust classifiers using the SVM is presented. The implementation of SVM software in an embedded resource-limited platform shows the restrictions for the use of SVM for mobile applications. The achieved results are directly compared with the results of the neural networks. Thereby the neural networks obtain better results in almost any quality criteria. The hypothesis that the SVM replaces neural networks can thus be disproven. From the findings of this thesis future research questions are identified.

Vorwort

Die vorliegende Arbeit entstand im Rahmen meiner Tätigkeit als wissenschaftliche Mitarbeiterin an der Hochschule Schmalkalden in der Fakultät Elektrotechnik. In der ehemaligen Forschungsgruppe Neuroinformatik begann ich meinen wissenschaftlichen Werdegang unter der Leitung von Prof. Dr. sc. nat. Dr.-Ing. Rudolf Baumgart-Schmitt. Ihm gilt mein besonderer Dank, da er mir die Aufnahme meiner wissenschaftlichen Tätigkeit ermöglichte und so mein Leben entscheidend beeinflusste. Außerdem danke ich ihm für die jahrelange Betreuung und Unterstützung meiner Arbeit. Großer Dank gilt Prof. Dr.-Ing. Andreas Wenzel, der nach dem Eintritt von Prof. Baumgart-Schmitt in den Ruhestand die Forschungsgruppe übernahm und die Betreuung meiner Arbeit fortführte. Seine Erfahrung und sein Rat waren mir eine große Hilfe für die Umsetzung meiner Arbeit.

Insbesondere möchte ich Prof. Dr.-Ing. Christoph Ament danken, der sich bereit erklärte meine Arbeit im Rahmen einer kooperativen Promotion mit der Technischen Universität Ilmenau zu betreuen. Ohne ihn wäre die Umsetzung dieser Arbeit nicht möglich gewesen. Ich danke auch Prof. Dr.-Ing. habil. Thomas Rauschenbach, der die Betreuung meiner Arbeit nach der Berufung von Prof. Ament nach Augsburg übernahm. Mein Dank gilt weiterhin Prof. Dr.-Ing. habil. Jürgen Wernstedt, der mich bei Veröffentlichungen und dem Abschluss dieser Arbeit mit Rat und Tat unterstützte.

Ebenfalls gilt mein Dank allen ehemaligen und aktiven Mitarbeiter der Forschungsgruppe Eingebettete Systeme, der ehemaligen Forschungsgruppe Neuroinformatik, sowie den anderen Promovenden der Hochschule Schmalkalden für die fachlichen Diskussionen und Anregungen. Besonders möchte ich nennen: Dr.-Ing. Christian Walther, Dipl.-Ing. (FH) Daniel Trommer, Manuel Schneider, M.Sc., und Christoph Menz, M.Sc., für Ihre Hilfe und fachlichen Rat. Weiterhin danke ich Dipl.-Betriebswirtin (FH) Mareike Hornung für ihre Hilfe bei den englischen Übersetzungen. Ich möchte auch meiner langjährigen Freundin Dipl.-Math. Franziska Schäfer danken, für ihre Hilfe in mathematischen Fragen.

Besonderer Dank gilt weiterhin meinen Eltern, die mir das Studium ermöglichten und mich in allem unterstützten. Nicht zuletzt möchte ich mich bei Martin, meiner Tochter Mila und meinem Sohn Nils bedanken für ihre Geduld und ihr Verständnis.

Ilmenau, im Sommer 2016

Maria Trommer

Inhaltsverzeichnis

1	Einleitung.....	1
1.1	Umfang und Zielstellung.....	2
1.2	Aufbau der Arbeit	5
2	Einführung in die Schlafstadienklassifikation.....	7
3	Theoretische Grundlagen der Support-Vektor-Maschine	11
3.1	Einführung in die Theorie der Support-Vektor-Maschinen.....	11
3.1.1	Ablauf einer Klassifikation und benötigte Datensätze	11
3.1.2	Kreuzvalidierung	12
3.1.3	Verteilung der Belehrungsdaten	12
3.1.4	Skalierung der Eingangswerte.....	13
3.1.5	Das Verfahren der Support-Vektor-Maschine	14
3.1.6	Herleitung der Entscheidungsfunktion	16
3.1.7	Anwendung der Kernel-Funktion.....	20
3.1.8	Optimierungsprozess	21
3.1.9	Arten von SVM	21
3.1.10	Abwandlungen der SVM	25
3.1.11	Mehr-Klassen-Klassifikation mit SVM.....	27
3.1.12	Kombination verschiedener SVM.....	32
3.1.13	Bewertung des Klassifikationsergebnisses	33

3.2	Entwicklung und aktueller Stand der SVM-Forschung.....	36
4	Realisierung einer Support-Vektor-Maschine zur robusten nichtlinearen Klassifikation biologischer Daten.....	39
4.1	Einfluss der Belehrungsdatenverteilung auf die SVM.....	39
4.1.1	Wirkung der Gleichverteilung und der untypischen Daten auf das SVM-Modell.....	39
4.1.2	Identifikation von untypischen Daten.....	50
4.1.3	Wirkung der Homogenität von Daten.....	59
4.2	Untersuchung zur Skalierung von Daten.....	60
4.3	Untersuchung der Merkmalsvektoren	63
4.4	Belehrung der SVM.....	65
4.4.1	Multikriterielle Parameteroptimierung mittels evolutionärer Algorithmen	66
4.4.2	Empirische Untersuchung der Kernel-Funktion.....	87
4.5	Klassifikation mittels SVM.....	88
4.5.1	Untersuchung der Entscheidungsstrategie.....	88
4.5.2	Kombination verschiedener SVM mittels Bagging.....	91
4.5.3	Kontextbasierte Klassifikation von Hypnogrammen.....	93
5	Anwendung der Untersuchungsergebnisse auf die realen EEG-Datensätze....	97
6	Einbindung der Support-Vektor-Maschine auf ein ressourcenarmes eingebettetes System.....	107
6.1	Timing-Anforderungen	110
6.2	Speicheranforderungen.....	112
7	Zusammenfassung und Ausblick	115
7.1	Zusammenfassung	115
7.2	Ausblick.....	118
Anhang A	Schlafklassifikation mittels EEG-Daten	121
Anhang B	Algorithmen.....	129

Anhang C	Algorithmus Parametrierungen	133
Literaturverzeichnis		141

Abkürzungsverzeichnis

AASM	American Academy of Sleep Medicine
ABK	Abbruchkriterium
BCI	Brain Computer Interface
BDS	Belehrungsdatensatz
DG	Durchschnittliche Klassen-Klassifikationsgüte
DNS	Desoxyribonukleinsäure
DSVM	Deep Support Vector Machine
dt.	deutsch
EA	evolutionärer Algorithmus
EEG	Elektroenzephalogramm
EMG	Elektromyogramm
EMI	Elektromagnetische Interferenz
engl.	englisch
EOG	Elektrookulogramm
FAT	File Allocation Table
GA	genetischer Algorithmus
GG	Gesamt-Klassifikationsgüte
GMG	Geometrisch-Mittlere Klassen-Klassifikationsgüte
GV	Gleichverteilung
Hz	Hertz
ICSD2	International Classification of Sleep Disorders, second edition
KB	Kilobyte
KKT	Karush-Kuhn-Tucker
Lsg.	Lösung
LS-SVM	Least Squares Support Vector Machine
MG	Minimale Klassen-Klassifikationsgüte
MHE	Mehrheitsentscheidung
MHz	Megahertz

Mio.	Million
MOEA	Multi-objective Evolutionary Algorithm
MOV	Movement
MW	Mittelwert
NB	Nebenbedingung
n. b.	nicht belehrt
NN	nächste Nachbarn
NSGA	Non-dominated Sorting Genetic Algorithm
PC	Personal Computer
PDS	Probandendatensatz
PM	Punktmenge
PSVM	Proximal Support Vector Machine
REM	Rapid Eye Movement
RS	Rastersuche
RTOS	Real Time Operating System
SD-Karte	Secure Digital (sicher digitale)-Speicherkarte
SIGA	Special Immigrative multi-criteria Genetic Algorithm
SMO	Sequential Minimal Optimization
SMOTE	Synthetic Minority Over-sampling Technique
SN	Signifikanzniveau
SQA	Semiquartilsabstand
SRAM	Static random-access memory
SV	Support-Vektor
SVC	Support Vector Classification
SVM	Support-Vektor-Maschine
SVR	Support-Vektor-Regression
TE	Trennebene
TSVM	Transductive Support Vector Machine
UI	unidentified (dt. unidentifiziert)
VDS	Validierungsdatensatz
XOR	Exklusiv-ODER-Verknüpfung

Begriffsverzeichnis

äquivalente Lösungen	Als äquivalente Lösungen werden die gefundene Parameterkombinationen für SVM durch Rastersuche oder genetische Algorithmen bezeichnet, die bei der Belehrung zu dem gleichen Belehrungsgrad führen.
Artefakt	Ein Artefakt ist ein durch die Eigenschaften der Methode hervorgerufenenes unechtes Ergebnis (hier: technische oder biologische Störung eines Signals).
Ausreißer	Ein Datum gilt als Ausreißer, wenn dieses nicht den Erwartungen entspricht und von den übrigen Daten abweicht. Dabei kann es sich um ein fehlerhaftes Datum handeln, welches durch eine Störung oder Messfehler entstanden ist.
Belehrungsdaten	Die Belehrungsdaten dienen der Belehrung des Klassifikators. Auf Grundlage dieser Daten wird der Klassifikator erzeugt. Sie sollten möglichst alle vorkommenden Beispiele des Klassifikationsproblems enthalten.
Belehrungsgrad	Der Belehrungsgrad ist die prozentuale Rate der richtig klassifizierten Daten der Belehrungsdatenmenge. Dieser entsteht häufig durch einfache Kreuzvalidation.
Diversität	Die Diversität bedeutet im Zusammenhang mit den multi-kriteriellen evolutionären Algorithmen die Erhöhung der Breite der pareto-optimalen Ergebnisse, z. B. durch die Nischentechnik der Crowding-Distanz.
Dominanz	Individuum I_2 dominiert Individuum I_1 ($I_1 \preceq I_2$), wenn mehr Gütekriterien von I_2 besser sind als von I_1 .
Echtzeitfähigkeit	„Unter Echtzeit versteht man den Betrieb eines Rechensystems, bei dem Programme zur Verarbeitung anfallender Daten ständig betriebsbereit sind, derart, dass die Verarbeitungsergebnisse innerhalb einer vorgegebenen Zeitspanne verfügbar sind. Die Daten können je nach Anwendungsfall nach einer zeitlich zufälligen Verteilung oder zu vorherbestimmten Zeitpunkten anfallen.“ (nach DIN 44300) [Sch05]
Epoche	Eine Epoche beschreibt den Zeitabschnitt des EEG-Signals von 10 oder 30 Sekunden, welcher mit einer Klasse bewertet wurde.
Fitness	Als Fitness eines Individuums wird der aggregierte Gütewert eines Individuums aus allen Zielfunktionen bezeichnet. [Wal12]

Front	Eine Front im Zusammenhang mit den multikriteriellen evolutionären Algorithmen ist die Gruppierung aller pareto-optimalen Lösungen.
Generalisierungsfähigkeit	Die Generalisierungsfähigkeit ist die Fähigkeit nicht belehrte Beispiele richtig zu klassifizieren.
Generation	Eine Generation beschreibt die Gesamtheit aller Individuen einer Population zu einem bestimmten Zeitpunkt innerhalb des multikriteriellen evolutionären Algorithmus. [Wal12]
Güte eines Klassifikators	Die Güte eines Klassifikators besteht aus mehreren Teilaspekten, welche durch die angewendeten Gütewerte erfasst werden. Sie wird zum einen mit der relativen Häufigkeit der richtig klassifizierten Validierungsdaten aller Klassen assoziiert, welche durch die Gesamt-Klassifikationsgüte GG repräsentiert wird. Zum anderen beinhaltet die Güte die gleichmäßige Klassifikationsleistung aller Klassen, repräsentiert durch die Durchschnittliche Klassen-Klassifikationsgüte DG. Weiterhin zeigen die relativen Häufigkeiten der richtig klassifizierten Validierungsdaten der schlecht erkannten Klassen (meist die in der Belehrungsmenge schwach vertretenen Klassen) die Güte eines Klassifikators. Dies werden repräsentiert durch die Geometrisch-Mittlere Klassen-Klassifikationsgüte GMG und die Minimale Klassen-Klassifikationsgüte MG. Ein Klassifikator mit hoher Güte erfordert je nach Anwendungsfall eine Reduzierung von bestimmten Vertauschungen. Dieser Aspekt wird durch den gewichteten Kappa-Koeffizient κ_w repräsentiert. Ein weiterer Teilaspekt der Güte, welcher nicht durch einen berechenbaren Gütewert bestimmt werden kann, ist die Generalisierungsfähigkeit, d. h. die Klassifikationsleistung anhand nicht belehrter Daten. Die Beurteilung der Generalisierungsfähigkeit wird durch die Anwendung des Klassifikators auf von der Belehrungsdatenmenge unabhängige Daten gewährleistet.
Hypnogramm	Ein Hypnogramm ist ein Diagramm der Schlafstadien über eine Nacht. Es wird auch Schlafprofil genannt.
Immigration	Die genetische Operation der Immigration beschreibt die zufällige Erzeugung von neuen Individuen in einer Generation.
Immigrationsrate	Die Immigrationsrate stellt die prozentuale Rate der immigrierenden Individuen dar.
Indifferenz	Individuum I_2 ist indifferent zu Individuum I_1 ($I_1 \sim I_2$), wenn die gleiche Anzahl an Gütekriterien von I_2 und I_1 besser bzw. schlechter sind.

Individuum	Ein Individuum im Zusammenhang mit den multikriteriellen evolutionären Algorithmen ist ein Lösungsbeispiel des Optimierungsproblems. Es besteht aus Parameterkombination und der zugehörigen Fitness.
invalide Merkmale	Merkmale sind invalid, wenn diese unabhängig von der Klasse gleiche oder ähnliche Werte zeigen.
Kernel	Ein Kernel stellt eine Funktion im niederdimensionalen Raum dar, die sich ähnlich wie das Skalarprodukt im höherdimensionalen Raum verhält. So muss der höherdimensionale Raum nicht bekannt sein. Eine Kernel-Funktion muss das Theorem von Mercer erfüllen. [Fis07][SS01]
Klasse	Die Klasse y_i ist ein numerischer Wert und beschreibt die Zugehörigkeit eines Datums i zu einer bestimmten Menge von Daten mit gleichen oder ähnlichen Eigenschaften. [Wen05]
Konvergenz	Die Konvergenz bedeutet im Zusammenhang mit den multikriteriellen evolutionären Algorithmen die Erhöhung der Güte der resultierenden Lösungen, z. B. durch Eliteprinzip, bei dem dominante Individuen überleben.
Kreuzung (crossover)	Die genetische Operation der Kreuzung beschreibt die Vermischung und Vereinigung von Eigenschaften verschiedener Individuen, wodurch wiederum neue Individuen entstehen. Dies wird auch Rekombination genannt. [Wal12]
Majoritätsklasse	Die Majoritätsklasse ist die meistvertretene oder eine häufig vertretene Klasse.
Minoritätsklasse	Die Minoritätsklasse ist eine schwach vertretene Klasse.
Mutation	Die genetische Operation der Mutation beschreibt die zufällige Abänderung von Eigenschaften eines Individuums und dient der intra-individuellen Variation der Population. [Wal12]
Mutationsrate	Die Mutationsrate stellt die prozentuale Rate von mutierenden Individuen dar.
Mutationsweite	Die Mutationsweite bestimmt den Wertebereich, in dem die Individuen mutieren.
pareto-optimal	Lösungen sind pareto-optimal, wenn diese unter Einfluss der Kriterien indifferente Ergebnisse darstellen.
Population	Eine Population im Zusammenhang mit den multikriteriellen evolutionären Algorithmen ist eine bestimmte Anzahl an Individuen in einer Generation. [Wal12]
Rekombination	siehe Kreuzung

robuster Klassifikator	Ein Klassifikator gilt als robust, wenn dieser für eine Vielzahl von unterschiedlichen Probanden und ungeachtet der Unsicherheiten der Experten den gestellten Güteanforderungen genügt. [Wen05]
Selektion	Die genetische Operation der Selektion beschreibt die Auswahl von Individuen für die nächste Generation anhand deren Fitness. [Wal12]
Signifikanz	Der Vergleich zweier Messreihen weist einen signifikanten Unterschied auf, wenn der aus dem zweiseitigen Wilcoxon-Vorzeichen-Rang-Test berechnete Signifikanzwert kleiner dem Signifikanzniveau α von 0,05 ist. Es treten hierbei zwei Faktoren auf, die einen bedeutenden Unterschied der verglichenen Daten für eine Signifikanz verlangen. Zum einen wird durch den verwendeten Test, den Wilcoxon-Vorzeichen-Rang-Test, die Verteilungsfreiheit der Daten vorausgesetzt. Zum anderen liegen in dieser Arbeit nur kleine Messreihen vor (je größer die Messreihen, desto kleiner muss der Unterschied für eine Signifikanz sein).
Testdaten	Die Testdaten dienen dem Testen des Klassifikators. Durch die Erprobung des Klassifikators mittels der Testdaten kann der Belehrungsgrad bestimmt werden. Die Testdaten müssen nicht unabhängig von den Belehrungsdaten sein.
untypischer Datenpunkt	Ein Datenpunkt ist untypisch, wenn dieser aufgrund seines Merkmalsvektors abseits der Haupt-Datenmenge seiner Klasse liegt. Dieser beeinträchtigt eine automatische Mustererkennung nachteilig (Modellkomplexität steigt, Klassifikationsleistung sinkt). Jedoch stellt ein untypischer Datenpunkt kein fehlerhaftes Datum dar und ist damit vom Begriff des Ausreißers klar zu trennen.
valide Merkmale	Merkmale sind valid, wenn diese einen Bezug zur Klasse aufweisen.
Validierungsdaten	Die Validierungsdaten dienen der Validation des Klassifikators. Durch Erprobung des Klassifikators mittels dieser Validierungsdaten können verschiedene Gütewerte bestimmt werden und damit eine Beurteilung der Güte des Klassifikators vorgenommen werden. Hierfür müssen die Validierungsdaten unabhängig von den Belehrungsdaten sein, d. h. im Fall der Schlafklassifikation müssen die Daten von unterschiedlichen Probanden stammen.

Symbolverzeichnis

$\%$	Modulo, Prozent
\bullet	Skalarprodukt
\wedge	UND-Verknüpfung
\succ	Dominanz
\sim	Indifferenz
\forall	alle
Σ	Summation
Π	Produktion
α	Lagrange-Multiplikator α
a	Abstand
a_{\min}	minimaler Abstand
\tilde{a}	Abstandsmedian
$aSQA$	Abstands-Semiquartilsabstand
AG	Abstandsgrenze
$a(E,P)$	Abstand zwischen Ebene E und Punkt P
$arg\ max(z_i)$	Argument des Maximums Funktion
β	Lagrange-Multiplikator β
b	Verschiebung
bm_{ij}	Gewichte der Bewertungsmatrix (i entspricht der Soll-Klasse und j entspricht der Ist-Klasse)
BG	Belehrungsgrad
BM	Bewertungsmatrix
BP	Brennpunkt
γ	Weite der gaußschen Glockenkurve
c	Konstante
c_0	additive Konstante des Polynoms
C	Fehlergewicht
CD	Crowding-Distanz
$\Delta\omega$	Winkelabweichung

\tilde{d}	Dichtemedian
dSQA	Dichte-Semiquartilsabstand
D	Differenz
DG	Dichtegrenze
ε	Genauigkeitsparameter
e_i	Error-Variable
E	Ebene
EC	Anzahl der ausgleichbaren Fehler bei der Error-Correcting Output Coding-Methode
EL	Ellipse
EP	Eltern-Population
EW_{ij}	Entscheidungswert der Trennebenen-Gleichung zwischen Klasse i und Klasse j
$exp(x)$	Exponentialfunktion
f	Anzahl der Dimensionen
F_i	i-te Front
FA	Faktor Abstand
FD	Faktor Dichte
$f(\underline{x})$	Entscheidungsfunktion
η	Lagrange-Multiplikator η
g_i	absolute Häufigkeiten der insgesamt zu klassifizierenden Daten einer Klasse i
gen	Generationszähler
G	Güte eines Klassifikators
G_{gew}	gewichtete Güte
$g(l_1, l_2)$	zu optimierende Funktion
HD	Hamming-Distanz
$h(l_1, l_2)$	Nebenbedingung
i	Zählvariable i
I	Individuum
j	Zählvariable j
κ_w	gewichteter Kappa-Koeffizient
k	Anzahl der nächsten Nachbarn
km_{ij}	Elemente der Kernel-Matrix

KP	Eltern- und Nachkommen-Population
$K(\underline{x}_i, \underline{x}_j)$	Kernel-Funktion
l	allgemeine Variable
$L(\underline{w}, b, \alpha)$	Lagrange-Funktion
$L_D(\underline{w}, b, \alpha)$	duale Lagrange-Funktion
m	Anzahl der Kriterien
max	Maximum
min	Minimum
M	Margin, Spannweite
Mm	Merkmal
$m(\underline{w})$	Zielfunktion der SVM
ν	Steuerparameter der ν -SVC (siehe Abschnitt 3.1.9 Arten von SVM)
\underline{n}	Normalenvektor
$\ \underline{n}\ $	Betrag des Normalenvektors
N	natürliche Zahl, Anzahl der Daten, Anzahl der Parameter
NP	Nachkommen-Population
ξ, ξ^*	Schlupfvariable (slack variable)
O	Anzahl an Knoten im gerichteten Graph
p_{0w}	gewichtete relative Häufigkeiten der richtig klassifizierten Epochen
p_{ew}	gewichtete relative Häufigkeiten der zufällig richtig klassifizierten Epochen
pop	alte Population
pop_new	neue Population
p_{Sig}	Signifikanzwert zweier Messreihen aus dem zweiseitigen Wilcoxon-Vorzeichen-Rang-Test
P	Punkt P
PK	Anzahl der Parameterkombinationen
PM	Punktemenge
q	Grad des Polynoms, Exponent
ϱ	Margin-Parameter
\mathbb{R}	Merkmalsraum, Menge der reellen Zahlen
r_i	absolute Häufigkeiten der richtig klassifizierten Daten der Klasse i

\mathbb{R}	Index für den realen Wertebereich
R_{emp}	empirisches Risiko
σ	Standardabweichung
S	Index für den skalierten Wertebereich
SP	Schwerpunkt
SQA	Semiquartilsabstand
$\text{sign}(x)$	Signumfunktion, Sprungfunktion
t	Zeit
t-SV	abgewandelter Support Vektor für die Proximal-SVM (in [FM01] als ϵ -support vector bezeichnet)
TE	Trennebene
TE_1, TE_2	parallele Ebenen zu TE
$\tanh(x)$	Tangens-Hyperbolicus-Funktion
U	Anzahl der Klassen
v, σ^2	Varianz
v_{ij}	absolute Häufigkeiten der klassifizierten Epochen der Vertauschungsmatrix (i entspricht der Soll-Klasse und j entspricht der Ist-Klasse)
$v_{\text{rel},ij}$	relative Häufigkeiten der klassifizierten Epochen der Vertauschungsmatrix normiert auf die Gesamtanzahl der Daten (i entspricht der Soll-Klasse und j entspricht der Ist-Klasse)
\mathbf{V}	Vertauschungsmatrix
VR	Validationsrate
\underline{w}	Normalenvektor der Trennebene
wert	Wert
W	Anzahl der möglichen Parameterwerte
\underline{x}	Merkmalsvektor
\bar{x}	Arithmetischer Mittelwert der Werte x_i
\tilde{x}	Median der Werte x_i
X	Menge der Belehrungsdaten
y	Ausgang oder Klasse
\mathbb{Z}	Menge der ganzen Zahlen
z_i	klassenspezifischer Wert der Klasse i gebildet aus den Trennebenen-Entscheidungen
Z	Anzahl der Trennebenen

ZF

Zufallsfaktor

Abbildungsverzeichnis

Abbildung 1-1: Gesamtübersicht des SVM-Generierungs- und Klassifikationsprozesses.....	4
Abbildung 2-1: Einteilung und Auflistung der Arten von UI-Epochen [Wal12]	8
Abbildung 2-2: Vereinfachte Darstellung eines Hypnogramms.....	10
Abbildung 3-1: Prinzip der linearen Skalierung	13
Abbildung 3-2: Prinzipielle Funktionsweise der SVM [Suy02].....	14
Abbildung 3-3: Prinzip der linearen Trennung von nichtlinearen Daten durch Erhöhung der Dimension	15
Abbildung 3-4: Bestimmung der maximalen Spannweite M	17
Abbildung 3-5: Übersicht der Arten von SVM.....	21
Abbildung 3-6: Bestimmung der Trennebenen bei nichtlinear trennbaren Daten mittels C-SVC [Fis07].....	22
Abbildung 3-7: Bestimmung der Trennebenen bei nichtlinear trennbaren Daten mittels Support-Vektor-Regression [Wan05]	24
Abbildung 3-8: Verlauf der ϵ -insensitivity Verlust-Funktion [SS01]	24
Abbildung 3-9: Funktionsweise der Proximal-SVM [FM01]	26
Abbildung 3-10: Auflistung der Methoden zur Mehr-Klassen-Klassifikation mit SVM	28
Abbildung 3-11: One-against-All-Methode [Mol03b].....	28
Abbildung 3-12: One-against-One-Methode [Mol03b].....	29
Abbildung 3-13: Directed Acyclic Graph-Methode [Mol03b]	30
Abbildung 4-1: „Soft-Margin“ (schwarz) und ursprüngliche SVM (grau) eines Zwei-Klassen-Problems von linear trennbaren Daten [AKJ04].....	40
Abbildung 4-2: Synthetisches Drei-Klassen-Problem	41
Abbildung 4-3: Anwendung des SMOTE-Algorithmus.....	42
Abbildung 4-4: Belehrungsdaten ohne untypische Daten und Trennebenen der SVM bei $C=32$ (links) und $C=16384$ (rechts).....	43
Abbildung 4-5: Belehrungsdaten ohne untypische Daten und Trennebenen der SVM bei $C=32$ und einer GV durch Duplikation (links) und SMOTE (rechts)	45
Abbildung 4-6: Synthetisches Drei-Klassen-Problem mit untypischen Daten	46
Abbildung 4-7: Belehrungsdaten mit untypischen Daten und Trennebenen der SVM bei $C=32$ (links) und $C=16384$ (rechts).....	47

Abbildung 4-8: Belehrungsdaten mit untypischen Daten und Trennebenen der SVM bei $C=32$ und einer GV durch Duplikation (links) und SMOTE (rechts)	48
Abbildung 4-9: Belehrungsdaten mit untypischen Daten und Trennebenen der SVM bei $C=16384$ und einer GV durch Duplikation (links) und SMOTE (rechts)	49
Abbildung 4-10: Programmablaufdiagramm der DiLa-Filterung	52
Abbildung 4-11: Synthetisches Drei-Klassen-Problem nach Löschung untypischer Daten durch den DiLa-Filter	53
Abbildung 4-12: Belehrungsdaten mit gelöschten untypischen Daten ($FD=0,8 FA=0,8 k=5$) nach SMOTE-GV und Trennebenen der SVM bei unterschiedlichen Zeitpunkten der Parameterbestimmung.....	55
Abbildung 4-13: Kastendiagramm der vier Gütewerte GG, DG, GMG und MG bei verschiedenen Datenbearbeitungsmethoden und Vorgehensweisen.....	57
Abbildung 4-14: Kastendiagramm der gewichteten Güte G_{gew} bei verschiedenen Datenbearbeitungsmethoden und Vorgehensweisen	57
Abbildung 4-15: Kastendiagramm zur Untersuchung der Datenhomogenität.....	59
Abbildung 4-16: Kastendiagramm zur Untersuchung der Skalierungsstrategie.....	61
Abbildung 4-17: Kastendiagramm zur Untersuchung der Skalierungsmethode.....	62
Abbildung 4-18: Kastendiagramm zur Untersuchung der Merkmalsvektoren	64
Abbildung 4-19: Diagramm für die Anzahl der Parameterkombinationen in Abhängigkeit der Auflösung für die Rastersuche.....	67
Abbildung 4-20: Diagramm für die Anzahl der Parameterkombinationen in Abhängigkeit des Suchbereichs für die Rastersuche	67
Abbildung 4-21: Darstellung zur Berechnung der Crowding-Distanz [DPAM00]	70
Abbildung 4-22: Programmablaufdiagramm des NSGA-II von [DPAM00]	72
Abbildung 4-23: Programmablaufdiagramm des SIGA	73
Abbildung 4-24: Klassisches Zwei-Klassen-XOR-Problem im zweidimensionalen Raum	77
Abbildung 4-25: Streudiagramm des Belehrungs- und Validierungsdatensatzes des klassischen Zwei-Klassen-XOR-Problems.....	77
Abbildung 4-26: Kastendiagramm für das klassische Zwei-Klassen-XOR-Problem des Vergleichs der Rastersuche mit den genetischen Algorithmen SIGA und NSGA-II	78
Abbildung 4-27: Acht-Klassen-XOR-Problem im zweidimensionalen Raum.....	79
Abbildung 4-28: Kastendiagramm für das Acht-Klassen-XOR-Problem des Vergleichs der Rastersuche mit den genetischen Algorithmen SIGA und NSGA-II.....	81
Abbildung 4-29: Oberflächendiagramm des Belehrungsgrades für die Belehrung mit realen Daten	82

Abbildung 4-30: Resultierende Belehrungsgrade für die Belehrung mit realen Daten durch die Rastersuche	83
Abbildung 4-31: Kastendiagramm des Vergleichs 1 der Rastersuche (RS) mit den genetischen Algorithmen SIGA und NSGA-II für die realen Daten.....	84
Abbildung 4-32: Kastendiagramm des Vergleichs 2 der Rastersuche (RS) mit den genetischen Algorithmen SIGA und NSGA-II für die realen Daten.....	84
Abbildung 4-33: Kastendiagramm des Vergleichs 3 der Rastersuche (RS) mit den genetischen Algorithmen SIGA und NSGA-II für die realen Daten.....	84
Abbildung 4-34: Kastendiagramme zur Untersuchung der Kernel-Funktion.....	87
Abbildung 4-35: Kastendiagramm zur Untersuchung der Entscheidungsstrategie.....	90
Abbildung 4-36: Kastendiagramme zur Untersuchung der Kombination mehrerer SVM.....	92
Abbildung 4-37: Kastendiagramme zur Untersuchung der kontextbasierten Klassifikation.....	95
Abbildung 5-1: Darstellung der angewendeten Methoden der Gesamtuntersuchung.....	98
Abbildung 5-2: Kastendiagramm zur Gesamtuntersuchung: die gewichtete Güte G_{gew}	99
Abbildung 5-3: Kastendiagramme der Gesamtuntersuchung: die Gütewerte GG, DG, GMG und MG.....	101
Abbildung 5-4: Klassifiziertes Hypnogramm mittels ursprünglicher Modellbildung (blau) und Expertenhypnogramm (orange) von Proband 351001	103
Abbildung 5-5: Klassifiziertes Hypnogramm mittels optimierter Modellbildung (blau) und Expertenhypnogramm (orange) von Proband 351001	103
Abbildung 5-6: Kastendiagramme des Vergleichs der neuronalen Netze aus [Wen05] mit der SVM-Klassifikation	105
Abbildung 6-1: Fotografie des Stellaris® LM3S3748 Evaluation Board [Tex10]	108
Abbildung 6-2: Funktion und Struktur des mit FreeRTOS eingebundenen Software Frameworks auf dem eingebetteten System [TMBST ⁺ 11]	109
Abbildung 6-3: Diagramm und Wertetabelle der durchschnittlichen Abarbeitungszeiten der SVM-Klassifikation auf dem LM3S3748 in Abhängigkeit der Klassenanzahl	110
Abbildung 6-4: Diagramm und Wertetabelle der durchschnittlichen Abarbeitungszeiten einer Acht-Klassen-Klassifikation auf dem LM3S3748 in Abhängigkeit der Anzahl der mit einmal ausgelesenen Bytes.....	111
Abbildung 6-5: Diagramm und Wertetabelle der durchschnittlichen Abarbeitungszeiten einer Acht-Klassen-Klassifikation auf dem LM3S3748 in Abhängigkeit der Merkmalsanzahl.....	112
Abbildung 6-6: Diagramm und Wertetabelle des SRAM-Bedarfs des SVM-Algorithmus auf dem LM3S3748 in Abhängigkeit der Klassenanzahl.....	112

Abbildung 6-7: Diagramm und Wertetabelle des SRAM-Bedarfs der Acht-Klassen-Klassifikation auf dem LM3S3748 in Abhängigkeit der Merkmalsanzahl	113
Abbildung A-1: EEG-Elektrodenpositionierung am menschlichen Kopf nach dem 10-20 System [Wen05]	123
Abbildung A-2: Signalverlauf des EEG-Signals und des zugehörigen Amplitudenspektrums im Wachzustand	124
Abbildung A-3: Signalverlauf des EEG-Signals und des zugehörigen Amplitudenspektrums im Zustand S2.....	125
Abbildung A-4: Signalverlauf des EEG-Signals und des zugehörigen Amplitudenspektrums im Zustand S4.....	125
Abbildung A-5: Signalverlauf des EEG-Signals und des zugehörigen Amplitudenspektrums im Zustand REM.....	126
Abbildung B-1: Programmablaufdiagramm der einfachen Kreuzvalidierung.....	129
Abbildung B-2: Programmablaufdiagramm des angepassten SMOTE-Algorithmus	130
Abbildung B-3: Programmablaufdiagramm des Datenwahlalgorithmus	131

Tabellenverzeichnis

Tabelle 2-1: Acht-Klassen-Problem der Schlafklassifikation.....	8
Tabelle 2-2: Sieben-Klassen-Problem der Schlafklassifikation.....	9
Tabelle 3-1: Auflistung möglicher Kernel zur Klassifikation [CL08]	16
Tabelle 3-2: Vertauschungsmatrix.....	33
Tabelle 4-1: Kennwerte des SVM-Klassifikators ohne untypische Daten bei $C=32$ und $C=16384$	43
Tabelle 4-2: Kennwerte des SVM-Klassifikators ohne untypische Daten bei $C=32$ und einer GV durch Duplikation und SMOTE.....	45
Tabelle 4-3: Kennwerte der SVM mit untypischen Daten bei $C=32$ und $C=16384$	47
Tabelle 4-4: Kennwerte der SVM mit untypischen Daten bei $C=32$ und einer GV durch Duplikation und SMOTE.....	48
Tabelle 4-5: Kennwerte der SVM mit untypischen Daten bei $C=16384$ und einer GV durch Duplikation und SMOTE.....	49
Tabelle 4-6: Kennwerte der SVM mit gelöschten untypischen Daten ($FD=0,8$ $FA=0,8$ $k=5$) nach SMOTE-GV bei unterschiedlichen Zeitpunkten der Parameterbestimmung ..	55
Tabelle 4-7: Signifikanzwerte für den Vergleich verschiedener Datenbearbeitungsmethoden und Vorgehensweisen	57
Tabelle 4-8: Signifikanzwerte für den Vergleich homogener Belehrungsdaten (1PDS) zu inhomogenen Belehrungsdaten	59
Tabelle 4-9: Signifikanzwert für den Vergleich der Skalierungsstrategien	61
Tabelle 4-10: Signifikanzwerte für den Vergleich der Skalierungsmethoden im Bezug auf die Min-Max-Methode.....	62
Tabelle 4-11: Signifikanzwerte zur Untersuchung der Merkmalsvektoren im Bezug auf 62 Merkmale.....	64
Tabelle 4-12: Signifikanzwert für den Vergleich SIGA gegen NSGA-II für das Zwei-Klassen-XOR-Problem	78
Tabelle 4-13: Signifikanzwert für den Vergleich SIGA gegen NSGA-II für das Acht-Klassen-XOR-Problem	81
Tabelle 4-14: Signifikanzwerte für den Vergleich SIGA gegen NSGA-II für reale Daten	85
Tabelle 4-15: Signifikanzwerte für den Vergleich der Kernel-Funktionen.....	87
Tabelle 4-16: Signifikanzwerte für den Vergleich der Entscheidungsstrategien	90
Tabelle 4-17: Prozentuale Auswertung der Klassenübergänge im gesamten Datensatz B1	94

Tabelle 4-18: Als „unerlaubt“ definierte Übergänge für die Schlafklassifikation.....	94
Tabelle 4-19: Signifikanzwerte für die kontextbasierte Klassifikation.....	95
Tabelle 5-1: Signifikanzwerte der Gesamtuntersuchung in Bezug auf die ursprüngliche Modellbildung für die gewichtete Güte G_{gew}	99
Tabelle 5-2: Signifikanzwerte der Gesamtuntersuchung in Bezug auf den vorhergehenden Schritt für die gewichtete Güte G_{gew}	99
Tabelle 5-3: Signifikanzwerte der Gesamtuntersuchung in Bezug auf die ursprüngliche Modellbildung für die Gütewerte GG, DG, GMG und MG.....	102
Tabelle 5-4: Signifikanzwerte der Gesamtuntersuchung in Bezug auf den vorhergehenden Schritt für die Gütewerte GG, DG, GMG und MG.....	102
Tabelle 5-5: Signifikanzwerte des Vergleichs der neuronalen Netze aus [Wen05] und SVM.....	105
Tabelle 6-1: Hardware-Restriktionen des LM3S3748 Mikrocontrollers [Tex11].....	107
Tabelle A-1: Auflistung der Epochen vom Datensatz B1.....	121
Tabelle A-2: Auflistung der Epochen vom Datensatz B2.....	122
Tabelle A-3: Auflistung der Epochen vom Datensatz B3.....	122
Tabelle A-4: Auflistung der verwendeten Merkmale zur Schlafklassifikation [Wen05].....	127
Tabelle C-1: Eingestellte Parameter zur Untersuchung der Datenbearbeitung	133
Tabelle C-2: Eingestellte Parameter zur Untersuchung der Datenhomogenität.....	133
Tabelle C-3: Eingestellte Parameter zur Untersuchung der Skalierungsstrategie	134
Tabelle C-4: Eingestellte Parameter zur Untersuchung der Skalierungsmethode	134
Tabelle C-5: Eingestellte Parameter zur Untersuchung des Merkmalsvektors	134
Tabelle C-6: Eingestellte Parameter der Rastersuche für die Vergleichsstudie zu den genetischen Algorithmen	134
Tabelle C-7: Resultierende SVM-Parameter aus der Rastersuche für das klassische Zwei- Klassen-XOR-Problem	135
Tabelle C-8: Eingestellte Parameter der genetischen Algorithmen (SIGA/NSGA-II) bei Vergleich 1.....	135
Tabelle C-9: Eingestellte Parameter der genetischen Algorithmen (SIGA/NSGA-II) bei Vergleich 2.....	136
Tabelle C-10: Eingestellte Parameter der genetischen Algorithmen (SIGA/NSGA-II) bei Vergleich 3.....	136
Tabelle C-11: Eingestellte Parameter zur Untersuchung der Kernel-Funktion.....	136
Tabelle C-12: Eingestellte Parameter zu den Untersuchungen der Entscheidungsstrategie, der Kombination mehrerer SVM und der kontextbasierten Klassifikation.....	137

Tabelle C-13: Eingestellte Parameter zur Gesamtuntersuchung für die ursprüngliche Modellbildung	137
Tabelle C-14: Eingestellte Parameter zur Gesamtuntersuchung Schritt 1	137
Tabelle C-15: Eingestellte Parameter zur Gesamtuntersuchung Schritt 2	138
Tabelle C-16: Eingestellte Parameter zur Gesamtuntersuchung Schritt 3	138
Tabelle C-17: Eingestellte Parameter zur Gesamtuntersuchung Schritt 4	139
Tabelle C-18: Eingestellte Parameter zur Gesamtuntersuchung Schritt 5	139

1 Einleitung

Durch die zunehmende Digitalisierung der Welt liegen immer mehr Daten vor, die von immer leistungsfähigeren Computern ausgewertet werden. Eine Auswertung durch den Menschen ist aufgrund der Datenmenge in vielen Fällen nicht mehr möglich. So werden z. B. Suchanfragen, Kaufverhalten und Interessen von Internetbenutzern für Marktanalysen oder gezielte Werbung erfasst und analysiert. Auch in der Industrie werden immer mehr Vorgänge von Maschinen übernommen und von rechenstarken Computern gesteuert, sei es in der Automobilindustrie, Medizintechnik, Metallverarbeitungsindustrie oder in der Elektronikindustrie. Die Überwachung von Maschinen, Anlagen und Prozessen wird zunehmend automatisiert (Condition Monitoring). Unter dem Begriff Industrie 4.0 sollen Fertigungsprozesse flexibler gemacht werden und somit „individualisierte Massenprodukte“ produzierbar sein [Sch13]. Die Fertigungstechnik von morgen soll flexibler und leichter zu bedienen sein. Bei all diesen Entwicklungen nehmen selbstlernende Verfahren eine wichtige Rolle ein.

Dennoch gibt es auch heute noch Aufgaben, speziell in der Medizin, die vom Menschen ausgeführt werden müssen. Gerade wenn es sich um Aufgaben handelt, bei denen Erfahrung und Intuition eine wichtige Rolle spielen. Der Grund dafür ist, dass ein Computer zwar bei der sequenziellen Abarbeitung von komplexen Rechenoperationen dem Menschen weit überlegen ist, aber bei kognitiven Problemen, dem Erkennen von Mustern beispielsweise, versagt. Das wohl bekannteste Beispiel der Mustererkennung ist die Klassifikation von handschriftlichen Zeichen. Zur Mustererkennung mittels intelligenter Algorithmen wird unter anderem das sogenannte „überwachte Lernen“ (engl. supervised learning) eingesetzt, d. h. das Lernen mit Lehrer. Dem System wird ein gewünschtes Verhalten durch einen Experten oder definierte Gesetze beigebracht.

Für einen erfolgreichen überwacht lernenden Klassifikationsalgorithmus sind einige Anforderungen zu erfüllen. Zum einen braucht man genügend unabhängige Daten zum Belehren und Validieren des Algorithmus. Diese Daten beinhalten Merkmale und Klasse¹ eines zugehörigen Beispiels. Gerade für die Belehrung ist es wichtig, möglichst viele eventuell vorkommende Beispiele erfasst zu haben, was aber oftmals sehr schwierig ist. Zum anderen müssen Merkmale zu

¹ siehe Begriffsverzeichnis

diesen Beispielen extrahiert werden (engl. Feature Engineering). Das Finden der treffenden Merkmale kann je nach Klassifikationsaufgabe allerdings problematisch sein. Weiterhin müssen die Daten durch einen Experten, in vielen Fällen dem Menschen, mit einer Klasse bewertet werden. Die Existenz eines Experten für das jeweilige Klassifikationsproblem ist dabei Voraussetzung. Der Experte selbst stellt eine weitere Fehlerquelle dar. Durch vages und teilweise widersprüchliches Expertenwissen entstehen fehlerhafte Bewertungen, die an den Klassifikator zu einem gewissen Grad weitergegeben werden. Es gilt, dass der Klassifikator maximal nur so gut sein kann wie die behelrenden Experten selbst. Anschließend soll der Algorithmus mit dem Belehrungsdatensatz das Muster erlernen. Daraufhin wird mit dem Validierungsdatensatz die Güte² des Klassifikators bestimmt.

Auch wenn alle möglichen Fehlerquellen, wie Signalstörungen, ungeeignete Merkmale oder abweichende Expertenbewertungen bei der Belehrung des Klassifikators vermieden werden, sind Fehlentscheidungen abhängig vom Klassifikationsproblem möglich. Um die Wahrscheinlichkeit für Fehlentscheidungen zu minimieren, forschen seit Jahren Wissenschaftler auf der ganzen Welt nach Methoden und Ansätzen zur Verbesserung intelligenter Algorithmen.

1.1 Umfang und Zielstellung

In dieser Arbeit soll die automatische Klassifikation von biologischen Daten mit SVM, speziell der Schlafstadienklassifikation anhand von Elektroenzephalogramm (EEG)-Daten, untersucht werden. Dieses Problem stellt eine sehr anspruchsvolle Aufgabe dar, da hierbei gleich mehrere Faktoren auftreten, die eine Klassifikation erschweren. Zum einen weisen biologische Signale, insbesondere das EEG, eine starke individuelle Ausprägung in Frequenz, Amplitude und Morphologie einzelner relevanter Merkmale auf, sowohl inter- als auch intraindividuell [ED11]. So ist das EEG nicht nur von Faktoren wie dem Alter abhängig, sondern wird auch durch Medikamenteneinnahme verändert. Damit ergeben sich für die Schlafklassifikation äußerst inhomogene Datensätze. Durch die kleinen Amplituden des EEG-Signals sind diese besonders stör anfällig. Durch die hohe Individualität können, außer den mit Störungen behafteten Daten, auch „untypische“ Daten³ in den Datensätzen auftreten. Diese Daten sind zwar nicht fehlerhaft, behindern jedoch die Bildung eines generalisierenden Klassifikators. Weiterhin sind die Klassen naturbedingt stark ungleichverteilt. Dazu kommt noch, dass die Expertenbewertungen abweichen können [Wen05]. Dies hat verschiedene Ursachen. Das EEG ist ein sehr komplexes Signal, dementsprechend sind auch die Bewertungsregeln für Schlaf komplex und nicht eindeutig anwendbar. Die Erfahrungen des Experten werden immer in die Bewertung mit einbezogen. Zudem ist der Schlaf ein fließender Prozess, d. h. es treten keine harten Stadienübergänge auf. So können vor allem in diesen Übergangsbereichen zwischen den Schlafstadien vermehrt ambivalente Klassifikationen auftreten. In [Wen05, S. 105] wurde bereits aus verschiedenen Studien Ergebnisse zur Expertenübereinstimmung zusammengetragen. Diese Expertenübereinstimmung liegt, abhängig vom Schlafstadium, zwischen 20 % und 90 %. Jeder dieser Faktoren zeigt die Komplexität der automatischen Schlafklassifikation auf.

² siehe Begriffsverzeichnis

³ siehe Begriffsverzeichnis

Die Support-Vektor-Maschine (SVM) ist eine anerkannte und oft empfohlene Klassifikationsmethode für verschiedene Aufgabenstellungen. So schreibt Bernhard Schölkopf in [SS01]: “At the same time, successful applications have demonstrated that SVMs not only have a more solid foundation than artificial neural networks, but are able to serve as a replacement for neural networks that perform as well or better, in a wide variety of fields.” Aus diesem Grund soll im Rahmen dieser Arbeit die Robustheit⁴ der Methode SVM für die vorliegende schwierige Klassifikationsaufgabe validiert werden. Diese ist gekennzeichnet durch:

- stark inhomogene Daten durch inter- und intraindividuelle Ausprägungen,
- ein störanfälliges Signal durch kleine Signalamplituden,
- eine starke Ungleichverteilung der Klassen,
- das Beinhalten ungenauer Daten und
- abweichende Expertenbewertungen.

Zumal genau diese Aufgabe mit identischen Daten bereits in [Wen05] mittels einem vierstufigen Prozess optimierter Populationen von künstlichen neuronalen Netzen untersucht wurde und somit als Vergleich dienen kann. Die SVM unterscheidet sich grundlegend von neuronalen Netzen, da diese Methode nur eingeschränkte Möglichkeiten zur Modellkomplexitätssteuerung aufweist. Welche Auswirkungen dies auf die Klassifikation hat und ob die SVM wirklich so gute Klassifikationsergebnisse liefern kann, soll in Laufe der Arbeit gezeigt werden.

Für eine zufriedenstellende robuste Lösung dieses komplexen Klassifikationsproblems gibt es zahlreiche Anwendungsgebiete, die medizinisch und auch ökonomisch von Bedeutung sind, wie zum Beispiel die komplexe Steuerung von Maschinen (Brain Computer Interface – BCI), eine automatische Schlafstadienklassifikation oder Narkosetiefenbestimmung. Sind solche Anwendungen auch auf mobilen Geräten verfügbar, können erheblich Kosten gespart werden. Um den Anwendungen der Klassifikatoren auf mobilen Geräten Rechnung zu tragen, soll der SVM-Klassifikationsalgorithmus im Rahmen dieser Arbeit echtzeitfähig in ein ressourcenarmes, eingebettetes System eingebunden werden.

In Abbildung 1-1 ist die Gesamtübersicht über den SVM-Klassifikationsprozess dargestellt. Jedes Feld beschreibt einen Teilprozess, wobei die schwarz geschriebenen Prozesse in dieser Arbeit untersucht werden. Die grau geschriebenen Prozesse werden aus vorherigen Arbeiten übernommen oder nicht näher untersucht. So werden, auch aus Gründen der Vergleichbarkeit, die EEG-Daten und deren Merkmalsextraktion sowie die Merkmalsselektion aus [Wen05] übernommen. Dennoch analysiert diese Arbeit den Einfluss der Merkmalsanzahl und den Einfluss von irrelevanten Merkmalen auf die Klassifikation mit SVM. Weiterhin sollen die Auswirkungen von ungleichverteilten Belehrungsdaten und untypischen Daten in der Belehrungsdatenmenge gezeigt werden. Zudem stellt die vorliegende Arbeit neue Methoden zur Bearbeitung der Belehrungsdaten, die der Vermeidung negativer Auswirkungen auf die Klassifikation dienen, vor und erprobt diese. So konnte im Rahmen dieser Arbeit der DiLa-Filter entwickelt werden, ein neuartiger Algorithmus zur Identifikation von untypischen Daten [TW16].

⁴ siehe Begriffsverzeichnis unter „robuster Klassifikator“

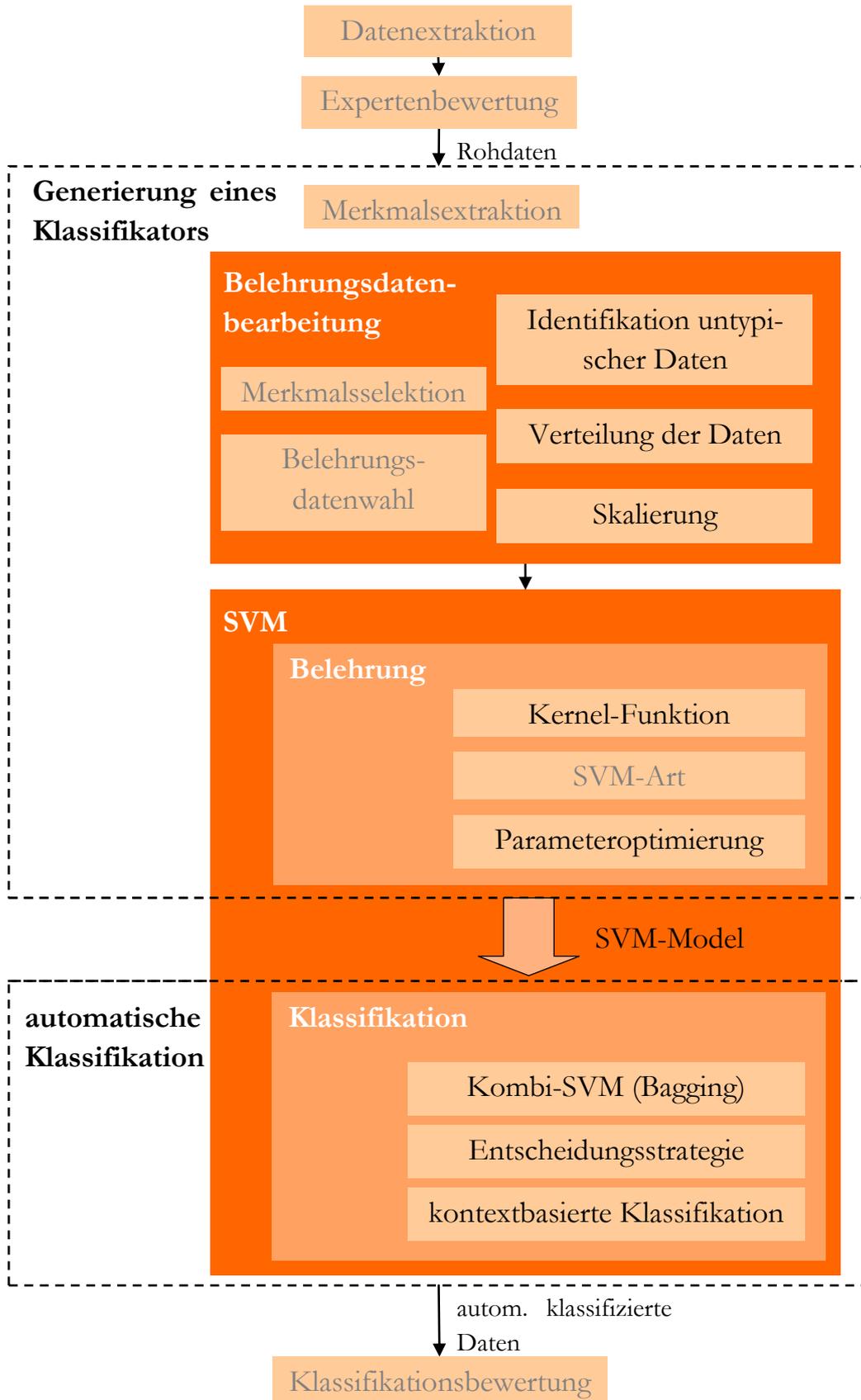


Abbildung 1-1: Gesamtübersicht des SVM-Generierungs- und Klassifikationsprozesses

Durch die aus den Untersuchungen gewonnen Erkenntnisse wird im Laufe dieser Arbeit ein Bedarf für einen Algorithmus zur Belehrungsdatenwahl zur gezielten Selektion von Belehrungsdaten für die SVM-Modellbildung deutlich. Untersuchungen zu verschiedenen Skalierungsvarianten und der Einfluss der Homogenität von Daten auf die Klassifikationsgüte vervollständigen das Thema der Belehrungsdatenbearbeitung. Bezüglich der Belehrung der SVM werden die Themen der Wahl der Kernel-Funktion und der Parameteroptimierung analysiert. Mit Hilfe von einem hierbei entwickelten neuen genetischen Algorithmus SIGA (Special Immigrative multi-criteria Genetic Algorithm), speziell für die SVM-Parameteroptimierung zur Schlafklassifikation entwickelt, wird erstmals die genetische Operation der Immigration eingesetzt. Damit konnte nicht nur zur verbesserten Konvergenz der genetischen Algorithmen beigetragen, sondern auch ein Beitrag für die Leistungssteigerung der SVM-Klassifikation geleistet werden.

Zudem werden erweiternde Klassifikationsmethoden untersucht. Für die Kombination mehrerer SVM-Modelle (Bagging) konnten spezielle Verknüpfungsvarianten für die Schlafklassifikation erprobt werden. Hierfür wurde eigens eine Ordinalskala der Schlafstadien festgelegt. Das Wissen um die speziellen Eigenschaften der Hypnogramme, den Schlafprofilen, kam auch bei der Umsetzung einer neuen kontextbasierten Klassifikation zum Einsatz. Weiterhin wurden vorhandene, aber auch erstmals in dieser Arbeit vorgestellte Verfahren zur Entscheidungsstrategie analysiert.

- Diese Arbeit untersucht die Methode der SVM eingehend und zeigt dabei ihre Stärken und Schwächen auf. Dabei wird an allen Stellen der Modellbildung, den Belehrungsdaten, der Belehrung und der SVM-Klassifikation eine Optimierung vorgenommen und durch experimentelle Untersuchungen ihre Wirkungsweise belegt. Diese Arbeit soll somit zum einen eine Vorgehensweise der SVM-Optimierung aufzeigen und zum anderen neuartige Methoden zur Optimierung vorstellen. Sie stellt damit einen Ansatz zur Erzeugung robuster Klassifikatoren mit Hilfe der SVM vor.

1.2 Aufbau der Arbeit

Im Folgenden führt Kapitel 2 in die gestellte Aufgabe der Schlafstadienklassifikation ein. Im Kapitel 3 wird die Methode der Support-Vektor-Maschine ausführlich theoretisch betrachtet. Weiterhin wird ein kurzer historischer Abriss über die Entwicklung der SVM-Methode gegeben sowie der aktuelle Stand dargelegt. Auf die theoretischen Grundlagen in Kapitel 3 aufbauend beschreibt Kapitel 4 erprobte Vorverarbeitungsmethoden und Modifikationen (siehe Abbildung 1-1), deren Auswirkungen auf die Klassifikationsgüte in Kapitel 5 dargestellt werden. In Kapitel 5 wird auch ein direkter Vergleich der erzielten Ergebnisse mit den Resultaten der neuronalen Netze vorgenommen.

Ein weiterer Teil der Arbeit beinhaltet die Einbindung des SVM-Algorithmus auf ein eingebettetes System. Nach einer kurzen Beschreibung der verwendeten Hardware, wird dessen Realisierung und die sich daraus ergebenden Restriktionen in Kapitel 6 erläutert. Am Ende der Arbeit soll Kapitel 7 eine Zusammenfassung der in der Arbeit behandelten Probleme und Lösungen sein und einen Ausblick auf künftige Problemstellungen geben.

2 Einführung in die Schlafstadienklassifikation

Die Klassifikationsaufgabe ist die automatische Klassifikation des menschlichen Schlafes anhand des einkanaligen EEG. Zur einfachen Anwendbarkeit des Klassifikators in einem mobilen Gerät soll das Signal durch nur einen Kanal aufgezeichnet werden, statt mit der üblichen aufwendigen, mehrkanaligen Ableitung nach dem 10-20-System (siehe Abbildung A-1 auf Seite 123). Da sich das einkanalige EEG optisch stark von dem üblichen mehrkanaligen EEG unterscheidet, wurden alle EEG-Signale parallel abgeleitet und die Bewertung mit Hilfe des gewohnten mehrkanaligen EEG, dem Elektrookulogramm (EOG) und dem Elektromyogramm (EMG) vorgenommen [RK68]. Dabei wird dem Zeitsignal typischerweise alle 10 oder 30 Sekunden eine Klasse zugeordnet. **Diese Zeitdauer wird im Folgenden als eine Epoche⁵ bezeichnet.**

Die aufgenommenen Daten sind in drei Datensätze unterteilt: B1 aus zehn gesunden Probanden vorwiegend zur Belehrung, B2 aus 16 schlafgestörten Probanden zur Validation des Klassifikators und B3 aus elf wiederum gesunden bzw. leicht schlafgestörten Probanden ebenfalls zur Validation.⁶

Der Schlaf des Menschen wird in sechs Schlafstadien eingeteilt [RK68]:

- Wach,
- S1, der Übergang zwischen wach und Schlaf, das Bewusstsein ist noch nicht komplett ausgeschaltet,
- S2, der leichte Schlaf, welcher den Hauptanteil des Schlafes einer Nacht ausmacht,
- S3, der Tiefschlaf mit nur wenigen Delta-Wellen,
- S4, der Tiefschlaf mit häufigen Delta-Wellen, indem der Körper des Schlafenden sehr entspannt ist; ein Aufwecken ist sehr schwierig und
- Rapid Eye Movement (REM), der sogenannte „Traumschlaf“, welcher durch schnelle Augenbewegungen und einen geringen Muskeltonus gekennzeichnet ist.

⁵ siehe Begriffsverzeichnis

⁶ Detaillierte Abgaben zu den verwendeten Datensätzen und deren Ableitung sind in Anhang A-A1 ab Seite 121 zu finden.

Hinzu kommen die Zustände:

- Movement (MOV), bei Bewegungen des Schlafenden und
- Unidentified (UI), bei Störungen im EEG-Signal.

Damit ergibt sich ein Acht-Klassen-Problem:

Klasse	1	2	3	4	5	6	7	8
Zustand	S1	S2	S3	S4	REM	WACH	MOV	UI

Tabelle 2-1: Acht-Klassen-Problem der Schlafklassifikation mit sechs Schlafstadien und zwei weiteren Zuständen

Dabei können viele Arten von Störungen eine Klassifikation in die Klasse UI bedingen. In Abbildung 2-1 sind die Arten der möglichen Störungen, sogenannte Artefakte⁷, dargestellt. Die Störungen werden in biologische und technische Artefakte, auch exogene Artefakte genannt, unterteilt. Zu den technischen Artefakten zählen zum einen instrumentelle Störungen. Das sind sämtliche Störungen, die durch das Gerät oder die Elektroden entstehen. Zum anderen sind elektromagnetische Interferenzen (EMI) für technische Störungen verantwortlich, d. h. Einstrahlungen von Signalen aus der Umwelt. Dazu gehören vor allem die 50 Hertz (Hz) aus unserem Stromnetz. Zu biologischen Artefakten gehören die biologisch-elektrischen Signale durch die Herzaktivität, durch Muskelpotentiale und durch Augenbewegungen. Weiterhin verursachen Lid- und Körperbewegungen sowie Pulswellen biologische Störungen des EEG-Signals. Ein zu hoher Hautwiderstand verschlechtert die Signalqualität des EEG, deshalb zählt auch er zu den Ursachen von biologischen Störungen. [Wal12]

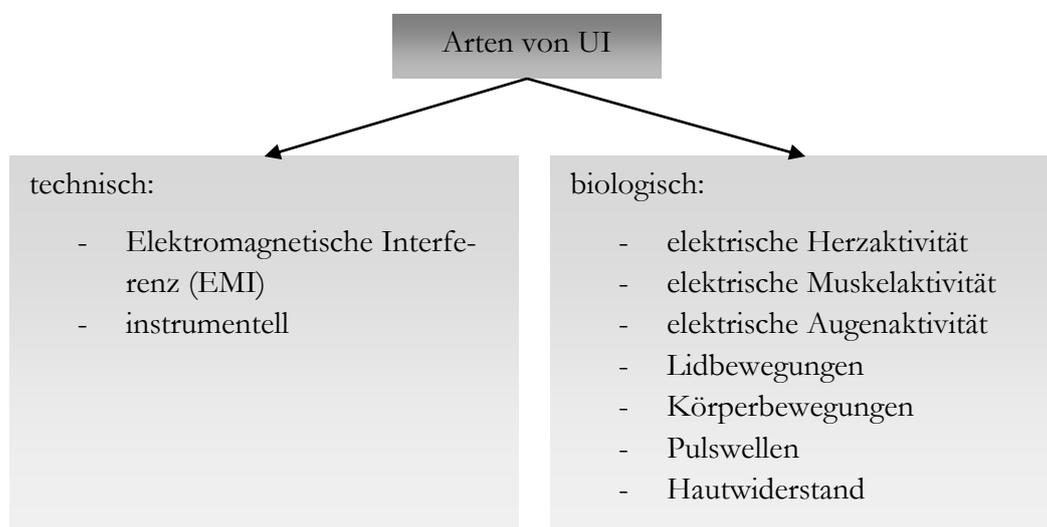


Abbildung 2-1: Einteilung und Auflistung der Arten von UI-Epochen [Wal12]

⁷ siehe Begriffsverzeichnis

Die Komplexität und Vielseitigkeit der Klasse UI bewirkt zunächst eine schlechte automatische Klassifikation dieser Klasse und eine starke negative Beeinflussung der gesamten Klassifikationsgüte. Aus diesem Grund und für die Vergleichbarkeit der Resultate mit [Wen05] wurde für die Untersuchungen diese Klasse aus den Daten entfernt, sodass ein Sieben-Klassen-Problem resultiert:

Klasse	1	2	3	4	5	6	7	-
Zustand	S1	S2	S3	S4	REM	WACH	MOV	UI

Tabelle 2-2: Sieben-Klassen-Problem der Schlafklassifikation mit sechs Schlafstadien und einem weiteren Zustand

Die Schlafstadien wurden mit Hilfe der von Rechtschaffen und Kales aufgestellten Regeln zur standardisierten Schlafklassifikation [RK68] anhand des EEG von Experten identifiziert. Dieser Klassifikationsstandard wurde bereits von anderen standardisierten Klassifikationssystemen abgelöst. Diese wurden durch die American Academy of Sleep Medicine (AASM) erarbeitet. Aktuell findet der ICSD2-Standard (International Classification of Sleep Disorders, second edition) [AAS05] in den meisten Schlaflaboren Anwendung. Dennoch wurden diese Daten nach dem Rechtschaffen und Kales-Standard verwendet, da für die Klassifikationsmethodenentwicklung der angewendete Standard unerheblich ist und diese Daten bereits in [Wen05] zur Klassifikation mit neuronalen Netzen Anwendung fanden. Ein Vergleich der Klassifikationsmethoden wurde damit ermöglicht.

Allgemein wird bei der manuellen Klassifikation der Schlafstadien zum einen die Grundaktivität des EEG-Zeitsignals und zum anderen temporäre Ereignisse betrachtet. Dies sind zum Beispiel K-Komplexe und Schlafspindeln, die ein typisches Merkmal für das Stadium S2 darstellen [Wen05]. Ebenso sind Augenbewegungen ein typisches Merkmal des Schlafstadiums REM, welches sich stark von den anderen Schlafstadien unterscheidet. Diese Augenbewegungen bewirken ein gering frequentes Signal mit großen Amplituden.⁸

Das EEG-Signal ist abhängig von Faktoren, wie zum Beispiel dem Alter, und weist damit eine hohe interindividuelle Ausprägung auf [ED11]. Weiterhin ändern Medikamente wie Schlafmittel und Narkotika die Form des EEG-Signals. Damit weist das EEG ebenfalls eine hohe intra-individuelle Ausprägung auf. Der automatischen Klassifikation werden aber, außer dem EEG-Signal selbst, keine weiteren Informationen über den Probanden zur Verfügung stehen. Aus diesem Grund ist die Entwicklung eines robusten Klassifikators⁹, welcher auf eine Vielzahl von Daten gute Klassifikationsleistungen erreicht, für diese Aufgabe besonders wichtig.

In Abbildung 2-2 ist ein typisches Hypnogramm eines gesunden Erwachsenen stark vereinfacht dargestellt. Das Leichtschlafstadium S1 tritt insgesamt zu 10% auf. Das Stadium S2 tritt zu 50% auf und stellt damit den Hauptanteil des Schlafes dar. Nur 20% entfallen auf die Tiefschlafstadien S3 und S4, die auch oft zu einem Zustand zusammengefasst werden. Der REM-Schlaf macht die übrigen 20% des Schlafes aus. Zusammen mit der Klasse Movement, die rund 1% der Nacht auftritt (nicht im Hypnogramm dargestellt), ergibt sich naturgemäß ein stark ungleichverteiltes Klassifikationsproblem.

⁸ nähere Informationen zu den Schlafstadien in Anhang A-A2 ab Seite 124

⁹ siehe Begriffsverzeichnis

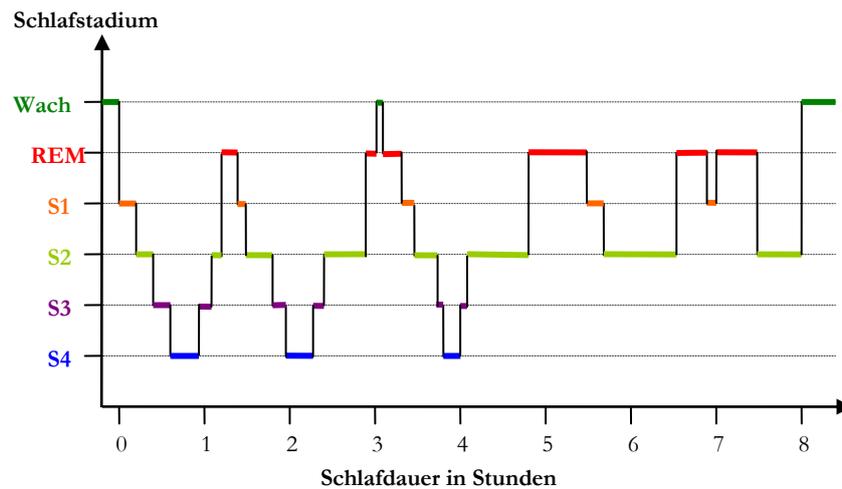


Abbildung 2-2: Vereinfachte Darstellung eines Hypnogramms eines gesunden Erwachsenen

Das Aufwachen während der Nacht ist auch bei einem „gesunden Schlaf“ nicht ungewöhnlich. Die Schlafstadien werden innerhalb einer Nacht zyklisch durchlaufen, wobei sich der Tiefschlaf hauptsächlich in der ersten Hälfte der Nacht konzentriert und in der zweiten Hälfte der REM-Schlaf mit immer längeren Perioden auftritt. Ob ein Mensch schlafgestört ist, kann anhand des Hypnogramms ermittelt werden. Schlafkrankheiten sind durch wenige Tiefschlafphasen, sehr häufige Wachphasen und besonders viele atypische Datenepochen gekennzeichnet. [Mül05][MM92]

Wie eingangs erwähnt, benötigt der Klassifikator Merkmale und das dazugehörige Schlafstadium (Klasse) eines Zeitabschnitts, um eine Zuordnung der Schlafstadien erlernen zu können. Die in dieser Klassifikationsaufgabe verwendeten Merkmale sind im Anhang unter Tabelle A-4 auf Seite 127 dargestellt. Es handelt sich dabei vor allem um Merkmale im Frequenzbereich, wie die relativen Leistungen in verschiedenen Frequenzbändern, Quantilfrequenzen, Frequenz der höchsten Amplituden in verschiedenen Frequenzbändern und akkumulierte Spektren. Weiterhin erwiesen sich die Hauptkomponenten des gesamten Spektrums und der Hüllkurvenspektren um verschiedene Frequenzbänder als geeignete Merkmale für die Schlafklassifikation [Wen05].

3 Theoretische Grundlagen der Support-Vektor-Maschine

3.1 Einführung in die Theorie der Support-Vektor-Maschinen

Die SVM ist heute eine Standardmethode zur Klassifikation und wird in vielen Büchern thematisiert [FP02][Wan05]. Als Ergebnis der Forschung auf diesem Gebiet entstehen immer mehr Abwandlungen und Erweiterungen der SVM [SSWB00][Suy02][FCL05]. Weiterhin wird die SVM auf immer mehr Anwendungsgebieten eingesetzt [Sri15][KS15]. Zum Verständnis der Methode und der in dem nachfolgenden Kapitel untersuchten Eigenschaften und Variationen der SVM soll zunächst auf die theoretischen Grundlagen eingegangen werden.

3.1.1 Ablauf einer Klassifikation und benötigte Datensätze

Für die SVM, wie auch für andere Klassifikatoren, gilt folgende Herangehensweise:

1. Akquisition von Daten und Berechnung des Merkmalvektors \underline{x}_i mit anschließender Bewertung dieser Daten, z. B. durch einen Experten mit einer Klasse y_i .
2. Belehren des Klassifikators und damit Finden einer Bildungsvorschrift, die möglichst viele Belehrungsdaten \underline{x}_i und die zugehörigen y_i beschreibt. Dies ist ein Optimierungsprozess, bei dem schrittweise das Modell des Klassifikators angepasst wird, bis die durch einen Testdatensatz erzeugten Resultate zufriedenstellend sind.
3. Mit einem von Belehrungs- und Testdatensatz unabhängigen Validierungsdatensatz kann die Güte des Klassifikators zur Bewertung der Robustheit bestimmt werden.

Die Datensätze sollten für einen robusten Klassifikator voneinander unabhängig sein, um realistische Gütewerte für unbekannte Daten zu erhalten. Das bedeutet im Fall der Schlafklassifikation, dass die Daten von unterschiedlichen Probanden stammen müssen.

Damit sind für die Erzeugung und Bewertung eines Klassifikators drei Datensätze notwendig:

- Belehrungsdaten,
- Testdaten und
- Validierungsdaten.¹⁰

3.1.2 Kreuzvalidierung

Häufig stehen allerdings nicht viele Daten zur Verfügung, da die Datenbewertung aufwendig ist oder, z. B. in der Medizin, Probanden nur begrenzt verfügbar sind. Aus diesem Grund wird der Testdatensatz oft mit Hilfe der Kreuzvalidierung eingespart. Bei der einfachen Kreuzvalidierung¹¹ (engl. cross-validation) [CL08] [Mik08] wird der Belehrungsdatensatz zufällig in N gleichgroße Teilmengen unterteilt. Nun finden mehrere Belehrungen mit jeweils $N-1$ dieser Teilmengen statt. Die Bewertung erfolgt durch den übrigen Teildatensatz als Testdatensatz. Dies geschieht, bis alle Teilmengen einmal als Testdatensatz verwendet wurden. Die Gesamtgüte G_{ges} ergibt sich nach:

$$G_{\text{ges}} = \frac{\sum_{i=1}^N G_i}{N} \quad (3.1)$$

aus dem Mittelwert aller ermittelten Gütewerte G_i . **Dieser wird im Folgenden auch als Belehrungsgrad¹² bezeichnet.** Der Algorithmus der einfachen Kreuzvalidierung ist in Abbildung B-1 auf Seite 129 in Form eines Programmablaufdiagramms dargestellt. Unter Anwendung der Kreuzvalidierung ergeben sich zur Erzeugung und Bewertung des Klassifikators nun zwei notwendige Datensätze:

- Belehrungsdaten und
- Validierungsdaten.

3.1.3 Verteilung der Belehrungsdaten

Da die Belehrungsdaten das Modell des Klassifikators festlegen, ist die Verteilung der Belehrungsdaten auf die einzelnen Klassen, aber auch innerhalb der Klassen entscheidend. Bei vielen Klassifikatoren ist eine identische Datenmenge für alle Klassen zwingend notwendig, da ansonsten die Majoritätsklassen überbelehrt und die Minoritätsklassen unterbelehrt werden würden¹³. Dies hat zur Folge, dass die Minoritätsklassen zugunsten der Majoritätsklassen permanent falsch klassifiziert werden. Die SVM hingegen kann auch mit ungleichverteilten Daten umgehen [Wan05], allerdings trifft dies nur auf moderat ungleichmäßig verteilte Daten zu [AKJ04]. Hochgradig ungleichverteilte Daten mit einem Verhältnis von etwa 50:1 führen dazu, dass nach dem Ockhams Rasiermesser Prinzip¹⁴ die Majoritätsklasse bevorzugt wird. In [AKJ04] wird die ungleichmäßige Datenverteilung für die SVM-Klassifikation thematisiert.

¹⁰ Definitionen der verwendeten Begriffe sind im Begriffsverzeichnis zu finden.

¹¹ Weitere Validierungstechniken sind unter [Mik08] nachzulesen.

¹² siehe Begriffsverzeichnis

¹³ siehe Begriffsverzeichnis

¹⁴ Ockhams Rasiermesser, oder auch Sparsamkeitsprinzip genannt, ist ein heuristisches Prinzip und besagt, dass die einfachste Theorie allen anderen vorzuziehen ist [Mol03a].

Zur Kompensation des Ungleichgewichts sind zwei Methoden anwendbar:

1. Zum einen kann eine Gewichtung der Klassen erreichen, dass die Minoritätsklassen mehr Aufmerksamkeit erhalten (interne Gewichtung).
2. Zum anderen können die Daten künstlich balanciert werden, indem die Majoritätsklasse durch Unterabtastung reduziert und/oder die Minoritätsklasse durch Überabtastung künstlich vergrößert wird (externe Gewichtung).

Ein entscheidender Nachteil der Unterabtastung ist der hohe Verlust an validen Daten. Bei biologischen Daten führt dies, durch deren hohe Individualität, meist zu einem ungenauem Modell und damit einer geringen Klassifikationsgüte. Aus diesem Grund ist das Überabtasten in diesem Fall zu bevorzugen.

3.1.4 Skalierung der Eingangswerte

Es ist festzustellen, dass bei der Klassifikation mit SVM die Merkmale mit großer Amplitude andere Merkmale dominieren [HCL10]. Werden die Merkmale skaliert, sodass alle Merkmale im gleichen Wertebereich liegen, kann dies verhindert werden. Meistens wird eine lineare Skalierung im Wertebereich zwischen $\min_S = -1$ und $\max_S = +1$ gewählt. Allerdings ist auch ein Wertebereich zwischen $\min_S = 0$ und $\max_S = 1$ oder einem anderen festen Minimum und Maximum möglich [CL08]. Die Überführung des realen Wertebereichs (Index R) in den skalierten Wertebereich (Index S) ist in Abbildung 3-1 dargestellt. Daraus ergibt sich für die Ermittlung des skalierten Wertes wert_S die Gleichung:

$$\text{wert}_S = (\text{wert}_R - \min_R) \frac{(\max_S - \min_S)}{(\max_R - \min_R)} + \min_S \quad (3.2)$$

Die Skalierung verhindert außerdem Probleme bei der Berechnung des Skalarprodukts, die bei sehr großen Werten auftreten könnten [HCL10]. Da aber der ursprüngliche Wertebereich der Klassifikationsdaten zum Zeitpunkt der Belehrung noch unbekannt ist, muss die Festlegung der Skalierungsfaktoren ausschließlich durch die Belehrungsdaten erfolgen. Die Skalierungsfaktoren jedes Merkmals werden gespeichert und in der gleichen Weise auf die Validierungsdaten und unbekannte Daten angewendet. Jedoch können bei dieser einfachen Art der Skalierung auch Probleme auftreten. Nur ein abweichendes Datum übt einen starken Einfluss auf die Skalierung aus.

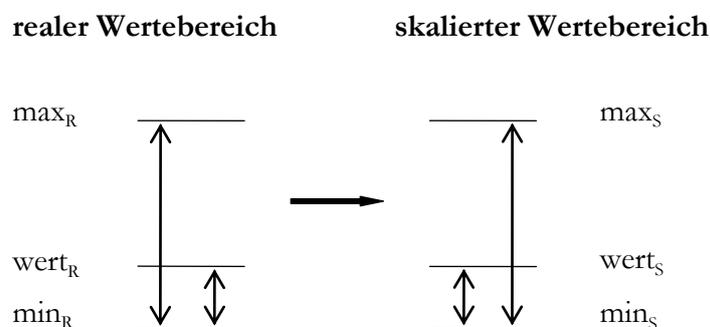


Abbildung 3-1: Prinzip der linearen Skalierung

3.1.5 Das Verfahren der Support-Vektor-Maschine

Die SVM ist keine physikalische Maschine, es handelt sich um ein rein mathematisches Verfahren. Aus einer Vielzahl von Informationsmaterial zum Thema SVM wurden die Veröffentlichungen [Vap95], [SS01], [FP02], [Suy02] und [Wan05] als Quellen des folgenden Abschnitts ausgewählt.

Das Prinzip der SVM soll in Abbildung 3-2 verdeutlicht werden. In einem f -dimensionalen Merkmalsraum sind Belehrungsdaten, die in zwei Klassen unterschieden werden, dargestellt. Die Daten der Belehrungsdatenmenge X setzen sich aus N Merkmalsvektoren mit \underline{x}_i , dem i -ten Merkmalsvektor und y_i , der zum i -ten Datum gehörigen Klasse, zusammen:

$$X = \{\underline{x}_i, y_i\}_{i=1}^N \quad (3.3)$$

Der Merkmalsvektor \underline{x}_i ist Element der reellen Zahlen im f -dimensionalen Raum \mathbb{R}^f . Die Anzahl der Dimensionen entspricht dabei der Anzahl der Merkmale. Die Klasse y kann bei dieser binären Entscheidung den Wert $+1$ oder -1 annehmen. Es gilt:

$$\underline{x}_i \in \mathbb{R}^f \text{ und} \quad (3.4)$$

$$y_i \in \{+1, -1\} \quad (3.5)$$

Ziel ist es eine Trennebene (TE) (engl. hyperplane) zu finden, welche die beiden Klassen trennt. Bedingung ist dabei eine möglichst große Spannweite M (engl. margin), d. h. den Abstand zwischen dem am nächsten liegenden Punkt zur Trennebene, zu erhalten.

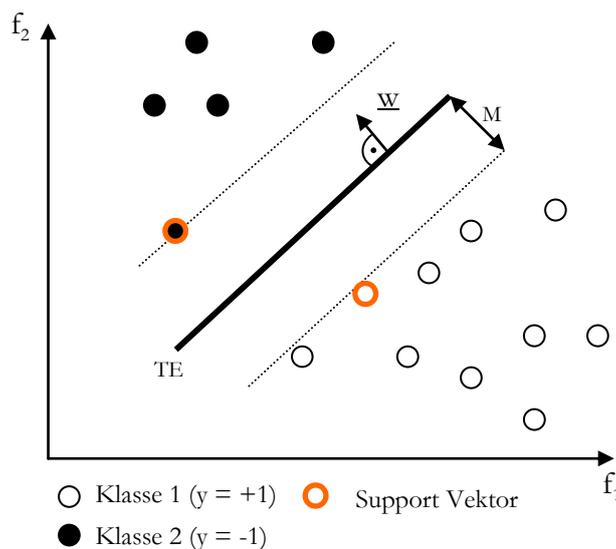


Abbildung 3-2: Prinzipielle Funktionsweise der SVM [Suy02]

Die Trennebene wird durch:

$$(\underline{w} \bullet \underline{x}_i) + b = 0 \quad (3.6)$$

beschrieben, wobei \underline{w} der Normalenvektor ist, welcher orthogonal zur Trennebene liegt und somit die Ausrichtung der Ebene bestimmt. Die Vektoren \underline{x}_i liegen auf der Trennebene, b ist die Verschiebung der Ebene und \bullet stellt das Skalarprodukt dar. Die Datenpunkte, die der Trennebene am nächsten liegen, werden Support-Vektoren (dt. Stützvektoren) genannt. Diese Support-Vektoren (SV) begründen auch den Namen der Methode.

Ist eine geeignete Trennebene gefunden, ergibt sich eine einfache lineare Entscheidungsfunktion, wonach neue Beispiele klassifiziert werden können:

$$f(\underline{x}_{\text{neu}}) = \text{sign}((\underline{w} \bullet \underline{x}_{\text{neu}}) + b) \quad (3.7)$$

Die Entscheidungsfunktion ist eine Sprungfunktion (Signum-Funktion) und wird nur durch die Support-Vektoren repräsentiert. Damit sind alle anderen Belegungsdaten für eine spätere Klassifikation unbedeutend und die SVM ist theoretisch unabhängig von der Anzahl der Belegungsdaten [FP02]. Die Dimension des Raumes, in dem die Trennebene aufgespannt wird, ist einerseits von der Anzahl der Merkmale abhängig. Zum anderen wird, um die in der Regel nichtlinearen Daten durch eine lineare Trennebene trennen zu können, die Dimension des Raumes erhöht, bis eine lineare Trennung möglich wird (siehe Abbildung 3-3). Dadurch kann die Dimension jedoch extrem ansteigen. Die Berechnung des Skalarprodukts in einem derartig extrem hochdimensionalen Raum, dem sogenannten „feature space“, kann durch die extrem hohe Rechenintensität sehr schwierig bis unmöglich sein. Deshalb wird der „Kernel-Trick“ angewendet. Ein Kernel¹⁵ ist eine Funktion im niederdimensionalen Raum, die sich ähnlich dem Skalarprodukt im hochdimensionalen Raum verhält. So muss der hochdimensionale Raum nicht bekannt sein. Die SVM zählt deshalb zu den Kernel-Methoden.

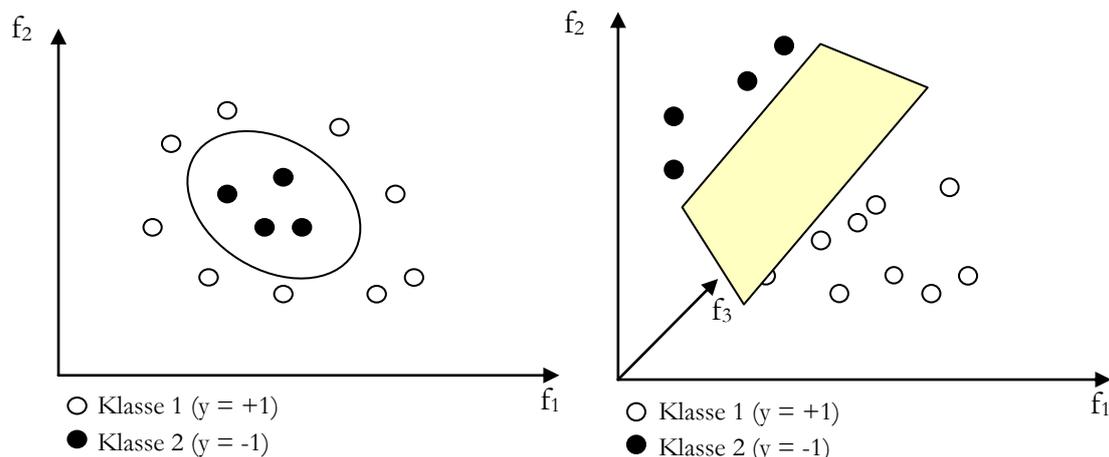


Abbildung 3-3: Prinzip der linearen Trennung von nichtlinearen Daten durch Erhöhung der Dimension: nichtlinear trennbarer Datensatz im zweidimensionalen Raum (links) und der selbe Datensatz linear trennbar im dreidimensionalen Raum (rechts) [SS01]

¹⁵ siehe Begriffsverzeichnis

In Tabelle 3-1 werden vier mögliche Kernel vorgestellt:

linear	$K(\underline{x}_1, \underline{x}_2) = (\underline{x}_1 \bullet \underline{x}_2)$
polynomielle	$K(\underline{x}_1, \underline{x}_2) = (\gamma(\underline{x}_1 \bullet \underline{x}_2) + c_0)^q$
sigmoide	$K(\underline{x}_1, \underline{x}_2) = \tanh[(\gamma(\underline{x}_1 \bullet \underline{x}_2) + c_0)]$
radial basis function (RBF)	$K(\underline{x}_1, \underline{x}_2) = \exp(-\gamma \ \underline{x}_1 - \underline{x}_2\ ^2)$

Tabelle 3-1: Auflistung möglicher Kernel zur Klassifikation [CL08]

Wobei \underline{x}_1 und \underline{x}_2 Vektoren sind, deren Ähnlichkeit bestimmt werden soll. Weiterhin wird der Grad des Polynoms mit q , die Weite der gaußschen Glockenkurve mit γ und die additive Konstante des Polynoms mit c_0 bezeichnet [SS01].

Die Kernel-Funktion und andere SVM-Parameter können jedoch nur empirisch für eine spezifische Klassifikationsaufgabe bestimmt werden. Weiterhin kann bei dieser Methode nur eine vollständige Belehrung erfolgen. Ein Nachbelehren einzelner Beispiele ist nicht möglich. Dafür liefert der SVM-Algorithmus unabhängig vom Startpunkt stabile Ergebnisse. Der Optimierungsprozess findet das globale Optimum [Mik08]. Auch das Thema der Über- oder Unteranpassung während des Optimierungsprozesses, welches bei einigen Klassifikatoren problematisch ist, spielt für die SVM keine Rolle. Da die einfache Entscheidungsfunktion nur auf den SV basiert, hat die SVM einen abschätzbaren Rechenaufwand und benötigt eine abschätzbare Speicherkapazität. Zudem ist das Arbeiten in hohen Dimensionen möglich. Somit können Klassifikationsprobleme mit vielen Merkmalen oder Nichtlinearitäten berechnet werden [Fis07]. Außerdem besitzt die SVM laut Literatur eine hohe Generalisierungsfähigkeit und ist gut auf reale Probleme anwendbar [BC00]. Inwieweit diese Aussagen auf die vorliegende Klassifikationsaufgabe zutrifft, soll in dieser Arbeit untersucht werden.

3.1.6 Herleitung der Entscheidungsfunktion

Zur Herleitung der Entscheidungsfunktion wurden die Werke [Vap95], [SS01], [FP02], [Suy02] und [Wan05] als Quellen verwendet. Wie in Abschnitt 3.1.5 bereits beschrieben, wird eine Trennebene TE: $(\underline{w} \bullet \underline{x}_i) + b = 0$ (3.6) gesucht, welche die Belehrungsdaten trennt und dabei die Spannweite M maximiert. Die Spannweite M wird durch die Support-Vektoren bestimmt, deshalb wird zunächst der Abstand a zwischen der Trennebene TE und einem SV bestimmt (siehe Abbildung 3-4). Die SV befinden sich auf zwei Ebenen TE_1 und TE_2 parallel zur Trennebene TE, die durch die Gleichungen:

$$(\underline{w} \bullet \underline{x}_1) + b = +1 \quad (3.8)$$

und

$$(\underline{w} \bullet \underline{x}_2) + b = -1 \quad (3.9)$$

beschrieben werden.

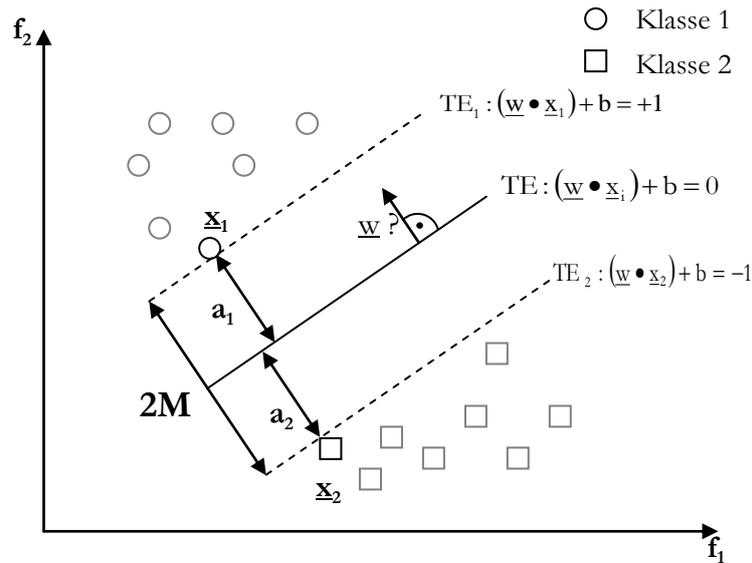


Abbildung 3-4: Bestimmung der maximalen Spannweite M über den Abstand der SV zur Trennebene TE [SS01]

Zur Bestimmung des Abstands dient die Hessesche Normalform [Gn001]:

$$a(E, P) = \frac{E(P)}{\|\underline{n}\|}, \quad (3.10)$$

wobei $a(E, P)$ der Abstand zwischen einer Ebene E und einem Punkt P ist, $E(P)$ die Ebenengleichung in Normalform und $\|\underline{n}\|$ der Betrag des Normalenvektors, welcher senkrecht auf der Ebene steht. Damit ergibt sich für den Abstand der SV zur Trennebene:

$$a(TE, \underline{x}_i) = \frac{(\underline{w} \bullet \underline{x}_i) + b}{\|\underline{w}\|}, \quad (3.11)$$

wobei $a < 0$ wird, wenn der Punkt auf der gleichen Seite von TE wie der Koordinatenursprung liegt. Im Beispiel aus Abbildung 3-4 hätte deshalb $a_1 := a(TE_1, \underline{x}_1)$ einen positiven Wert:

$$a_1 = \left(\frac{\underline{w}}{\|\underline{w}\|} \bullet \underline{x}_1 \right) + \frac{b}{\|\underline{w}\|} = \frac{1}{\|\underline{w}\|} \quad (3.12)$$

und $a_2 := a(TE_2, \underline{x}_2)$ einen negativen Wert:

$$a_2 = \left(\frac{\underline{w}}{\|\underline{w}\|} \bullet \underline{x}_2 \right) + \frac{b}{\|\underline{w}\|} = \frac{-1}{\|\underline{w}\|}. \quad (3.13)$$

Um diesen Effekt auszugleichen, wird die Klasse y_i eingefügt und man schreibt anstelle von Gleichung (3.11):

$$a(TE, \underline{x}_i) = y_i \left(\left(\frac{\underline{w}}{\|\underline{w}\|} \bullet \underline{x}_i \right) + \frac{b}{\|\underline{w}\|} \right) \quad (3.14)$$

Damit ergibt sich für die Spannweite M :

$$2M = a_1 + a_2 = \left(\frac{\underline{w}}{\|\underline{w}\|} \bullet \underline{x}_1 \right) + \frac{b}{\|\underline{w}\|} + (-1) \left(\left(\frac{\underline{w}}{\|\underline{w}\|} \bullet \underline{x}_2 \right) + \frac{b}{\|\underline{w}\|} \right) = \frac{1}{\|\underline{w}\|} + (-1) \frac{-1}{\|\underline{w}\|} = \frac{2}{\|\underline{w}\|}$$

nach Auflösung des Skalarprodukts:

⇔

$$2M = \sum_{j=1}^f \frac{w_j}{\|\underline{w}\|} x_{1j} - \sum_{j=1}^f \frac{w_j}{\|\underline{w}\|} x_{2j} = \sum_{j=1}^f \frac{w_j}{\|\underline{w}\|} (x_{1j} - x_{2j}) = \frac{2}{\|\underline{w}\|}$$

$$2M = \left(\frac{\underline{w}}{\|\underline{w}\|} \bullet (\underline{x}_1 - \underline{x}_2) \right) = \frac{2}{\|\underline{w}\|} \quad (3.15)$$

Um die Spannweite M zu maximieren, muss $\|\underline{w}\|$ minimiert werden, d. h. es wird die quadratische Zielfunktion [FP02]:

$$m(\underline{w}) = \frac{1}{2} \|\underline{w}\|^2 \quad (3.16)$$

unter der Nebenbedingung (NB):

$$y_i ((\underline{w} \bullet \underline{x}_i) + b) \geq 1 \quad (3.17)$$

minimiert. Die Nebenbedingung ist nur im Falle einer „richtigen“ Zuordnung der Punkte zur Klasse erfüllt. Dies wird aus den folgenden vier Fällen ersichtlich:

1. $y_i = 1$ und $((\underline{w} \bullet \underline{x}_i) + b) \geq 1 \rightarrow$ NB erfüllt \rightarrow korrekte Klassifikation,
2. $y_i = 1$ und $((\underline{w} \bullet \underline{x}_i) + b) \leq -1 \rightarrow$ NB nicht erfüllt \rightarrow inkorrekte Klassifikation,
3. $y_i = -1$ und $((\underline{w} \bullet \underline{x}_i) + b) \geq 1 \rightarrow$ NB nicht erfüllt \rightarrow inkorrekte Klassifikation und
4. $y_i = -1$ und $((\underline{w} \bullet \underline{x}_i) + b) \leq -1 \rightarrow$ NB erfüllt \rightarrow korrekte Klassifikation.

Zur Lösung des Optimierungsproblems mit Nebenbedingung wird der Lagrange-Ansatz verwendet, welcher allgemein lautet:

$$L(l_1, l_2, \alpha) = g(l_1, l_2) - \alpha [h(l_1, l_2) - c], \quad (3.18)$$

wobei $g(l_1, l_2)$ die zu optimierende Gleichung, $h(l_1, l_2) = c$ die Nebenbedingung und α der Lagrange-Multiplikator ist [EED15].

Damit ergibt sich für das vorhandene Problem durch Einsetzen der Gleichungen (3.16) und (3.17):

$$L(\underline{w}, b, \alpha) = \frac{1}{2} \|\underline{w}\|^2 - \sum_{i=1}^N \alpha_i (y_i ((\underline{w} \bullet \underline{x}_i) + b) - 1) \quad (3.19)$$

wobei α_i die Lagrange-Multiplikatoren sind und N die Anzahl der Belegungsdaten darstellt. Gesucht ist der Sattelpunkt der Lagrange-Funktion $L(\underline{w}, b, \alpha)$, bei dem die Funktion im Hinblick auf \underline{w} und b minimiert und gleichzeitig im Hinblick auf α_i maximiert wird [SS01].

Zur Lösung des Problems erfolgen die ersten Ableitungen der Lagrange-Funktion:

$$\frac{\partial L(\underline{w}, b, \alpha)}{\partial b} = - \sum_{i=1}^N \alpha_i y_i = 0 \quad \rightarrow \quad \sum_{i=1}^N \alpha_i y_i = 0 \quad (3.20)$$

und

$$\frac{\partial L(\underline{w}, b, \alpha)}{\partial \underline{w}} = \underline{w} - \sum_{i=1}^N \alpha_i y_i \underline{x}_i = 0 \quad \rightarrow \quad \underline{w} = \sum_{i=1}^N \alpha_i y_i \underline{x}_i \quad . \quad (3.21)$$

Da die Nebenbedingung eine Ungleichung (siehe Gleichung (3.17)) darstellt und der ursprüngliche Lagrange-Ansatz nur für Gleichungen gilt, wird das Kuhn-Tucker Theorem angewendet, welches eine Verallgemeinerung der Lagrange-Multiplikatoren darstellt und auch Ungleichungen zulässt. Deshalb werden Gleichung (3.20) und (3.21) auch Karush-Kuhn-Tucker (KKT)-Bedingungen genannt. Diese ergeben durch das Einsetzen in Gleichung (3.19) das duale Problem:

$$L_D = \frac{1}{2} \left\| \sum_{i=1}^N \alpha_i y_i \underline{x}_i \right\|^2 - \sum_{i=1}^N \alpha_i y_i ((\underline{w} \bullet \underline{x}_i) + b) + \sum_{i=1}^N \alpha_i$$

nach Ausmultiplizieren

$$\Leftrightarrow L_D = \frac{1}{2} \left\| \sum_{i=1}^N \alpha_i y_i \underline{x}_i \right\|^2 - \sum_{i=1}^N \alpha_i y_i (\underline{w} \bullet \underline{x}_i) - \sum_{i=1}^N \alpha_i y_i b + \sum_{i=1}^N \alpha_i$$

nach Einsetzen von Gleichung (3.20) und (3.21)

$$\Leftrightarrow L_D = \frac{1}{2} \left\| \sum_{i=1}^N \alpha_i y_i \underline{x}_i \right\|^2 - \sum_{j=1}^N \sum_{i=1}^N \alpha_i \alpha_j y_i y_j (\underline{x}_i \bullet \underline{x}_j) - 0 + \sum_{i=1}^N \alpha_i$$

Betrag quadrieren

$$\Leftrightarrow L_D = \frac{1}{2} \sum_{j=1}^N \sum_{i=1}^N \alpha_i \alpha_j y_i y_j (\underline{x}_i \bullet \underline{x}_j) - \sum_{j=1}^N \sum_{i=1}^N \alpha_i \alpha_j y_i y_j (\underline{x}_i \bullet \underline{x}_j) - 0 + \sum_{i=1}^N \alpha_i$$

$$L_D = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i,j=1}^N \alpha_i \alpha_j y_i y_j (\underline{x}_i \bullet \underline{x}_j) \quad (3.22)$$

mit den Nebenbedingungen:

$$\sum_{i=1}^N \alpha_i y_i = 0 \quad (3.23)$$

und

$$\alpha_i \geq 0 \quad . \quad (3.24)$$

Das vorliegende Optimierungsproblem besitzt keine lokalen Optima, weshalb das globale Optimum sicher ermittelt wird. Ist eine Lösung gefunden, werden nur einige der Lagrange-Multiplikatoren $\alpha_i \neq 0$ sein, dies sind die Support-Vektoren. Alle anderen Belegungsdatenpunkte ($\alpha_i = 0$) haben keinen Einfluss auf das gefundene Modell.

Die Verschiebung b kann durch

$$b = y_i - \sum_{i=j}^N y_j \alpha_j (\underline{x}_i \bullet \underline{x}_j) \quad (3.25)$$

beschrieben werden, was sich aus der Ebenengleichung

$$(\underline{w} \bullet \underline{x}_i) + b = y_i \quad (3.26)$$

und $\underline{w} = \sum_{i=1}^N \alpha_i y_i \underline{x}_i$ (3.21) ergibt, wobei es sich dabei um andere Indizes i handelt, weshalb stattdessen j verwendet wird. Die Verschiebung b wird somit durch Einsetzen eines SV-Datenpunktes bestimmt. Die resultierende Entscheidungsfunktion aus Gleichung (3.7) auf Seite 15 mit der Gleichung (3.21) auf Seite 19 lautet nun:

$$f(\underline{x}_{\text{neu}}) = \text{sign} \left(\sum_{i=1}^N \alpha_i y_i (\underline{x}_i \bullet \underline{x}_{\text{neu}}) + b \right), \quad (3.27)$$

wobei $\underline{x}_{\text{neu}}$ einen unbekanntes und zu klassifizierenden Punkt darstellt.

Diese Herleitung und Gleichungen beziehen sich auf den einfachsten Fall der SVM, der auf linear trennbare Daten zutrifft. Dieser Fall ist in der Realität selten, selbst wenn man die Dimensionen erheblich erhöht. Hierfür findet die „Soft-Margin“-SVM Anwendung, die anschließend in Abschnitt 3.1.9 unter dem Absatz C-SVC beschrieben wird.

3.1.7 Anwendung der Kernel-Funktion

Die Kernel-Funktion soll, wie zuvor in Abschnitt 3.1.5 beschrieben, die Berechnung des Skalarprodukts im Hochdimensionalen vermeiden. Eine Funktion muss Bedingungen erfüllen, um als Kernel-Funktion $K(\underline{x}_i, \underline{x}_j)$ verwendet werden zu können. Dies ist als das Theorem von Mercer bekannt [SS01].

1. Die Funktion muss symmetrisch sein: $K(\underline{x}_i, \underline{x}_j) = K(\underline{x}_j, \underline{x}_i)$.
2. Die Kernel-Matrix $\text{km}_{ij} := K(\underline{x}_i, \underline{x}_j)$ muss (semi-) positiv definit für alle Belegungsdaten sein: $\sum_{i,j} \alpha_i \alpha_j \text{km}_{ij} \geq 0$ für $\forall \alpha \in \mathbb{R}^f$.

Die Anwendung dieses Kernel-Tricks ändert die Lösung des Optimierungsproblems (vergleiche Gleichung (3.22))[FP02]:

$$L_D = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i,j=1}^N \alpha_i \alpha_j y_i y_j K(\underline{x}_i, \underline{x}_j) \quad (3.28)$$

(mit den Nebenbedingungen $\sum_{i=1}^N \alpha_i y_i = 0$ (3.23) und $\alpha_i \geq 0$ (3.24)).

Die Entscheidungsfunktion $f(\underline{x}_{\text{neu}})$ ergibt sich zu:

$$f(\underline{x}_{\text{neu}}) = \text{sign} \left(\sum_{i=1}^N \alpha_i y_i K(\underline{x}_i, \underline{x}_{\text{neu}}) + b \right) \quad (3.29)$$

3.1.8 Optimierungsprozess

Der Optimierungsprozess garantiert das Finden des globalen Optimums [Mik08]. Um das quadratische Problem, d. h. das Finden der Lagrange-Multiplikatoren α_i , wie es zuvor in Abschnitt 3.1.6 beschrieben wurde, auch für große Belegungsdatenmengen lösen zu können, wird in [CL08] ein Zerlegungsverfahren (engl. decomposition method) angewendet. Bei dieser Methode wird das große quadratische Problem in viele kleine quadratische Probleme zerlegt und damit lediglich eine Teilmenge von α pro Iteration verändert. Diesen Vorgang nennt man „chunking“. Der extrem Fall dieses Verfahrens ist die „Sequential Minimal Optimization“ [Pla99], kurz SMO, bei der die minimale Anzahl α pro Iteration verändert werden. Das bedeutet in diesem Fall zwei α pro Iteration. Durch die SMO wird die analytische Lösung des quadratischen Problems ermöglicht und erspart eine aufwendige numerische Lösung. Der SMO-Algorithmus ist einfach zu implementieren und oft schneller als andere Zerlegungsverfahren. Das macht diese Methode für große Belegungsdatenmengen geeignet [HCL10]. Basierend auf dem SMO-Prinzip wurde der verwendete Optimierungsalgorithmus gestaltet [FCL05].

3.1.9 Arten von SVM

Die SVM wird unterschieden in die bisher behandelte Support-Vektor-Klassifikation (engl. Support Vector Classification - SVC) und die Support-Vektor-Regression (SVR). Die Abbildung 3-5 zeigt verschiedenen Arten der SVM, welche im Folgenden vorgestellt werden [Sta11][SS01]. Jede dieser SVM-Arten unterscheidet sich in Berechnung und Anwendung.

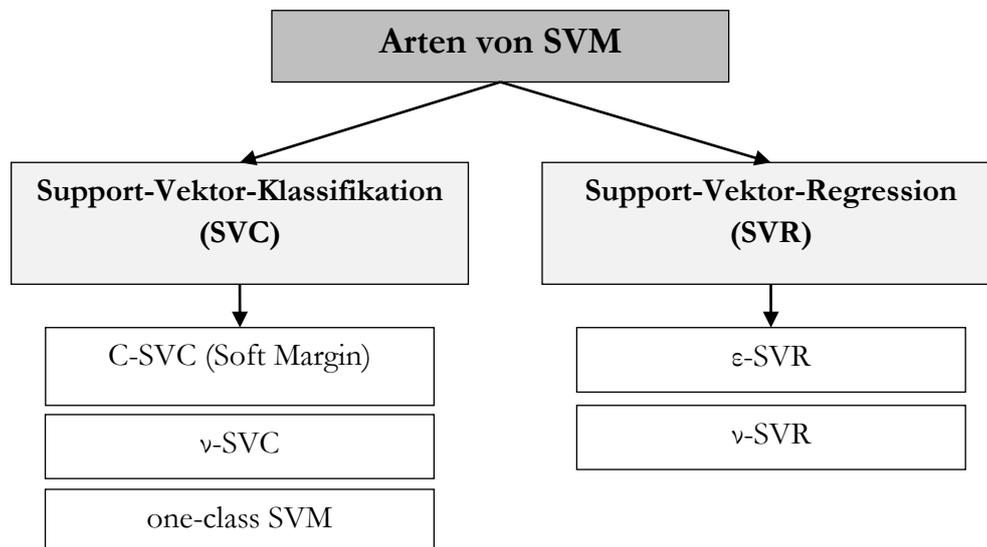


Abbildung 3-5: Übersicht der Arten von SVM

C-SVC

Die C-SVC entspricht der sogenannten „Soft-Margin“-SVM, die zu den SVM-Klassifikationen gehört und 1995 von Cortes und Vapnik vorgestellt wurde [CV95]. Hierbei werden bei nicht-linear trennbaren Daten „Ausreißer“ in den Belehrungsdaten zugelassen, um diese dennoch linear zu trennen. Dabei stellt ξ die Schlupfvariable (engl. slack variable) dar. Diese ist:

- $\xi = 0$ im Falle der korrekten Klassifikation,
- $0 < \xi < 1$, wenn der Datenpunkt innerhalb der Spannweite, aber auf der richtigen Seite liegt und
- $\xi > 1$, wenn der Datenpunkt auf der falschen Seite der Trennebene liegt.

Abbildung 3-6 stellt diesen Sachverhalt graphisch dar [Wan05]. Über das Fehlergewicht C wird jeder „Ausreißer“ proportional zum Ausmaß der Fehlklassifikation bestraft und somit die Balance bestimmt zwischen einer großen Spannweite bei vielen bzw. erheblichen Fehlern und einer kleinen Spannweite bei wenig bzw. geringeren Fehlern. Damit ist die zu minimierende Gleichung (vergleiche Gleichung (3.16) auf Seite 18):

$$m(\underline{w}, \xi) = \frac{1}{2} \|\underline{w}\|^2 + C \sum_{i=1}^N \xi_i^q, \quad (3.30)$$

wobei der Exponent q häufig gleich eins gewählt wird. Es gilt weiterhin:

$$C > 0 \quad (3.31)$$

unter der Nebenbedingung, dass (vergleiche Gleichung (3.17) auf Seite 18):

$$y_i((\underline{w} \bullet \underline{x}_i) + b) \geq 1 - \xi_i \quad (3.32)$$

mit

$$\xi_i \geq 0 \quad (3.33)$$

und $i = 1, 2, \dots, N$.

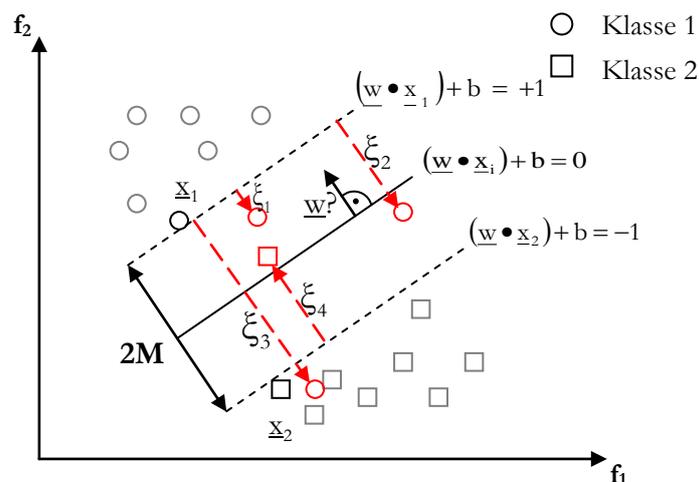


Abbildung 3-6: Bestimmung der Trennebenen bei nichtlinear trennbaren Daten mittels C-SVC [Fis07]

Bei Verwendung des Lagrange-Ansatzes (vergleiche Gleichung (3.19) auf Seite 18) ergibt sich:

$$L(\underline{w}, b, \xi, \alpha, \beta) = \frac{1}{2} \|\underline{w}\|^2 + C \sum_{i=1}^N \xi_i - \sum_{i=1}^N \alpha_i (y_i ((\underline{w} \bullet \underline{x}_i) + b) - (1 - \xi_i)) - \sum_{i=1}^N \beta_i \xi_i, \quad (3.34)$$

wobei β ein weiterer Lagrange-Multiplikator ist.

Mit den Ableitungen nach \underline{w} , b und ξ ergeben sich die folgenden Gleichungen (vergleiche Gleichung (3.20) und (3.21) auf Seite 18):

$$\underline{w} = \sum_{i=1}^N \alpha_i y_i \underline{x}_i, \quad (3.35)$$

$$\sum_{i=1}^N \alpha_i y_i = 0 \quad \text{und} \quad (3.36)$$

$$\alpha_i = C - \beta_i. \quad (3.37)$$

Die Lösung des dualen Problems führt zu (vergleiche Gleichung (3.22) auf Seite 19):

$$L_D = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i,j=1}^N \alpha_i \alpha_j y_i y_j (\underline{x}_i \bullet \underline{x}_j), \quad (3.38)$$

wobei

$$\sum_{i=1}^N \alpha_i y_i = 0 \quad \text{und} \quad (3.39)$$

$$C \geq \alpha_i \geq 0 \quad (3.40)$$

als Nebenbedingungen definiert sind (vergleiche Gleichung (3.23) und (3.24) auf Seite 19).

Die Belehrungsdatenpunkte, bei denen $\alpha_i = 0$ bestimmt wird, stellen keine SV dar und haben auf das Modell keinen Einfluss. Die Datenpunkte mit $\alpha_i = C$ sind die „Ausreißer“, welche bewusst als Fehler für die Bestimmung der Trennebenen zugelassen werden. Alle anderen Daten mit $0 < \alpha_i < C$ stellen die SV dar [Pla99]. Wie aus Gleichung (3.38) ersichtlich ist, verschwinden die Schlupfvariablen aus dem Optimierungsproblem. Nur das Fehlergewicht C bleibt und wird, wie der anzuwendende Kernel, empirisch ermittelt. Wie auch bei der ursprünglichen SVM wird die Verschiebung b durch Einsetzen eines Belehrungspunktes ermittelt (vergleiche Gleichung (3.25) auf Seite 19). Dieser Datenpunkt muss zu den SV gehören und damit die Bedingung $\xi_i = 0$ aufweisen. Die Entscheidungsfunktion entspricht der Gleichung (3.27) (siehe Seite 20) und unterscheidet sich somit nicht von der ursprünglichen SVM mit komplett linear trennbaren Daten.

v-SVC

Die ν -SVC wurde von [SSWB00] vorgestellt und ersetzt das schwierig zu wählende Fehlergewicht C der C -SVC durch den natürlicheren Parameter $\nu \in (0,1]$, welcher die Obergrenze der Trainingsfehler und gleichzeitig die Untergrenze der SV darstellt [CL08][SS01]. Im Unterschied zur C -SVC

ergibt sich eine zusätzliche Bedingung zum Lösen der dualen Gleichung [SSWB00][CL01]. Die resultierende Entscheidungsfunktion ist wiederum identisch zu den vorherigen Methoden.

One-class SVM

Die One-class SVM ist eine natürliche Erweiterung der SVC auf Daten, die keiner Klasse zugeordnet sind (engl. unlabeled data) [SS01]. Das heißt, es gibt statt zwei oder mehreren Klassen nur noch eine Klasse. Einsatzgebiet ist unter anderem die Identifikation von unbekanntem Daten (engl. novelty detection) und die Quantil-Schätzung (engl. quantile estimation), d. h. „nicht überwacht“ Lernen (engl. unsupervised learning).

Support-Vektor-Regression (SVR)

Bei der SVR wird anhand des Merkmalsvektors \underline{x} die Klasse y geschätzt, dabei ist $y \in \mathbb{R}$ statt $y \in \{+1, -1\}$ im Falle der SVC [SS01][Wan05]. Dementsprechend wird die Trennebene zuvor bei der Belehrung in einem Raum, der zusätzlich zu den Merkmalsinformationen auch die Zielwerte y enthält, aufgespannt. Um diese Trennebene ist bei der SVR eine „Röhre“ (engl. tube) mit dem Radius ϵ als Genauigkeitsparameter (engl. accuracy parameter) definiert (siehe Abbildung 3-7). Alle Punkte in dieser Röhre sind normale Datenpunkte mit einem Verlust von null und folglich mit einer Schlupfvariablen von $\xi = 0$.

Mit anderen Worten, hier ist die Differenz zwischen vorhergesagtem Wert $f(x)$ und dem tatsächlichen Wert y kleiner als ϵ . Die Datenpunkte am Rand dieser Röhre sind freie Support-Vektoren (engl. free support vectors). Auch diese SV weisen einen Verlust von null und $\xi = 0$ auf. Die Punkte außerhalb der Röhre sind ebenfalls Support-Vektoren, die sogenannten beschränkten Support-Vektoren (engl. bounded support vectors). Jedoch verursachen diese einen Verlust, der je nach verwendeter Verlust-Funktion (z. B. Vapnik's ϵ -insensitivity loss function in Abbildung 3-8) bestimmt wird. Die Schlupfvariable ξ entspricht dem Abstand zwischen dem Datenpunkt und dem Rand der Röhre.

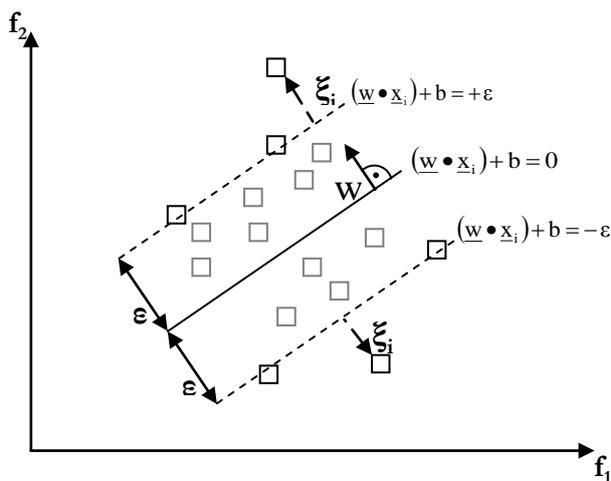


Abbildung 3-7: Bestimmung der Trennebenen bei nichtlinear trennbaren Daten mittels Support-Vektor-Regression [Wan05]

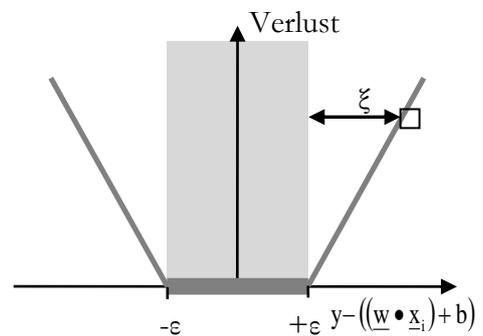


Abbildung 3-8: Verlauf der ϵ -insensitivity Verlust-Funktion [SS01]

ϵ -SVR

Die ϵ -SVR ist die in [Vap98] vorgestellte, ursprüngliche Variante der Regression. In der ϵ -SVR soll die Minimierung von $\|\underline{w}\|^2$ und des empirischen Risikos R_{emp} unter Anwendung der ϵ -insensitivity Verlust-Funktion erfolgen [SS04][Wan05]. Die Optimierung stellt sich aufwendiger als in der Klassifikation dar, da zwei Fälle (über und unter der Röhre) auftreten und doppelt so viele Lagrange-Multiplikatoren (α und α^*) bezogen auf die Anzahl der SV gefunden werden müssen. Weiterhin können das Fehlergewicht C sowie die Größe der Röhre ϵ nur empirisch ermittelt werden. Die resultierende Regression-Trennebene stellt keine Signum-Entscheidungsfunktion wie im Falle der Klassifikation dar.

 ν -SVR

Die ν -SVR, wie die ν -SVC ebenfalls in [SSWB00] vorgestellt, ersetzt den Genauigkeitsparameter ϵ durch den Parameter ν und ist eine Erweiterung der ϵ -SVR. Das vorher nur empirisch ermittelbare ϵ wird bei dieser Methode automatisch minimiert, dabei stellt (wenn $\epsilon \neq 0$ ist) ν zum einen die Obergrenze der Fehler und zugleich die Untergrenze der SV dar. Auch bei der ν -SVR wird die ϵ -insensitivity Verlust-Funktion angewendet.

Welche der beschriebenen Arten für die vorliegende Aufgabe genutzt werden sollte, konnte nicht aus theoretischen Überlegungen entnommen werden. Letztlich muss die SVM-Art an einem konkreten Problem empirisch auf ihre Eignung überprüft werden.

3.1.10 Abwandlungen der SVM

Abgesehen von der ursprünglichen SVM von Vapnik existieren bereits einige Abwandlungen. Diese verfolgen das Ziel, die Optimierung zu vereinfachen, jedoch die Vorteile der SVM beizubehalten. Da für die vorliegende Anwendung ein kürzerer Optimierungsprozess nicht zwingend notwendig ist, jedoch größten Wert auf die Klassifikationsleistung gelegt wird, stellen diese Abwandlungen keine Alternative zur klassischen SVM dar. Einige dieser Abwandlungen sind die Least Squares Support Vector Machine (LS-SVM), die Proximal Support Vector Machine (PSVM) und die Transductive Support Vector Machine (TSVM). Diese sollen im Folgenden diskutiert werden.

LS-SVM

Die LS-SVM wurde in [SV99] von Suykens und Vandewalle vorgestellt und in dem Buch [Suy02] beschrieben. Sie ist eine neue Darlegung der Standard-SVM. Die LS-SVM löst Gleichungsbedingungen statt Ungleichungsbedingungen. Dafür wird eine Error-Variable e_i eingeführt, welche bei überlappenden Daten Fehlklassifikationen erlaubt, ähnlich der Schlupfvariablen ξ_i der klassischen SVM. Für diese Error-Variable wird eine quadratische Verlustfunktion gewählt. Diese Änderungen führen zu einem Satz linearer Gleichungen anstatt einer quadratischen Programmierung und vereinfachen somit das Problem. Durch die Gleichungsbedingungen können nun die Lagrange-Multiplikatoren α_i sowohl positiv als auch negativ sein. Im Gegensatz zur klassischen SVM sind bei der LS-SVM typischerweise keine der Lagrange-Multiplikatoren null. Dies führt zu

dem Schluss, dass alle Belehrungsdatenpunkte je nach Lagrange-Multiplikator α_i in das Modell eingehen, aber mehr oder weniger wichtig sind. Damit vereinfacht sich bei der LS-SVM die Belehrung. Jedoch erhöht sich die Komplexität des Modells, wodurch die Einbindung auf eine ressourcenarme Plattform erschwert wird. Die Entscheidungsfunktion der LS-SVM entspricht der klassischen SVM.

Die experimentellen Untersuchungen verschiedener Quellen zeigen unterschiedliche Ergebnisse in der Klassifikationsleistung im Vergleich zur klassischen SVM. So untersuchten [VGBV⁺10] die Klassifikation von Kehlkopfbeschwerden anhand der menschlichen Stimme und verglichen dabei die klassische SVM mit der LS-SVM. In allen Tests, in denen sie unterschiedliche Datenverarbeitungsmethoden anwendeten, schnitt die LS-SVM schlechter ab. Sie fassten zusammen: „...The ordinary SVM ... was superior to the LS-SVM..., in all the tests...“ [VGBV⁺10]. In [YX07] jedoch ergibt sich ein anderes Bild. Hier werden verschiedene SVM-Methoden auf die Klassifikation von Leukämie (Zwei- bzw. Drei-Klassen-Problem) und auf eine Texterkennung (Vier-, Fünf- und Sieben-Klassen-Problem) angewendet. In fast allen Tests waren die Ergebnisse der klassischen SVM und der LS-SVM ähnlich. Aufgrund dieser Ergebnisse schrieben sie: „...LS-SVM and Hard M-SVM achieve the same classification performance ...“ und „...Soft SVM and rLS-SVM are comparable...“ [YX07].

PSVM

Die Proximal SVM führt laut [FM01] zu einem schnellen und einfachen Algorithmus. Ähnlich der LS-SVM werden die Ungleichungen der klassischen SVM durch Gleichungen ersetzt, was die Optimierung vereinfacht. Anstatt die Klassifikation der Vektoren nach ihrer Lage im Bezug auf eine Trennebene durchzuführen, klassifiziert die PSVM die Vektoren anhand der Zuordnung zur nächstgelegenen Ebene von zwei parallelen Ebenen. Abbildung 3-9 soll diesen Sachverhalt verdeutlichen. Die Ebenen stellen bei diesem Modell keine Grenzen mehr dar. Sie werden als Proximal-Ebenen bezeichnet, wobei der Abstand der Ebenen eine wichtige Eigenschaft des Modells ist. Sie werden so weit wie möglich auseinander geschoben.

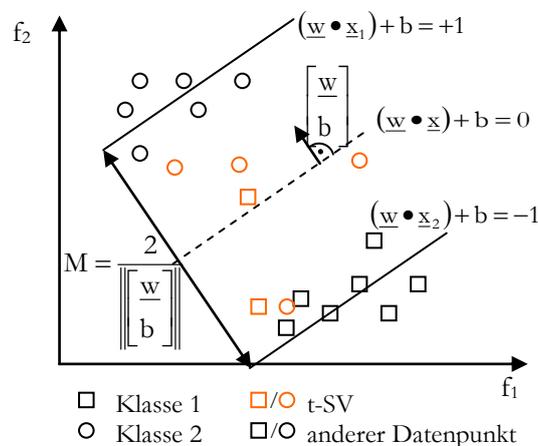


Abbildung 3-9: Funktionsweise der Proximal-SVM [FM01]

Da normalerweise keiner der Datenpunkte direkt auf den Proximal-Ebenen liegen und damit keiner der Lagrange-Multiplikatoren null ist, wird das Prinzip der SV abgewandelt. Es werden die t-SV¹⁶ eingeführt. Ein Datenpunkt ist dann ein t-SV, wenn die Error-Variable e_i (entspricht der Schlupfvariablen ξ) kleiner als t ist. Dabei wird t typischerweise so festgelegt, dass nur 1% der Datenpunkte zu einem t-SV werden. Die Versuche von [FM01] zeigen vergleichbare Ergebnisse zur klassischen SVM, aber mit einer kürzeren Berechnungszeit.

TSVM

Die Transduktive-SVM (TSVM) wurde von Vapnik in [Vap98] vorgestellt. Diese dient dem teilweise überwachten Lernen (engl. semi-supervised learning). Hierbei sind nur ein Teil der Daten durch einen Experten bewertet. Bei der TSVM wird versucht, den unbewerteten Daten eine Klasse zuzuordnen, sodass die Fehlklassifikationen einer Testdatenmenge minimiert werden. Die Optimierung dieser SVM gestaltet sich schwierig und bedarf spezieller Vorgehensweisen. So schlug Joachims in [Joa99] vor, beginnend mit einem kleinen Fehlergewicht C , teilweise den unbewerteten Daten Klassen zuzuordnen. Anschließend wird nach und nach das Fehlergewicht C erhöht und so weiteren unbewerteten Daten eine Klasse zugeordnet. Für die Texterkennung konnte [Joa99] eine gute Klassifikationsleistung bestätigen. Diese Art der SVM ist als teilweise überwachtetes Lernen sehr speziell und für das vorliegende Problem nicht zutreffend.

3.1.11 Mehr-Klassen-Klassifikation mit SVM

Die SVM wurde ursprünglich für Zwei-Klassen-Probleme entworfen. Die Anwendung der SVM auf Mehr-Klassen-Probleme erfordert daher besondere Herangehensweisen. Bei der Mehr-Klassen-Klassifikation mit SVM unterscheidet man zwei Methoden. Zum einen kann das Mehr-Klassen-Modell durch Lösung mehrerer Zwei-Klassen-Modelle erzeugt werden. Zum anderen gibt es die sogenannten All-Together-Methoden, bei denen nur eine Optimierungsgleichung zu lösen ist. Die Abbildung 3-10 zeigt die Zuordnung der wichtigsten Methoden.

Zu den Methoden der Mehr-Klassen-Klassifikation gehören die One-against-All [SS01][HL02][MCS06], die One-against-One [SS01][HL02][MCS06], die Directed Acyclic Graph-SVM [HL02] und die Error-Correcting Output Coding-Methode [DB95] als binäre Methoden, außerdem die All-Together-Methoden von Vapnik [WW99], Crammer und Singer [CS01] und Lee, Lin und Wahba [LLW04]. Diese werden im Folgenden kurz vorgestellt und bezüglich ihrer Vor- und Nachteile diskutiert.

¹⁶ in [FM01] als ϵ -support vector bezeichnet

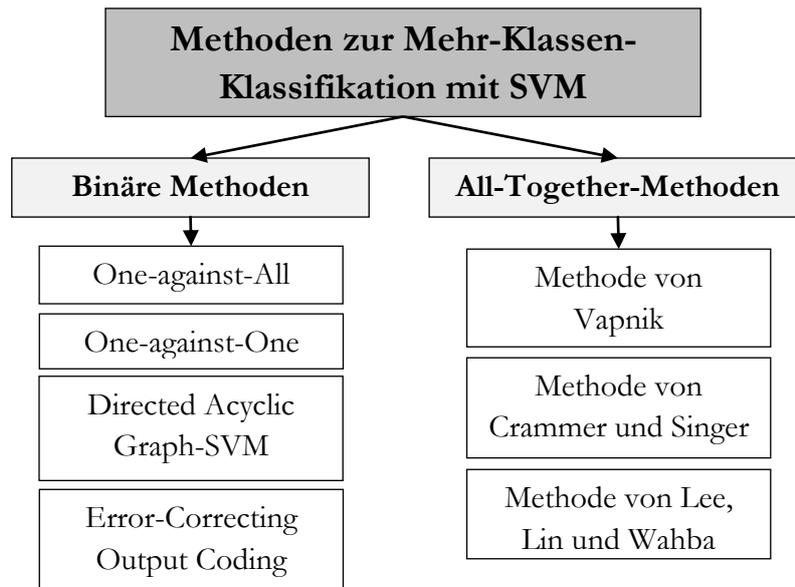


Abbildung 3-10: Auflistung der Methoden zur Mehr-Klassen-Klassifikation mit SVM

One-against-All

Die One-against-All-Methode (auch One-versus-All oder One-versus-the-Rest genannt) ermittelt das Ergebnis aufgrund mehrerer binärer Entscheidungen [SS01][HL02][MCS06]. Jede Entscheidung erfolgt durch Betrachtung einer Klasse gegen alle anderen Klassen. Damit ergibt sich für die Anzahl der benötigten Trennebenen Z :

$$Z = U, \tag{3.41}$$

wobei U die Anzahl der Klassen darstellt. Die Lösung eines Drei-Klassen-Problems mit dieser Methode ist in Abbildung 3-11 gezeigt. Zur Anwendung kommt die „winner takes all“-Strategie, bei der sich die Klasse durchsetzt, die einmalig beim Vergleich Klasse gegen alle anderen gewinnt. Der Vorteil dieser Methode ist die überschaubare Anzahl an zu lösenden binären Entscheidungen. Nachteil ist ein relativ großer Bereich (Grauzone), in dem eine genaue Zuordnung zu einer Klasse durch das Modell nicht möglich ist. In diesem Fall erfolgt die Zuordnung zu einer Klasse zufällig.

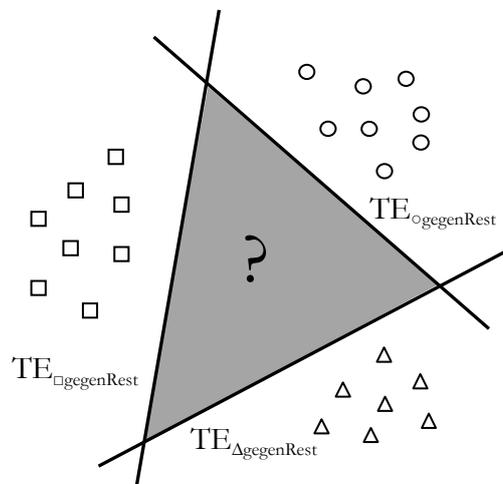


Abbildung 3-11: One-against-All-Methode [Mol03b]

One-against-One

Bei der One-against-One-Methode (auch One-versus-One genannt) führen binäre Entscheidungen zwischen allen Klassen untereinander zum Ergebnis [SS01][HL02]. Damit ergibt sich die Anzahl der Trennebenen Z aus der Gleichung:

$$Z = \frac{U(U-1)}{2}, \quad (3.42)$$

wobei U die Anzahl der Klassen darstellt.

Die Lösung des gleichen Drei-Klassen-Problems aus Abbildung 3-11 erfolgt damit wie in Abbildung 3-12 dargestellt. Hierbei findet die „max wins“-Strategie Anwendung. Es wird folglich die Klasse gewählt, welche die meisten Entscheidungen für sich entschieden hat. Vorteil ist die geringere Grauzone. Nachteil ist eine erhöhte Anzahl an Trennebenen und damit eine verlängerte Belehrungs- und Klassifikationszeit. Diese Methode erwies sich bereits als brauchbarer Ansatz für praktische Anwendungen [MCS06].

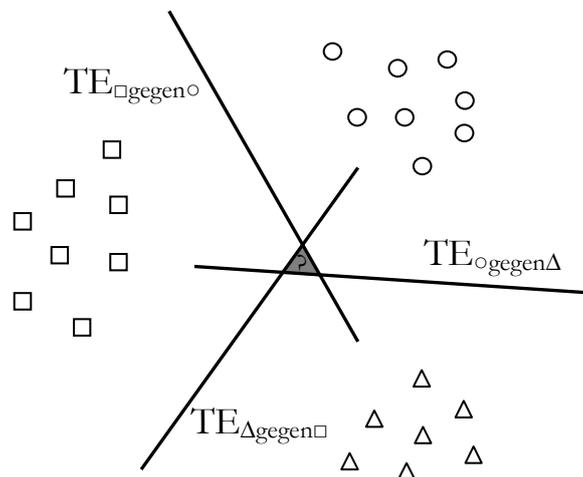


Abbildung 3-12: One-against-One-Methode [Mol03b]

Directed Acyclic Graph-SVM

Die Directed Acyclic Graph-SVM entspricht in der Belehrungsphase der One-against-One-Methode, d. h. es werden Modelle von allen Klassen gegeneinander erzeugt. In der Klassifikationsphase werden im Gegensatz zu One-against-One nicht alle Trennebenen des Modells genutzt [HL02]. Dieses Prinzip ist in Abbildung 3-13 am Beispiel einer Vier-Klassen-Klassifikation verdeutlicht. Der gerichtete Graph besteht aus einer Anzahl an Knoten O , welche sich durch die Gleichung:

$$O = \frac{U(U-1)}{2} \quad (3.43)$$

bestimmen lässt, wobei U die Anzahl der Klassen darstellt. Die endgültige Entscheidung ergibt sich aus dem Durchlaufen eines gerichteten Graphs. Wurde sich einmal gegen eine Klasse ent-

schieden, bleiben alle weiteren Trennebenen dieser Klasse unberücksichtigt. Damit sind bei Validierung und Klassifikation

$$Z = U - 1 \tag{3.44}$$

Trennebenen auszuwerten. Vorteil dieser Methode ist eine verkürzte Klassifikationszeit, jedoch bedingt eine einzige Fehlentscheidung Folgefehler, wodurch die Fehlerwahrscheinlichkeit steigt.

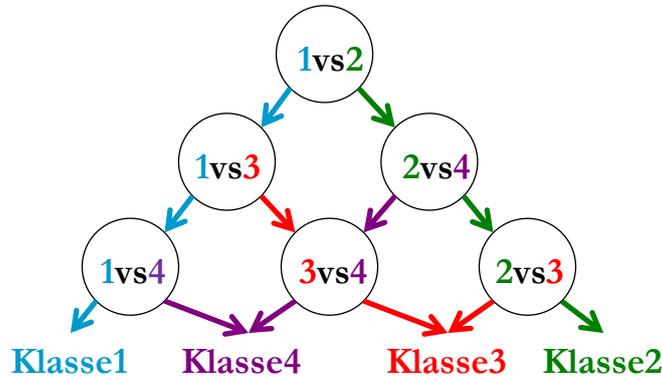


Abbildung 3-13: Directed Acyclic Graph-Methode [Mol03b]

Error-Correcting Output Coding-Methode

Die Error-Correcting Output Coding-Methode ordnet jeder Klasse ein N-bit großes binäres Code-Wort zu [DB95]. Dieses Code-Wort ergibt sich aus N binären Klassifikationen. Damit entsteht eine Matrix der Größe N x U. Diese Methode ist dazu in der Lage, einzelne Fehlentscheidungen der binären Klassifikatoren auszugleichen, da die Klasse gewählt wird, deren Code dem resultierenden Code am ähnlichsten ist. Die Anzahl der ausgleichbaren Fehler hängt von der Hamming-Distanz HD ab, d. h. von der minimalen Anzahl der sich unterscheidenden Bits aller Klassen untereinander. Die ausgleichbaren Fehler EC berechnen sich durch:

$$EC = \frac{HD-1}{2} \tag{3.45}$$

Voraussetzung dafür ist die Unabhängigkeit der Bitpositionen von allen anderen. Die Error-Correcting-Output-Coding-Methode ist für überlappende oder zu kleine Datensätze ungeeignet, da die Code-Worte jeder Klasse sich ausreichend unterscheiden müssen, um einen Fehlerausgleich bewirken zu können.

Methode von Vapnik

Die Methode von Vapnik [Vap98][WW99] basiert auf der One-against-All-Methode. Im Gegensatz zu dieser werden nicht mehrere binäre Optimierungsprobleme gelöst, sondern nur ein einziges Optimierungsproblem. Die Optimierung der Vapnik Methode gestaltet sich mit (U-1)·N Variablen aufwendiger als die der binären Methoden mit N Variablen. In der Validierung bzw. Klassifikation ist die Methode von Vapnik laut experimenteller Ergebnisse der One-against-All-Methode gleichwertig [HL02][ZT07].

Methode von Crammer und Singer

Die Crammer und Singer-Methode [CS01] ist ähnlich der Methode von Vapnik. Das Optimierungsproblem verwendet weniger Schlupfvariablen ξ und die Entscheidungsfunktion enthält nicht mehr die Verschiebung b . Das quadratische Problem ist mit $U \cdot N$ Variablen bei dieser Methode größer als bei der Methode von Vapnik mit $(U-1) \cdot N$ Variablen. Aus diesem Grund sieht [CS01] die Einteilung des einen großen quadratischen Problems in mehrere kleine quadratische Probleme vor. So kann der Berechnungsaufwand erheblich reduziert werden [ZT07].

Methode von Lee, Lin und Wahba

Die Methode von Lee, Lin und Wahba wurde in [LLW04] vorgestellt. Sie geht davon aus, dass die Lösung von Mehr-Klassen-Problemen den Bayes-Entscheidungsregeln ähnlich ist. Wie auch bei Vapnik's Methode besteht das quadratische Problem aus $(U-1)N$ Variablen [ZT07].

Die Wahl der anzuwendenden Methode basiert auf beschriebenen experimentellen Ergebnissen der Literatur und theoretischen Überlegungen. So wäre wohl von den binären Methoden die One-against-One Methode zu bevorzugen. Zum einen, weil diese durch die hohe Anzahl an Trennebenen zwar längere Belehrungs- und Klassifikationszeiten als die One-against-All aufweist, dafür aber die bessere Klassifikation durch einen geringen Graubereich liefert. Zum anderen, weil die Directed-Acyclic-Graph-Methode eine erhöhte Fehlerwahrscheinlichkeit durch Fehlerfortpflanzung aufweist und die Error-Correcting-Output-Coding-Methode für überlappende Daten, wie sie bei biologischen Signalen auftreten, nicht geeignet ist. Die All-Together-Methoden sind bei der Lösung des quadratischen Problems aufwendiger als die binären Methoden, deshalb werden binäre Methoden oft bevorzugt angewendet. In dem zu untersuchenden Anwendungsfall der Klassifikation mittels einer belehrten SVM ist die Belehrungszeit zweitrangig. Deshalb ist die viel bedeutendere Frage, welche Methode die „beste“ SVM liefern könnte. Die aufwendigste der All-Together-Methoden, die Methode von Crammer und Singer, wurde in [Wan05] auf die Vorhersage von Proteinen in Desoxyribonukleinsäure (DNS) mit der One-against-One Methode verglichen und geschlussfolgert: „... we found that the multiclass SVMs using Crammer and Singer's scheme [60] and the group of binary SVMs using „one-against-one“ scheme [59] obtained similar results” [Wan05]. In [ZT07] wurden die Methoden One-against-All, Vapnik, Crammer und Singer sowie Lee, Lin und Wahba auf das Problem der Genselektion in DNA angewendet und festgestellt, dass „MSVM-RFE (Lee, Lin und Wahba) is not as good as other three algorithms“. Weiterhin lieferten die One-against-All- und die Vapnik-Methode in diesem Anwendungsfall ähnliche Ergebnisse. Die Crammer und Singer-Methode konnte abhängig von der Anzahl der ausgewählten Gene vergleichbare bis bessere Resultate als die One-against-All und Vapnik-Methode erzielen [ZT07]. Dies zeigt, dass die bisher vorgestellten All-Together-Methoden zwar einen erhöhten Optimierungsaufwand haben, aber wahrscheinlich keine deutliche Verbesserung der Klassifikationsgüte erreichen werden. Aufgrund dessen wurde für alle experimentellen Untersuchungen die One-against-One-Methode ausgewählt.

3.1.12 Kombination verschiedener SVM

Zur Umsetzung der Kombination mehrerer Klassifikatoren existiert eine Reihe von Ansätzen. Drei dieser Ansätze werden im Folgenden kurz vorgestellt:

- das „Boosting“,
- das „Bagging“ und
- das „Deep Learning“.

Boosting

Das „Boosting“ (dt. Verstärken) ist eine von Robert Schapire in [Sch90] 1990 erstmals beschriebene Methode, wobei aus mehreren „schwachen“ Klassifikatoren ein guter Klassifikator entstehen soll. Die „schwachen“ Klassifikatoren, oder auch Basisklassifikatoren genannt, berücksichtigen meist nur ein Merkmal. Aus diesem Grund weisen diese Klassifikatoren einzeln für sich nur geringe Klassifikationsgüten auf, sind aber dafür sehr schnell ausgewertet. Die „schwachen“ Klassifikatoren werden mittels Gewichtung zu einem Klassifikator zusammengeführt. Das Finden der Gewichte, die zu einem guten Klassifikator führen, stellt nun wiederum eine Optimierungsaufgabe dar.

Bagging

Das Wort „Bagging“ ist vom englischen „bootstrap aggregating“ abgeleitet und wurde 1996 von Leo Breiman in [Bre96] erstmals beschrieben. Bei dieser Methode werden die resultierenden Klassen mehrerer Klassifikatoren unter Anwendung einer geeigneten Variante zu einer Klasse kombiniert. Im einfachsten Fall, dem Mehrheitsentscheid, resultiert die Klasse, die von den meisten Klassifikatoren bestimmt wurde. Die Wirkung dieser Methode ist von der Stabilität des Klassifikators abhängig. Je mehr sich eine kleine Änderung in der Belehrungsdatenmenge auf den Klassifikator auswirkt, desto instabiler ist der Klassifikator und desto höher ist die Wirkung der Methode.

Deep Learning

Das „Deep Learning“ kombiniert mehrere nichtlineare Verarbeitungseinheiten in einem Netz aus mehreren Schichten, wobei die Ausgänge der vorherigen Schicht die Eingänge der nachfolgenden Schicht darstellen [Gol16]. Damit ähnelt das Deep Learning den neuronalen Netzen. Ein wesentlicher Unterschied liegt in der Anzahl der Schichten. Wo neuronale Netze häufig nur eine verdeckte Schicht aufweisen, haben Deep Learning-Algorithmen viele Schichten. Die Menge der Schichten bedingt aber auch den entscheidenden Nachteil der Deep Learning-Methode. Denn die vielen Schichten, welche das Deep Learning ausmachen, erfordern eine große Datenmenge, um eine ausreichende Belehrung vornehmen zu können. Dementsprechend ist eine hohe Rechenkapazität notwendig. Aus diesem Grund wurde diese Methode, obwohl schon länger bekannt, erst in den letzten Jahren wieder aufgegriffen, da nun durch leistungsstärkere Computer die Rechenintensität handhabbar sowie die Akquisition und Speicherung großer Datenmengen möglich geworden ist. Die Methode wird vor allem zur Bild- und Texterkennung eingesetzt, da Bilder und Texte besonders preiswert gesammelt und aufbereitet werden können. Unter dem Begriff „deep

support vector machine“ (DSVM) oder auch Multi-Layer SVM wird der Ansatz des Deep Learning in Kombination mit der SVM verstanden [SSA15][WSMM⁺13]. Prinzipiell wäre die Entwicklung eines Deep Learning-Verfahrens zur Klassifikation von biologischen Signalen unter der Voraussetzung, dass eine große Menge an Daten existiert, möglich. Die Anwendung des Deep Learning auf die vorliegenden EEG-Daten scheint jedoch nicht sinnvoll, da die zur Verfügung stehende Datenmenge begrenzt ist und sich durch die starke Individualität der in dieser Arbeit verwendeten EEG-Daten der Datenbedarf noch zusätzlich erhöht.

3.1.13 Bewertung des Klassifikationsergebnisses

Das Ergebnis der Klassifikation ist die Vertauschungsmatrix \mathbf{V} mit U für die Anzahl der Klassen:

$$\mathbf{V} = \begin{pmatrix} v_{11} & v_{12} & \dots & v_{1U} \\ v_{21} & v_{22} & \dots & v_{2U} \\ \vdots & \vdots & \ddots & \vdots \\ v_{U1} & v_{U2} & \dots & v_{UU} \end{pmatrix}. \quad (3.46)$$

Sie stellt die absoluten Häufigkeiten der klassifizierten Daten v_{ij} nach Soll- und Ist-Klassifikation dar, wie in Tabelle 3-2 veranschaulicht.

Soll \ Ist	Klasse 1	Klasse 2	...	Klasse U
Klasse 1	v_{11}	v_{12}	v_{1j}	v_{1U}
Klasse 2	v_{21}	v_{22}	v_{2j}	v_{2U}
...	v_{i1}	v_{i2}	v_{ij}	v_{iU}
Klasse U	v_{U1}	v_{U2}	v_{Uj}	v_{UU}

Tabelle 3-2: Vertauschungsmatrix, wobei v_{ij} die absoluten Häufigkeiten der klassifizierten Daten darstellen mit i als Soll-Klasse und j als Ist-Klasse

Aus der Vertauschungsmatrix berechnete Gütewerte sollen Aufschluss über die Qualität der Klassifikation geben und die Klassifikationsergebnisse vergleichbar machen. Die nächstliegende Assoziation der Güte¹⁷ eines Klassifikators ist wohl die relativen Häufigkeiten der richtig klassifizierten Validierungsdaten aller Klassen. Tatsächlich setzt sich die Güte eines Klassifikators aus mehreren Teilaspekten zusammen. Für die in dieser Arbeit betrachtete Klassifikationsaufgabe ist das gleichmäßige Klassifizieren aller Beispiele, auch der schwach vertretenen Klassen, für die Güte des Klassifikators von großer Bedeutung. Aus diesem Grund wird die Güte eines Klassifikators ebenso mit der relativen Häufigkeit der richtig klassifizierten Validierungsdaten der am schlechtesten klassifizierten Klassen, welche meist die in der Belehrungsmenge schwach vertretenen Klassen sind, assoziiert. Ein weiterer Teilaspekt der Güte eines Klassifikators ist die Reduzierung bestimmter Vertauschungen. Die Anwendung des Klassifikators erfolgt auf Validierungsdaten, die von den Belehrungsdaten völlig unabhängig sind. Damit ist das Auftreten von nicht belehrten Beispielen garantiert. Der Klassifikator sollte auch in der Lage sein, solche Bei-

¹⁷ siehe Begriffsverzeichnis

spiele richtig zu klassifizieren. Dabei spricht man von der Generalisierungsfähigkeit¹⁸ eines Klassifikators. Die Generalisierungsfähigkeit ist damit auch ein Bestandteil der Güte eines Klassifikators und Voraussetzung für einen robusten Klassifikator¹⁹. Ein Klassifikator gilt als robust, wenn dieser für eine Vielzahl von unterschiedlichen Probanden und ungeachtet der Unsicherheiten der Experten den gestellten Güteanforderungen genügt [Wen05].

Die aus der Vertauschungsmatrix bestimmbaren Gütewerte wurden bereits in [BS96], [Wen05] und [Wal12] vorgestellt und angewendet. Die sechs benutzten Gütewerte sind in den folgenden Gleichungen beschrieben.

1. Die Gesamt-Klassifikationsgüte (GG)

$$GG = 100\% \cdot \frac{\sum_{i=1}^U v_{ii}}{\sum_{i=1}^U \sum_{j=1}^U v_{ij}} = 100\% \cdot \frac{\sum_{i=1}^U r_i}{\sum_{i=1}^U g_i} = 100\% \cdot \frac{\sum_{i=1}^U r_i}{N} \quad (3.47)$$

wobei U der Anzahl der Klassen und v_{ij} der absoluten Häufigkeiten der klassifizierten Epochen entspricht. Dabei ist i die Soll-Klasse und j die Ist-Klasse. Somit stellt die GG das Verhältnis der Summe der absoluten Häufigkeiten der richtig klassifizierten Daten r_i der Klassen i zur Gesamtanzahl der Daten N in Prozent dar, wobei N die Summe aus den absoluten Häufigkeiten der insgesamt zu klassifizierenden Daten g_i der Klassen i bildet.

2. Die Durchschnittliche Klassen-Klassifikationsgüte (DG)

$$DG = 100\% \cdot \frac{1}{U} \sum_{i=1}^U \frac{v_{ii}}{\sum_{j=1}^U v_{ij}} = 100\% \cdot \frac{1}{U} \sum_{i=1}^U \frac{r_i}{g_i} \quad (3.48)$$

entspricht der durchschnittlich richtig klassifizierten Daten aller Klassen in Prozent.

3. Die Geometrisch-Mittlere Klassen-Klassifikationsgüte (GMG)

$$GMG = 100\% \cdot \sqrt[U]{\prod_{i=1}^U \frac{v_{ii}}{\sum_{j=1}^U v_{ij}}} = 100\% \cdot \sqrt[U]{\prod_{i=1}^U \frac{r_i}{g_i}} \quad (3.49)$$

gibt das geometrische Mittel der Klassen-Klassifikationsgüten wieder. Auf dieses Maß üben die schlecht klassifizierten Klassen einen hohen Einfluss aus. Wurden alle Klassen gleichmäßig gut erkannt, entspricht der GMG dem DG.

4. Die Minimale Klassen-Klassifikationsgüte (MG)

$$MG = 100\% \cdot \min_{v_i} \frac{v_{ii}}{\sum_{j=1}^U v_{ij}} = 100\% \cdot \min_{v_i} \frac{r_i}{g_i} \quad (3.50)$$

ist der Anteil der richtig klassifizierten Daten r_i der schlechtesten Klasse in Prozent.

¹⁸ siehe Begriffsverzeichnis

¹⁹ siehe Begriffsverzeichnis

5. Die gewichtete Güte G_{gew}

$$G_{\text{gew}} = 0,6 \cdot GG + 0,2 \cdot DG + 0,1 \cdot GMG + 0,1 \cdot MG \quad (3.51)$$

ist eine Gewichtung der vier vorherigen Gütewerte, wobei die GG das größte Gewicht und die Gütewerte der schlecht klassifizierten Klassen (GMG und MG) geringere Gewichte erhalten. Somit spiegelt die gewichtete Güte G_{gew} die für die Bewertung wichtigen Aspekte der Generalisierungsfähigkeit wider. Das Verhältnis der Gewichte in Gleichung (3.51) wurde in zahlreichen experimentellen Versuchen als angemessen zur Bewertung der Übereinstimmung von zwei Profilen ermittelt. Hierbei stand die Verwertbarkeit der automatisch erzeugten Hypnogramme für diagnostische Aussagen im Fokus. Die gewählte Gewichtung repräsentiert die notwendigen Teilaspekte sehr gut und wurde zusammen mit den verwendeten Daten bereits in vielen Versuchen sehr erfolgreich eingesetzt, was eine Vergleichbarkeit der Ergebnisse sicherstellt [Wen05].

6. Der gewichtete Kappa-Koeffizient (κ_w) [Wen05][BLB00]

$$\kappa_w = \frac{p_{0w} - p_{ew}}{1 - p_{ew}} \quad (3.52)$$

ergibt sich aus den gewichteten relativen Häufigkeiten der richtig klassifizierten Daten p_{0w} und der gewichteten relativen Häufigkeiten der zufällig richtig klassifizierten Daten p_{ew} , welche mittels einer Bewertungsmatrix **BM** problemspezifisch gewichtet werden. Somit können Schweregrade der Vertauschungen durch dieses Gütekriterium berücksichtigt werden.

Die Bewertungsmatrix setzt sich aus den Gewichten bm_{ij} zusammen nach

$$\mathbf{BM} = \begin{pmatrix} 1 & bm_{12} & \dots & bm_{1U} \\ bm_{21} & 1 & \dots & bm_{2U} \\ \vdots & \vdots & \ddots & \vdots \\ bm_{U1} & bm_{U2} & \dots & 1 \end{pmatrix} \quad bm_{ij} \in [0,1], \quad (3.53)$$

wobei kleine Werte für bm_{ij} für sehr unähnliche Klassen und hohe Werte für ähnliche Klassen stehen. Die gewichteten relativen Häufigkeiten ergeben sich aus

$$p_{0w} = \sum_{i=1}^U \sum_{j=1}^U bm_{ij} \cdot vrel_{ij} \quad (3.54)$$

und

$$p_{ew} = \sum_{i=1}^U \sum_{j=1}^U \left(bm_{ij} \sum_{l=1}^U vrel_{il} \sum_{l=1}^U vrel_{lj} \right) \quad (3.55)$$

wobei $vrel_{ij}$ die relativen Häufigkeiten der klassifizierten Epochen v_{ij} sind. Diese sind definiert als

$$vrel_{ij} = \frac{v_{ij}}{N} \quad (3.56)$$

mit N als Gesamtanzahl der Daten. Das Gütekriterium des gewichteten Kappa-Koeffizienten wurde von [BLB00] übernommen, die zugehörige Bewertungsmatrix von [Wen05]. Der Wert des gewichteten Kappa-Koeffizienten ist Maximal $\kappa=1$, wenn nur konkordante Klassifikationen auftreten. Wenn die Klassifikation der Zufallswahrscheinlichkeit entspricht, nimmt

der gewichtete Kappa-Koeffizient den Wert $\kappa=0$ an. Ein Wert von $\kappa=-1$ bedeutet, dass keine konkordanten Klassifikationen auftreten, obwohl die Wahrscheinlichkeit dafür maximal ist. Die Klassifikation ist schlechter als der Zufall.

3.2 Entwicklung und aktueller Stand der SVM-Forschung

Die Entwicklung der Support-Vektor-Maschinen ist in [CS00], [Sew10a] und [Sew10b] aufgezeigt. Sie beginnt bereits 1936 als Fischer in [Fis36] den ersten Algorithmus zur Mustererkennung vorzuschlag, der auf die Taxonomie von Schwertlilienarten angewendet wurde. Als 1957 Frank Rosenblatt das „Perzeptron“ vorstellte [Ros58], welches die Basis aller neuronaler Netze ist, löste dies eine regelrechte Euphorie in der Entwicklung von Klassifikatoren aus.

Der erste Schritt zur SVM wurde 1963 von Vapnik und Lerner gemacht, als sie den „generalized portrait algorithm“ veröffentlichten [VL63], welcher Grundlage für die SVM-Algorithmen ist. Diesen entwickelten Vapnik und Chervonenkis 1964 weiter und beschrieben dies in „A note on one class of perceptrons“ [VC64]. Der Kernel wurde zwar bereits seit Beginn des 20. Jahrhunderts angewendet, doch erst der Artikel [ABR64], in dem Aizerman, Braverman und Rozonoer den Kernel als Skalarprodukt im Merkmalsraum (feature space) vorstellten, bildete die Basis für den „Kernel-Trick“ zur linearen Klassifikation nichtlinearer Probleme. Ein Kernel ist eine Funktion im niederdimensionalen Raum, die sich ähnlich wie das Skalarprodukt im höherdimensionalen Raum verhält, so muss der höherdimensionale Raum nicht bekannt sein. 1965 wurde durch Cover das sogenannte „Theorem von Cover“ aufgestellt, welches die lineare Separierung nichtlinearer Daten durch Erhöhung der Dimension beschreibt [Cov65]. Die Verwendung von „Schlupfvariablen“ (slack variables) wurde 1968 von Smith vorgeschlagen, um das Problem überlappender Daten zu lösen [Smi68].

„Theory of pattern recognition“ von Vapnik und Chervonenkis [VC74] erschien 1974 in Russland und wurde 1979 zunächst in die deutsche Sprache [VC79], anschließend 1982 in die englische Sprache [VC82] übersetzt. Man kann sagen, dass die Entwicklung der SVM erst zu dieser Zeit mit den Arbeiten von Vapnik richtig begann. Aus diesem Grund wird Vapnik als Begründer der SVM angesehen.

Ein weiterer bedeutender Schritt in der Entwicklung der SVM geschah 1990, als Poggio, Girosi und Wahba Kernel zum Belehren von Netzen verwendeten [PG90]. Heute werden Kernel bei Support-Vektor-Maschinen eingesetzt, deshalb wird die SVM auch als „Kernel-basierte“ Lernmethode bezeichnet. Zu ihrer heutigen Form gelangten Support-Vektor-Maschinen 1992, als Boser, Guyon und Vapnik bei der COLT-Konferenz ihren Algorithmus vorstellten, der die Spannweite (engl. margin), die Differenz zwischen den Stützvektoren, maximiert und die Komplexität des Problems automatisch anpasst [BGV92]. So entstand die SVM erstmals durch Zusammenführung der zuvor entwickelten Methoden. 1995 wurde von Cortes und Vapnik noch eine weitere Methode zur Klassifikation mit SVM vorgestellt, der „Soft-Margin“-Klassifikator. Diese Methode lässt bei der Ermittlung der Spannweite Fehler zu, d. h. es müssen nicht alle Belegungsdaten getrennt werden [CV95]. Somit können auch überlappende Daten klassifiziert werden.

Ein bedeutendes Werk "The Nature of Statistical Learning Theory", ebenfalls von Vapnik, erschien 1995 [Vap95]. Es diskutiert die fundamentalen Ideen, die hinter der statistischen Lerntheorie stehen. Weiterhin erweitert es den SVM-Algorithmus um die SVM-Regression. Das 1998 von Vapnik erschienene Buch „Statistical Learning Theory“ stellt eines der bedeutendsten Werke dar. Von den ursprünglichen Zwei-Klassen-Problemen konnte die SVM 1998 in diesem Buch [Vap98] und 1999 in [WW99] auch auf die Anwendung auf Mehr-Klassen-Probleme ergänzt werden.

Mittlerweile ist das Thema Klassifikation mit SVM ein populäres Forschungsgebiet auf der ganzen Welt. So entstanden und entstehen eine Vielzahl von Abwandlungen und Kombinationen der SVM mit anderen Methoden, u. a. die „Least Square SVM“, die „Maxi-Min Margin Machine“ oder die „Transductive SVM“.

Suykens ist ein bekannter Wissenschaftler, der viele Artikel auf dem Gebiet der Least-Square-SVM sowie das Buch „Least Square Support Vector Machines“ 2002 veröffentlichte [Suy02]. Auch Bernhard Schölkopf gehört zu den bekannten Wissenschaftlern der heutigen Zeit. Er veröffentlichte 2002 ein Buch "Learning with Kernels" [SS01] und ist auf dem Gebiet der Support-Vektor-Regression sehr aktiv.

Heute gilt die SVM als eine Standardmethode zur Klassifikation, die in vielen Büchern über Maschinelles Lernen behandelt wird [LR15]. Sie wird für die verschiedensten Anwendungen genutzt [MG14], wie in der Bildverarbeitung zur Gesichtserkennung [Sri15] oder Diagnostik von bildgebenden Verfahren [KS15]. Auch auf EEG-Daten fand die SVM bereits Anwendung. Unter dem Thema „Gehirn-Computer-Interface“ (engl. Brain Computer Interface – BCI) soll aus dem EEG Intentionen für Muskelaktivitäten herausgelesen werden, was anschließend zur Steuerung verschiedenster Vorgänge genutzt werden kann [CCT10] [MCE15]. Unter dem Thema der Emotionserkennung (Emotion Recognition) sollen anhand der EEG-Daten emotionale Zustände, wie fröhlich oder traurig, erkannt werden [LLGC15]. Auch für die Schlafklassifikation mit SVM lassen sich wissenschaftliche Arbeiten finden. Dabei existieren Arbeiten, die sich auf die Erkennung einzelner EEG-Ereignisse, wie Augenbewegungen, konzentrieren [MTSMS15]. Aber auch Schlafstadienklassifikationen wurden bereits mit Hilfe der SVM durchgeführt [AF14][KD12]. Jedoch beschränkte sich die Klassifikation meist auf vier oder fünf Schlafstadien, indem Schlafstadien zusammengefasst oder zuvor herausgefiltert wurden. So wurden in [AF14] die Klassen REM, Movement und UI aus der Klassifikation ausgeschlossen. Weiterhin wurden nur wenige Daten zur Validierung benutzt, wobei unklar ist, ob diese auch wirklich unabhängig von den Belehrungsdaten waren. Es wurden keine Angaben zu der Schlafqualität der Probanden gemacht, wodurch anzunehmen ist, dass es sich um gesunde Probanden handelte. Die in [KD12] vorgestellte Klassifikation beschränkte sich ebenfalls auf fünf Klassen, wobei die Klassen Movement und UI herausgenommen und die Stadien S3 und S4 zusammengefasst wurden. Weiterhin wurde die Klassifikation, bezogen auf nur einen Experten, bewertet, wodurch das Ergebnis erheblich besser erscheint. Der Vergleich mit den in dieser Arbeit vorgestellten Resultaten ist somit nicht möglich.

4 Realisierung einer Support-Vektor-Maschine zur robusten nichtlinearen Klassifikation biologischer Daten

4.1 Einfluss der Belehrungsdatenverteilung auf die SVM

Wie bereits in Abschnitt 1.1 Umfang und Zielstellung ab Seite 2 dargestellt, weisen biologische Daten, und auch speziell die verwendeten Daten zur Schlafklassifikation, Eigenschaften auf, die eine optimale Modellbildung erschweren. Die von Natur aus stark ungleichverteilten Daten sind zusätzlich stark inhomogen und durch inter- und intraindividuelle Ausprägungen gekennzeichnet. Aufgrund dessen sind abweichende Expertenbewertungen wahrscheinlich. Sie beinhalten somit auch für ihre Klasse untypische Daten. Inwieweit sich diese Eigenschaften auf die Modelle der SVM auswirken, soll im Folgenden an einem synthetischen Problem anschaulich gezeigt werden.

4.1.1 Wirkung der Gleichverteilung und der untypischen Daten auf das SVM-Modell

Viele Klassifikationsmethoden, wie zum Beispiel die neuronalen Netze, setzen für eine akzeptable Güte eine Gleichverteilung der Belehrungsdaten voraus [Wen05]. Verschiedene Literaturquellen geben an, dass die SVM jedoch auch mit ungleichverteilten Daten umgehen kann [Wan05]. So schreibt Akbani in [AKJ04], dass die SVM bei leicht bis mittelgradig ungleichverteilten Daten gute Ergebnisse liefert. Erst bei hochgradig ungleichverteilten Daten, d. h. bei einem Verhältnis von rund 1:50, dominiert die Majoritätsklasse. Dieser Sachverhalt kann theoretisch belegt werden und wird als „Schwäche der Soft-Margin“ bezeichnet. Durch das Fehlergewicht C werden bei beiden Klassen Fehler zugunsten einer größeren Spannweite M zugelassen [BP12]. Liegt nun ein starkes Ungleichgewicht in der Datenmenge beider Klassen vor, führt eine ähnliche Fehleranzahl bei beiden Klassen dazu, dass die Trennebene (TE) unweigerlich in Richtung der Minoritätsklasse verschoben wird (siehe Abbildung 4-1).

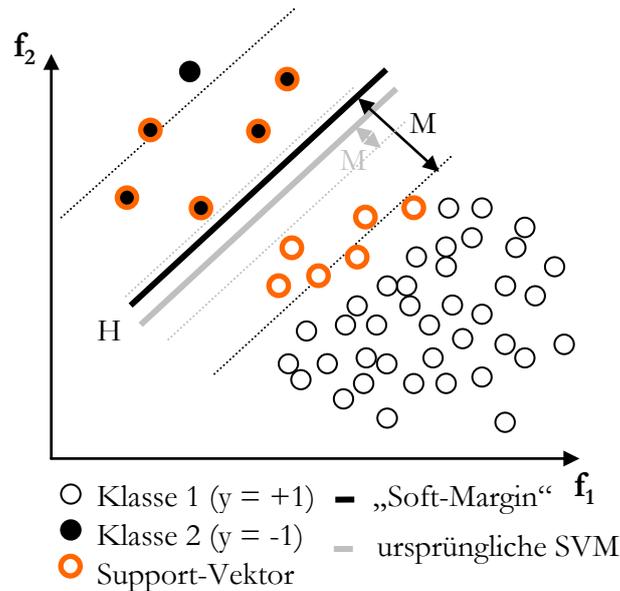


Abbildung 4-1: „Soft-Margin“ (schwarz) und ursprüngliche SVM (grau) eines Zwei-Klassen-Problems von linear trennbaren Daten [AKJ04]

Dem entgegen wirkt der Umstand, dass die Summe der Lagrange-Multiplikatoren α beider Klassen gleich ist (siehe Gleichung (3.36) auf Seite 23) und somit eine Gewichtung der Support-Vektoren erfolgt. Wodurch bei leicht ungleichverteilten Daten dennoch gute Ergebnisse erzielt werden können.

Untypische Daten sind ein häufiges Problem in der Klassifikation. Bei einem untypischen Datenpunkt handelt es sich um einen Datenpunkt, der aufgrund seines Merkmalsvektors abseits der Haupt-Datenmenge seiner Klasse liegt und damit eine optimale Modellbildung verhindert. Sie stellen natürliche Abweichungen dar. Diese Abweichungen haben verschiedene Ursachen, beispielsweise wenn nicht alle Merkmale eines Klassifikationsproblems erfasst wurden oder messbar waren. Es bedeutet aber nicht, dass es sich um einen fehlerhaften Datenpunkt handelt, weshalb der Begriff „Ausreißer“²⁰ nicht ganz zutreffend ist und daher auch nicht verwendet wird.

Zur Erprobung und Veranschaulichung der tatsächlichen Einflüsse von ungleichverteilten Belegungsdaten auf den SVM-Klassifikator wurde ein synthetisches ungleichverteiltes Drei-Klassen-Problem, wie in Abbildung 4-2, erzeugt.²¹ Die Klasse 1 mit 100 Datenpunkten (rot), die Klasse 2 mit 2500 Datenpunkten (grün) und die Klasse 3 mit 5000 Datenpunkten (blau) sind in einem zweidimensionalen Merkmalsraum im Wertebereich zwischen 0 und 1 angeordnet. Damit ergibt sich eine extreme Ungleichverteilung mit einem Verhältnis von 1:25:50. Die Punkte liegen ohne Ausnahmen innerhalb von definierten ellipsenförmigen Punktwolken und enthalten damit keine untypischen Daten.

²⁰ siehe Begriffsverzeichnis

²¹ Die zu diesem Thema gemachten Untersuchungen und Ergebnisse wurden bereits teilweise in [TSW14] veröffentlicht.

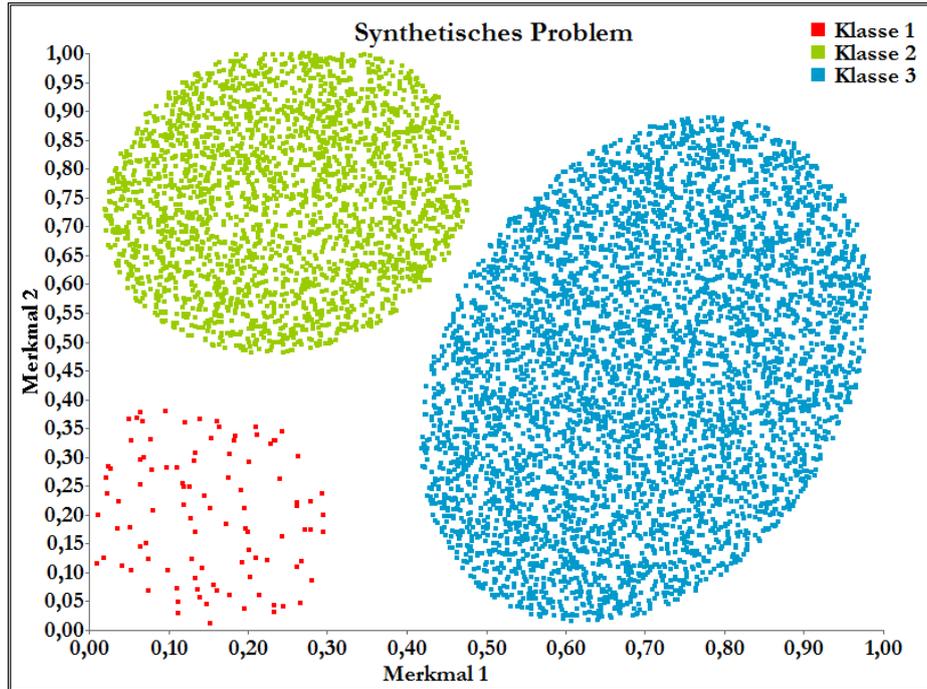


Abbildung 4-2: Synthetisches Drei-Klassen-Problem

Die Definition einer Ellipse EL besagt, dass für jeden Punkt P innerhalb einer Ellipse gilt:

$$EL := a(BP_1, P) + a(BP_2, P) \leq c \quad (4.1)$$

Der Punkt P liegt demnach dann innerhalb der Ellipse EL , wenn die Summation der Abstände a zu den Brennpunkten BP_1 und BP_2 der Ellipse kleiner oder gleich einen konstanten Wert c ist. Für die drei Ellipsen EL des dargestellten synthetischen Problems gelten folgende Gleichungen:

$$EL_1 := a\left(\begin{pmatrix} 0,1 \\ 0,3 \end{pmatrix}, \begin{pmatrix} p_1 \\ p_2 \end{pmatrix}\right) + a\left(\begin{pmatrix} 0,2 \\ 0,1 \end{pmatrix}, \begin{pmatrix} p_1 \\ p_2 \end{pmatrix}\right) \leq 0,4 \quad (4.2)$$

$$EL_2 := a\left(\begin{pmatrix} 0,2 \\ 0,6 \end{pmatrix}, \begin{pmatrix} p_1 \\ p_2 \end{pmatrix}\right) + a\left(\begin{pmatrix} 0,3 \\ 0,9 \end{pmatrix}, \begin{pmatrix} p_1 \\ p_2 \end{pmatrix}\right) \leq 0,55 \quad \text{und} \quad (4.3)$$

$$EL_3 := a\left(\begin{pmatrix} 0,6 \\ 0,1 \end{pmatrix}, \begin{pmatrix} p_1 \\ p_2 \end{pmatrix}\right) + a\left(\begin{pmatrix} 0,8 \\ 0,8 \end{pmatrix}, \begin{pmatrix} p_1 \\ p_2 \end{pmatrix}\right) \leq 0,9 \quad (4.4)$$

Anhand dieses synthetischen Problems, zunächst ohne untypische Daten, sollen die SVM-Modelle, die anhand der ungleichverteilten und der künstlich gleichverteilten Daten erzeugt wurden, verglichen werden und damit der Einfluss von ungleichverteilten Daten und des Fehlergewichts C gezeigt werden. Anschließend zeigen die gleichen Versuche des synthetischen Problems mit untypischen Daten, welche Auswirkungen die untypischen Daten auf die Modelle der SVM haben.

Das Problem der Gleichverteilung (GV) von Daten lässt sich auf verschiedenste Weise lösen (siehe Abschnitt 3.1.3 auf Seite 12). Neben der internen Ausbalancierung über eine Gewichtung der Fehlergewichte stehen unterschiedliche externe Ausbalancierungsmethoden zur Verfügung. Bei der externen Ausbalancierung werden die Daten künstlich gleichverteilt. Dies kann durch Verringerung der Datenmenge der Majoritätsklasse, dem Unterabtasten (engl. Undersampling) und durch Erzeugung synthetischer Datenpunkte für die Minoritätsklasse, dem Überabtasten (engl. Oversampling), erfolgen [BP12][Jap00].

Das Unterabtasten ist jedoch nicht immer empfehlenswert, da valide Daten verworfen werden und damit ein Informationsverlust einhergeht. Dies tritt insbesondere dann ein, wenn nur wenige Daten vorliegen oder die Daten aufgrund ihrer Variabilität einen hohen Informationsgehalt besitzen. Infolge des Informationsverlustes werden die resultierenden SVM-Trennebenen in ihrer Orientierung verändert und damit das Klassifikationsmodell in seiner Güte beeinträchtigt [AKJ04]. Aus diesem Grund wurden für die Versuche zwei Überabtastungsmethoden erprobt. Zum einen die Methode der Duplikation, eine sehr einfache Methode, bei der einzelne Datenpunkten lediglich kopiert werden.

Zum anderen den SMOTE-Algorithmus (Synthetic Minority Oversampling TEchnique) aus [CBHK02], welcher auf der Geraden zwischen einem Datenpunkt und seinen k nächsten Nachbarn zufällig neue synthetische Datenpunkte erzeugt. Der ursprüngliche Algorithmus sieht maximal fünf zu suchende nächste Nachbarn vor, was auf die vorliegenden synthetischen Daten angewendet zu Problemen führt. Die Abbildung 4-3 (links) zeigt eine Punktwolke von 100 Punkten. Bei einer Erzeugungsraterate von 49 synthetischen Punkten auf einen realen Punkt und nur 5 nächsten Nachbarn (Abbildung 4-3-mitte) werden zu viele Punkte zwischen zu wenigen nächsten Nachbarn erzeugt. Es entsteht eine ungleichmäßige Verteilung der erzeugten Punkte innerhalb der Punktwolke. Wird hingegen zwischen einem Punkt und seinen 99 nächsten Nachbarn neue Punkte erzeugt, verschwindet der Effekt. Der auf das vorliegende synthetische Problem angepasste SMOTE-Algorithmus ist in Anhang B: Abbildung B-2 auf Seite 130 in Form eines Programmablaufplans dargestellt.



Abbildung 4-3: Anwendung des SMOTE-Algorithmus: Punktwolke ohne GV (links), mit GV durch SMOTE bei 5 nächsten Nachbarn (Mitte) und bei 99 nächsten Nachbarn (rechts)[TSW14]

Um in den folgenden Versuchen die Qualität eines SVM-Modells auch ohne Validierungsdaten beurteilen und damit Vergleiche anstellen zu können, sind zugehörige Kennwerte zu den Modellen berechnet worden. Ein für die Komplexitätsbeurteilung wichtiger Kennwert ist die Anzahl der Support-Vektoren. Diese erhöhen mit steigender Anzahl die Komplexität des Klassifikators. Ein weiteres Qualitätsmerkmal ist die Fläche des Graubereichs, das heißt der Bereich zwischen den Schnittpunkten der Trennebenen (TE), in dem keine Klassenzugehörigkeit definiert ist. Diese Fläche sollte möglichst gering sein. Die Abweichung der Trennebene zu einem für dieses Problem ermitteltes Idealmodell in Bezug auf Winkelabweichung $\Delta\omega$ soll Aufschluss über die Orientierungsänderung der Trennebene geben und damit über die Beeinträchtigung der Leistungsfähigkeit des Klassifikators zeigen. Weiterhin soll der Minimalabstand der Trennebene a_{\min} zu den zwei von ihr getrennten Klassen die Verschiebung der Trennebene in Richtung einer Klasse zeigen.

Der erste Versuch vergleicht zwei SVM-Klassifikatoren, die anhand des ungleichverteilten Behlungsdatensatzes ohne untypische Daten (siehe Abbildung 4-2) mit unterschiedlichen Fehlergewichten C erzeugt wurden. Die Abbildung 4-4 zeigt die beiden SVM-Modelle, wobei die SV als dicke Datenpunkte dargestellt sind. Die Tabelle 4-1 zeigt die zugehörigen Kennwerte.

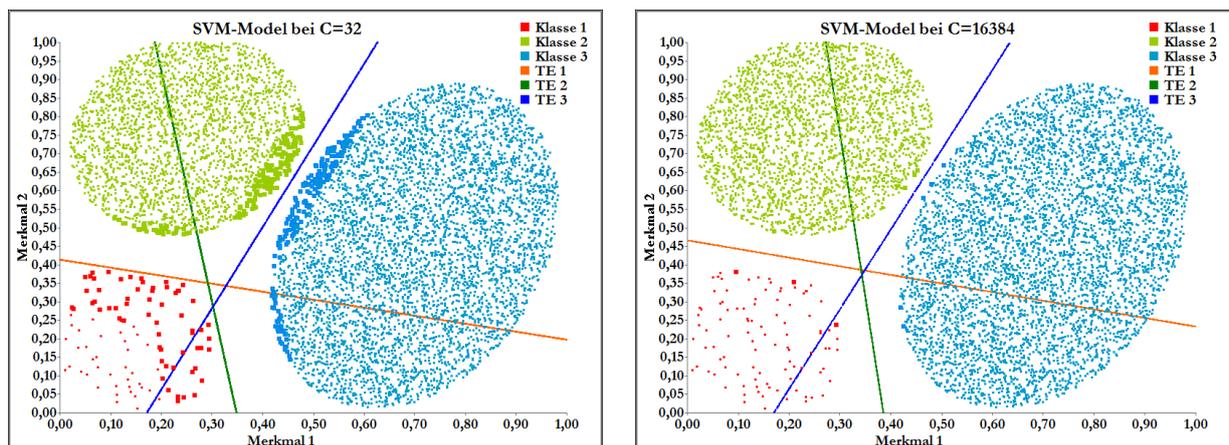


Abbildung 4-4: Behlungsdaten ohne untypische Daten und Trennebenen der SVM bei $C=32$ (links) und $C=16384$ (rechts) (SV als dicke Datenpunkte dargestellt) - Dabei ist eine deutliche Verschiebung der Trennebenen bei dem geringen Fehlergewicht von $C=32$ zu erkennen.

ohne untypische Daten			$C=32$	$C=16384$
Anzahl der SV			471	12
Fläche des Graubereichs			0,001103	0,000047
Winkelabweichung	$\Delta\omega$	Trennebene 1	2,96°	2,03°
Minimalabstand	$a_{\min K1}$		0,000013	0,037
	$a_{\min K2}$	0,107	0,059	
Winkelabweichung	$\Delta\omega$	Trennebene 2	1,12°	3,80°
Minimalabstand	$a_{\min K1}$		0,000052	0,069
		$a_{\min K3}$	0,111	0,068
Winkelabweichung	$\Delta\omega$	Trennebene 3	3,03°	2,60°
Minimalabstand	$a_{\min K1}$		0,014	0,019
		$a_{\min K2}$	0,023	0,019

Tabelle 4-1: Kennwerte des SVM-Klassifikators ohne untypische Daten bei $C=32$ und $C=16384$

Es wird deutlich, dass durch ein geringes Fehlergewicht C viele Fehler zugelassen werden. Dies macht sich in der Anzahl der Support-Vektoren (471 bei $C=32$ zu 12 bei $C=16384$) und in der Verschiebung der Trennebene bemerkbar. Es verschieben sich die TE 1 (orange) zwischen Klasse 1 und Klasse 2, die TE 2 (dunkelgrün) zwischen Klasse 1 und Klasse 3 und auch die TE 3 (dunkel blau) zwischen Klasse 2 und Klasse 3 jeweils in Richtung der schwach vertretenen Klasse. Dies ist sowohl optisch in Abbildung 4-4 als auch aus den Kennwerten in Tabelle 4-1 deutlich ablesbar. Der Minimalabstand a_{\min} zur Minoritätsklasse ist stets deutlich geringer als zur Majoritätsklasse (0,000013 zu 0,107 bei TE1; 0,000052 zu 0,111 bei TE2 und 0,014 zu 0,023 bei TE3), wobei die Ausprägung dieses Effekts direkt mit dem Verhältnis des Ungleichgewichts zusammenhängt. Betrachtet man die Fläche des Graubereichs (0,001103 bei $C=32$ zu 0,000047 bei $C=16384$), wird diese bei einem hohen Fehlergewicht C nahezu null. Weiterhin ergeben sich bei beiden SVM-Klassifikatoren lineare Trennebenen mit ähnlichen Orientierungen.

→ Insgesamt wird bei dem synthetischen Problem ohne untypische Daten mit einem hohem Fehlergewicht C bereits ein sehr viel besserer SVM-Klassifikator erreicht, sowohl qualitativ als auch auf die Komplexität bezogen. Es ist zu schlussfolgern, dass durch ein geeignetes Fehlergewicht C bereits die „Schwäche der Soft-Margin“ vermieden und eine Ausbalancierung unnötig werden kann.

Die Auswirkungen der Gleichverteilung (GV) soll nun am synthetischen Problem ohne untypische Daten bei einem Fehlergewicht von $C=32$ aufgezeigt werden. In Abbildung 4-5 sind die SVM-Klassifikatoren mit GV durch Duplikation und SMOTE-Algorithmus dargestellt, in Tabelle 4-2 befinden sich die zugehörigen Kennwerte.

Der Vergleich zwischen den Gleichverteilungsmethoden zeigt kaum Unterschiede. Während der Graubereich durch Duplikation etwas mehr reduziert werden kann (0,000212 bei Duplikation zu 0,000522 bei SMOTE), steigt die Anzahl der Support-Vektoren beim SMOTE nicht ganz so sehr an (750 bei Duplikation zu 698 bei SMOTE). Der Vergleich zwischen den Modellen ohne Gleichverteilung in Abbildung 4-4 (links) und mit Gleichverteilung der Daten in Abbildung 4-5 macht deutlich, dass durch die Gleichverteilung der Klassifikator qualitativ besser wird. Zum einen kann die Verschiebung zur Minoritätsklasse kompensiert werden, wobei sogar eine leichte Überkompensation auftritt. Dies zeigt sich in ähnlichen Abstandswerten für die einzelnen Trennebenen, wobei die Abstandswerte der Minoritätsklassen leicht höher sind als die der Majoritätsklassen (z. B. Duplikation 0,049 zu 0,046 bei TE1; 0,067 zu 0,056 bei TE2 und 0,024 zu 0,014 bei TE3). Zum anderen kann die Fläche des Graubereichs erheblich gesenkt werden (0,001103 ohne GV zu 0,000212 bzw. 0,000522 mit GV). Jedoch macht die GV die SVM komplexer. Zwar bleiben die Trennebenen linear, dennoch steigt die Anzahl der Support-Vektoren (471 ohne GV zu 750 bzw. 698 mit GV).

→ Die Gleichverteilung behebt zwar die „Schwäche der Soft-Margin“, macht aber das Modell deutlich komplexer.

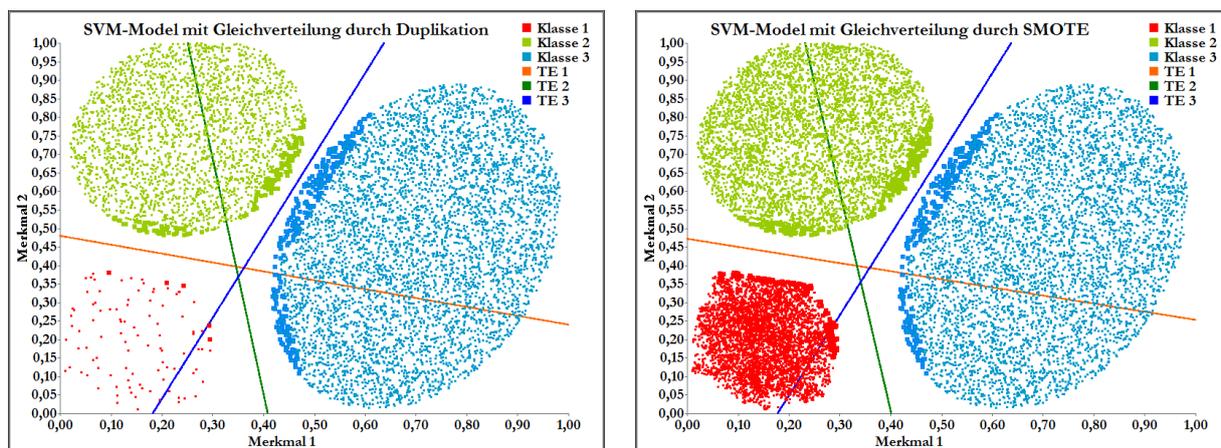


Abbildung 4-5: Belehrungsdaten ohne untypische Daten und Trennebenen der SVM bei $C=32$ und einer GV durch Duplikation (links) und SMOTE (rechts) (SV als dicke Datenpunkte dargestellt) - Beide GV-Methoden können die Verschiebung der Trennebenen kompensieren.

ohne untypische Daten bei $C=32$		Dupl.	SMOTE
Anzahl der SV		750	698
Fläche des Graubereichs		0,000212	0,000522
Winkelabweichung	$\Delta\omega$	Trennebene 1	1,67°
Minimalabstand	$a_{\min K1}$		0,049
	$a_{\min K2}$		0,046
Winkelabweichung	$\Delta\omega$	Trennebene 2	1,30°
Minimalabstand	$a_{\min K1}$		0,067
	$a_{\min K3}$		0,056
Winkelabweichung	$\Delta\omega$	Trennebene 3	3,07°
Minimalabstand	$a_{\min K1}$		0,024
	$a_{\min K2}$		0,014

Tabelle 4-2: Kennwerte des SVM-Klassifikators ohne untypische Daten bei $C=32$ und einer GV durch Duplikation und SMOTE

Es sollen nun die Auswirkungen von untypischen Daten auf den SVM-Klassifikator gezeigt und insbesondere der Einfluss dieser untypischen Daten bei der Ausbalancierung betrachtet werden. Dafür wurde das gleiche Drei-Klassen-Problem wie zuvor erzeugt, jedoch mit einer Rate an untypischen Daten von 5%. Dieses ist in Abbildung 4-6 zu sehen. Die untypischen Daten entstanden in einem pro Klasse definierten Wertebereich, wodurch die Klassen sich nun überlappen. Anhand dieses neuen Problems soll zunächst der Einfluss des Fehlergewichts C auf die SVM gezeigt werden.

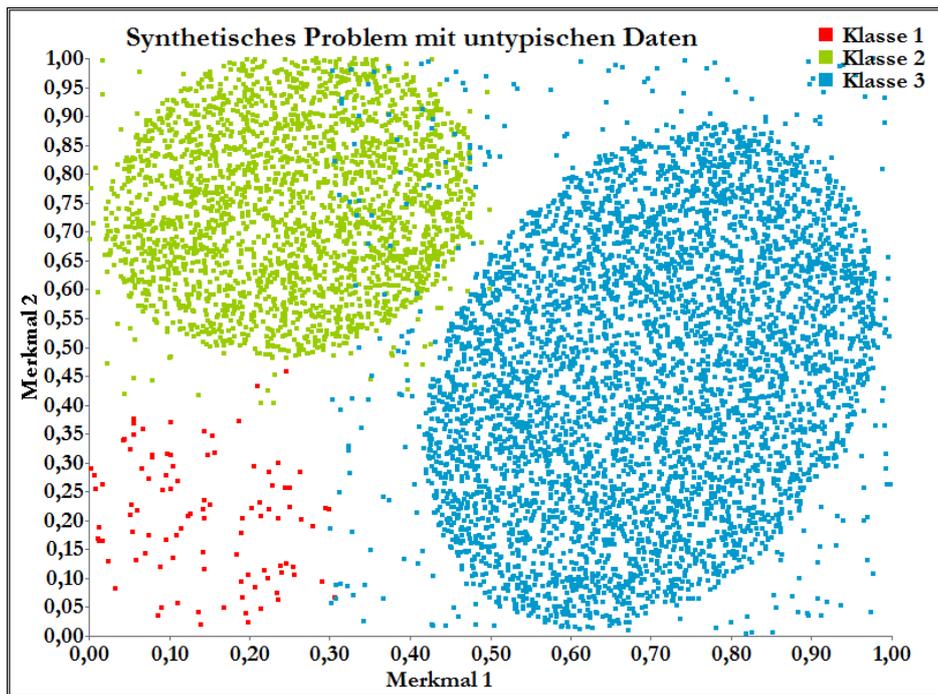


Abbildung 4-6: Synthetisches Drei-Klassen-Problem mit untypischen Daten

Die Abbildung 4-7 und die Tabelle 4-3 zeigen die entstandenen Trennebenen mit Kennwerten bei einem Fehlergewicht von $C=32$ und $C=16384$. Wie auch im Problem ohne untypische Daten kann durch das geringe Fehlergewicht C die Verschiebung der Trennebenen in Richtung der Minoritätsklasse beobachtet werden. Die extrem unterschiedlichen Minimalabstände a_{\min} pro Trennebene bei der SVM mit $C=32$ erfassen diesen Sachverhalt (0,00043 zu 0,113 bei TE1; 0,000008 zu 0,134 bei TE2 und 0,008 zu 0,029 bei TE3). Weiterhin erhöht sich die Anzahl der Support-Vektoren durch ein geringes Fehlergewicht C (675 bei $C=32$ zu 299 bei $C=16384$), was eine Steigerung der Komplexität des SVM-Klassifikators mit sich bringt. Im Allgemeinen kann durch das hohe Fehlergewicht C der SVM-Klassifikator verbessert werden. Die reduzierte Fläche des Graubereichs ist ein Hinweis darauf (0,003052 bei $C=32$ zu 0,000045 bei $C=16384$).

→ Jedoch zeigt sich an diesem Problem, dass bei Datensätzen mit untypischen Daten die Erhöhung des Fehlergewichts C nicht die „Schwäche der Soft-Margin“ vermeiden kann.

Denn auch bei $C=16384$ ist die Verschiebung in Richtung der Minoritätsklasse deutlich sichtbar (0,007 zu 0,87 bei TE1; 0,000026 zu 0,13 bei TE2 und 0,02 zu 0,017 bei TE3). Zusätzlich ändern sich die Orientierungen der Trennebenen spürbar (von $1,44^\circ$ auf $8,06^\circ$ bei TE1 und von $2,33^\circ$ auf $8,85^\circ$ bei TE2). Die TE 3 wird durch die untypischen Daten, die mit Klasse 2 überlappen, nichtlinear. Im Vergleich zu dem Problem ohne untypische Daten (Tabelle 4-1) zeigt sich, dass die untypischen Daten zwangsläufig die Anzahl der Support-Vektoren erhöhen, da jeder abweichende Datenpunkt als Support-Vektor in den Klassifikator aufgenommen wird (471 und 12 ohne untypische Daten zu 675 und 299 mit untypischen Daten).

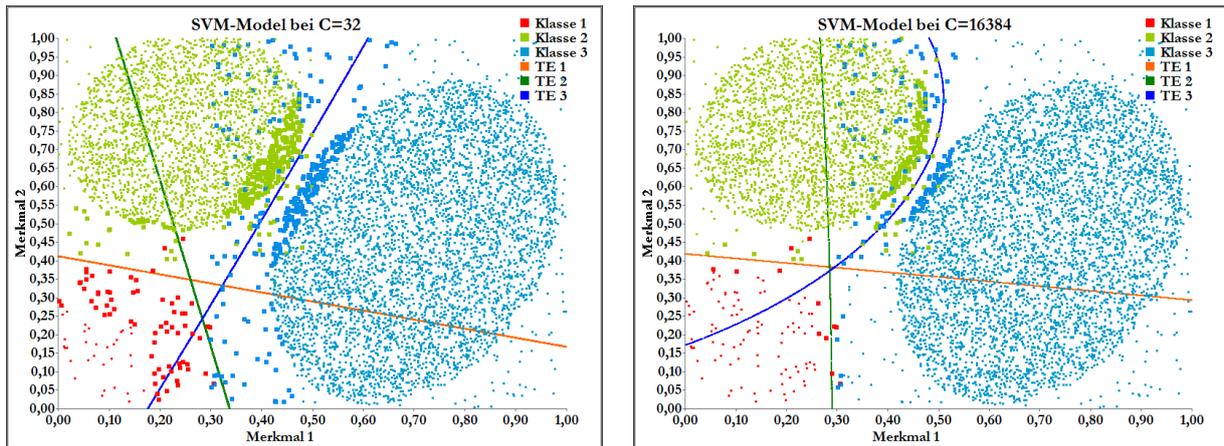


Abbildung 4-7: Belehrungsdaten mit untypischen Daten und Trennebenen der SVM bei C=32 (links) und C=16384 (rechts) (SV als dicke Datenpunkte dargestellt) - Dabei ist bei beiden Fehlergewichten eine Trennebenenverschiebung erkennbar.

mit untypischen Daten			C=32	C=16384
Anzahl der SV			675	299
Fläche des Graubereichs			0,003052	0,000045
Winkelabweichung	$\Delta\omega$	Trennebene 1	1,44°	8,06°
Minimalabstand	$a_{\min K1}$		0,000043	0,007
	$a_{\min K2}$	0,113	0,087	
Winkelabweichung	$\Delta\omega$	Trennebene 2	2,33°	8,85°
Minimalabstand	$a_{\min K1}$		0,000008	0,000026
		$a_{\min K3}$	0,134	0,130
Winkelabweichung	$\Delta\omega$	Trennebene 3	4,19°	-
Minimalabstand	$a_{\min K1}$		0,008	0,020
		$a_{\min K2}$	0,029	0,017

Tabelle 4-3: Kennwerte der SVM mit untypischen Daten bei C=32 und C=16384

Die künstliche Gleichverteilung (GV) der Datensätze bei C=32 kann die Verschiebung der Trennebenen (TE) auch bei Datensätzen mit untypischen Daten kompensieren und dadurch die Leistungsfähigkeit der SVM erheblich verbessern, wie in Abbildung 4-8 und Tabelle 4-4 dargestellt. Dies zeigen die Minimalabstände a_{\min} zu den Trennebenen mit ähnlichen Werten (zum Beispiel durch Duplikation mit 0,051 zu 0,043 bei TE1; 0,051 zu 0,055 bei TE2 und 0,023 zu 0,014 bei TE3).

→ Die Gleichverteilung bewirkt außerdem bei diesem Problem mit untypischen Daten eine Änderung der Orientierung der Trennebenen.

Jedoch steigt dadurch die Anzahl der Support-Vektoren (675 zu 1139 durch Duplikation bzw. 1004 durch SMOTE), die aufgrund der untypischen Daten bereits erhöht sind. Der Graubereich kann durch die Gleichverteilung verringert werden (0,0031 vor GV zu 0,0016 bzw. 0,002 nach GV). Die Orientierungsänderung der Trennebenen wird besonders an der TE 1 deutlich (1,44° vor GV zu 6,7° bzw. 5,37° nach GV).

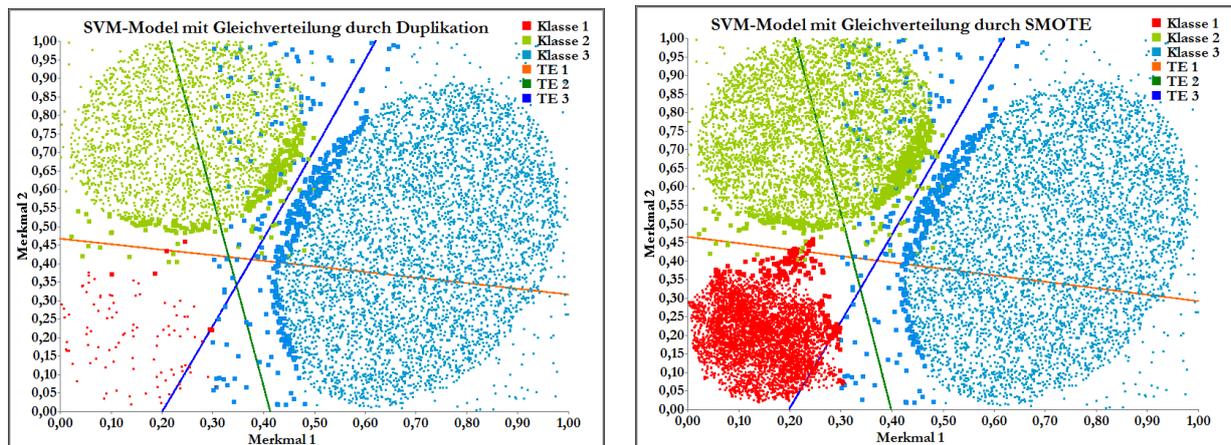


Abbildung 4-8: Belehrungsdaten mit untypischen Daten und Trennebenen der SVM bei $C=32$ und einer GV durch Duplikation (links) und SMOTE (rechts) (SV als dicke Datenpunkte dargestellt) - Beide GV-Methoden können die Verschiebung der Trennebenen kompensieren, jedoch verändert sich die Orientierung der Trennebenen.

mit untypischen Daten bei $C=32$			Dupl.	SMOTE
Anzahl der SV			1139	1004
Fläche des Graubereichs			0,001646	0,001969
Winkelabweichung	$\Delta\omega$	Trennebene 1	6,70°	5,37°
Minimalabstand	$a_{\min K1}$		0,051	0,046
	$a_{\min K2}$	0,043	0,049	
Winkelabweichung	$\Delta\omega$	Trennebene 2	1,00°	0,43°
Minimalabstand	$a_{\min K1}$		0,051	0,039
	$a_{\min K3}$	0,055	0,067	
Winkelabweichung	$\Delta\omega$	Trennebene 3	4,95°	4,85°
Minimalabstand	$a_{\min K1}$		0,023	0,021
	$a_{\min K2}$	0,014	0,016	

Tabelle 4-4: Kennwerte der SVM mit untypischen Daten bei $C=32$ und einer GV durch Duplikation und SMOTE

Der Grund dafür ist die Einbeziehung der untypischen Daten in die Gleichverteilung. Diese werden demnach dupliziert oder dienen als Ausgangspunkt für neue Datenpunkte. Dadurch steigt der Einfluss der untypischen Daten, was zu einer Orientierungsänderung der Trennebenen führt. Auf die Linearität der Trennebenen hat die Gleichverteilung keinen Einfluss. Der Vergleich der beiden Gleichverteilungsmethoden führt zu dem Schluss, dass beide ähnliche Ergebnisse liefern. Jedoch kommt die SVM durch die SMOTE-Gleichverteilung mit weniger Support-Vektoren aus (1139 durch Duplikation zu 1004 durch SMOTE).

Die Auswirkungen der künstlichen Gleichverteilung bei dem Fehlgewicht von $C=16384$ zeigt Abbildung 4-9 und Tabelle 4-5 mit den zugehörigen Kennwerten.

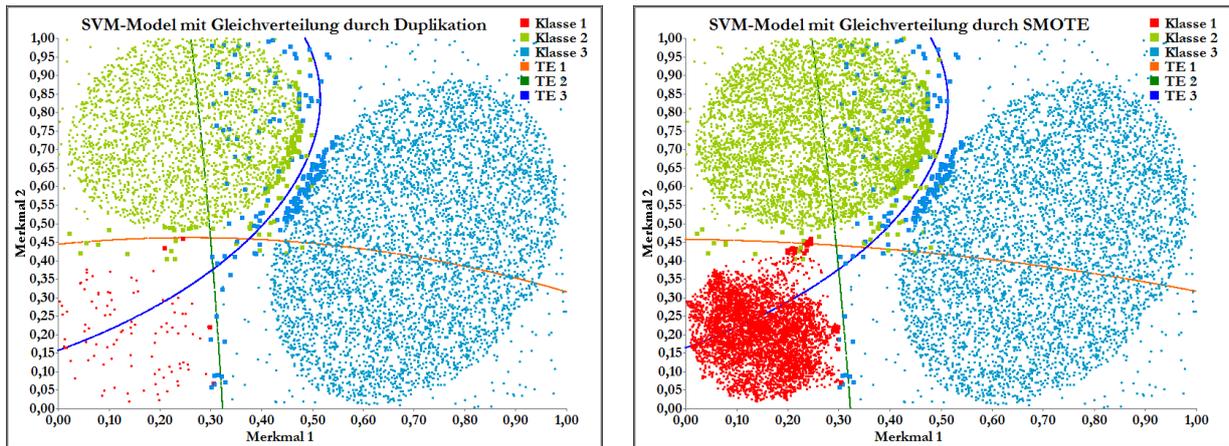


Abbildung 4-9: Belehrungsdaten mit untypischen Daten und Trennebenen der SVM bei $C=16384$ und einer GV durch Duplikation (links) und SMOTE (rechts) (SV als dicke Datenpunkte dargestellt) - Beide GV-Methoden können die Verschiebung der Trennebenen nicht mehr vollständig kompensieren.

mit untypischen Daten bei $C=16384$			Dupl.	SMOTE
Anzahl der SV			441	361
Fläche des Graubereichs			0,003249	0,002151
Winkelabweichung	$\Delta\omega$	Trennebene 1	-	-
Minimalabstand	$a_{\min K1}$		0,062	0,058
	$a_{\min K2}$	0,019	0,032	
Winkelabweichung	$\Delta\omega$	Trennebene 2	-	-
Minimalabstand	$a_{\min K1}$		0,000206	0,000062
	$a_{\min K3}$	0,108	0,112	
Winkelabweichung	$\Delta\omega$	Trennebene 3	-	-
Minimalabstand	$a_{\min K1}$		0,027	0,026
	$a_{\min K2}$	0,009	0,010	

Tabelle 4-5: Kennwerte der SVM mit untypischen Daten bei $C=16384$ und einer GV durch Duplikation und SMOTE

Die Minimalabstände a_{\min} sind bei diesem hohen Fehlergewicht C dennoch sehr unterschiedlich (z. B. Duplikation: 0,062 zu 0,019 bei TE1; 0,0002 zu 0,108 bei TE2 und 0,027 zu 0,009 bei TE3). Die Anzahl der Support-Vektoren steigen, wie in bisher allen Versuchen, durch die Gleichverteilung an (299 vor GV zu 441 bzw. 361 nach GV). Der Graubereich vergrößert sich durch die Gleichverteilung sogar (0,000045 zu 0,003 bzw. 0,002). Auch die Orientierung der Trennebenen, besonders der TE 1, ändert sich. Diese Werte sind jedoch durch die Nichtlinearität der Trennebenen nicht mehr auswertbar. Auch hier ist die Ursache wieder in der Einbeziehung der untypischen Daten in der Gleichverteilung zu suchen. Insgesamt verschlechtert sich die Leistungsfähigkeit der SVM sogar durch die Gleichverteilung bei diesem hohen Fehlergewicht C .

→ Die Gleichverteilung kann nicht immer die Verschiebung der Trennebenen kompensieren.

Es lässt sich zusammengefasst sagen:

- Das Fehlergewicht C beeinflusst die SVM sehr, sowohl in Komplexität als auch in Qualität. Bereits ein geeignetes Fehlergewicht C kann die „Schwäche der Soft-Margin“ vermeiden und eine Gleichverteilung unnötig machen. Dieser Einfluss ist jedoch datenabhängig. Das Fehlergewicht allein kann die Schwäche nicht mehr ausgleichen, wenn die Belehrungsdaten mit untypischen Daten behafteten sind.
 - Die künstliche Gleichverteilung der Datensätze kann die Leistung der SVM erheblich verbessern, jedoch erhöht sie die Komplexität durch Erhöhung der Anzahl der Support-Vektoren. Unter den getesteten Überabtastungsmethoden waren nur geringe Unterschiede feststellbar. Aufgrund der geringeren Anzahl an Support-Vektoren und damit einer geringeren Komplexität wäre der SMOTE-Algorithmus der Duplikation vorzuziehen.
 - Die untypischen Daten stellen ein großes Problem in der Klassifikation dar. Sie verursachen eine höhere Komplexität durch Zunahme der Support-Vektoren und durch Forcierung von nichtlinearen Trennebenen. Sie bewirken eine Verzerrung, welche sich in den Orientierungen der Trennebenen bemerkbar macht. Bei der Gleichverteilung gewinnen diese untypischen Daten weiter an Einfluss und verzerren die Trennebenen zusätzlich. Die Gleichverteilung erreicht dadurch keine Qualitätssteigerung mehr, die Verschiebung kann nicht kompensiert werden und der Graubereich vergrößert sich.
- Um nun zu einem optimalen Klassifikator zu gelangen, müssen die untypischen Daten unbedingt identifiziert und gelöscht werden. Alternativ sollten die untypischen Daten zumindest aus der Gleichverteilung ausgeschlossen werden. Erst anschließend sollten ungleichverteilte Datensätze ausbalanciert werden.

4.1.2 Identifikation von untypischen Daten

Zur Identifikation „untypischer Daten“ wurde ein neuartiger Datenbearbeitungsalgorithmus, der DiLa-Filter, entworfen [TW16]. Im Gegensatz zu vielen anderen Filterungsmethoden wird der DiLa nicht auf das Rohsignal angewendet, sondern auf die berechneten Merkmalsvektoren. Dabei werden erstmals anhand der Dichte- und Lage-Information die untypischen Datenpunkte aus den Belehrungsdaten entfernt. Dieser Algorithmus lässt sich auf eine Vielzahl von Problemstellungen anwenden.

Ausgangspunkt für die Identifikation von untypischen Daten sind die folgenden Annahmen:

1. Untypische Daten befinden sich in Bereichen mit einer geringen Dichte an Datenpunkten.
2. Untypische Daten kommen vornehmlich außerhalb der von typischen Daten gebildeten Punktwolken vor.

Der durchschnittliche euklidische Abstand eines Punktes zu seinen nächsten Nachbarn (NN) wird als Maß für die Dichte verwendet. Dieses Dichtemaß ist somit lokal und kann durch die Wahl der Anzahl der nächsten Nachbarn k nahezu ohne Einschränkungen variiert werden. Die Ermittlung des Abstandsdurchschnitts bewirkt jedoch eine Reduzierung des Einflusses von k . Wie lokal die Dichteinformation ist, wird damit durch die Anzahl der nächsten Nachbarn k be-

stimmt. Es gilt: „Je kleiner k , desto lokaler die Dichteinformation.“ Die Anzahl der nächsten Nachbarn k muss als freier Parameter für eine konkrete Anwendung empirisch bestimmt werden. Die Dichteberechnung mittels der nächsten Nachbarn kann aufgrund der hohen Dimensionalität problematisch sein. Mit steigender Dimension reduziert sich der Distanzunterschied zum nächsten Nachbarn. Jedoch kann durch die klassenspezifischen Ausprägungen der Merkmale die Dimension keinen zu großen Einfluss ausüben. Deshalb ist es wichtig, dass die Daten eine hohe Korrelation zur Klasse aufweisen.

Das Lagemaß stellt der euklidische Abstand des Punktes zu einem zuvor ermittelten Dichtezentrum dar. Dieses Dichtezentrum wird über den Schwerpunkt der Punktwolke, nachdem die untypischen Daten nach Dichte entfernt wurden, bestimmt.

Beide Maße, Dichte und Lage, sind zunächst sehr einfach gewählt. Dadurch steigt der Berechnungsaufwand des Algorithmus linear mit der Anzahl der Dimensionen. Der DiLa-Filter ist somit für große und hochdimensionale Datenmengen geeignet. Die Maße könnten jedoch beliebig durch andere Methoden ersetzt werden. Die Berechnungskomplexität müsste dann erneut betrachtet werden.

Im DILA-Filter werden folglich Dichte- und Lage-Information kombiniert, um untypische Daten sicher identifizieren zu können. Das Programmablaufdiagramm des DILA-Filters ist in Abbildung 4-10 dargestellt. Da es sich um die Bearbeitung von Belehrungsdaten der SVM handelt, werden für jede Klasse y die folgenden Schritte separat durchgeführt. Zunächst werden von jedem Punkt i der Klasse y die nächsten Nachbarn gesucht. Als Maß für die Dichte wird der durchschnittliche euklidische Abstand bestimmt. Problematisch gestaltet sich nun die Festlegung der Dichtegrenze, welche einen untypischen Datenpunkt von einem typischen Datenpunkt trennt.

Da je nach Verteilung der Punkte verständlicherweise variierende Wertebereiche auftreten würden, berechnet sich die Dichtegrenze nicht über einen absoluten Wert. Die Dichtegrenze DG wird für jede Klasse y mittels des Medianes \tilde{d}_y und des Semiquartilsabstandes $dSQA_y$ des Dichtemaßes und einem Faktor-Dichte FD wie folgt definiert:

$$DG_y = \tilde{d}_y + (FD \cdot dSQA_y) \quad (4.5)$$

Das Dichtezentrum wird mittels des Schwerpunktes SP_y bestimmt, nachdem zunächst alle Datenpunkte, die durch die DG als untypisch gelten, gelöscht wurden. Es gilt die Formel:

$$SP_y = \frac{\sum_{i=0}^N \underline{x}_i}{N_y} \quad (4.6)$$

wobei N die Anzahl der Punkte einer Klasse y und \underline{x} die mehrdimensionalen Merkmalsvektoren der Punkte i einer Klasse darstellen. Für jeden Punkt i bestimmt der DiLa-Filter nun den Abstand zu diesem Zentrum.

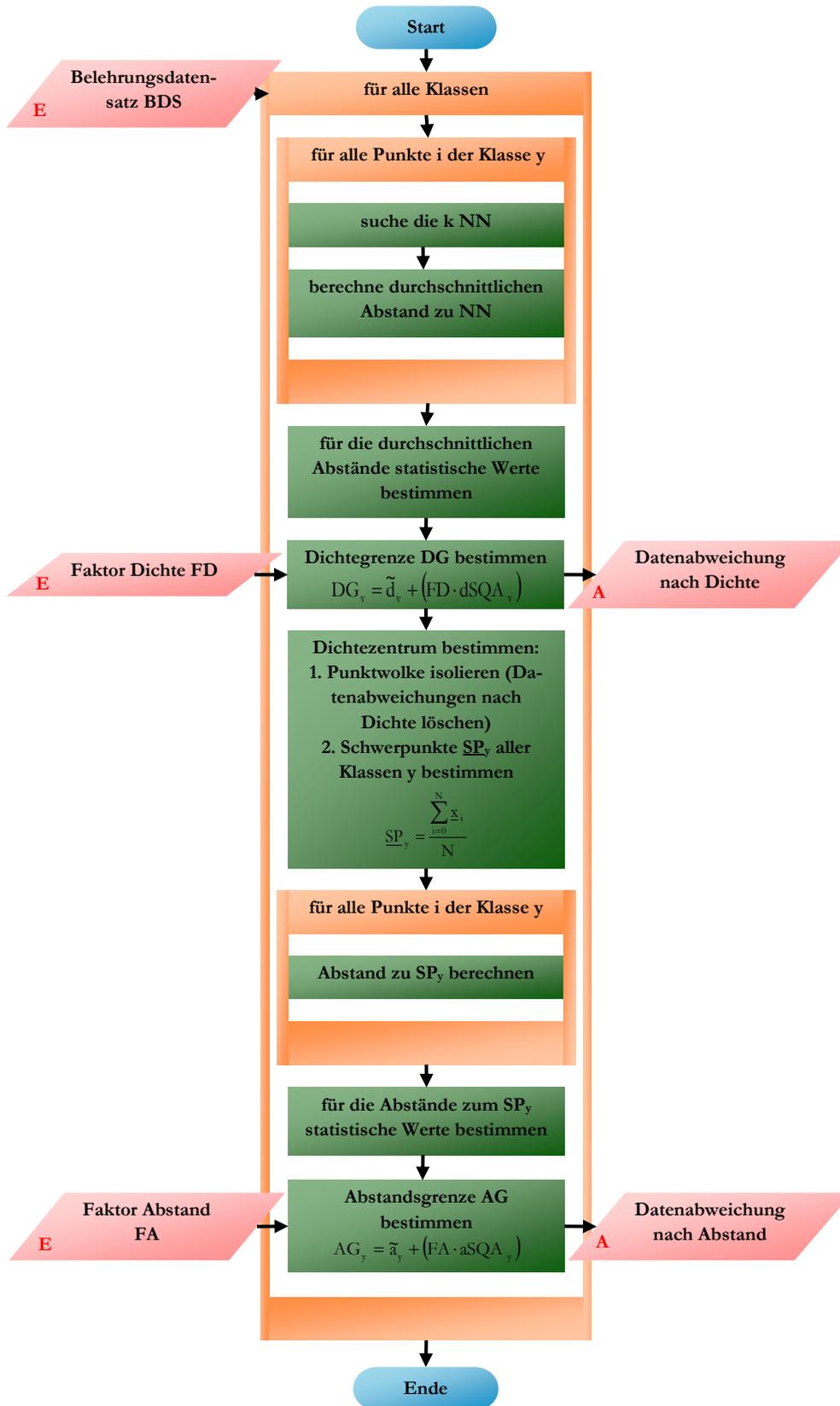


Abbildung 4-10: Programmablaufdiagramm der DiLa-Filterung

Analog zur Dichtegrenze DG berechnet sich anschließend die Abstandsgrenze AG mit Hilfe der statistischen Werte Median $\tilde{\alpha}_y$ und Semiquartilsabstand $aSQA_y$ sowie mittels eines Faktor-Abstand FA. Es gilt:

$$AG_y = \tilde{\alpha}_y + (FA \cdot aSQA_y) \quad (4.7)$$

Ein Datenpunkt wird genau dann durch den DiLa-Filter als untypischer Datenpunkt identifiziert, wenn der Punkt zum einen nach einem festzulegenden Faktor-Dichte FD als separat stehender Punkt erkannt wird und sich zum anderen zusätzlich nach einem festzulegenden Faktor-Abstand FA weit vom Dichtezentrum entfernt befindet.

Auf das in Abbildung 4-6 auf Seite 46 dargestellt synthetische Problem mit untypischen Daten wurde der DiLa-Filter mit den Faktoren $FD=0,8$ und $FA=0,8$ sowie $k=5$ angewendet. Daraus resultierten die in Abbildung 4-11 (links) dargestellten Daten.

Es können durch den DiLa-Filter bei dieser Faktorenkombination die meisten untypischen Daten identifiziert und gelöscht werden. Grund für die verbliebenen untypischen Daten ist die Ellipsenform der Punktwolken, die bei Klasse 3 besonders ausgeprägt ist. Höhere Faktoren würden die Anzahl der verbleibenden untypischen Daten erhöhen, niedrigere Faktoren würden vermehrt auch Datenpunkte innerhalb der Punktwolke löschen.

Nach den gewonnenen Erkenntnissen aus Abschnitt 4.1.1 kann nun die Gleichverteilung (GV) der Daten mittels SMOTE-Algorithmus durchgeführt werden. Die in Abbildung 4-11 (rechts) dargestellten Daten sind durch den SMOTE-Algorithmus gleichverteilt, nachdem untypische Daten mit dem DiLa gelöscht wurden.

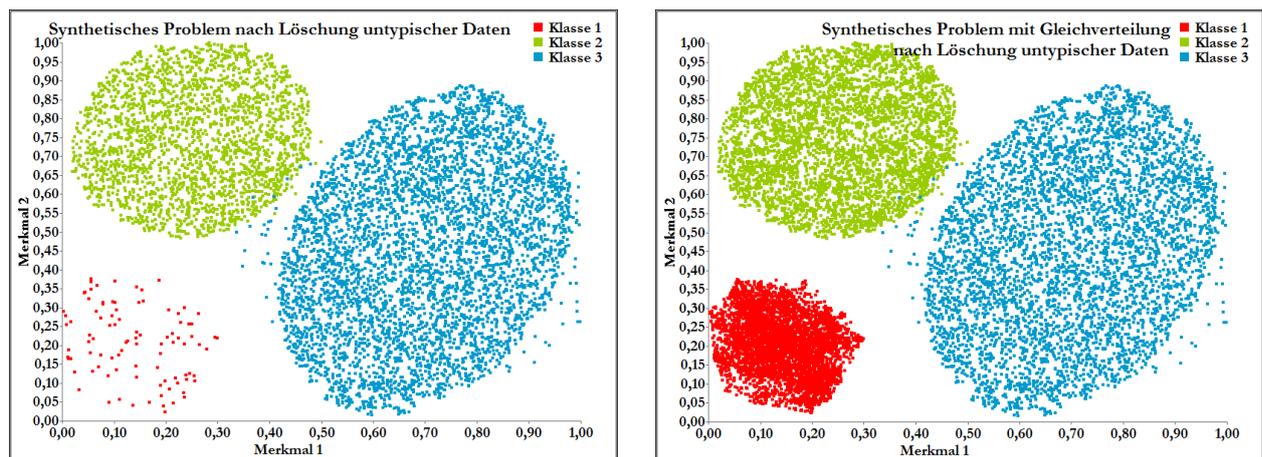


Abbildung 4-11: Synthetisches Drei-Klassen-Problem nach Löschung untypischer Daten durch den DiLa-Filter mit $FD=0,8$, $FA=0,8$ und $k=5$ ohne GV (links) und nach GV (rechts) - Dabei konnten die meisten untypischen Daten gelöscht werden.

Für die Bestimmung der SVM-Parameter stehen nun drei mögliche Zeitpunkte der Datenbearbeitung zur Wahl, die betrachtet werden müssen.

1. Die Parameterbestimmung könnte anhand der Ursprungsdaten, d. h. vor Gleichverteilung und vor Datenlöschung mit dem DiLa-Filter, vorgenommen werden. Hierbei müsste nur einmal die Parameterbestimmung vorgenommen werden, bevor die Faktoren für die DiLa-Filterung gesucht werden. Fraglich ist jedoch die zu erreichende Güte dieser Parameter, wenn diese auf die gefilterten und gleichverteilten Daten angewendet werden.
2. Die Parameterbestimmung könnte anhand der bearbeiteten Daten, d. h. vor der Gleichverteilung und nach DiLa-Filterung, erfolgen. Hierbei müsste die Parameterbestimmung bei jeder Faktorenkombination des DiLa-Filters erneut vorgenommen werden.
3. Die Parameterbestimmung könnte anhand der gleichverteilten Daten, d. h. nach der Gleichverteilung und nach der DiLa-Filterung, gemacht werden. Hierbei ist die Parameterbestimmung, die bei jeder Faktorenkombination erfolgt, durch die erhöhte Datenmenge rechenintensiv. In diesem Fall sind die Daten zur Bestimmung der Parameter und die Daten, auf welche die Parameter angewendet werden, identisch.

Der Rechenaufwand ist folglich bei Reihenfolge 1 am geringsten und bei Reihenfolge 3 am höchsten. Entscheidend sollte jedoch die Güte des resultierenden Klassifikators sein.

Die Ergebnisse für das synthetische Problem sind in Abbildung 4-12 und Tabelle 4-6 zu sehen. Bei allen drei Modellen liegen die Trennebenen infolge der angewendeten Gleichverteilung (GV) mittig zwischen den Klassen, was die ähnlichen Minimalabstände a_{\min} der Trennebenen (TE) zu den jeweiligen Klassen verdeutlichen. Auch die Anzahl der Support-Vektoren (SV) sind bei allen Modellen ähnlich. Bei Reihenfolge 1 (oben links im Bild), d. h. Parameterbestimmung anhand der Ursprungsdaten, resultiert das augenscheinlich beste Modell. Es gibt nahezu keinen Graubereich (0,000044). Die Trennebenen sind ungefähr linear und in ihrer Orientierung ähnlich dem Idealmodell. Jedoch treten mehrere Parameterkombinationen mit dem gleichen Belehrungsgrad von 99,95% auf. Die aus den anderen Parameterkombinationen resultierenden Modelle ähneln dem Modell von Reihenfolge 2 (oben rechts). Bei Reihenfolge 2, d. h. eine Parameterbestimmung nach Datenlöschung mittels DiLa-Filter aber noch vor der Gleichverteilung, entstand ein Modell mit kleinem Graubereich (0,000718) und nichtlinearen Trennebenen, sodass eine Winkelauswertung nicht möglich ist. Scheinbar erfolgt bei der Parameterbestimmung nach der Datenfilterung eine erhöhte Anpassung an die Belehrungsdaten, wodurch die verbliebenen untypischen Daten einen größeren Einfluss auf den Verlauf der Trennebenen haben. Die TE 3 macht einen leichten Bogen im Bereich der verbliebenen untypischen Daten. Bei Reihenfolge 3 (unten im Bild), d. h. Parameterbestimmung nach Datenfilterung mit dem DiLa-Algorithmus und GV, entstand ebenfalls ein Modell mit kleinem Graubereich (0,000465) und nichtlinearen Trennebenen. Jedoch ist der Verlauf der TE 3 (blau) besonders auffällig. Durch die Parameterbestimmung nach der Gleichverteilung verstärkt sich der Effekt, der in Modell von Reihenfolge 2 zu sehen war. In diesem Fall ist ein deutlicher Bogen in der TE 3 zu erkennen.

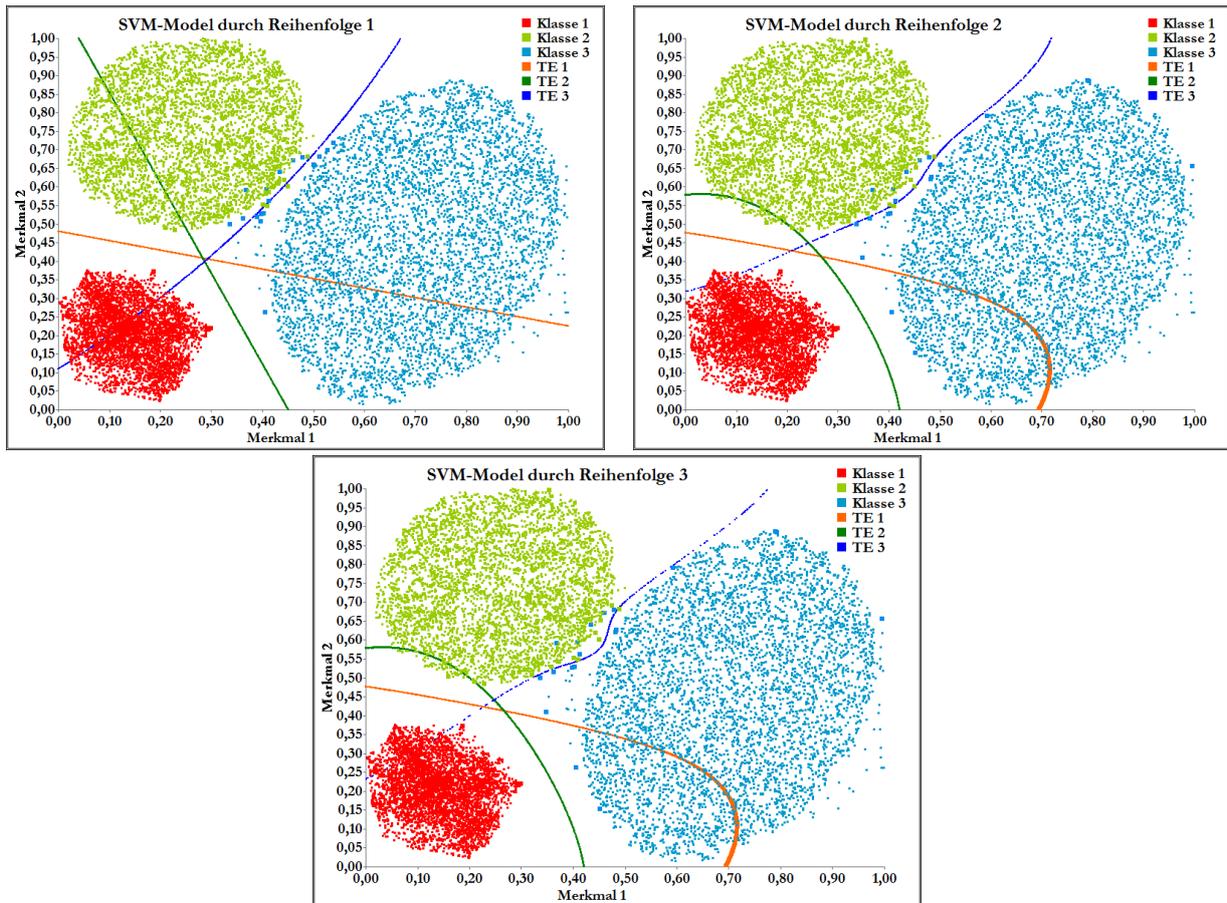


Abbildung 4-12: Belehrungsdaten mit gelöschten untypischen Daten ($FD=0,8 \mid FA=0,8 \mid k=5$) nach SMOTE-GV und Trennebenen der SVM bei unterschiedlichen Zeitpunkten der Parameterbestimmung (SV als dicke Datenpunkte dargestellt):

- Reihenfolge 1, d. h. vor Datenlöschung und GV (oben links), erzeugt ein sehr gutes SVM-Modell.
- Reihenfolge 2, d. h. nach Datenlöschung aber vor GV (oben rechts), erzeugt ein akzeptables SVM-Modell mit einer verstärkten Anpassung auf die verbliebenen untypischen Daten.
- Reihenfolge 3, d. h. nach Datenlöschung und GV (unten), erzeugt ein akzeptables SVM-Modell mit einer noch stärkeren Anpassung auf die verbliebenen untypischen Daten.

bei Parameterbestimmung zu unterschiedlichen Zeitpunkten			Reihenfolge 1	Reihenfolge 2	Reihenfolge 3
Anzahl der SV			38	40	40
Fläche des Graubereichs			0,000044	0,000718	0,000465
Winkelabweichung	$\Delta\omega$	Trennebene 1	0,83°	-	-
Minimalabstand	$a_{\min K1}$		0,048	0,048	0,048
	$a_{\min K2}$	0,048	0,047	0,047	
Winkelabweichung	$\Delta\omega$	Trennebene 2	12,05°	-	-
Minimalabstand	$a_{\min K1}$		0,063	0,107	0,094
	$a_{\min K3}$	0,127	0,219	0,199	
Winkelabweichung	$\Delta\omega$	Trennebene 3	5,19°	-	-
Minimalabstand	$a_{\min K1}$		0,014	0,008	0,004
	$a_{\min K2}$	0,012	0,008	0,003	

Tabelle 4-6: Kennwerte der SVM mit gelöschten untypischen Daten ($FD=0,8 \mid FA=0,8 \mid k=5$) nach SMOTE-GV bei unterschiedlichen Zeitpunkten der Parameterbestimmung

→ In Anbetracht der Resultate dieser synthetischen Daten ist Reihenfolge 1 vorzuziehen, sowohl wegen des besseren Modells als auch wegen des geringsten Rechenaufwandes. Reihenfolge 3 ist hingegen aufgrund der Qualität des erreichten Modells und dem erhöhten Rechenaufwand nicht sinnvoll.

Anhand von realen Daten werden die Datenbearbeitungsmethoden Gleichverteilung und DiLa-Filterung sowie die drei Reihenfolgen der Parameterbestimmung ebenfalls erprobt und verglichen.

1. Ohne Gleichverteilung und vor DiLa-Filterung der Belehrungsdaten (weiß), d. h. keine Bearbeitung der Daten als Referenz,
2. mit Gleichverteilung, aber noch vor DiLa-Filterung (gelb), um die Auswirkungen der untypischen Daten einschätzen zu können,
3. mit Gleichverteilung und DiLa-Filterung nach Reihenfolge 1 (orange), d. h. Parameterbestimmung vor Gleichverteilung und Filterung,
4. mit Gleichverteilung und DiLa-Filterung nach Reihenfolge 2 (rot), d. h. Parameterbestimmung vor Gleichverteilung aber nach Datenlöschung,
5. mit Gleichverteilung und DiLa-Filterung nach Reihenfolge 3 (violett), d. h. Parameterbestimmung nach Gleichverteilung und Datenlöschung.

Hierfür finden die zehn Probandendatensätze von B1 (Tabelle A-1 auf Seite 121) Verwendung. Dabei werden jeweils ein Datensatz zum Belehren und die restlichen neun zum Testen eingesetzt.²² Es sind in Abbildung 4-13 in Form eines Kastendiagramms die vier Güterwerte GG, DG, GMG und MG in Prozent und in Abbildung 4-14 die daraus resultierende gewichtete Güte G_{gew} in Prozent jeweils für die fünf Datenbearbeitungsmethoden und Vorgehensweisen dargestellt. Dazu zeigt Tabelle 4-7 die aus dem zweiseitigen Wilcoxon-Vorzeichen-Rang-Test resultierenden Signifikanzwerte p_{sig} für den Vergleich der Ergebnisse, bezogen auf die ursprüngliche Vorgehensweise ohne Datenbearbeitung (ohne Gleichverteilung und vor DiLa-Filterung). Die signifikanten Unterschiede sind grau hinterlegt. Eine Signifikanz²³ liegt dann vor, wenn sich der errechnete Wert unter dem Signifikanzniveau α von 0,05 befindet. Durch die Annahme der Verteilungsfreiheit des Wilcoxon-Vorzeichen-Rang-Tests und der nur kleinen Messreihen müssen die Daten einen bedeutenden Unterschied aufweisen, um als signifikant zu gelten.

Die Gleichverteilung der Daten bewirkt eine signifikante Reduzierung des GG ($p_{\text{sig}}=0,005$), aber auch eine Stärkung der schlecht erkannten Klassen, welche meist die schwach vertretenden Klassen sind. Dies zeigen die restlichen drei Güterwerte DG, GMG und MG. Diese Güterwerte steigen an, wobei sich die GMG ($p_{\text{sig}}=0,009$) und die MG ($p_{\text{sig}}=0,005$) signifikant verbessern (gelb dargestellt). Dies entspricht den Erkenntnissen aus den künstlichen Daten. Durch die Gleichverteilung werden die Trennebenen in Richtung der Majoritätsklassen verschoben, wodurch die Minoritätsklassen besser erkannt werden. Aus den künstlichen Daten resultiert aber auch eine Verzerrung des Modells durch Ausreißer, die durch die Gleichverteilung noch verstärkt wird.

²² Detaillierte Angaben zu den eingestellten Parametern sind in Tabelle C-1 auf Seite 133 zu finden.

²³ siehe Begriffsverzeichnis

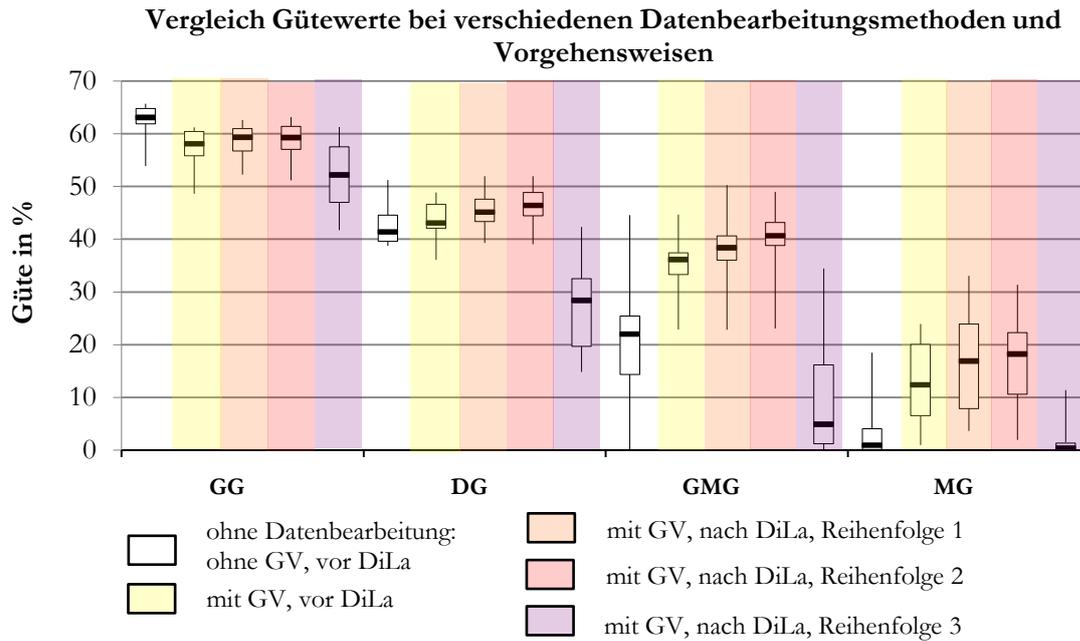


Abbildung 4-13: Kastendiagramm der vier Gütewerte GG, DG, GMG und MG bei verschiedenen Datenbearbeitungsmethoden und Vorgehensweisen - Dabei zeigt Reihenfolge 2 die besten Ergebnisse.

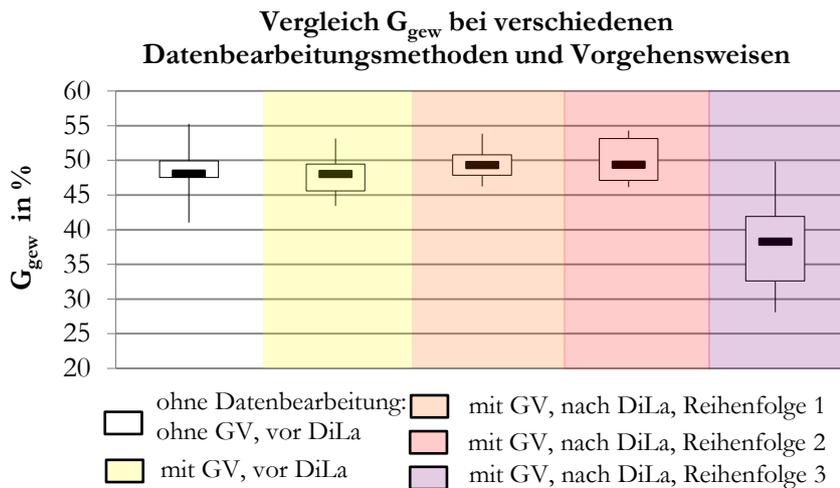


Abbildung 4-14: Kastendiagramm der gewichteten Güte G_{gew} bei verschiedenen Datenbearbeitungsmethoden und Vorgehensweisen - Dabei zeigt Reihenfolge 2 die besten Ergebnisse.

Signifikanzwerte p_{sig} im Vergleich zu: ohne Datenbearbeitung	GG	DG	GMG	MG	G_{gew}
mit GV, vor DiLa	0,005	0,139	0,009	0,005	0,959
mit GV, nach DiLa, Reihenfolge 1	0,005	0,005	0,007	0,007	0,074
mit GV, nach DiLa, Reihenfolge 2	0,013	0,005	0,007	0,007	0,092
mit GV, nach DiLa, Reihenfolge 3	0,005	0,005	0,036	0,398	0,005

Tabelle 4-7: Signifikanzwerte für den Vergleich verschiedener Datenbearbeitungsmethoden und Vorgehensweisen in Bezug auf die Resultate ohne Datenbearbeitung (zweiseitiger Wilcoxon-Vorzeichen-Rang-Test, SN=0,05, signifikante Unterschiede grau hinterlegt)

Demnach sollten die realen Daten, die durch eine Datenlöschung mit anschließender Gleichverteilung bearbeitet werden, bessere Ergebnisse erzielen als mit alleiniger Gleichverteilung. Dies kann durch die Versuche mit realen Daten bestätigt werden. Es wird die DiLa-Filterung angewendet und wiederum die drei Reihenfolgen zur Parameterbestimmung erprobt. Dabei erreicht die Reihenfolge 2 (rot dargestellt), mit einer Parameterbestimmung nach der DiLa-Datenfilterung aber vor der Gleichverteilung, die besten Ergebnisse. Auch bei dieser Vorgehensweise wird, im Vergleich zu „ohne Datenbearbeitung“ (weiß dargestellt), zunächst die GG ($p_{\text{Sig}}=0,013$) signifikant schlechter. Dennoch ist die GG hier besser als durch die alleinige GV (gelb dargestellt). Die Gütwerte DG ($p_{\text{Sig}}=0,005$), GMG ($p_{\text{Sig}}=0,007$) und MG ($p_{\text{Sig}}=0,007$) jedoch werden durch die DiLa-Filterung signifikant verbessert. Nur wenig schlechtere Ergebnisse zeigt Reihenfolge 1, der Parameterbestimmung vor DiLa-Filterung und vor GV (orange dargestellt). Diese Ergebnisse weichen nur leicht von den Erkenntnissen aus den künstlichen Daten ab, wo die Modelle der Reihenfolge 1 und Reihenfolge 2 oft ähnlich sind und nur ein Modell der Reihenfolge 1 besser ist. Wie auch durch die Versuche anhand der künstlichen Daten vermutet, erreicht Reihenfolge 3, die Parameterbestimmung nach DiLa-Filterung und nach GV (violett dargestellt), vergleichsweise sehr geringe Gütwerte.

Obwohl die GG sich signifikant verschlechtert, zeigt die gewichteten Güte G_{gew} mit ($p_{\text{Sig}}=0,092$) in Summe eine nicht signifikante Steigerung der Güte des Klassifikators durch die DiLa-Filterung mittels Reihenfolge 2 (in Abbildung 4-14 und Tabelle 4-7 rot dargestellt).

- Aus den Versuchen mit realen Daten kann geschlussfolgert werden, dass die Parameterbestimmung nach Reihenfolge 2 anderen Vorgehensweisen vorzuziehen ist. Durch den geringeren Rechenaufwand und nur geringe Unterschiede zu Reihenfolge 2 wäre jedoch auch die Reihenfolge 1 vertretbar.
- Auch wenn durch die Gleichverteilung die Robustheit des Klassifikators gesteigert wird, da die Klassifikation der schlechtesten Klassen verbessert wird, sinkt durch die Gleichverteilung alleine zunächst die gewichtete Güte leicht. Erst durch die DiLa-Filterung vor der Gleichverteilung können sowohl die Klassifikation der schlechtesten Klassen, als auch die resultierende gewichtete Güte G_{gew} gesteigert werden und damit ein insgesamt verbesserter SVM-Klassifikator geschaffen werden.
- Die Verteilung der Belehrungsdaten ist ein wichtiger Punkt in der Modellbildung mit SVM. Durch sie lässt sich das Modell in Komplexität und Qualität direkt beeinflussen, was über die SVM-Parameter nur sehr eingeschränkt möglich ist. Insgesamt ist die Gleichverteilung bei stark ungleichverteilten und komplexen Problemen unabdingbar. Jedoch macht dies eine zusätzliche Datenfilterung notwendig. Mit dem neuartigen DiLa-Filter konnte ein wirkungsvoller Algorithmus zur Datenbearbeitung vorgestellt werden.

4.1.3 Wirkung der Homogenität von Daten

Ein weiteres Thema, welches bei realen Daten eine Rolle spielt, ist die Frage der Homogenität der Daten. Im Falle der EEG-Klassifikation heißt das, wie die Belehrungsdatenwahl eines Probanden (homogen) oder mehrerer Probanden (inhomogen) das Modell beeinflusst. Bei den Untersuchungen zu neuronalen Netzen in [Wen05] resultierte die Erkenntnis, dass eine Inhomogenität der Daten negative Auswirkungen auf die entstandene Klassifikatoren hat. Ob dies auch auf Modelle der SVM zutrifft, soll in den folgenden Untersuchungen gezeigt werden.

Es wurden zehn Belehrungsdatensätze (BDS) aus jeweils einem Probandendatensatz (PDS) von B1 (siehe Tabelle A-1 auf Seite 121) erstellt, anschließend zehn Belehrungsdatensätze aus jeweils zwei Probandendatensätzen und zwei Belehrungsdatensätze aus fünf Probandendatensätzen. Dabei wiesen die Belehrungsdatensätze trotz Mischung etwa den gleichen Umfang auf. Die Modelle wurden ansonsten durch einfache Kreuzvalidierung mit den gleichen SVM-Parametern erzeugt und gegen die Datensätze von B2 und B3 (Tabelle A-2 und Tabelle A-3 ab Seite 122) validiert.²⁴ Abbildung 4-15 stellt die Resultate dieser Untersuchung in Form eines Kastendiagramms dar. Tabelle 4-8 zeigt dazu die aus dem zweiseitigen Wilcoxon-Vorzeichen-Rang-Test resultierende Signifikanzwerte p_{Sig} der Ergebnisse in Bezug auf einen homogenen Belehrungsdatensatz (1PDS). Die signifikanten Unterschiede sind dabei grau hinterlegt.

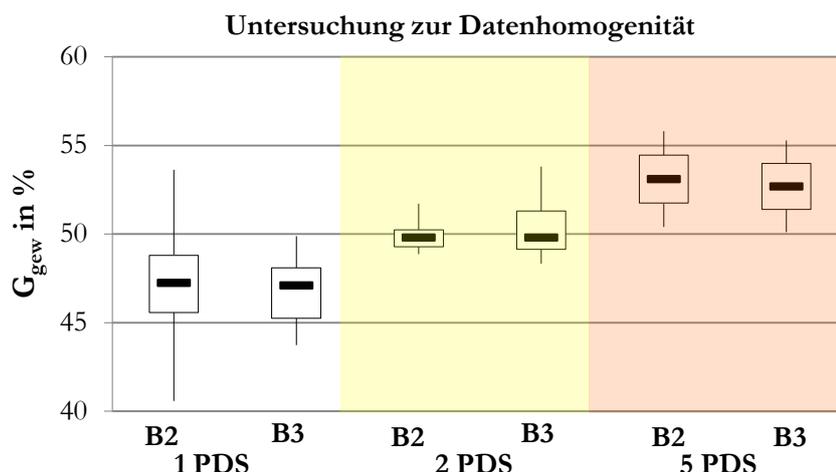


Abbildung 4-15: Kastendiagramm zur Untersuchung der Datenhomogenität: die gewichtete Güte G_{gew} bei Belehrungsdatensätzen aus einem, zwei und fünf Probandendatensätzen (PDS) - Dabei steigt mit wachsender Inhomogenität auch die Klassifikationsgüte.

Signifikanzwerte p_{Sig} der Untersuchung zur Datenhomogenität im Vergleich zu: 1 PDS	G_{gew}	
	B2	B3
2 PDS	0,037	0,005
5 PDS	0,009	0,005

Tabelle 4-8: Signifikanzwerte für den Vergleich homogener Belehrungsdaten (1PDS) zu inhomogenen Belehrungsdaten (zweiseitiger Wilcoxon-Vorzeichen-Rang-Test, $SN=0,05$, signifikante Unterschiede grau hinterlegt)

²⁴ Detaillierte Angaben zu den eingestellten Parametern sind in Tabelle C-2 auf Seite 133 zu finden.

Es ist die gewichtete Güte G_{gew} als Maß für die Gesamtgüte des Modells über die Versuche mit einem (weiß), zwei (gelb) und fünf (orange) Probandendatensätzen zu sehen. Dazu ist jeweils das Ergebnis von B2 und B3 als Validierungsdatsatz abgebildet. Deutlich zeigt sich, dass mit steigender Anzahl der verwendeten Probandendatensätze auch die Güte des Klassifikators steigt, wobei die B3-Datensätze gegenüber der B2-Datensätze etwas schlechter abschneiden. Dabei ist die Verbesserung stets signifikant zum homogenen Belehrungsdatsatz (Tabelle 4-8).

→ Demnach wirken sich inhomogene Daten, anders als bei neuronalen Netzen, auf die SVM nicht negativ aus. Im Gegenteil erhöht ein inhomogener Datensatz die Klassifikationsgüte und erzeugt somit einen robusteren Klassifikator.

4.2 Untersuchung zur Skalierung von Daten

Die Skalierung oder auch Normalisierung der Daten, wie bereits in Abschnitt 3.1.4 ab Seite 13 erläutert, dient der Angleichung des Wertebereichs der Merkmale. So gehen alle Merkmale bei der späteren Belehrung und Klassifikation möglichst gleichwertig ein. Dabei werden die Skalierungsfaktoren anhand der Belehrungsdaten bestimmt und anschließend auf die Validierungsdaten angewendet. Bei der Skalierung kommt üblicherweise eine lineare Skalierung auf den Wertebereich von $[-1, +1]$ oder $[0, 1]$ zur Anwendung. Im einfachsten Fall der Min-Max-Skalierung wird der Maximalwert eines Merkmals $+1$ und der Minimalwert eines Merkmals -1 beziehungsweise 0 gesetzt und alle Werte dazwischen entsprechend skaliert. Treten abweichende Merkmalsdaten in der Belehrungsdatenmenge auf, d. h. Merkmalsvektoren, in denen ein oder mehrere Merkmale extrem von den üblichen Werten dieser Klasse abweichen, haben diese einen starken Einfluss auf die Skalierungsfaktoren. Damit wird die Wirkung der Skalierung, die Merkmale gleichwertig zu machen, reduziert. Skalierungsvarianten, bei denen nicht auf einen festen Wert, sondern auf statistisch beeinflusste Werte skaliert wird, sollen diesen starken Einfluss einzelner abweichender Merkmale auf die Skalierung mindern und so zu einem besseren Klassifikationsergebnis führen. Es findet somit eine Glättung des Wertebereichs der skalierten Merkmale statt.

Bei biologischen Datensätzen stellt sich die Frage der Skalierungsstrategie, ob eine Skalierung probandenspezifisch anzuwenden ist oder ob alle Belehrungsdatsätze zusammen skaliert werden können. Da die Probandendatsätze untereinander abweichen könnten, würden entsprechend die Skalierungsfaktoren abweichen. Zur Untersuchung der Skalierungsstrategie kam der Datensatz 970502 von B1 aufgrund der typischen Klassenverteilung zum Einsatz. Dieser Datensatz wurde durch die Min-Max Methode skaliert und ein SVM-Modell erzeugt. Die Skalierung der restlichen neun Datensätze von B1 erfolgte zum einen durch eine feste Skalierung, d. h. mit den gleichen Skalierungsfaktoren wie der Belehrungsdatsatz 970502. Zum anderen wurden für die neun Datensätze jeweils neue Skalierungsfaktoren ermittelt und eingesetzt, dies beschreibt die probandenspezifische Skalierung.²⁵ In Abbildung 4-16 ist das Kastendiagramm mit der resultierenden gewichteten Güte G_{gew} zur Skalierungsstrategien zu sehen. Tabelle 4-9 zeigt dazu den aus dem zweiseitigen Wilcoxon-Vorzeichen-Rang-Test berechneten Signifikanzwert p_{Sig} .

²⁵ Detaillierte Angaben zu den eingestellten Parametern sind in Tabelle C-3 auf Seite 134 zu finden.

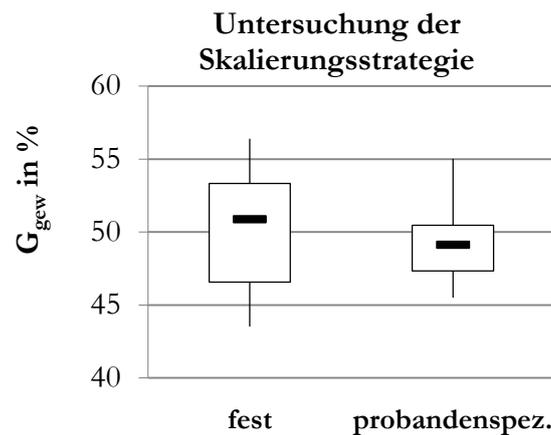


Abbildung 4-16: Kastendiagramm zur Untersuchung der Skalierungsstrategie: die gewichtete Güte G_{gew} - Dabei zeigt die feste Skalierung einen leicht höheren Median.

Signifikanzwert p_{Sig} zur Untersuchung der Skalierungsstrategie	G_{gew}
	0,260

Tabelle 4-9: Signifikanzwert für den Vergleich der Skalierungsstrategien (zweiseitiger Wilcoxon-Vorzeichen-Rang-Test, $SN=0,05$)

Die feste Skalierung schneidet mit einem höheren Median besser ab, weist aber auch einer höheren Streuung auf. Der Unterschied zwischen fester und probandenspezifischer Skalierung ist mit $p_{\text{Sig}}=0,26$ nicht signifikant. In den folgenden Versuchen wird aufgrund dieser Erkenntnis die Skalierungsstrategie der festen Skalierung angewendet.

Des Weiteren ist die Frage der Skalierungsmethode zu klären. Im Folgenden werden vier Skalierungsmethoden vorgestellt und anhand realer EEG-Datensätze untersucht.

1. Die Min-Max-Methode ist eine lineare Skalierungsmethode, bei der der Maximalwert eines Merkmals +1 und der Minimalwert eines Merkmals -1 gesetzt wird.
2. Die 10-90-Methode ist ebenfalls eine lineare Skalierungsmethode, bei der der 0,9 Quartil der Merkmalswerte und alle Werte darüber auf +1 sowie das 0,1 Quartil der Merkmalswerte und alle Werte darunter auf -1 gesetzt werden. Die Werte dazwischen werden entsprechend skaliert. Somit haben einzelne Ausreißer keinen Einfluss auf die Skalierungsfaktoren, da jeweils 10 % an den Rändern des Merkmalsraumes von der Skalierung ausgeschlossen sind.
3. Die Standardisierung ist eine statistische Skalierungsmethode, bei der der Mittelwert gleich null gesetzt wird [AH01]. Für diese Methode ergibt sich ein Wertebereich von $[\bar{x} - \sigma, \bar{x} + \sigma]$, wobei \bar{x} den arithmetischen Mittelwert und σ die Standardabweichung darstellt. Die Werte werden skaliert nach:

$$x_s = \frac{x_R - \bar{x}_R}{\sigma} \quad (4.8)$$

4. Die Vektorlängen-Methode skaliert den Merkmalsvektor auf seine Länge nach [GB01]:

$$x_{is} = \frac{x_{iR}}{\|x\|} \quad (4.9)$$

Bei dieser Methode ergeben sich somit für jeden Datenpunkt unterschiedliche Skalierungsfaktoren.

Für die Untersuchungen zu den Skalierungsmethoden wurden die zehn Probandendatensätze von B1 zur Belehrung und die Datensätze B2 und B3 zur Validierung eingesetzt.²⁶ In Abbildung 4-17 ist das Kastendiagramm der resultierenden gewichteten Güte der vier Skalierungsmethoden dargestellt, in Tabelle 4-10 die zugehörigen Signifikanzwerte p_{sig} aus dem zweiseitigen Wilcoxon-Vorzeichen-Rang-Test in Bezug auf die Min-Max-Methode.

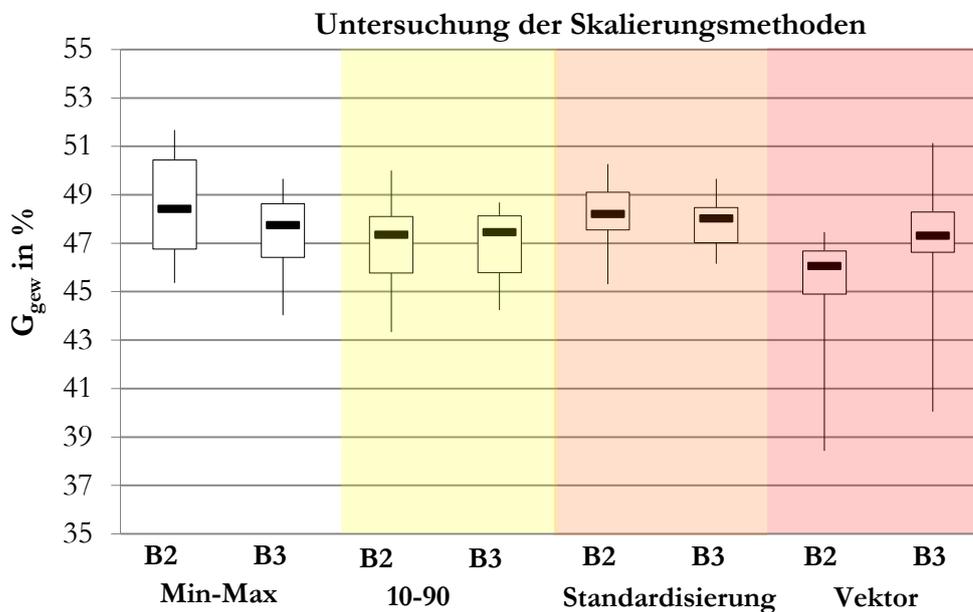


Abbildung 4-17: Kastendiagramm zur Untersuchung der Skalierungsmethode: die gewichtete Güte G_{gew} - Dabei zeigt die Standardisierung einen hohen Median und eine geringe Streuung.

Signifikanzwerte p_{sig} zur Untersuchung der Skalierungsmethode im Vergleich zu: Min-Max	B2	B3
10-90	0,037	0,114
Standardisierung	0,139	0,799
Vektor	0,013	0,508

Tabelle 4-10: Signifikanzwerte für den Vergleich der Skalierungsmethoden im Bezug auf die Min-Max-Methode (zweiseitiger Wilcoxon-Vorzeichen-Rang-Test, $SN=0,05$, signifikante Unterschiede grau hinterlegt)

²⁶ Detaillierte Angaben zu den eingestellten Parametern sind in Tabelle C-4 auf Seite 134 zu finden.

Insgesamt sind zwischen den Methoden keine enormen Unterschiede ersichtlich, wodurch auch kaum signifikante Werte in Tabelle 4-10 zu finden sind. Die vorhandenen signifikanten Unterschiede in Tabelle 4-10 stellen Verschlechterungen dar. Die Methode der Vektorlänge zeigte trotz geringer Streuung vor allem nach unten Ausreißer. Durch einen geringen Median schneidet die Vektorlängenmethode auch insgesamt am schlechtesten ab. Die 10-90-Methode erreicht durch geringere Mediane und relativ hohe Streuungen den dritten Rang. Die Min-Max Methode ist aufgrund des hohen Medianes aber der ebenfalls hohen Streuung nur als zweit beste Methode zu bewerten.

→ Tatsächlich ist die Methode der Standardisierung zur Skalierung für die verwendeten EEG-Daten wegen des hohen Medianes und der geringen Streuung der Ergebnisse vorzuziehen.

Die hier verglichenen Ergebnisse umfassen die zehn Ergebnisse aus den zehn Probandendatensätzen von B1. Diese geringe Menge an Daten erfordert einen bedeutenden Unterschied zweier Messreihen, um als signifikant gelten zu können. Im Falle der Untersuchung der Skalierung konnten aber bei diesen geringen Daten nur leichte Verbesserungen erreicht werden, wodurch diese nicht signifikant sind.

4.3 Untersuchung der Merkmalsvektoren

Es wurde in dieser Arbeit keine vollständige Merkmalsselektion vorgenommen, da dies bereits in [Wen05] für die Schlafklassifikation mittels neuronaler Netze erfolgte und die Vergleichbarkeit erhalten werden soll. Lediglich der Einfluss der Merkmalsanzahl und der Einfluss irrelevanter Merkmale auf die Klassifikation mit SVM soll im Folgenden gezeigt werden.

Einige Klassifikatoren, wie die neuronalen Netze, reagieren empfindlich auf zu viele Merkmale [Wen05]. Im Falle der neuronalen Netze wird die Netzwerkstruktur durch viele Merkmale erhöht. Dies beeinflusst die Generalisierungsfähigkeit nachteilig.

Bei der Methode SVM erhöhen viele Merkmale die Dimension, wobei die SVM mit hohen Dimensionen umgehen kann [Fis07]. Dennoch macht es Sinn, Merkmale, die für die Klassifikation irrelevant sind, zu identifizieren und die Anzahl der Merkmale zu reduzieren. Damit wird zum einen der Berechnungsaufwand reduziert und zum anderen das Risiko gesenkt, durch verrauschte oder abweichende Merkmale die Klassifikationsleistung zu mindern [NT10]. Inwieweit dies bei der vorliegenden EEG-Schlafklassifikationsaufgabe eine Rolle spielt, soll in den folgenden Untersuchungen gezeigt werden.

Insgesamt wurden vier Datensatzvarianten mit 65, 62, 22 und 12 Merkmalen verglichen. Außer den Datensätzen mit den 62 validen Merkmalen (siehe Tabelle A-4 auf Seite 127) wurden weitere drei invalide Merkmale (insgesamt mit 65 Merkmale) erzeugt. Valide Merkmale weisen einen Bezug zur Klasse auf, invalide Merkmale dagegen zeigen unabhängig von der Klasse gleiche oder ähnliche Werte.²⁷ Zur Relevanzbewertung der Merkmale kam der Operator „Weight by SVM“ des frei zugänglichen Programms „Rapid Miner“ [Rap13] zum Einsatz. Dieser ersetzt sicher nicht eine umfassende Untersuchung zur Merkmalsselektion, sondern kann lediglich Hinweise auf

²⁷ Die Definitionen der Begriffe valide und invalide Merkmale sind im Begriffsverzeichnis zu finden.

relevante Merkmale liefern. Für den zu untersuchenden Einfluss der Merkmalsanzahl ist dies jedoch ausreichend. Die drei invaliden Merkmale konnten durch den Operator sicher erkannt werden. So wurden 22 der 62 Merkmale anhand der vom Programm ermittelten Werte ausgewählt, um die Klassifikationsergebnisse mit den Ergebnissen der 62 Merkmale zu vergleichen. Des Weiteren dienten Datensätze mit 12 Merkmalen, die bereits im Schlaftauswertungsgerät Quisi [BSRW⁺98] verwendet wurden, als weitere Versuchsdaten.

Für die Untersuchungen dienten die zehn Probandendatensätze von B1 als Belehrungsdaten. Mit jedem Probandendatensatz von B1 wurde ein Model erzeugt und dessen Parameter durch einen Rastersuchalgorithmus mittels der restlichen neun Probandendatensätze von B1 als Testdaten ermittelt. Zur Validierung dienten die Datensätze B2 und B3.²⁸

In Abbildung 4-18 ist das Kastendiagramm des Vergleichs der Klassifikation mit 65, 62, 22 und 12 Merkmalen zu sehen, in Tabelle 4-11 die zugehörigen Signifikanzwerte aus dem zweiseitigen Wilcoxon-Vorzeichen-Rang-Test in Bezug auf die Ergebnisse mit 62 Merkmalen.

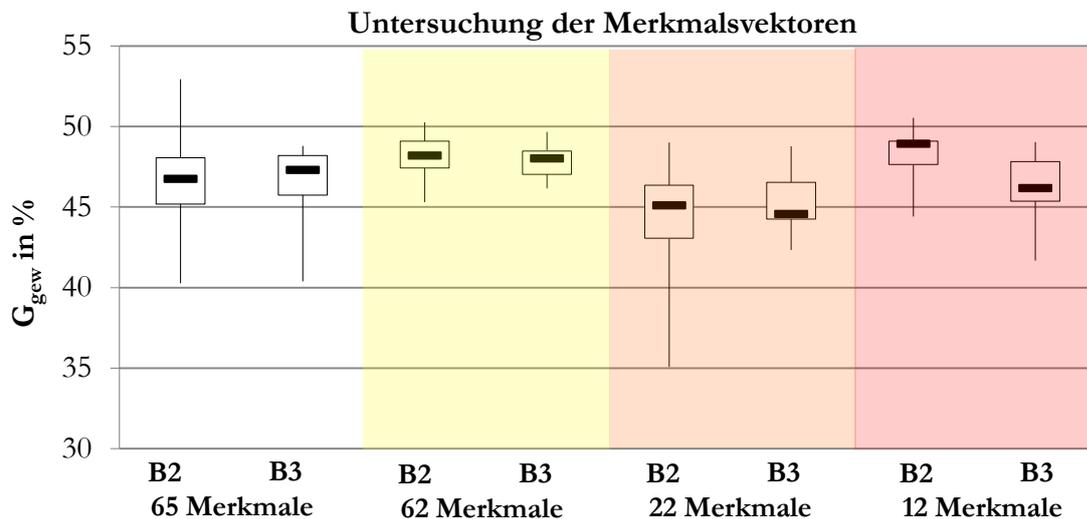


Abbildung 4-18: Kastendiagramm zur Untersuchung der Merkmalsvektoren: die gewichtete Güte G_{gew} – Dabei wird der negative Einfluss von invaliden Merkmalen bei 65 Merkmalen (weiß) deutlich. Außerdem zeigt sich die SVM gegen eine hohe Merkmalsanzahl unempfindlich.

Signifikanzwerte p_{Sig} zur Untersuchung der Merkmalsvektoren im Vergleich zu: 62 Merkmalen	B2	B3
65 Merkmale	0,074	0,009
22 Merkmale	0,005	0,007
12 Merkmale	0,508	0,007

Tabelle 4-11: Signifikanzwerte zur Untersuchung der Merkmalsvektoren im Bezug auf 62 Merkmale (zweiseitiger Wilcoxon-Vorzeichen-Rang-Test, $\text{SN}=0,05$, signifikante Unterschiede grau hinterlegt)

²⁸ Detaillierte Angaben zu den eingestellten Parametern sind in Tabelle C-5 auf Seite 134 zu finden.

Zunächst kann aus dem Vergleich der 65 Merkmale (inklusive drei invalider Merkmale) gegen die 62 validen Merkmale die Erkenntnis gewonnen werden, dass diese drei invaliden Merkmale tatsächlich einen negativen Einfluss auf die Modellbildung und spätere Klassifikation haben. Insgesamt wird die Klassifikation durch die invaliden Merkmale leicht schlechter, die Klassifikation der B3-Daten sogar signifikant schlechter ($p_{\text{Sig}}=0,009$). Weiterhin tritt eine höhere Streuung der Klassifikationsergebnisse auf. Durch die Verringerung der Merkmale auf 22 wird eine signifikante Verschlechterung der Güte bewirkt ($p_{\text{Sig}}=0,005$ für B2 und $p_{\text{Sig}}=0,007$ für B3). Durch die Verringerung auf 12 Merkmale kann für die B2-Daten mit $p_{\text{Sig}}=0,508$ nur eine leichte, aber keine signifikante Verbesserung der Klassifikationsgüte in Bezug zur Klassifikation mit 62 Merkmalen erreicht werden. Die Klassifikation der B3-Daten wird hingegen mit $p_{\text{Sig}}=0,007$ signifikant schlechter.

Alle Versuche mit reduzierten Merkmalsvektoren zeigen ähnliche bzw. schlechtere Klassifikationsergebnisse. Was jedoch nicht heißt, dass die Klassifikationsgüte nie durch eine Merkmalsreduktion steigen kann. Durch eine gezielte Selektion der geeigneten Merkmale ist dies trotz der Resultate dennoch denkbar.

→ Die Klassifikationsmethode der SVM ist folglich gegenüber einer hohen Anzahl an Merkmalen unempfindlich. Jedoch wirken sich invalide Merkmale negativ aus.

Damit ist eine ausführliche Merkmalsauswahl nicht nur zur Rechenaufwandsreduktion, sondern auch zur Qualitätssteigerung des Klassifikators durchaus sinnvoll, aber nicht zwingend notwendig, solange die Merkmale valide sind. Da auf eine ausführliche Merkmalsauswahl in dieser Arbeit verzichtet wurde, werden in weiteren Versuchen, soweit das die der Rechenaufwand zulässt, die Merkmalsätze mit 62 Merkmalen (siehe Tabelle A-4 auf Seite 127) verwendet.

4.4 Belehrung der SVM

Wie bereits in Abschnitt 4.1 erläutert, ist die Wahl der Belehrungsdaten von elementarer Bedeutung für die spätere Klassifikationsgüte und Robustheit eines Klassifikators. Hierbei sollten einige Grundsätze eingehalten werden.

→ Es muss auf die Gleichverteilung der Belehrungsepochen pro Klasse geachtet werden, um eine gleichwertige Klassifikation aller Klassen zu gewährleisten.

→ Untypische Daten verzerren das Modell und wirken sich negativ auf die Klassifikationsgüte aus, besonders wenn diese in eine Gleichverteilung mit einbezogen werden.

Beide Faktoren wurden bereits in Abschnitt 4.1 untersucht. Weiterhin sollten die Belehrungsepochen möglichst alle auftretenden Muster dieser Klasse widerspiegeln, was sich bei biologischen Signalen als äußerst schwierig gestaltet, da durch die interindividuelle Streuung nie alle Muster bekannt sein können. Hinzu kommt, dass sich die Daten gesunder Probanden deutlich von denen kranker Probanden unterscheiden. Bei der Belehrung sollte deshalb bereits die spätere Anwendung im Blick sein. Um aber mit den vorhandenen Datensätzen ein optimales Modell erzielen zu können, entstand die Idee der Belehrungsdatenoptimierung durch einen Datenwahlalgorithmus, der zufällig oder nach einer geeigneten Strategie Belehrungsdatensätze

erprobt. Das Programmablaufdiagramm eines solchen Datenwahlalgorithmus ist in Abbildung B-3 auf Seite 131 gezeigt. Der Belehrungsdatensatz BDS wird dabei so lange geändert, wie eine Verbesserung des Belehrungsgrades BG eintritt. Erfolgt keine Verbesserung, wird die Änderung des Belehrungsdatensatzes verworfen. Um eine Überanpassung an die Belehrungsdaten zu vermeiden, wird alle fünf Iterationen die Güte mittels eines Validierungsdatensatzes VDS ermittelt. Im Falle einer Verschlechterung werden die letzten fünf Änderungen des Belehrungsdatensatzes verworfen. Eventuell könnte auch noch ein zweiter Validierungsdatensatz eingesetzt werden, um so die Generalisierungsfähigkeit des resultierenden SVM-Modells zu erhöhen. Die Wirkung und Effektivität eines solchen Algorithmus und auch die Art der Belehrungsdatenmodifikation ist jedoch nicht Gegenstand dieser Arbeit.

Vor der Belehrung der SVM müssen die zu verwendende SVM-Art, die Kernel-Funktion und die sich daraus ergebenden Parameter festgelegt werden.²⁹ Dies gestaltet sich schwierig, da hierfür keine analytische Methode bekannt ist. Diese Festlegungen können nur empirisch für die jeweilige Anwendung getroffen werden. Die Untersuchungen zur Auswahl der geeigneten SVM-Art (siehe Abschnitt 3.1.9) für die gestellte Klassifikationsaufgabe konnte im Rahmen dieser Arbeit nicht bearbeitet werden. Für alle weiteren Analysen fand daher stets die C-SVC Verwendung. Die Parameteroptimierung, obwohl auch nur empirisch ermittelbar, kann jedoch mittels verschiedener Vorgehensweisen erfolgen. Diese sollen im Folgenden dargelegt werden.

4.4.1 Multikriterielle Parameteroptimierung mittels evolutionärer Algorithmen

Trotz intensiver Forschung und der Entwicklung von immer leistungsfähigeren Methoden und Algorithmen müssen bei jedem Klassifikator bestimmte Parameter empirisch ermittelt werden. Bei der SVM ist die Anzahl dieser Parameter abhängig von der gewählten SVM-Art und Kernel-Funktion, wobei auch die Art und der Kernel nur empirisch an einem konkreten Problem bestimmbar sind. Zur empirischen Bestimmung von Parametern können folgende Methoden angewendet werden:

- die zufällige Suche,
- die Rastersuche und
- die evolutionäre Suche.

Die zufällige Suche ist das zufällige Testen von Parameterkombinationen. Sie stellt keine sichere Methode zum Finden des Optimums innerhalb einer akzeptablen Zeitspanne dar und wird daher im wissenschaftlichen Bereich nicht genutzt.

Bei der Rastersuche (RS) werden gezielt alle Parameterkombinationen in einem Raster mit einer bestimmten Auflösung getestet. Damit wird innerhalb der gewählten Auflösung und dem gewählten Suchbereich auf jeden Fall das Optimum gefunden. Daraus ergibt sich aber auch ein Nachteil, denn je nach Parameteranzahl, Auflösung und Suchbereich kann diese Methode sehr lange dauern. Erhöht man die Auflösung, steigt die Anzahl der Parameterkombinationen und damit der Berechnungsaufwand exponentiell. Ähnlich verhält es sich mit dem Suchbereich, dem

²⁹ siehe Abbildung 1-1: Gesamtübersicht des SVM-Generierungs- und Klassifikationsprozesses auf Seite 4

Wertebereich, in dem die Parameter gesucht werden. Die Anzahl der Parameterkombinationen PK ergibt sich aus:

$$PK = \prod_{i=2}^N W_i, \quad (4.10)$$

wobei W_i die Anzahl der möglichen Werte der Parameter i und N die Anzahl der Parameter angibt. Ist die Anzahl der möglichen Werte für alle Parameter gleich, lässt sich die Gleichung zu:

$$PK = W^N \quad (4.11)$$

vereinfachen. Dies sollen die Abbildung 4-19 und Abbildung 4-20 verdeutlichen (unter Beachtung der logarithmisch eingeteilten y-Achse). Aus diesem Grund ist die Methode der Rastersuche nur für Probleme mit überschaubaren Parameterkombinationen geeignet.

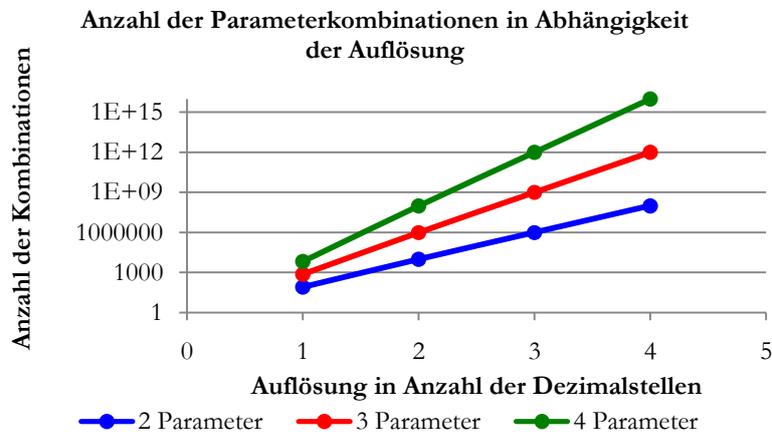


Abbildung 4-19: Diagramm für die Anzahl der Parameterkombinationen in Abhängigkeit der Auflösung für die Rastersuche - Dabei besteht eine exponentielle Abhängigkeit.

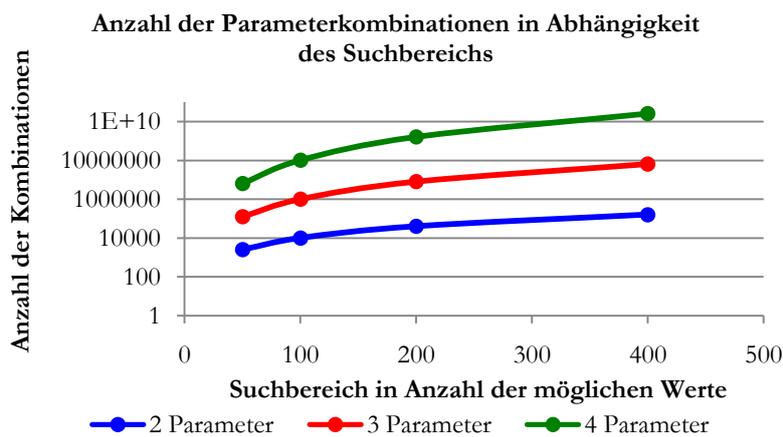


Abbildung 4-20: Diagramm für die Anzahl der Parameterkombinationen in Abhängigkeit des Suchbereichs für die Rastersuche - Dabei besteht eine exponentielle Abhängigkeit.

Die evolutionäre Suche bedient sich der Methode der Evolution. Zu Beginn werden zufällig Parameterkombinationen erzeugt, die Individuen der Startpopulation und ihre Qualität, die Fitness, geprüft. Ein Individuum besteht somit aus den Parametern und der damit erreichten Klassifikationsgüte. Erreichen die Individuen gute Ergebnisse, dürfen sie weiter bestehen (Selektion) und können durch Kreuzung³⁰ (engl. crossover) mit anderen Individuen oder durch Mutation neue Individuen mit ihren Anlagen erschaffen. Auf diese Weise entsteht Generation um Generation, bis ein Abbruchkriterium erfüllt oder eine maximale Anzahl an Generationen erreicht ist. Eine Generation beschreibt die Population zu einem bestimmten Zeitpunkt.³¹ Damit ist die evolutionäre Suche das gezielte Testen bestimmter Parameterkombinationen, die durch Anwendung evolutionärer Aspekte Erfolg versprechen. Der Vorteil gegenüber der Rastersuche gerade bei sehr vielen möglichen Parameterkombinationen ist die Geschwindigkeit, da nicht alle Kombinationen getestet werden müssen. Nachteilig ist, dass der Prozess der evolutionären Suche stark von stochastischen Vorgängen abhängig ist und somit jeder Suchvorgang anders verläuft. Damit können unterschiedliche Lösungen zu unterschiedlichen Zeitpunkten im Algorithmus gefunden werden. Weiterhin wird der Algorithmus wiederum durch empirisch ermittelte Parameter eingestellt. Dennoch ist die evolutionäre Suche in der Wissenschaft eine weit verbreitete und noch immer weiterentwickelte Methode zur Parametersuche. Der Aufwand ist bei der Rastersuche von Auflösung, Suchbereich und der Anzahl der Parameter abhängig. Sie steigt, wie aus Abbildung 4-19 und Abbildung 4-20 abzuleiten, exponentiell an. Der Aufwand der evolutionären Suche hingegen zeigt Abhängigkeiten von der Anzahl der Individuen pro Generation und von der Maximalanzahl der Generationen, die vom Anwender gewählt werden und den stochastischen Einfluss bestimmen. Damit ist die evolutionäre Suche verglichen mit der Rastersuche vor allem bei komplexen Problemen in der Lage, schneller zu vergleichbaren Resultaten zu gelangen.

Die Beurteilung der Qualität eines Individuums kann im Hinblick auf mehrere Kriterien erfolgen, damit ergeben sich multikriterielle Probleme. Die multikriterielle Optimierung bietet nicht nur ein Ergebnis, sondern mehrere pareto-optimale Ergebnisse. Dies bedeutet unter Einfluss der Kriterien „gleich-gute“ Ergebnisse. Dies ist für die Schlafklassifikation äußerst wichtig, da nicht nur die Gesamt-Klassifikationsgüte (GG) betrachtet werden muss, sondern auch die Klassifikation der am schlechtesten erkannten Klassen (meist die schwach vertretenen Klassen) oder eine minimale Vertauschung einiger Klassen für die Qualität des Klassifikators wichtig sind.

Um in möglichst kurzer Zeit das oder ein Optimum mit bis zu drei SVM-Parametern in hoher Auflösung zu finden, wurde ein multikriterieller genetischer Algorithmus, der SIGA (Special Immigrative multi-criteria Genetic Algorithm), für den Anwendungsfall der Schlafklassifikation entworfen [KTBSW⁺10]. Im Folgenden soll zunächst die Theorie der evolutionären Algorithmen (EA) umrissen und anschließend der SIGA sowie der zugrunde liegende Non-dominated Sorting Genetic Algorithm (NSGA)-II vorgestellt werden. Abschließend zeigen experimentelle Ergebnisse die Leistungsfähigkeit des SIGA.

³⁰ auch Rekombination genannt

³¹ Detaillierte Definitionen zu den Begriffen der evolutionären Algorithmen sind im Begriffsverzeichnis zu finden.

Die evolutionären Algorithmen werden in:

- Evolutionäre Programmierung,
- evolutionäre Strategien [Rec73][Sch77] und
- genetische Algorithmen (GA)[Gol89]

unterteilt [Wal12][Bäc96]. Obgleich auch Mischformen dieser drei Arten möglich sind.³²

Nach dem Vorbild der Natur spricht man im Zusammenhang mit evolutionären Algorithmen, wie auch in der Biologie, vom Geno- und Phänotyp der Individuen. Der Genotyp enthält die genetische Information der Individuen, die kodierten Parameter. Der Phänotyp beschreibt das Erscheinungsbild der Individuen, die dekodierte Information des Genotyps.

Die evolutionäre Programmierung ist durch reellwertige Parameter und skalierte Fitnesswerte geprägt. Weiterhin findet keine Kreuzung statt. Das Hauptaugenmerk liegt auf der Mutation. Zur Selektion werden alle Individuen nach ihrem Fitnesswert sortiert und die besten Individuen für die nächste Generation gespeichert.

Die evolutionären Strategien weisen ebenfalls reellwertige Parameter, jedoch keine skalierten Fitnesswerte auf. Neben verschiedenster Kreuzungsvarianten von mindestens zwei Individuen findet die Mutation als genetischer Operator Anwendung. Selektiert wird eine feste Anzahl an Individuen.

Die genetischen Algorithmen arbeiten im Gegensatz zur evolutionären Programmierung und den evolutionären Strategien nicht mit reellen Werten, sondern nach biologischem Vorbild mit binär codierten Werten. Im genetischen Algorithmus tritt die Kreuzung als genetischer Operator in den Vordergrund und die Mutation entgegen der anderen beiden Formen zurück.

Der in dieser Arbeit vorgestellte Algorithmus SIGA arbeitet auf Basis reeller Parameterwerte und widerspricht damit der klassischen Definition des genetischen Algorithmus. Tatsächlich vereint der SIGA Merkmale der evolutionären Programmierung und der genetischen Algorithmen. Da er sowohl die Mutation als auch die Kreuzung verwendet, lässt er sich zu den genetischen Algorithmen zuordnen. Das Prinzip der Selektion ähnelt jedoch eher dem der evolutionären Programmierung. Der SIGA [KTBSW⁺10] basiert auf dem von [DPAM00] veröffentlichten NSGA-II. Die Wahl des NSGA-II basiert auf den in [Wal12] gewonnenen Ergebnissen bei der multikriteriellen Optimierung von Fuzzy Regelwerken zur automatischen Anästhesie-Klassifikation anhand des EEG. Der NSGA-II soll im Folgenden beschrieben und anschließend sollen die für den SIGA vorgenommenen Änderungen erläutert werden.

Der NSGA-II ist einer der multikriteriellen evolutionären Algorithmen, die in [Wal12] für eine ähnliche Anwendung, die automatische Klassifikation von Narkosestadien anhand des EEG mittels unscharfer Regelwerke, untersucht wurden. Dieser Algorithmus zeichnet sich durch eine nicht-dominierende Rangsortierung aus, wodurch nach pareto-optimalen Lösungen sortiert wird. Zur Erhöhung der Konvergenz, d. h. die Erhöhung der Güte der resultierenden Lösungen, kommt das Eliteprinzip zum Einsatz, wodurch dominante Individuen überleben. Zur Erhöhung der Diversität, d. h. die Erhöhung der Breite der pareto-optimalen Ergebnisse, ist die

³² Detaillierte Definitionen zu den Begriffen der evolutionären Algorithmen und multikriteriellen Optimierung sind im Begriffsverzeichnis zu finden.

Nischentechnik der Crowding-Distanz vorgesehen. Weiterhin erfolgt eine Wettkampfselektion, bei der sich die Individuen mit der höchsten Fitness durchsetzen. Auch durch dieses Prinzip soll die Konvergenz erhöht werden. Aufgrund von experimentellen Ergebnissen sprach der Autor in [Wal12] eine Empfehlung für diesen Algorithmus aus.

Der NSGA-II arbeitet mit drei Populationen:

- Der Eltern-Population (EP),
- der Nachkommen-Population (NP) und
- der Kombination aus Eltern und Nachkommen (KP).

Zur Einteilung einer Population in Fronten wird der Non-dominated Sorting-Ansatz genutzt. Das heißt, alle Individuen, die sich nicht dominieren, also indifferent zueinander sind, werden einer Front zugeordnet. Dabei gilt:

- Individuum I_2 dominiert Individuum I_1 ($I_1 \preceq I_2$), wenn mindestens ein Kriterium von I_2 besser ist als von I_1 und alle anderen gleich sind³³, und
- Individuum I_2 ist indifferent zu Individuum I_1 ($I_1 \sim I_2$), wenn einige Kriterien besser und einige schlechter sind.

Wenn die Individuen indifferent sind, sind die durch die Individuen repräsentierten Lösungen pareto-optimal.³⁴

Um auch innerhalb einer Front die Individuen einteilen zu können, wird die Crowding-Distanz CD definiert:

$$CD_i = \sum_{j=1}^m |I_{j,i+1} - I_{j,i-1}|, \quad (4.12)$$

wobei I ein Individuum und m die Anzahl der Kriterien darstellt (siehe Abbildung 4-21).

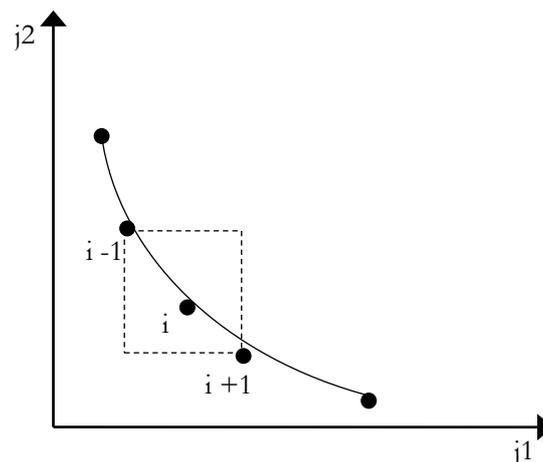


Abbildung 4-21: Darstellung zur Berechnung der Crowding-Distanz [DPAM00]

³³ Hierbei handelt es sich um die Definition der „schwachen Dominanz“. Die starke Dominanz wird, da sie sehr unwahrscheinlich ist, nicht betrachtet.

³⁴ Detaillierte Definitionen zu den Begriffen der evolutionären Algorithmen sind im Begriffsverzeichnis zu finden.

Die Crowding-Distanz eines Individuums CD_i ist der Abstand zwischen den Nachbarindividuen eines Individuums I_i aufsummiert über alle Kriterien. Dabei ist auf gleiche Wertebereiche aller Kriterien zu achten, da ansonsten Kriterien mit hohen Wertebereichen andere Kriterien dominieren würden. Im Falle unterschiedlicher Wertebereiche bietet sich eine Skalierung an. Die Randindividuen einer Front erhalten zur Ermittlung der Crowding-Distanz CD einen unendlichen Wert und werden damit immer ausgewählt. Somit wird gewährleistet, dass die gewählten Individuen über die ganze Pareto-Front verteilt sind. In Abbildung 4-21 ist eine solche Pareto-Front und die Berechnung der Crowding-Distanz am Beispiel zweier Kriterien gezeigt. Die Punkte, die jeweils ein Individuum mit zwei Kriterien darstellen, liegen alle in einer Front und sind damit pareto-optimal.

Der gesamte Ablauf des NSGA-II ist in Abbildung 4-22 graphisch in Form eines Programmablaufdiagramms dargestellt. Nachdem eine Startpopulation zufällig erzeugt wurde, beginnt die Generationenschleife so lange zu laufen, bis das Abbruchkriterium, wie zum Beispiel eine Mindestgüte oder der maximale Generationszähler, erreicht wird. Zu Beginn der Generationenschleife werden die Populationen EP_{gen} und NP_{gen} in KP_{gen} kombiniert und alle Fronten von KP_{gen} gesucht. Je nach Front und Crowding-Distanz selektiert der Algorithmus die besten Individuen für die nächste Eltern-Population EP_{gen+1} , bis diese vollständig gefüllt ist. Besondere Aufmerksamkeit gebührt der letzten Front, die nicht mehr komplett in die Population EP_{gen+1} passt. Die Individuen dieser Front werden nach ihrer zuvor berechneten Crowding-Distanz sortiert und so nur die besten Individuen in die nächste Eltern-Population übernommen. Durch Kreuzung, der Vermischung von Eigenschaften zweier dieser Eltern-Individuen, und Mutation, der zufälligen Abweichung einer Eigenschaft eines Individuums, werden nun Individuen für die neue Nachkommen-Population NP_{gen+1} geschaffen.

Der speziell für die Parameteroptimierung der SVM zur Schlafklassifikation entwickelte SIGA ist in Abbildung 4-23 in Form eines Programmablaufdiagramms abgebildet. Dieser wurde programmtechnisch in C realisiert und optimiert nach mehreren Kriterien, die zur Bewertung der Güte der Parameter herangezogen werden.

Die Kriterien sind:

1. die Gesamt-Klassifikationsgüte (GG)³⁵,
2. die Durchschnittliche Klassen-Klassifikationsgüte (DG)³⁶,
3. die Geometrisch-Mittlere Klassen-Klassifikationsgüte (GMG)³⁷,
4. die Minimale Klassen-Klassifikationsgüte (MG)³⁸ und
5. alle Elemente der Vertauschungsmatrix (bei maximal acht Klassen: 64 Kriterien).

Damit ergibt sich eine maximale Gesamtanzahl der Kriterien von 68. Damit gehört der SIGA wie der NSGA-II zu den Multi-objective Evolutionary Algorithm (MOEA).

³⁵ siehe Gleichung (3.47) auf Seite 34

³⁶ siehe Gleichung (3.48) auf Seite 34

³⁷ siehe Gleichung (3.49) auf Seite 34

³⁸ siehe Gleichung (3.50) auf Seite 34

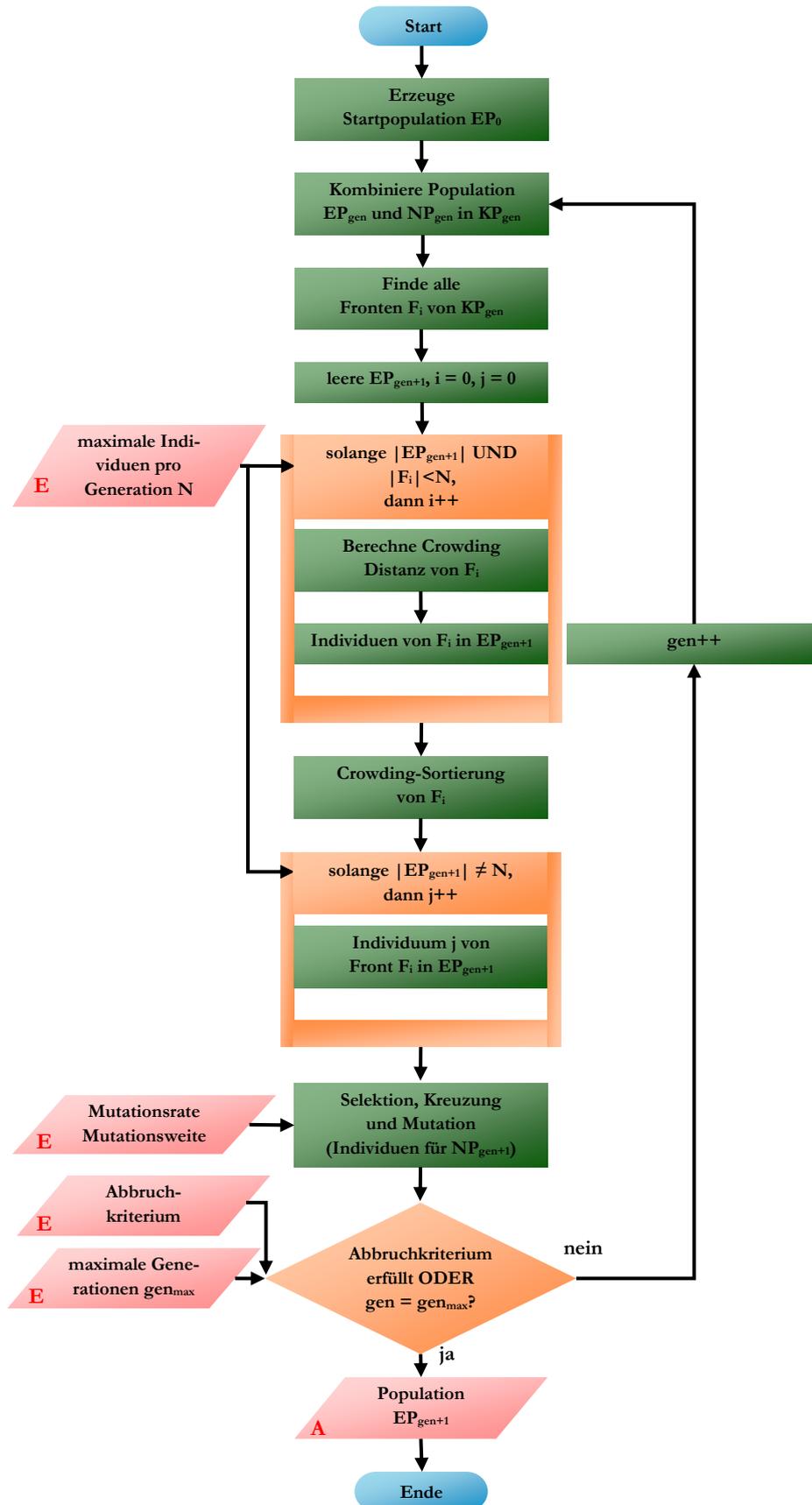


Abbildung 4-22: Programmablaufdiagramm des NSGA-II von [DPAM00]

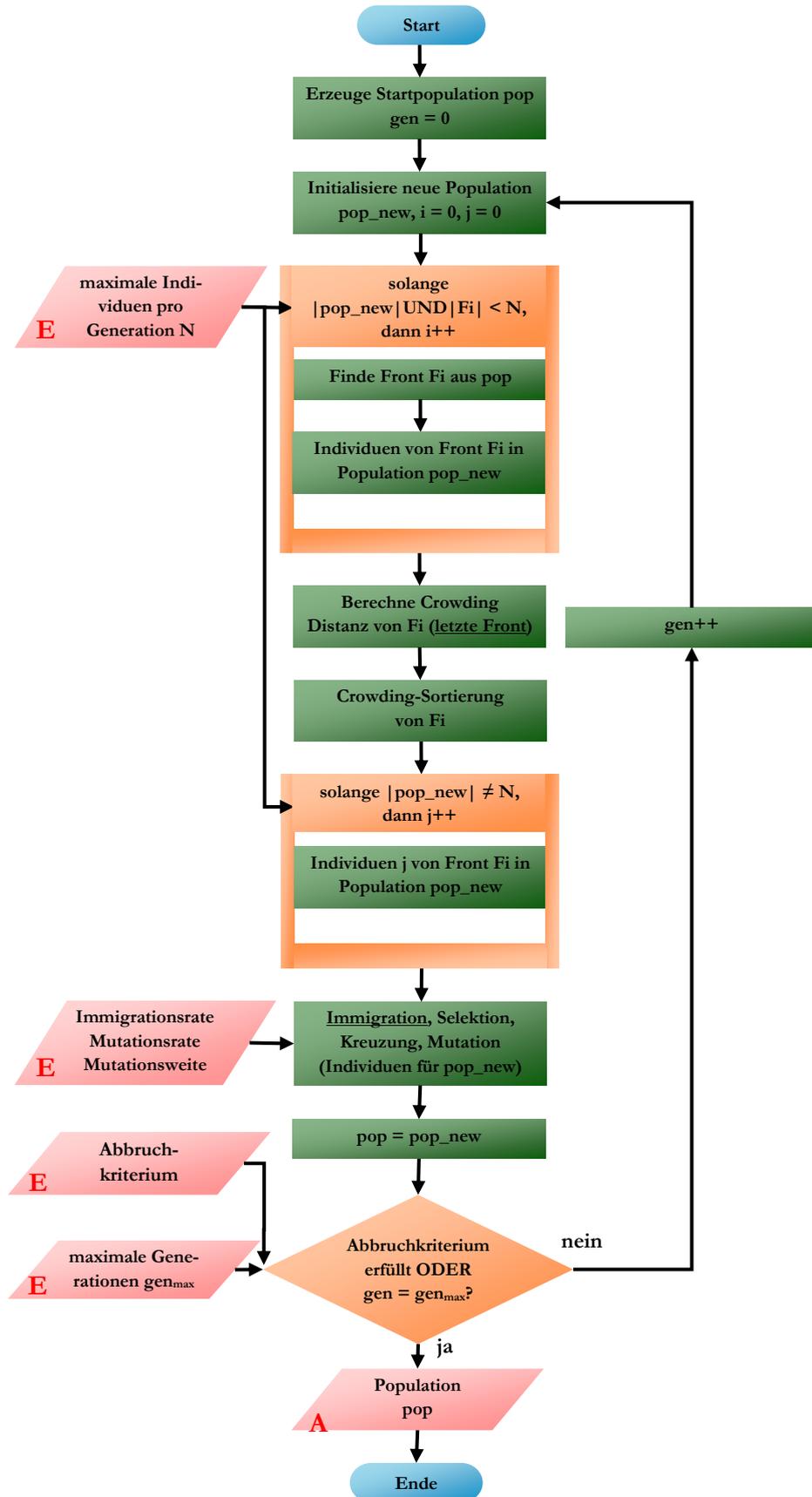


Abbildung 4-23: Programmablaufdiagramm des SIGA

Genau wie der NSGA-II beginnt auch der SIGA mit einer zufälligen Startpopulation und einigen Initialisierungen, bevor die Generationenschleife so lange abgearbeitet wird, bis das Abbruchkriterium (Mindestgüte oder maximaler Generationszähler) erreicht wurde. Jedoch entfällt beim SIGA die Kombination der Eltern- und Nachkommen-Population, da der SIGA diese nicht unterscheidet. Eltern und Nachkommen werden gemeinsam in einer Population `pop_new` gespeichert, die an dieser Stelle initialisiert wird.

Damit benötigt der SIGA nur zwei Populationen:

- `pop` zum Speichern der alten Population und
- `pop_new` zum Speichern der neuen Population.

Als nächstes folgt das Finden der Fronten F_i mittels des „Non-dominated Sorting“-Ansatzes.

Bei der zuvor beim NSGA-II beschriebenen Definition der Dominanz und der sehr großen Anzahl an Kriterien ergaben sich sehr große Fronten. Es waren demnach sehr viele Individuen indifferent zueinander, aber für die Schlafklassifikation sehr unterschiedlich geeignet. Dies resultiert daraus, dass für die Anwendung der Schlafklassifikation nicht alle Kriterien gleich bedeutsam sind. Deshalb erwies sich dieser Ansatz als nicht geeignet für die vorliegende Anwendung. Durch Abänderung der Definition der Dominanz konnte dieses Problem behoben werden.

Die neue Definition der Dominanz lautet:

- Individuum I_2 dominiert Individuum I_1 ($I_1 \preceq I_2$), wenn mehr Kriterien von I_2 besser sind als von I_1 , und
- Individuum I_2 ist indifferent zu Individuum I_1 ($I_1 \sim I_2$), wenn die gleiche Anzahl an Kriterien von I_2 und I_1 besser bzw. schlechter sind.

Beim NSGA-II werden zunächst alle Fronten gesucht und anschließend von jedem Individuum die Crowding-Distanz berechnet. Dann wird die Eltern-Population gefüllt. Beim SIGA hingegen wird jeweils eine Front gesucht und komplett in die neue Population ohne Berechnung der Crowding-Distanz übernommen, bis die gefundene Front nicht mehr vollständig in die Population passt. Diese letzte Front wird nach der Crowding-Distanz sortiert und nur die besten Individuen für die neue Population übernommen. So konnte der SIGA gegenüber dem NSGA-II effektiver gemacht werden. Anschließend folgt die Erzeugung der Nachkommen. Neben den bewährten Methoden der Kreuzung und Mutation wurde im SIGA die Immigration implementiert. Dieser neue genetische Operator basiert ebenfalls auf dem Vorbild der Natur. So kommt es bei einer gehäuften Paarung genetisch verwandter Lebewesen zu einem vermehrten Auftreten von Erbkrankheiten. Eine Auffrischung des Genpools durch andere Lebewesen ist daher unabdingbar und der Grund, warum Zoos und Tierparks Tiere untereinander tauschen müssen. So soll auch das Prinzip der Immigration im SIGA wirken, bei dem völlig neue Individuen zufällig in die Population eingebracht werden und somit neue Eigenschaften einbringen. Auf diese Weise weist der Algorithmus eine bessere Konvergenz auf und die Wahrscheinlichkeit für das Erreichen des globalen Optimums wird erhöht. Damit ist die Immigration der Mutation ähnlich, aber in ihrer Wirkung viel stärker. Durch die Immigrationsrate kann bestimmt werden, wie viele Individuen pro Generation immigrieren. Die Kreuzung geschieht ausschließlich mit den zuvor selektierten Individuen. Zufällig werden Individuen ausgewählt und deren Parameter gekreuzt. Auf diese Weise pflanzen sich nur die besten Individuen fort. Für die anschließende Mutation

werden ebenfalls aus den selektierten Individuen zufällig Individuen gewählt. Zwei Größen bedingen den Einfluss der Mutation auf den Optimierungsprozess: Die Mutationsrate und die Mutationsweite. Die Mutationsrate bestimmt, wie viele Individuen mutieren. Die Mutationsweite gibt an, in welchem Wertebereich der zufällig gewählte Parameter sich ändert.

Der SIGA lässt sich von außen durch den Benutzer steuern, so kann der Optimierungsprozess direkt auf den Anwendungsfall und die gewünschte Optimierungstiefe angepasst werden.

Von außen durch den Benutzer einzustellen sind zum einen SVM-Parameter, wie:

- SVM-Typ,
- SVM-Kernel,
- Anzahl der Klassen,
- Wertebereich, in dem die Parameter gesucht werden,
- Auflösung der Parameter und
- Anzahl der Datenpakete für die einfache Kreuzvalidierung.

Zum anderen sind genetische Parameter, wie:

- Anzahl der Individuen pro Generation,
- Anteil an Nachkommen einer Population in Prozent,
- Mutationsrate (Anteil der mutierten Individuen einer Population in Prozent),
- Mutationsweite (Wertebereich in dem die Parameter mutieren) und
- Immigrationsrate (Anteil der zufällig erzeugten Individuen einer Population in Prozent)

einstellbar.³⁹ Des Weiteren können die genannten fünf Kriterien von Seite 71 beliebig aktiviert oder deaktiviert werden. Zusätzlich ist eine Gewichtung der Kriterien einschließlich der Vertauschungsmatrix über eine Gewichtungsmatrix möglich. So können gezielt bestimmte Vertauschungen in der Matrix reduziert werden.

Um die Wirkung des SIGA auf die Klassifikationsqualität zu zeigen, wurden Tests mit realen Daten und mit künstlichen Daten anhand des Exklusiv-Oder(XOR)-Problems durchgeführt. Zum Vergleich steht die Rastersuche (RS) dem SIGA und dem ursprünglichen NSGA-II gegenüber. Zur Vergleichbarkeit der drei Algorithmen sollten Auflösung und/oder Zeit identisch sein. Der Zeitfaktor wird dabei in Anzahl der zu erprobenden Lösungen gemessen.

Es ergeben sich drei Vergleichsvarianten:

1. Variante: Zeit (Anzahl der erprobten Lösungen) und Auflösung

Hierbei wird bei allen Algorithmen die gleiche Auflösung eingestellt. Alle Algorithmen erproben die gleiche Anzahl an Lösungen. Hierzu arbeiten die genetischen Algorithmen mit einer festen Individuen- und Generationenanzahl ohne Abbruchkriterium. Daraufhin können die Ergebnisse im Hinblick auf den besten Belehrungsgrad⁴⁰ oder, im Falle vieler äquivalenter Lösungen⁴¹ mit gleichem Belehrungsgrad, im Hinblick auf die Anzahl der gefundenen äquivalenten Lösungen ausgewertet werden.

³⁹ Detaillierte Definitionen zu den Begriffen sind im Begriffsverzeichnis zu finden.

⁴⁰ siehe Begriffsverzeichnis

⁴¹ siehe Begriffsverzeichnis

2. Variante: Zeit (Anzahl der erprobten Lösungen)

Alle Algorithmen erproben hierbei die gleiche Anzahl an Lösungen. Die genetischen Algorithmen laufen ohne Abbruchkriterium mit einer festen Individuen- und Generationenanzahl. Allerdings werden sie mit einer höheren Auflösung gestartet. Zur Auswertung kann hier, wie in Variante 1, der beste Belehrungsgrad oder die Anzahl der gefundenen äquivalenten Lösungen ermittelt werden.

3. Variante: Auflösung

Hierbei wird bei allen Algorithmen die gleiche Auflösung eingestellt. Die genetischen Algorithmen laufen mit Abbruchkriterium. Die Auswertung zeigt, wann eine äquivalente Lösung zur Rastersuche durch den genetischen Algorithmus und wann eine bessere Lösung gefunden wurde.

Da die genetischen Algorithmen stochastischen Prozessen unterworfen sind, wurden die Ergebnisse aus jeweils 15 Läufen der genetischen Algorithmen benutzt und Median, Minimum, Maximum sowie erstes und drittes Quartil zur Auswertung bestimmt.

Im Weiteren werden zunächst zwei Versuche zum Exklusiv-ODER (XOR)-Problem beschrieben und ausgewertet. Anschließend werden die Algorithmen an den realen Daten erprobt und verglichen.

Versuch 1 stellt das klassische Zwei-Klassen-XOR-Problem im zweidimensionalen Raum dar, wobei sich die beiden Merkmale Mm_1 und Mm_2 im reellen Wertebereichen aller vier Quadranten bis zum Wert sieben bewegen.

Es gilt: $y \in [1,2]$, $y \in \mathbb{Z}$, $Mm_i \in [-7,7]$ und $Mm_i \in \mathbb{R}$, wobei y die Klasse und Mm_i die Merkmale sind. Abbildung 4-24 veranschaulicht dieses Problem graphisch. Die Daten für diesen Versuch wurden zufällig erzeugt und bestehen aus 100 Belehrungsvektoren sowie 320 Validierungsvektoren pro Klasse mit der gleichen Datenanzahl pro Quadrant. Der Rastersuch(RS)-Algorithmus ergab sechs gleichwertige Ergebnisse mit einem Belehrungsgrad von 99%.⁴² Die Validierung dieser sechs Ergebnisse ergab Gesamt-Klassifikationsgüten im Bereich von 96,5625% bis 96,875%. Unter Anwendung der Parameter mit einer der besten Klassifikationsraten entstand das in Abbildung 4-25 gezeigte Diagramm.⁴³ Es sind Belehrungs- und Validierungsdaten dieses Versuchs dargestellt, weiterhin sind die nicht generalisierten Datenpunkte gekennzeichnet. Besonders fällt auf, dass in Klasse 1 im ersten Quadranten die Belehrungsdaten nahe dem Koordinatenursprung Lücken aufweisen. An dieser Stelle häufen sich die Fehlklassifikationen. Dies verdeutlicht die Bedeutung der Belehrungsdaten. Fehlen Belehrungsdaten, besonders an den Grenzen der Klassen, sind Fehlklassifikationen die Folge.

⁴² Detaillierte Angaben zu den eingestellten Parametern sind in Tabelle C-6 auf Seite 134 zu finden.

⁴³ Detaillierte Angaben zu den eingestellten Parametern sind in Tabelle C-7 auf Seite 135 zu finden.

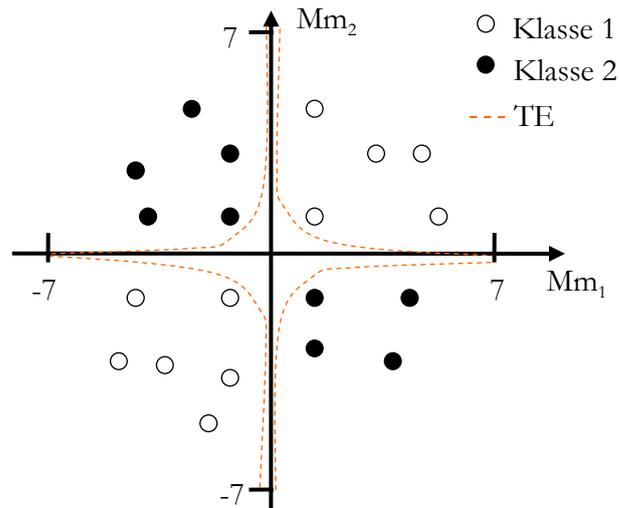


Abbildung 4-24: Klassisches Zwei-Klassen-XOR-Problem im zweidimensionalen Raum

Belehrungs- und Validierungsdaten des klassischen Zwei-Klassen-XOR-Problems

- Belehrung Klasse 1 × Validierung Klasse 1 ● NIO Klasse 1
- Belehrung Klasse 2 × Validierung Klasse 2 ● NIO Klasse 2

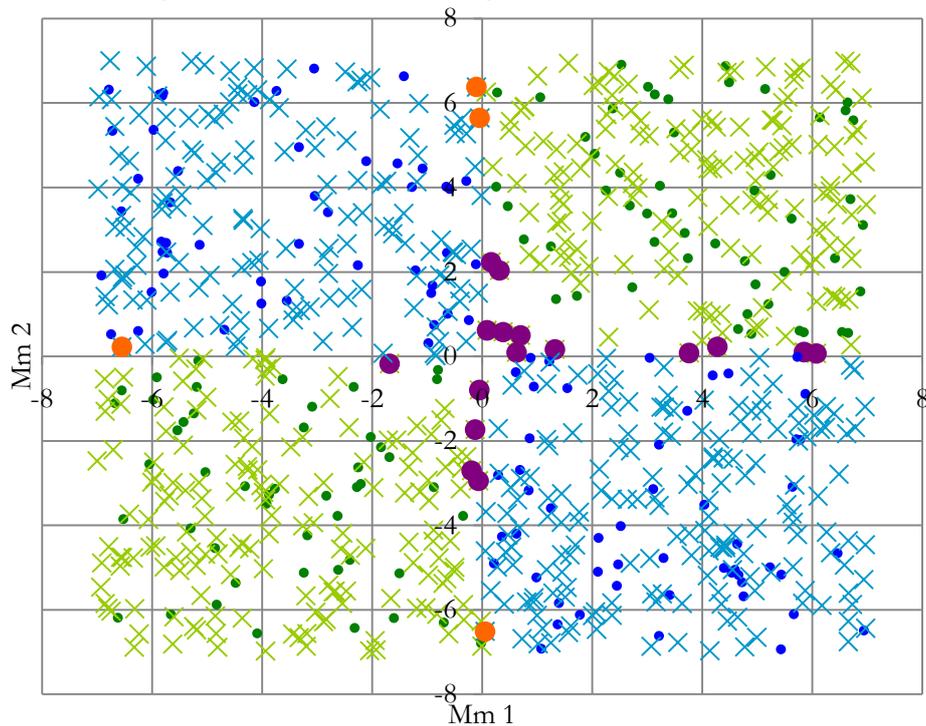


Abbildung 4-25: Streudiagramm des Belehrungs- und Validierungsdatensatzes des klassischen Zwei-Klassen-XOR-Problems mit Kennzeichnung der nicht generalisierten Datenpunkte - Dabei häufen sich Fehlklassifikationen (NIO) im Grenzbereich zwischen den Klassen, besonders wenn die Belehrungsdaten dort Lücken aufweisen.

Zum Vergleich der Rastersuche mit den genetischen Algorithmen (GA) wurde die erste Variante mit der gleichen Lösungsanzahl und gleichen Auflösung gewählt und die Anzahl an besten Lösungen ausgewertet.⁴⁴ Grund hierfür ist, dass dieser Versuch durch Rastersuche und genetische Algorithmen verschiedener Auflösungen zwar viele Lösungen mit äquivalentem Belehrungsgrad (99%) finden konnte, aber keine bessere Lösung. Auch die Qualität der gefundenen Lösungen, d. h. die Gesamt-Klassifikationsgüten der anschließenden Validation, sind bei Rastersuche und genetischen Algorithmen ähnlich. Die Ergebnisse in Form eines Kastendiagramms zeigt Abbildung 4-26, den zugehörigen Signifikanzwert p_{sig} zwischen SIGA und NSGA-II aus dem zweiseitigen Wilcoxon-Vorzeichen-Rang-Test zeigt Tabelle 4-12.

Im Vergleich zur Rastersuche konnten die genetischen Algorithmen im Hinblick auf den Median acht bis neun Mal mehr äquivalente Lösungen finden, im Hinblick auf das Maximum sogar rund 15-mal mehr. Allerdings wurde bei jeweils einem Lauf der genetischen Algorithmen kein besseres Ergebnis gefunden. Der NSGA-II im Vergleich zum SIGA weist mit $p_{sig}=0,004$ (Tabelle 4-12) signifikant mehr äquivalente Lösungen auf. Dies ist damit zu begründen, dass bei der SIGA-spezifischen Immigration häufig keine guten Lösungen gefunden werden, im Gegensatz zu Kreuzung und Mutation.

→ Insgesamt sind die genetischen Algorithmen durch den stochastischen Einfluss in diesem Versuch oft deutlich besser, jedoch mit Ausnahmen. Der SIGA ist gegenüber dem NSGA-II bei diesem relativ einfachen Problem offensichtlich nicht im Vorteil.

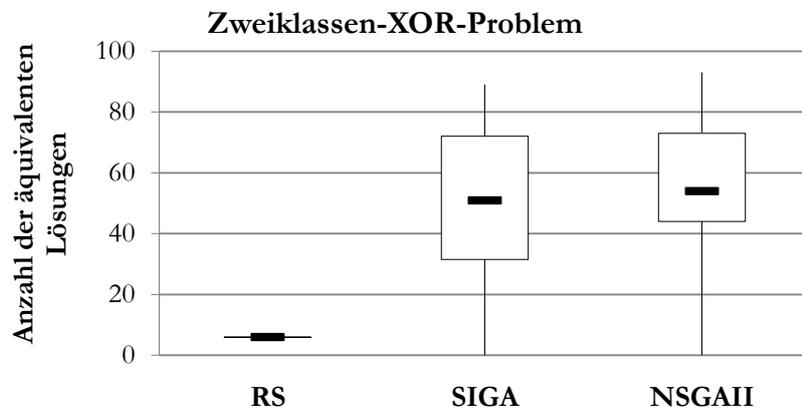


Abbildung 4-26: Kastendiagramm für das klassische Zwei-Klassen-XOR-Problem des Vergleichs der Rastersuche mit den genetischen Algorithmen SIGA und NSGA-II, dabei finden die GA deutlich mehr äquivalente Lösungen als die RS

Signifikanzwert p_{sig} für das Zwei-Klassen-XOR-Problem	Anzahl der äquivalenten Lösungen
SIGA gegen NSGA-II	0,004

Tabelle 4-12: Signifikanzwert für den Vergleich SIGA gegen NSGA-II für das Zwei-Klassen-XOR-Problem (zweiseitiger Wilcoxon-Vorzeichen-Rang-Test, $SN=0,05$, signifikante Unterschiede grau hinterlegt)

⁴⁴ Detaillierte Angaben zu den eingestellten Parametern für SIGA und NSGA-II sind in Tabelle C-8 auf Seite 135 zu finden.

Die definierten 14 „nicht belehrten“-Punkte, wie in Abbildung 4-27 zu sehen, berücksichtigen die folgenden Situationen.

1. Zwei Punkte gleicher Klasse nebeneinander (Punkt [4,3] und Punkt [5,3] aus Klasse 4 (gelb)):

Diese Situation kann nur teilweise generalisiert werden. Offensichtlich fehlt in diesem Fall zu viel Information über die Klasse. Ein Punkt wird gar nicht richtig zugeordnet, der andere ist parameterabhängig richtig klassifizierbar.

2. Zwei Punkte verschiedener Klassen nebeneinander (Punkt [0,5] aus Klasse 5 (rot) und Punkt [0,6] aus Klasse 6 (hellblau)) und
3. Zwei Punkte gleicher Klasse diagonal innerhalb einer Gruppierung (Punkt [1,4] und Punkt [0,5] aus Klasse 5 (rot)):

Diese Situationen generalisiert die SVM in jedem Fall. Offensichtlich können die Trennebenen richtig positioniert werden, da die ausschlaggebenden Punkte an den Ecken der Gruppierungen (Punkt [1,5] und Punkt [1,6]) vorhanden sind.

4. Zwei Punkte gleicher Klasse diagonal über zwei Gruppierungen (Punkt [5,5] und Punkt [6,6] aus Klasse 7 (violett)):

In dieser Situation sind die „nicht belehrte“-Punkte ausschlaggebend für die Trennebenen-Positionierung. Aus diesem Grund lassen sich diese Punkte nur abhängig von den verwendeten SVM-Parametern generalisieren.

5. Zwei Punkte verschiedener Klassen diagonal (Punkt [2,3] aus Klasse 1 (weiß) und Punkt [1,4] aus Klasse 5 (rot)):

Obwohl in Situation 5 die „nicht belehrten“-Fälle Eckpunkte der Gruppierungen darstellen, werden sie in allen gefundenen Parameterkombinationen dennoch richtig klassifiziert.

6. Ein Punkt weit weg von einer Klassengrenze (Punkt [0,0] aus Klasse 1 (weiß)) und
7. Ein Punkt an der Klassengrenze nicht in ausschlaggebender Position (Punkt [7,1] aus Klasse 4 (gelb) und Punkt [7,4] sowie Punkt [4,7] aus Klasse 8 (grün)):

Die Punkte aus Situation 6 und Situation 7 konnten immer von der SVM richtig zugeordnet werden. Diese Punkte haben somit keinen großen Einfluss auf die Trennebenen-Position.

8. Ein Punkt an der Klassengrenze in ausschlaggebender Position (Punkt [5,1] aus Klasse 3 (dunkelblau)):

Da der „nicht belehrte“-Punkt ein Eckpunkt in dieser Situation direkt an der Entscheidungsgrenze ist, bewirkt sein Fehlen bei der Belehrung eine verschobene Trennebene und somit war keine Generalisierung möglich.

→ Auch dieser Versuch macht deutlich, wie wichtig die Belehrungsdaten sind. Überall dort, wo die Eckpunkte zwischen zwei Klassen bei der Belehrung fehlen, können keine sicheren Generalisierungen stattfinden. Wenn doch richtig klassifiziert werden kann, wird die Trennebene infolge der geeigneten Parameter doch eher zufällig korrekt gelegt.

Da die Belehrungsgrade für viele gefundene Parameterkombinationen identisch waren, konnte erst im Validationsprozess die Eignung dieser Parameterkombinationen festgestellt werden.

Wie schon in Versuch 1 empfiehlt sich hier die Variante 1 zum Vergleich der Methoden, da sehr viele äquivalente Lösungen mit ähnlichen Gesamt-Klassifikationsgüten in der Validierung von Rastersuche und genetischen Algorithmen gefunden wurden.⁴⁶ Die Ergebnisse sind in Abbildung 4-28 in Form eines Kastendiagramms dargestellt, die Tabelle 4-13 enthält den Signifikanzwert p_{sig} zwischen SIGA und NSGA-II aus dem zweiseitigen Wilcoxon-Vorzeichen-Rang-Test.

Die Abbildung 4-28 zeigt vier- bis fünfmal mehr Lösungen im Median durch die genetischen Algorithmen, welche, durch die Simplizität dieses XOR-Problems, Lösungen mit 100% Belehrungsgrad darstellen. Die genetischen Algorithmen liegen damit deutlich über der Rastersuche. Der NSGA-II im Vergleich zum SIGA ist auch in diesem Versuch mit $p_{\text{sig}} < 0,001$ signifikant besser, was ebenfalls durch die schlechten Belehrungsgrade der Immigrationslösungen bedingt ist.

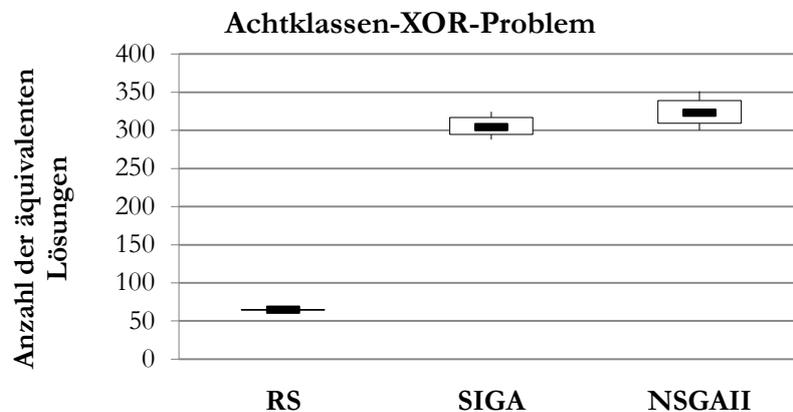


Abbildung 4-28: Kastendiagramm für das Acht-Klassen-XOR-Problem des Vergleichs der Rastersuche mit den genetischen Algorithmen SIGA und NSGA-II - Dabei finden die genetischen Algorithmen deutlich mehr äquivalente Lösungen als die RS.

Signifikanzwert p_{sig} für das Acht-Klassen-XOR-Problem	Anzahl der äquivalenten Lösungen
SIGA gegen NSGA-II	<0,001

Tabelle 4-13: Signifikanzwert für den Vergleich SIGA gegen NSGA-II für das Acht-Klassen-XOR-Problem (zweiseitiger Wilcoxon-Vorzeichen-Rang-Test, SN=0,05, signifikante Unterschiede grau hinterlegt)

- Der Vergleich anhand der künstlichen Daten konnte zwar die Überlegenheit der genetischen Algorithmen zur RS in Bezug auf die Anzahl der gefundenen Lösungen zeigen, aber durch die Simplizität der Probleme keine signifikanten Unterschiede in der Qualität der Lösungen aufweisen.
- Der NSGA-II war in der Anwendung auf die künstlichen Daten gegenüber dem SIGA im Vorteil.

⁴⁶ Detaillierte Angaben zu den eingestellten Parametern sind in Tabelle C-8 auf Seite 135 zu finden.

Anhand der realen Daten zur Schlafklassifikation sollen nun bessere Vergleiche möglich sein. Die Belehrungsdaten bestehen aus 100 Epochen dieser Daten pro Klasse. Die Abbildung 4-29 zeigt den Belehrungsgradverlauf über die SVM-Parameterkombinationen als Oberflächendiagramm. Die dargestellten Parameter sind das Fehlergewicht C als Parameter der C-SVC und γ die Weite der gaußschen Glockenkurve als Parameter des Kernels⁴⁷. Dadurch wird zunächst eine Vorstellung über die lokalen Optima und deren Lage gewonnen. Das Diagramm in Abbildung 4-29 entstand durch den Rastersuch-Algorithmus mit Auflösungen von 0,5 bis 100, dabei wurde die Anzahl der Parameterkombinationen durch den Logarithmus Dualis dargestellt.⁴⁸ Die Parameterkombinationen bilden diagonal über den Parameterbereich einen Gebirgskamm. Dementsprechend korrelieren γ und das Fehlergewicht C . Es ist deutlich eine Falte um den Nullpunkt des Fehlergewichtparameters zu sehen. Parameterkombinationen mit einem Fehlergewicht unter null schneiden im Belehrungsgrad schlechter ab.

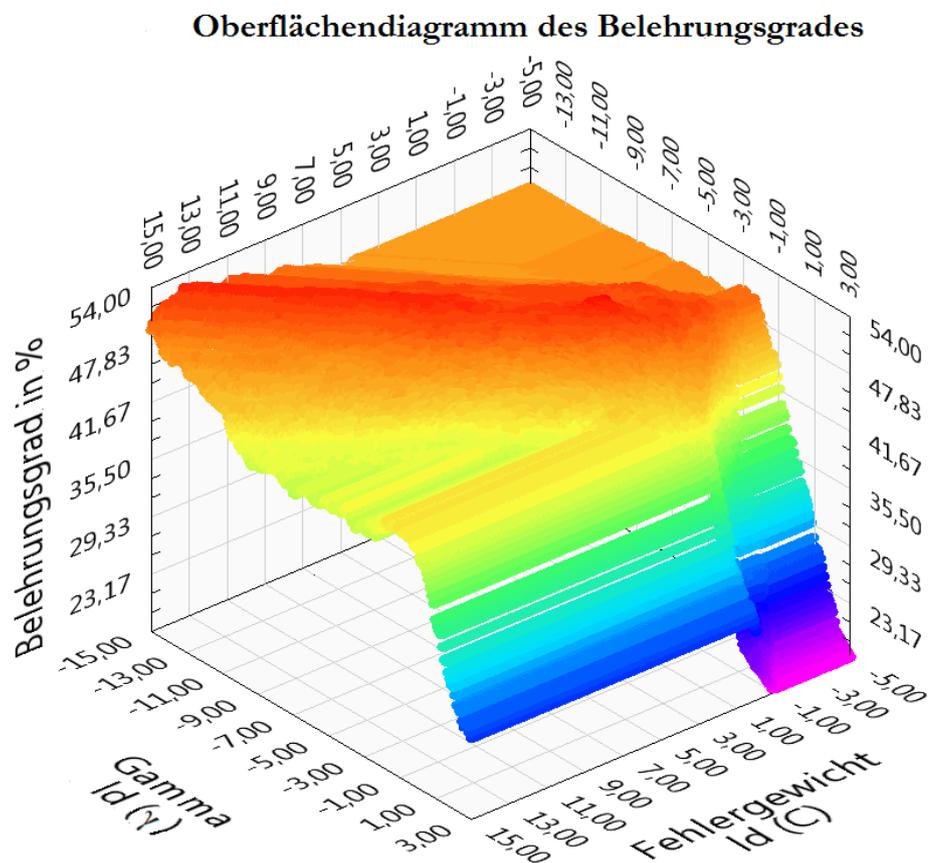


Abbildung 4-29: Oberflächendiagramm des Belehrungsgrades für die Belehrung mit realen Daten: der Belehrungsgrad über die SVM-Parameter⁴⁹

⁴⁷ in diesem Fall der RBF-Kernel, siehe Tabelle 3-1 auf Seite 16

⁴⁸ Bis auf die variable Auflösung entsprechen die verwendeten Parameter der Tabelle C-6 auf Seite 134.

⁴⁹ erstellt durch den Rastersuch-Algorithmus

In gleicher Weise ergab sich auch die Abbildung 4-30 als Ergebnis für die Rastersuche. Es sind die maximalen Belehrungsgrade über die Anzahl der erprobten Parameterkombinationen dargestellt, die direkt mit der eingestellten Auflösung zusammenhängen. Da die Anzahl der möglichen Parameterkombinationen in der Auflösung exponentiell ansteigt, ist die x-Achse logarithmisch dargestellt.

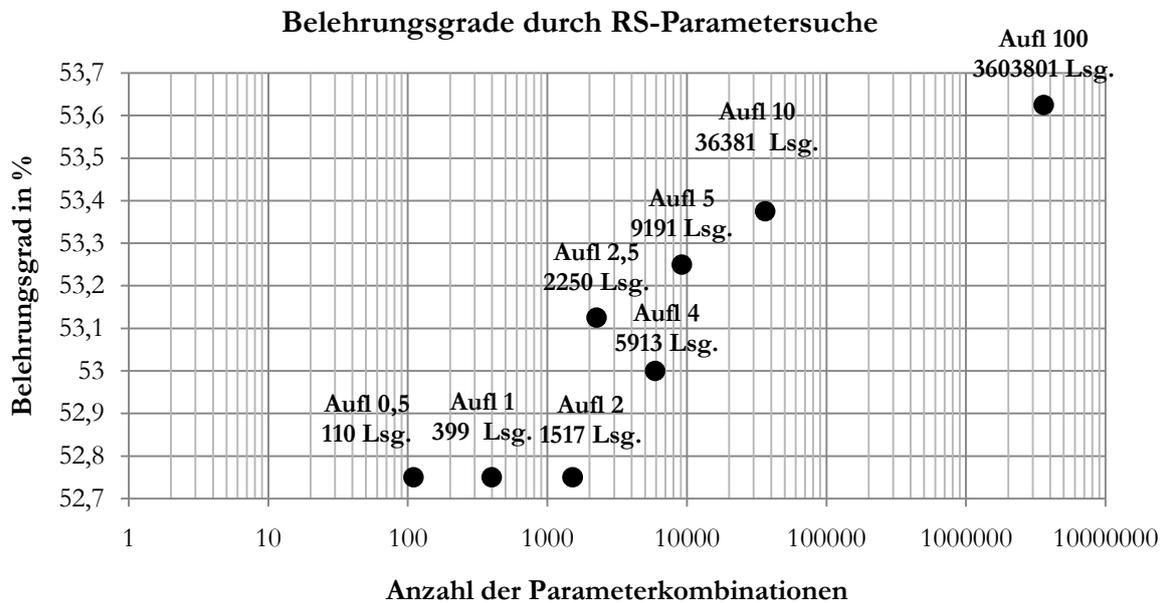


Abbildung 4-30: Resultierende Belehrungsgrade für die Belehrung mit realen Daten durch die Rastersuche - maximale Belehrungsgrade über die Anzahl der erprobten Parameterkombinationen (Auflösung) - Dabei zeigt sich eine nicht-stetige, exponentielle Abhängigkeit.

Tendenziell steigt der Belehrungsgrad mit der Anzahl der Parameterkombinationen an. Dieser Anstieg ist jedoch nicht stetig, da bei den verschiedenen Auflösungen teilweise andere Parameterkombinationen entstehen. Diese Ergebnisse der Rastersuche sollen in den nächsten Versuchen mit realen Daten als Vergleich mit den Ergebnissen der genetischen Algorithmen herangezogen werden.

Die Vergleichsvariante 1, welche die gleiche Auflösung und gleiche Zeit kennzeichnet, ergab die in Abbildung 4-31 abgebildeten Ergebnisse. Die zugehörigen Signifikanzwerte aus dem zweiseitigen Wilcoxon-Vorzeichen-Rang-Test sind in Tabelle 4-14 zu finden. Für diese Ergebnisse wurde eine Auflösung von 1 mit 399 auszuprobierenden Lösungen gewählt.⁵⁰ Bei dieser Vergleichsvariante sollte die Rastersuche im Vorteil sein, da diese den gesamten Suchraum erprobt. Die genetischen Algorithmen könnten jedoch zufällig, z. B. durch Mutation, dem Optimum etwas näher kommen. Die Ergebnisse entsprechen den Erwartungen. Im Vergleich zur Rastersuche liegt der Median der genetischen Algorithmen mit einem Belehrungsgrad von 52,5% leicht unter dem Ergebnis der Rastersuche mit 52,75%.

⁵⁰ Detaillierte Angaben zu den eingestellten Parametern sind in Tabelle C-8 auf Seite 135 zu finden.

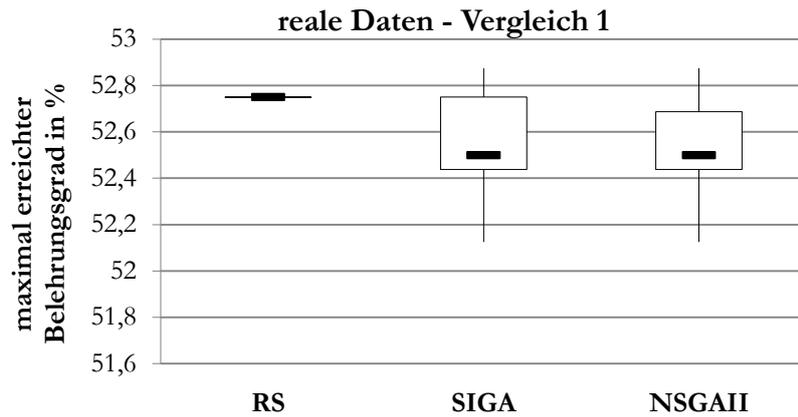


Abbildung 4-31: Kastendiagramm des Vergleichs 1 der Rastersuche (RS) mit den genetischen Algorithmen SIGA und NSGA-II für die realen Daten - maximal erreichte Belehrungsgrade - Dabei ist die RS gegenüber der GA wie erwartet im Vorteil.

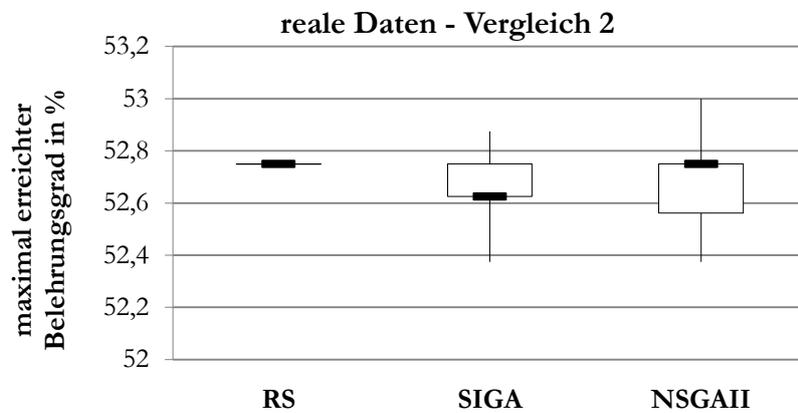


Abbildung 4-32: Kastendiagramm des Vergleichs 2 der Rastersuche (RS) mit den genetischen Algorithmen SIGA und NSGA-II für die realen Daten - maximal erreichte Belehrungsgrade - Dabei ist die RS gegenüber der GA wider Erwarten im Vorteil, aufgrund des stochastischen Einflusses auf die GA.

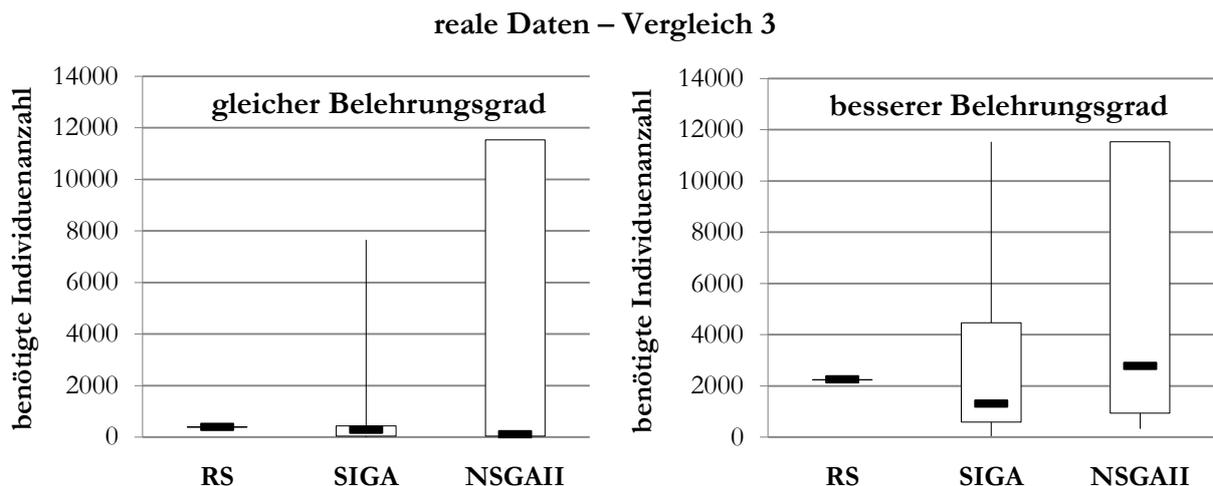


Abbildung 4-33: Kastendiagramm des Vergleichs 3 der Rastersuche (RS) mit den genetischen Algorithmen SIGA und NSGA-II für die realen Daten - benötigte Anzahl an Individuen für den gleichen Belehrungsgrad von 52,75% (links) und für einen besseren Belehrungsgrad (rechts) - Dabei wird der Vorteil des SIGA gegenüber des NSGA-II deutlich.

Signifikanzwerte p_{Sig} des Vergleichs SIGA gegen NSGA-II für die realen Daten	max. erreichter Belehrungsgrad	benötigte Individuen für	
		gleichen Belehrungsgrad	besseren Belehrungsgrad
Vergleich 1	0,655		
Vergleich 2	0,414		
Vergleich 3		0,099	0,001

Tabelle 4-14: Signifikanzwerte für den Vergleich SIGA gegen NSGA-II für reale Daten (zweiseitiger Wilcoxon-Vorzeichen-Rang-Test, $SN=0,05$, signifikante Unterschiede grau hinterlegt)

Jedoch werden durch den Zufallseinfluss sowohl bessere (maximaler Belehrungsgrad 52,875%) als auch schlechtere (minimaler Belehrungsgrad 52,125%) Parameter in der gleichen Zeit und der gleichen Auflösung gefunden. Die genetischen Algorithmen SIGA und NSGA-II zeigen kaum Unterschiede, was durch den Signifikanzwert von $p_{\text{Sig}}=0,655$ aus Tabelle 4-14 deutlich wird.

Vergleichsvariante 2, in der die gleiche Anzahl an erprobten Lösungen jedoch mit unterschiedlicher Auflösung zum Vergleich der Rastersuche und der genetischen Algorithmen angewandt wird, ergab die Resultate aus Abbildung 4-32.⁵¹ Die zugehörigen Signifikanzwerte aus dem zweiseitigen Wilcoxon-Vorzeichen-Rang-Test sind ebenfalls in Tabelle 4-14 zu finden. Bei diesem Vergleich wird sich durch die höhere Auflösung der genetischen Algorithmen ein Vorteil gegenüber der Rastersuche erhofft. Wie die Ergebnisse jedoch zeigen, wird diese Erwartung nicht erfüllt. Tatsächlich ähneln die Ergebnisse denen aus Vergleich 1. Die genetischen Algorithmen sind mit 52,625% beim SIGA und 52,75% Belehrungsgrad beim NSGA-II nur gleich bzw. leicht unter der Rastersuche mit einem Belehrungsgrad von 52,75%. Durch die stochastischen Prozesse wurden wiederum bei den 15 Läufen der genetischen Algorithmen bessere (maximal 52,875% beim SIGA bzw. 53% beim NSGA-II) und schlechtere Parameter (minimal 52,375% Belehrungsgrad) gefunden. Die Erklärung dafür liegt in der großen Anzahl an möglichen Parameterkombinationen, die durch die hohe Auflösung bedingt ist. Durch die vielen möglichen Parameterkombinationen steigt der stochastische Einfluss auf den genetischen Prozess. Die Wahrscheinlichkeit in die Nähe eines Optimums zu kommen, sinkt. In Summe bewirkt dies keinen Vorteil für die genetischen Algorithmen. Die genetischen Algorithmen untereinander weisen nur leichte, keine signifikanten Unterschiede auf, was durch den Signifikanzwert von $p_{\text{Sig}}=0,414$ aus Tabelle 4-14 deutlich wird. Dabei liegt der NSGA-II trotz einer höheren Streuung im Median noch vor dem SIGA.

Die Vergleichsvariante 3 vergleicht bei gleicher Auflösung die Zeit, die für einen bestimmten Belehrungsgrad benötigt wird. Die Abbildung 4-33 (links) zeigt die benötigten Lösungen bis zum gleichen Belehrungsgrad der Rastersuche von 52,75%. Die Abbildung 4-33 (rechts) zeigt die benötigten Lösungen zum nächst besseren Belehrungsgrad von 52,875%.⁵² Die zugehörigen Signifikanzwerte aus dem zweiseitigen Wilcoxon-Vorzeichen-Rang-Test sind wiederum in Tabelle 4-14 zu finden. Als Abbruchkriterium der genetischen Algorithmen wurde ein Belehrungsgrad von größer als 52,75% gewählt. Damit die Algorithmen dennoch terminieren, auch wenn das

⁵¹ Detaillierte Angaben zu den eingestellten Parametern sind in Tabelle C-9 auf Seite 136 zu finden.

⁵² Detaillierte Angaben zu den eingestellten Parametern der genetischen Algorithmen sind in Tabelle C-10 auf Seite 136 zu finden.

Abbruchkriterium nicht erreicht werden konnte, wurde eine maximale Generationenanzahl von 500 und damit eine maximale Anzahl an Individuen von 11531 eingestellt. Konnte bis dahin kein gleiches bzw. besseres Ergebnis erreicht werden, wurde dieses Maximum von 11531 als Wert für die Auswertung angenommen. Die Werte aus Abbildung 4-33 (links) zeigen, dass der SIGA gegenüber der Rastersuche im Median schneller zu einer vergleichbaren Lösung findet. Der Semiquartilsabstand ist gering, jedoch trat ein Ausreißer auf, der nun das Maximum bildet. Die Auswertung von Vergleich 3 der realen Daten zeigt den Vorteil des SIGA gegenüber dem NSGA-II, denn der NSGA-II kann in vielen Läufen keine vergleichbaren Lösungen finden. Dadurch ist der Semiquartilsabstand sehr hoch und der Median liegt über der Rastersuche. Der Unterschied zwischen SIGA und NSGA-II ist mit $p_{\text{sig}}=0,099$ (Tabelle 4-14) dennoch nicht signifikant.

Die Resultate der genetischen Algorithmen sind stark von der Anfangspopulation abhängig. Hat die Anfangspopulation hauptsächlich schlechte Lösungen, kann auch Mutation und Kreuzung oft keine viel besseren hervorbringen. Die Immigrationsindividuen des SIGA verhindern diese einseitige Entwicklung, dadurch zeigte der SIGA in diesem Versuch bessere Ergebnisse. Die Ergebnisse aus Abbildung 4-33 (rechts), welche die benötigten Individuen bis zu einer besseren Lösung als 52,75% zeigen, bestätigen die vorherigen Ergebnisse in allen Punkten. Der SIGA findet im Median schneller eine Lösung als die Rastersuche, welche erst bei 2250 Parameterkombinationen einen höheren Belehrungsgrad erreichen kann (siehe Abbildung 4-30). Jedoch kann der SIGA auch einmal keine bessere Parameterkombination finden, was das Maximum mit der Maximalanzahl an Individuen von 11531 zeigt. Der NSGA-II kann, wie schon zuvor, in einigen Läufen keine bessere Lösung finden und schneidet dadurch mit $p_{\text{sig}}=0,001$ (Tabelle 4-14) in allen Werten signifikant schlechter ab.

- Es lassen sich durch die genetischen Algorithmen bei simplen Problemen mehr äquivalente Lösungen mit ähnlichen Klassifikationsraten, verglichen mit der Rastersuche, finden.
- Bei komplexeren Problemen macht sich die Abhängigkeit von stochastischen Prozessen bemerkbar. Obwohl die genetischen Algorithmen oft schneller äquivalente Lösungen finden, können durch den stochastischen Einfluss manchmal auch keine äquivalenten Lösungen gefunden werden. Eine höhere Auflösung bewirkt bei den genetischen Algorithmen durch den Anstieg des stochastischen Einflusses eher eine Verschlechterung der Ergebnisse.
- Der enorme Vorteil des SIGA mit dem Immigrationsprinzip wird bei komplexeren Problemen besonders deutlich. Dieses Prinzip reduziert die Abhängigkeit von der Anfangspopulation und verhindert somit eine einseitige Entwicklung. Damit konvergiert der SIGA gegenüber der Rastersuche schneller.

Rückblickend könnte eine nicht vollständig zufällige Anfangspopulation zu besseren Ergebnissen führen, zum Beispiel durch eine weite Verteilung der Anfangsindividuen innerhalb des Suchraums.

4.4.2 Empirische Untersuchung der Kernel-Funktion

Die Bestimmung der Kernel-Funktion kann nur empirisch für eine konkrete Aufgabe erfolgen. Für die Untersuchung des geeigneten Kernel fanden die zehn Probandendatensätze von B1 Anwendung. Aus jedem der Datensätze wurden vier Modelle mit unterschiedlichen Kernel-Funktionen durch einfache Kreuzvalidierung erzeugt und gegen die restlichen neun Datensätze validiert.⁵³ Untersucht wurden die vier Kernel-Funktionen in Tabelle 3-1 aus Abschnitt 3.1.5 auf Seite 16, der Radial Basis Funktion (RBF)-Kernel, der polynomielle Kernel, der sigmoide Kernel und der lineare Kernel. Zur Beurteilung der Güte wurde die gewichtete Güte nach Gleichung (3.51) auf Seite 35 (Abbildung 4-34 links) und der gewichtete Kappa-Koeffizient κ_w nach Gleichung (3.52) auf Seite 35 (Abbildung 4-34 rechts) gewählt. Die zugehörigen Signifikanzwerte p_{sig} aus dem zweiseitigen Wilcoxon-Vorzeichen-Rang-Test zeigt Tabelle 4-15.

→ Anhand der Ergebnisse kann der sigmoide Kernel als der geeignetste Kernel für diese Daten identifiziert werden.

Die gewichtete Güte lässt zunächst kaum einen Unterschied zu den anderen Kernel-Funktionen erkennen. Einzig der RBF-Kernel weist eine leichte verringerte Güte auf. Jedoch ist keiner dieser Unterschiede zum sigmoiden Kernel signifikant (siehe Tabelle 4-15). Anhand des gewichteten Kappa-Koeffizienten werden die Unterschiede deutlicher.

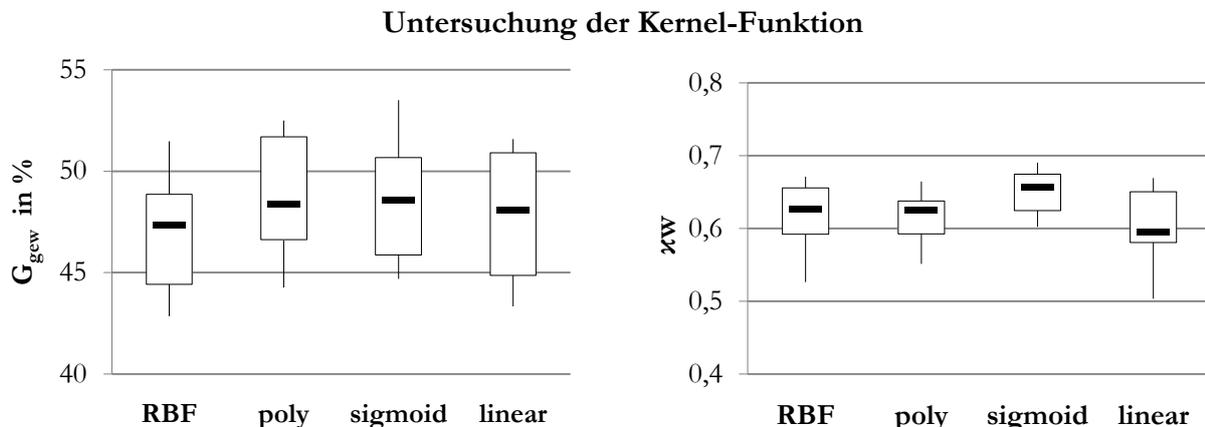


Abbildung 4-34: Kastendiagramme zur Untersuchung der Kernel-Funktion: die gewichtete Güte G_{gew} (links) und der gewichtete Kappa-Koeffizient κ_w (rechts) - Dabei zeigt der sigmoide Kernel die besten Resultate.

Signifikanzwerte p_{sig} im Vergleich zum: sigmoiden Kernel	G_{gew}	κ_w
RBF	0,114	0,009
polynomiell	0,721	0,005
linear	0,169	0,007

Tabelle 4-15: Signifikanzwerte für den Vergleich der Kernel-Funktionen in Bezug auf den sigmoiden Kernel (zweiseitiger Wilcoxon-Vorzeichen-Rang-Test, SN=0,05, signifikante Unterschiede grau hinterlegt)

⁵³ Detaillierte Angaben zu den eingestellten Parametern zur Kernel-Untersuchung sind in Tabelle C-11 auf Seite 136 zu finden.

Der sigmoide Kernel ist den anderen Kernel-Funktionen, sowohl durch den höchsten Median als auch durch die geringe Streuung, überlegen. Dies zeigen auch die Signifikanzwerte aus Tabelle 4-15, welche die signifikante Verbesserung des gewichteten Kappa-Koeffizienten durch den sigmoiden Kernel gegenüber allen anderen Kernel-Funktionen belegen. Der gewichtete Kappa-Koeffizient beinhaltet eine Gewichtung der Vertauschungsmatrix. Somit zeigt dieses Resultat, dass durch den sigmoiden Kernel schwere Vertauschungen besser vermieden werden können.

4.5 Klassifikation mittels SVM

Zum Zeitpunkt der Klassifikation sind die Belehrungsdaten bearbeitet, d. h. untypische Daten identifiziert, die Daten gleichverteilt und mit einer geeigneten Variante skaliert. Weiterhin wurden geeignete Parameter sowie eine Kernel-Funktion bestimmt und in einem Optimierungsprozess ein Modell erstellt. Doch auch zu diesem Zeitpunkt, wenn der Klassifikator bereits vorliegt, können noch Optimierungen vorgenommen werden. Resultat der Entscheidungsfunktion ist ein skalarer Wert, welcher ursprünglich aufgrund seines Vorzeichens die Klasse festlegt. Jedoch könnte auch der Betrag des Wertes für eine Klassenzuordnung ausgewertet werden. In welcher Weise die Auswertung erfolgt, beschreibt die Entscheidungsstrategie. Drei solcher Entscheidungsstrategien sollen im Folgenden untersucht und mit der ursprünglichen Vorgehensweise verglichen werden. Desweiteren werden Varianten der Ergebnisverknüpfung von mehreren Klassifikatoren (Bagging) erprobt und eine kontextbasierte Vorhersage für die Schlafklassifikation mit SVM analysiert.

4.5.1 Untersuchung der Entscheidungsstrategie

Die eingesetzte Mehr-Klassen-SVM basiert auf der im Abschnitt 3.1.11 auf Seite 29 vorgestellten One-against-One-Methode. Dabei werden nach Gleichung (3.42) auf Seite 29 abhängig von der Anzahl der Klassen mehrere Trennebenen gesucht. Für eine Sieben-Klassen-Klassifikation werden demnach 21 Trennebenen ermittelt und anhand der 21 Signum-Entscheidungsfunktionen je nach Vorzeichen des resultierenden Entscheidungswertes eine Klasse bestimmt. Dabei ermittelt der Algorithmus die Klasse, auf die die meisten Entscheidungen fallen, die sogenannte **Max-Wins-Strategie**. Diese Strategie lässt sich beschreiben durch:

$$y = \arg \max_{i \in \{1, \dots, U\}, i \in \mathbb{N}} (z_i) \quad \text{und} \quad (4.13)$$

$$EW_{ij} = \begin{cases} > 0 & z_i = z_j + 1 \\ \leq 0 & z_j = z_i + 1 \end{cases} \quad \text{für } \forall i, j = \{1, \dots, U\} \wedge j > i. \quad (4.14)$$

Es wird die Klasse y ausgewählt, bei der der klassenspezifische Wert z_i am größten ist (vergleiche Gleichung (4.13)). Für alle Trennebenen zwischen Klasse i und Klasse j erhöht sich der Wert z_i um eins, wenn der Entscheidungswert EW_{ij} nur durch sein Vorzeichen das Datum der jeweiligen Klasse zuordnet (vergleiche Gleichung (4.14)). Der Abstand des Punktes von der Trennebene spielt bei dieser Strategie keine Rolle. Es zählt lediglich, auf welcher Seite der Trennebene der Punkt liegt. Bei linearen Trennebenen kann jedoch ein direkter Zusammenhang zwischen Ent-

scheidungswert und Abstand zur Trennebene hergestellt werden.⁵⁴ Je größer der Entscheidungswert, desto größer ist auch der Abstand zur Trennebene und desto sicherer ist die Entscheidung auch. Es liegt nahe, diesen Abstand von der Trennebene in die Entscheidungsfindung mit einfließen zu lassen, um Fehlentscheidungen zu minimieren. Bei nichtlinearen Trennebenen ist der Zusammenhang zwischen dem Entscheidungswert und dem Abstand zur Trennebene jedoch nicht mehr linear, wodurch aus dem Entscheidungswert nicht mehr direkt auf den Abstand zur Trennebene geschlossen werden kann. Dennoch wurden verschiedene Entscheidungsstrategien erprobt, bei denen auch der Betrag der Werte zur Ergebnisfindung Verwendung fand. Die Entscheidung für eine Klasse wird bei allen erprobten Strategien, wie bei der Max-Wins-Strategie, nach Gleichung (4.13) über das Maximum von z_i getroffen. Nur die Vorgehensweise bei der Berechnung von z_i wird abgewandelt.

Bei der **MaxSum-Wins-Strategie** sollen statt eines konstanten Werts die Beträge der Entscheidungswerte EW_{ij} aufaddiert werden. Diese Strategie wurde bereits in [Bau09] vorgestellt. Es gilt:

$$EW_{ij} = \begin{cases} > 0 & z_i = z_i + |EW_{ij}| \\ \leq 0 & z_j = z_j + |EW_{ij}| \end{cases} \quad \text{für } \forall i, j = \{1, \dots, U\} \wedge j > i. \quad (4.15)$$

Damit werden bei dieser Strategie die Entscheidungswerte direkt zur Entscheidungsfindung mit einbezogen.

Die **MaxSumLim-Wins-Strategie** addiert ebenfalls die Entscheidungswerte auf, jedoch maximal einen Wert von eins. Damit wird der Einfluss hoher Entscheidungswerte begrenzt. Auch diese Strategie wurde in [Bau09] beschrieben. Dies ist in Gleichung (4.16) mathematische formuliert:

$$EW_{ij} = \begin{cases} > 0 \wedge < 1 & z_i = z_i + |EW_{ij}| \\ \geq 1 & z_i = z_i + 1 \\ \leq 0 \wedge > -1 & z_j = z_j + |EW_{ij}| \\ \leq -1 & z_j = z_j + 1 \end{cases} \quad \text{für } \forall i, j = \{1, \dots, U\} \wedge j > i. \quad (4.16)$$

Bei der **MaxSumNorm-Wins-Strategie** werden die Entscheidungswerte EW_{ij} zunächst bezogen auf den maximalen Wert der Klasse normiert, bevor sie zu dem klassenspezifischen Wert z_i aufaddiert werden. Wichtig dabei ist, dass der maximale Entscheidungswert für die Normierung für jede Klasse separat zu bestimmen ist. Das bedeutet, dass der normierte Entscheidungswert $EW_{ij, \text{norm}}$ für die Klasse i und die Klasse j unterschiedlich ist. Andernfalls würde diese Methode der MaxSum-Wins-Strategie gleichen. Diese Strategie wurde erstmals für die vorliegende Arbeit entwickelt. Es gelten die Gleichungen:

$$EW_{ij} = \begin{cases} > 0 & z_i = z_i + EW_{ij, \text{norm}_i} \\ \leq 0 & z_j = z_j + EW_{ij, \text{norm}_j} \end{cases} \quad \text{für } \forall i, j = \{1, \dots, U\} \wedge j > i \quad \text{und} \quad (4.17)$$

⁵⁴ Eine ausführliche Beschreibung der Interpretation des Entscheidungswertes ist [Mon05] zu entnehmen.

$$EW_{ij, \text{norm}_i} = \frac{|EW_{ij}|}{\max(EW_i)} \quad (4.18)$$

Damit soll wiederum der Einfluss hoher Entscheidungswerte gegenüber kleiner Entscheidungswerte gemindert werden.

Die Auswirkung dieser Maßnahmen auf das Klassifikationsergebnis ist in Abbildung 4-35 zu sehen. Als Versuchsdaten wurden wiederum die zehn Probanden des B1-Datensatzes verwendet. Jeder Proband diente einzeln als Belehrungsdatensatz eines SVM-Modells, welches an den restlichen neun Datensätzen validiert wurde.⁵⁵ In Abbildung 4-35 sind die Kastendiagramme für die gewichtete Güte G_{gew} der vier erprobten Entscheidungsstrategien abgebildet, Tabelle 4-16 enthält die zugehörigen Signifikanzwerte p_{sig} aus dem zweiseitigen Wilcoxon-Vorzeichen-Rang-Test.

Deutlich ist die Verschlechterung der Güte durch die MaxSum-Wins-Strategie zu sehen. Diese Strategie ist mit $p_{\text{sig}}=0,005$ (Tabelle 4-16) gegenüber allen anderen erprobten Strategien signifikant schlechter. Das alleinige Aufaddieren der Entscheidungswerte kann durch die Nichtlinearität der Hyperebenen offensichtlich nicht genutzt werden. Eine leichte Verbesserung hingegen kann durch die MaxSumLim- und die MaxSumNorm-Wins-Strategie erreicht werden. Diese Verbesserung ist mit $p_{\text{sig}}=0,333$ (Tabelle 4-16) jedoch nicht signifikant.

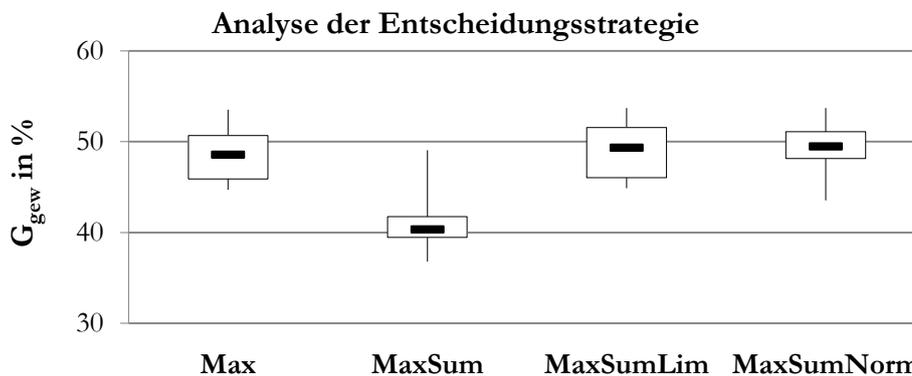


Abbildung 4-35: Kastendiagramm zur Untersuchung der Entscheidungsstrategie: die gewichtete Güte G_{gew} - Dabei können durch Einbeziehung der Entscheidungswerte leichte Verbesserungen erreicht werden, wenn diese normiert oder zumindest begrenzt werden.

Signifikanzwerte p_{sig} zur Untersuchung der Entscheidungsstrategie	G_{gew}		
	MaxSum	MaxSumLim	MaxSumNorm
Max	0,005	0,333	0,333
MaxSum		0,005	0,005
MaxSumLim			0,799

Tabelle 4-16: Signifikanzwerte für den Vergleich der Entscheidungsstrategien (zweiseitiger Wilcoxon-Vorzeichen-Rang-Test, SN=0,05, signifikante Unterschiede grau hinterlegt)

⁵⁵ Die detaillierte Parametrierung der Modellbildung zur Entscheidungsstrategieanalyse richtet sich nach den gewonnenen Erkenntnissen aus den vorherigen Abschnitten und ist in Tabelle C-12 auf Seite 137 zu finden.

Dabei ist die Normierung (MaxSumNorm-Wins-Strategie) der Begrenzung (MaxSumLim-Wins-Strategie) aufgrund von einem leicht höheren Median und einem geringeren Semiquartilsabstand vorzuziehen. Der Signifikanzwert von $p_{\text{Sig}}=0,799$ des Vergleichs beider Methoden zeigt deren Ähnlichkeit.

- Die Abwandlung der klassischen Entscheidungsstrategie ist durchaus sinnvoll, wobei jedoch Aufwand und Nutzen abgewogen werden sollten.
- Bezieht man den Abstand zur Trennebene mit ein, sollte der Einfluss hoher Entscheidungswerte durch eine Normierung des Wertes oder zumindest durch eine Begrenzung abgeschwächt werden.

4.5.2 Kombination verschiedener SVM mittels Bagging

Zur Kombination mehrerer SVM-Modelle wird das „Bagging“ verwendet. Dieses Wort ist vom englischen „bootstrap aggregating“ abgeleitet und wurde 1996 von Leo Breiman in [Bre96] erstmals beschrieben. Die Wirkung dieser Methode ist von der Stabilität des Klassifikators abhängig. Je mehr sich eine kleine Änderungen in der Belehrungsdatenmenge auf den Klassifikator auswirkt, desto instabiler ist der Klassifikator und desto höher ist die Wirkung der Methode. Für den Klassifikator SVM konnte bereits in Abschnitt 4.1 ab Seite 39 gezeigt werden, dass dieser extrem von den vorliegenden Belehrungsdaten abhängig ist. Aufgrund dessen muss angenommen werden, dass die „Bagging“-Methode eine Verbesserung erreichen wird. Im einfachsten Fall wird zur Kombination der Ergebnisse aus den verschiedenen Klassifikatoren die Mehrheitsentscheidung (MHE) angewendet. Es wird die Klasse gewählt, die Resultat der meisten Klassifikatoren war.

Für die Klassifikation von Schlaf bietet sich jedoch noch eine andere Variante an. Da Schlaf ein fließender Prozess ist, können nicht alle Klassenübergänge auftreten. Beispielsweise wird nach dem Leichtschlafstadium S1 nicht direkt das Tiefschlafstadium S4 auftreten.⁵⁶ Anhand des Wissens über die auftretenden Schlafstadienübergänge lassen sich die Schlafstadien, außer REM, in einer Ordinalskala darstellen. Damit kann aus den resultierenden Klassen der verschiedenen Klassifikatoren der Mittelwert oder der Median bestimmt werden. Die Reihenfolge der Schlafstadien für eine solche Ordinalskala, von keinem Schlaf über leichten Schlaf bis zum Tiefschlaf, festzulegen, ist jedoch wegen der Sonderrolle des Schlafstadiums REM nicht einfach. In diesem Stadium sind viele Übergänge möglich, deshalb kann REM sowohl zwischen Wach und S1 als auch zwischen S1 und S2 eingeordnet werden.

Für die Untersuchungen wurden die Datensätze von B1 zum Belehren der SVM-Modelle genutzt und die Datensätze von B2 und B3 zum Validieren.⁵⁷ In Abbildung 4-36 sind die Ergebnisse der Untersuchung zur Kombination mehrerer SVM-Modelle zu sehen. Im oberen Diagramm ist die gewichtete Güte G_{gew} in Prozent und im unteren Diagramm ist der gewichtete Kappa-Koeffizient κ_w dieser Untersuchung dargestellt.

⁵⁶ siehe Abbildung 2-2: Vereinfachte Darstellung eines Hypnogramms auf Seite 10

⁵⁷ Detaillierte Angaben zu den eingestellten Parametern sind in Tabelle C-12 auf Seite 137 zu finden.

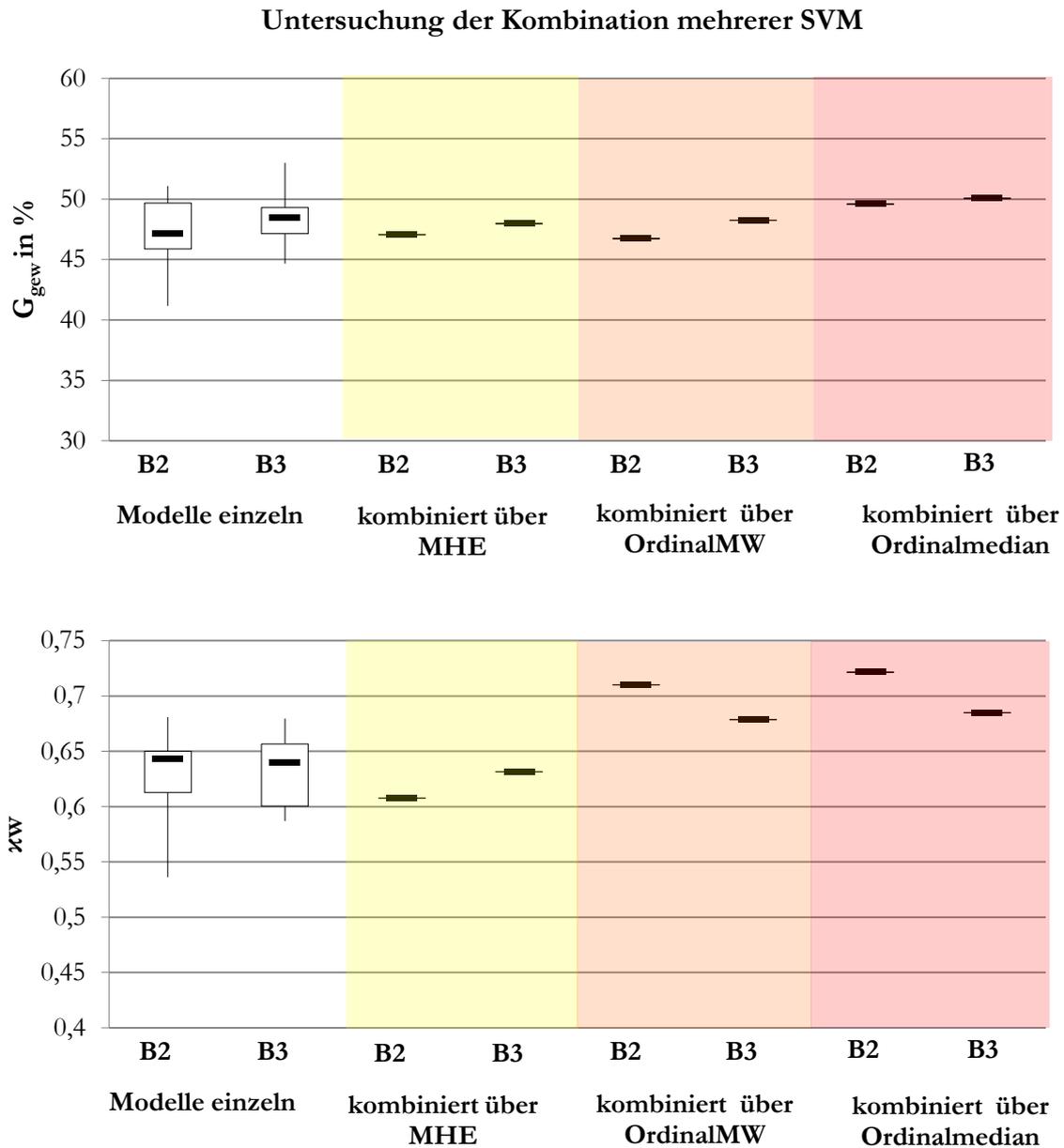


Abbildung 4-36: Kastendiagramme zur Untersuchung der Kombination mehrerer SVM: die gewichtete Güte G_{gew} (oben) und der gewichtete Kappa-Koeffizient κ_w (unten); Modelle einzeln (weiß), Modelle kombiniert mittels Mehrheitsentscheidung (MHE) (gelb), Modelle kombiniert mittels Ordinalmittelwert (OrdinalMW) (orange) und Modelle kombiniert mittels Ordinalmedian (rot) - Dabei ist die Kombination mehrerer SVM-Modelle für Schlafdaten sinnvoll, besonders mittels Ordinalmedian.

Zum Vergleich sind zunächst die Ergebnisse der einzelnen Modelle (weiß) in Form eines Kastendiagramms abgebildet.⁵⁸ Die Ergebnisse der kombinierten Modelle durch Mehrheitsentscheidung (MHE), Ordinalmittelwert und Ordinalmedian sind in den Farben gelb, orange und rot unterlegt.

Am besten schneidet bei den verwendeten Daten die Variante des Ordinalmedians ab. Die gewichteten Güte G_{gew} zeigt (Abbildung 4-36 oben), dass eine Kombination der Modelle durch Mehrheitsentscheidung oder Ordinalmittelwert keine Auswirkung auf das Klassifikationsergebnis hat. Lediglich der Ordinalmedian kann die Klassifikationsleistung verbessern. Anhand des gewichteten Kappa-Koeffizienten κ_w (Abbildung 4-36 unten) ist die Wirkung der Modellkombination noch deutlicher. Der gewichtete Kappa-Koeffizient κ_w beinhaltet nicht nur die richtig klassifizierten Daten, sondern kann auch die mehr oder weniger schweren Vertauschungen wiedergeben. Durch die Ermittlung eines Ordinalmittelwertes oder eines Ordinalmedians können schwere Vertauschungen stark reduziert werden, da einzelne abweichende Ergebnisse durch die anderen Modelle kompensiert werden. Der Ordinalmedian zeigt auch im gewichteten Kappa-Koeffizienten κ_w die besten Resultate. Der Median ist unempfindlicher gegenüber stark abweichenden Ergebnissen. Offensichtlich ist dies für die Art der vorliegenden Daten und deren Klassifikation mittels SVM von Vorteil.

→ Die Kombination von mehreren SVM-Modellen kann die Klassifikationsgüte erhöhen. Die Möglichkeit für die Schlafstadien eine Ordinalskala festlegen zu können, sollte dabei genutzt werden.

Eine zusätzliche Gewichtung der einzelnen Modelle ist naheliegend. So könnten „gute“ Modelle hohe Gewichte und „schlechte“ Modelle niedrige Gewichte erhalten. Dies sollte in künftigen Arbeiten untersucht werden.

4.5.3 Kontextbasierte Klassifikation von Hypnogrammen

Wie in Abschnitt 2 ab Seite 7 und Abschnitt 4.5.2 ab Seite 91 ersichtlich, handelt es sich beim Schlaf um einen fließenden Prozess. Abrupte Übergänge von sehr unähnlichen Schlafklassen kommen nicht vor. Daher ist für die Klassifikation von Hypnogrammen eine kontextbasierte Klassifikation naheliegend.⁵⁹ Für die Klassifikation eines Zustandes wird der vorherige Zustand betrachtet und in die Entscheidung einbezogen. Hierfür erfolgte zunächst die Betrachtung der Datensätze von B1 und der darin enthaltenen Klassenübergänge. Das Ergebnis der Auswertung ist prozentual zur Ausgangsklasse in Tabelle 4-17 dargestellt. Mit Hilfe dieser Werte, sowie der Einbeziehung der Kenntnisse zu den Schlafstadien und deren Übergänge im Hypnogramm, konnten einige Klassenübergänge als „unerlaubte“ Übergänge identifiziert werden. Die für die kontextbasierte Klassifikation als „unerlaubt“ definierten Übergänge sind in Tabelle 4-17 farbig hinterlegt und in Tabelle 4-18 nochmals aufgelistet. So ist der Übergang von und zu den Tiefschlafstadien nur durch die, nach der Ordinalskala angrenzenden Schlafstadien möglich. Der Übergang von und zur Klasse Movement ist hingegen immer möglich.

⁵⁸ Eine Untersuchung auf Signifikanz ist hier nicht möglich, da bei der Kombination der Modelle nur ein Wert resultiert.

⁵⁹ Dieses Prinzip wurde bereits in [Wal12] auf die Klassifikation von Narkosedaten angewendet.

zu \ von	S1	S2	S3	S4	REM	Wach	Mov
S1		56,54	0,18	0,00	33,64	48,05	13,73
S2	48,73		47,61	1,05	49,09	22,82	47,06
S3	0,25	25,73		98,60	0,45	0,90	9,31
S4	0,00	0,32	50,92		0,00	0,00	2,45
REM	7,76	9,08	0,00	0,00		8,11	23,53
Wach	35,62	3,68	0,55	0,00	3,64		3,92
Mov	7,63	4,65	0,74	0,35	13,18	20,12	

Tabelle 4-17: Prozentuale Auswertung der Klassenübergänge im gesamten Datensatz B1 (sehr unwahrscheinliche Übergänge sind orange gekennzeichnet)

von	zu
S1	→ S3, S4
S2	→ S4
S3	→ S1, REM, Wach
S4	→ S1, S2, REM, Wach
REM	→ S3, S4
Wach	→ S3, S4

Tabelle 4-18: Als „unerlaubt“ definierte Übergänge für die Schlafklassifikation

Ausgehend von den Erkenntnissen zu den Schlafstadienübergängen wurde die Klassifikation dahingehend modifiziert, dass der letzte Zustand zunächst gespeichert wird. Bei Klassifikation einer Klasse, die einen solchen „unerlaubten“ Übergang entspricht, wird anhand der Anzahl der Trennebenen-Entscheidungen die nächstmögliche Klasse gewählt. Falls diese ebenfalls einen „unerlaubten“ Übergang bedingt, wird wiederum die nächste mögliche Klasse gewählt und so weiter. Bei einem Übergang in die Klasse Movement wird als vorhergehender Zustand die letzte gültige Klasse gespeichert, da von der Klasse Movement alle Übergänge möglich sind.

Für Untersuchungen mit realen Daten wurden die Datensätze von B1 einzeln als Belehrungsdaten und die Datensätze B2 und B3 zur Validierung verwendet.⁶⁰ Abbildung 4-37 zeigt die Resultate dieser Untersuchung in Form von Kastendiagrammen, die Tabelle 4-19 die zugehörigen Signifikanzwerte des Vergleichs mit und ohne kontextbasierter Klassifikation aus dem zweiseitigen Wilcoxon-Vorzeichen-Rang-Test.

Die gewichtete Güte G_{gew} (Abbildung 4-37 links) steigt durch die Verwendung der kontextbasierten Klassifikation bei beiden Datensätzen an. Wobei der Datensatz B3 mit gesunden und leicht schlafgestörten Probanden stärker auf diese Methode anspricht und mit $p_{sig}=0,022$ sogar signifikant ansteigt. Der gewichtete Kappa-Koeffizient κ_w lässt die Wirkung noch deutlicher erkennen. Die kontextbasierte Klassifikation kann die schweren Vertauschungen bei beiden Datensätzen reduzieren, wodurch der gewichtete Kappa-Koeffizient κ_w ansteigt. Der Datensatz B3 weist dabei mit $p_{sig}=0,005$ einen signifikanten Anstieg des gewichteten Kappa-Koeffizienten κ_w auf.

⁶⁰ Detaillierte Angaben zu den eingestellten Parametern sind in Tabelle C-12 auf Seite 137 zu finden.

kontextbasierte Klassifikation

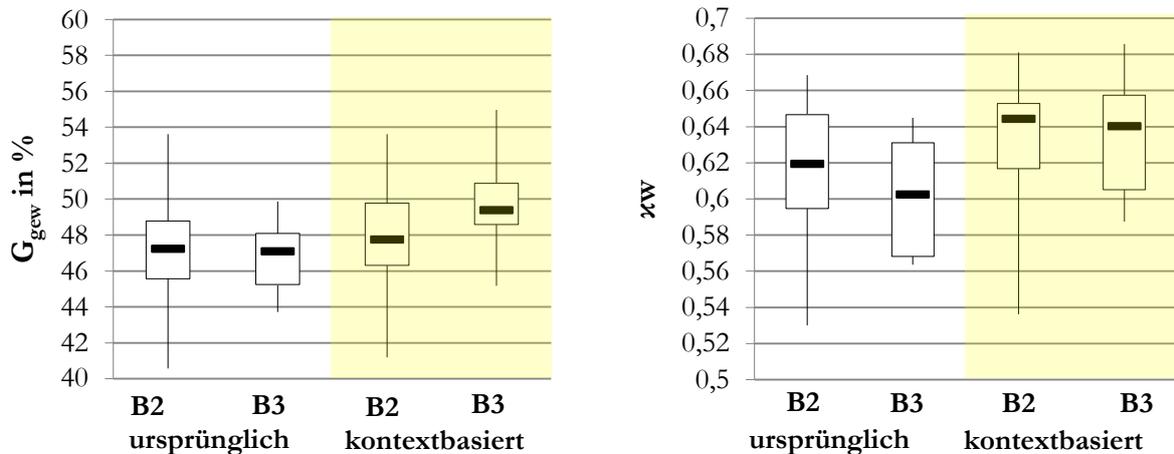


Abbildung 4-37: Kastendiagramme zur Untersuchung der kontextbasierten Klassifikation: die gewichtete Güte G_{gew} (links) und der gewichtete Kappa-Koeffizient κ_w (rechts) - Dabei reduziert die Einbeziehung des vorhergehenden Zustandes die schweren Vertauschungen und steigert damit die Güte des Klassifikators.

Signifikanzwerte p_{sig} der kontextbasierten Klassifikation	G_{gew}		κ_w	
	B2	B3	B2	B3
	0,203	0,022	0,074	0,005

Tabelle 4-19: Signifikanzwerte für die kontextbasierte Klassifikation (zweiseitiger Wilcoxon-Vorzeichen-Rang-Test, $SN=0,05$, signifikante Unterschiede grau hinterlegt)

→ Die Anwendung der kontextbasierten Vorhersage reduziert die schweren Vertauschungen. Sie steigert damit die Güte des Klassifikators und somit dessen Robustheit.

Nun ist die vorgestellte kontextbasierte Vorhersage eine sehr einfache Variante, da nur der letzte gültige Zustand gespeichert wird. Zudem sind bei einer Fehlentscheidung Folgefehler wahrscheinlich. Tatsächlich aber beziehen die Experten bei der manuellen Bewertung der Daten alle vorherigen Zustände und ihr Expertenwissen um den typischen Verlauf eines Hypnogramms mit ein. Dies könnte algorithmisch erfasst werden, indem mehrere vorherige Zustände, bestenfalls sogar alle vorherigen Zustände, gespeichert und einbezogen würden. Zudem müssten die Übergangswahrscheinlichkeiten der einzelnen Klassen wie in Tabelle 4-17 bei der Entscheidung berücksichtigt werden. Somit würde die Berechnung der Wahrscheinlichkeiten für die künftigen Zustände möglich. Es könnte sogar der klassenspezifische Wert z_i aus Abschnitt 4.5.1 ab Seite 88, welcher sich aus den Entscheidungswerten EW_{ij} ergibt, einbezogen werden. Jedoch sind die Übergangswahrscheinlichkeiten zeitlich nicht konstant. Sie ändern sich im Laufe einer Nacht. So ist in der zweiten Nachthälfte der Zustand REM wahrscheinlicher als in der ersten (siehe Abbildung 2-2: Vereinfachte Darstellung eines Hypnogramms auf Seite 10). Auch dies müsste für eine optimale Klassifikation analysiert und algorithmisch umgesetzt werden. Auf diese Art würde die Bewertung eines Experten am besten nachempfunden werden. Dieses Thema sollte in künftigen Arbeiten eingehender behandelt werden.

5 Anwendung der Untersuchungsergebnisse auf die realen EEG-Datensätze

Die Erkenntnisse aus Kapitel 4 sollen nun gebündelt auf die vorliegenden realen Daten angewendet werden. Dies soll die Verbesserung der Klassifikationsleistung durch die angewendeten Methoden verdeutlichen. Im Folgenden werden in fünf Schritten die zuvor untersuchten Methoden angewendet. Die graphische Darstellung dieser Methoden der Gesamtuntersuchung ist in Abbildung 5-1 zu sehen. Die ursprüngliche Modellbildung beinhaltet eine Min-Max-Skalierung der zehn homogenen Belehrungsdatensätze von B1 mit 62 Merkmalen und die Anwendung des RBF-Kernels. Die Parameterbestimmung erfolgt für die ursprüngliche Modellbildung über einen einfachen Rastersuchalgorithmus unter Anwendung der einfachen Kreuzvalidation.⁶¹

1. Im ersten Schritt soll dieser ursprünglichen Modellbildung die Modellbildung mit zehn inhomogenen Datensätzen gegenüber gestellt werden (Abschnitt 4.1.3 Wirkung der Homogenität von Daten ab Seite 59). Weiterhin wurde die Standardisierung zur Skalierung der Daten (Abschnitt 4.2 Untersuchung zur Skalierung von Daten ab Seite 60) und der sigmoide Kernel (Abschnitt 4.4.2 Empirische Untersuchung der Kernel-Funktion ab Seite 87) verwendet.⁶²
2. Im zweiten Schritt wurden für die Modellbildung zusätzlich der DiLa-Algorithmus zur Identifikation untypischer Daten mit anschließender SMOTE-Gleichverteilung der Belehrungsdaten (Abschnitt 4.1.2 Identifikation von untypischen Daten ab Seite 50) angewendet. Der SIGA diente zur genetischen Parameteroptimierung (Abschnitt 4.4.1 Multikriterielle Parameteroptimierung mittels evolutionärer Algorithmen ab Seite 66).⁶³

⁶¹ Detaillierte Angaben zu den eingestellten Parametern sind in Tabelle C-13 auf Seite 137 zu finden.

⁶² Detaillierte Angaben zu den eingestellten Parametern sind in Tabelle C-14 auf Seite 137 zu finden.

⁶³ Detaillierte Angaben zu den eingestellten Parametern sind in Tabelle C-15 auf Seite 138 zu finden.

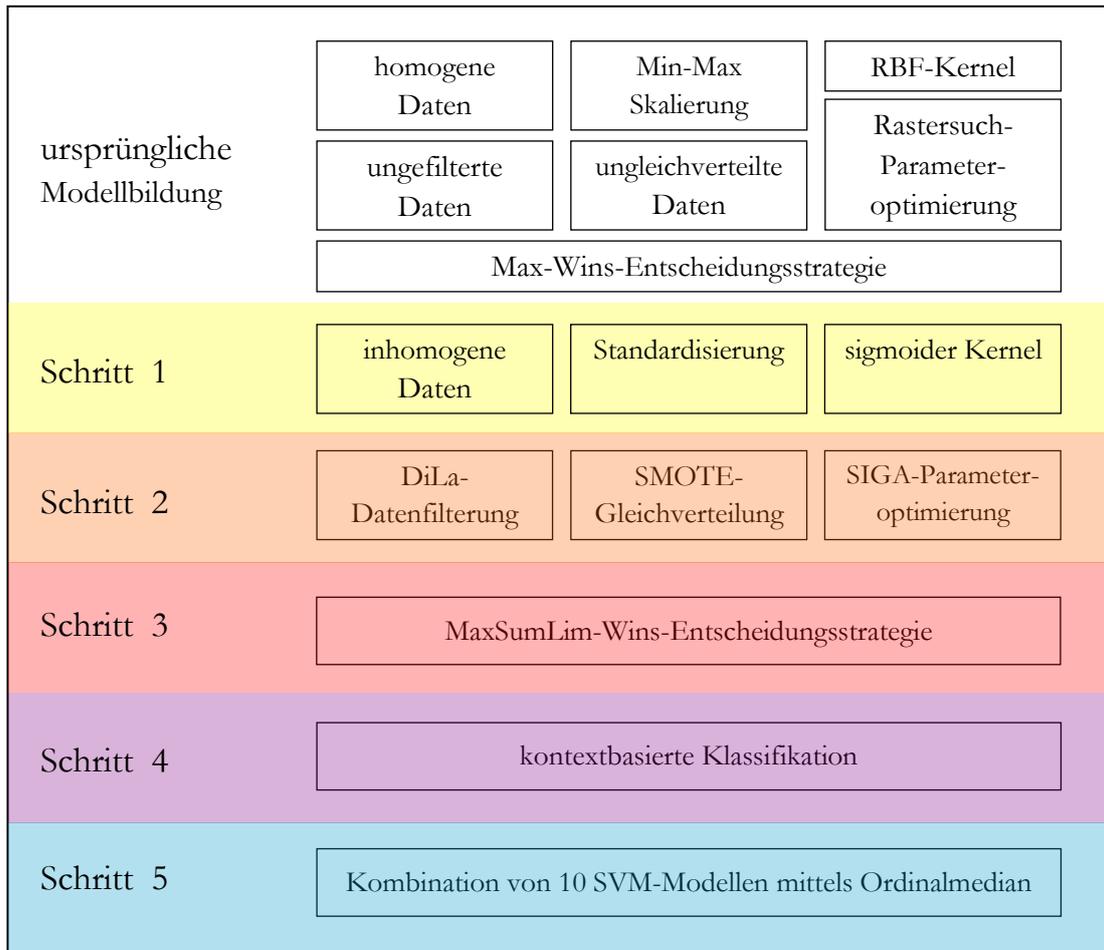


Abbildung 5-1: Darstellung der angewendeten Methoden der Gesamtuntersuchung

- In Schritt 3 fand zusätzlich die MaxSumLim-Wins-Entscheidungsstrategie (Abschnitt 4.5.1 Untersuchung der Entscheidungsstrategie ab Seite 88) Anwendung.⁶⁴
- In Schritt 4 ist für die Klassifikation von Hypnogrammen die kontextbasierte Klassifikation (Abschnitt 4.5.3 Kontextbasierte Klassifikation von Hypnogrammen ab Seite 93) eingesetzt worden.⁶⁵
- Schritt 5, die Kombination der zehn Modelle über Bildung des Ordinalmedians (Abschnitt 4.5.2 Kombination verschiedener SVM mittels Bagging ab Seite 91), vervollständigt die Untersuchungen.⁶⁶

Als Validierungsdatensätze wurden die Datensätze von B2 und B3 verwendet. Die Ergebnisse der Gesamtuntersuchung sind in Form eines Kastendiagramms der gewichteten Güte G_{gew} in Abbildung 5-2 dargestellt. Die Tabelle 5-1 zeigt die zugehörigen Signifikanzwerte in Bezug auf die ursprüngliche Modellbildung aus dem zweiseitigen Wilcoxon-Vorzeichen-Rang-Test, die Tabelle 5-2 zeigt die Signifikanzwerte in Bezug auf den vorhergehenden Bearbeitungsschritt.

⁶⁴ Detaillierte Angaben zu den eingestellten Parametern sind in Tabelle C-16 auf Seite 138 zu finden.

⁶⁵ Detaillierte Angaben zu den eingestellten Parametern sind in Tabelle C-17 auf Seite 139 zu finden.

⁶⁶ Detaillierte Angaben zu den eingestellten Parametern sind in Tabelle C-18 auf Seite 139 zu finden.

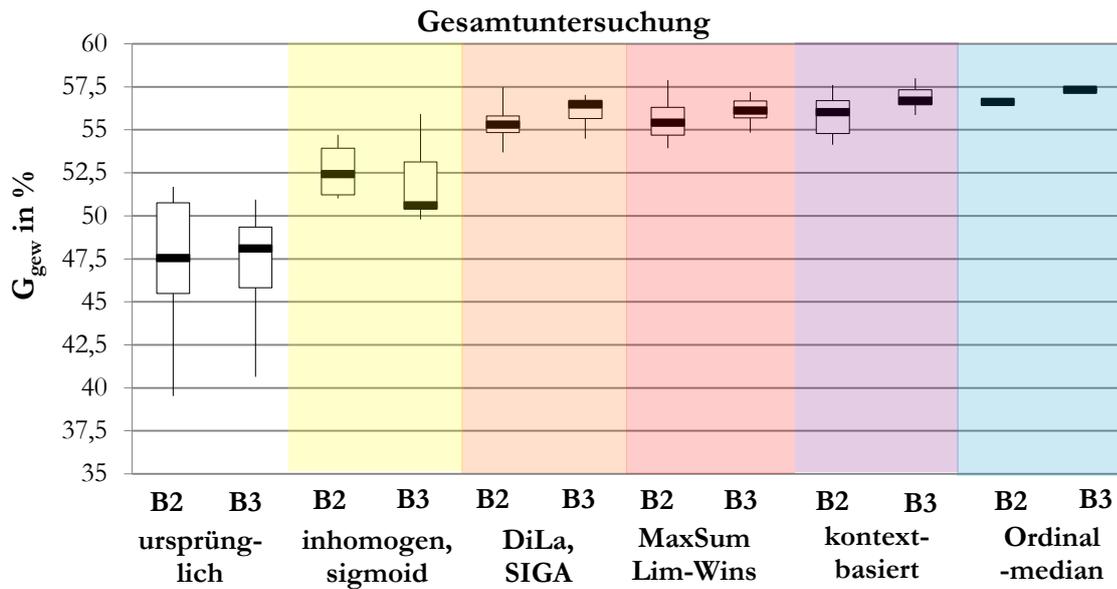


Abbildung 5-2: Kastendiagramm zur Gesamtuntersuchung: die gewichtete Güte G_{gew} ; ursprüngliche Modellbildung (weiß), Schritt 1: Modellbildung inhomogener Daten und sigmoiden Kernel (gelb), Schritt 2: DiLa-Datenfilterung und SIGA-Parameteroptimierung (orange), Schritt 3: MaxSumLim-Wins Entscheidungsstrategie (rot), Schritt 4: kontextbasierte Klassifikation (violett), Schritt 5: Kombination der Modelle mittels Ordinalmedian (blau) - Dabei ist eine stetige Verbesserung der Klassifikationsgüte über die angewendeten Methoden erkennbar.

Signifikanzwerte p_{sig} der Gesamtuntersuchung im Vergleich zur ursprünglichen Modellbildung	G_{gew}	
	B2	B3
Schritt 1: inhomogen, sigmoid	0,007	0,007
Schritt 2: DiLa, SIGA	0,005	0,005
Schritt 3: MaxSumLim-Wins	0,005	0,005
Schritt 4: kontextbasiert	0,005	0,005

Tabelle 5-1: Signifikanzwerte der Gesamtuntersuchung in Bezug auf die ursprüngliche Modellbildung für die gewichtete Güte G_{gew} (zweiseitiger Wilcoxon-Vorzeichen-Rang-Test, $SN=0,05$, signifikante Unterschiede grau hinterlegt)

Signifikanzwerte p_{sig} der Gesamtuntersuchung im Vergleich zu dem vorhergehenden Bearbeitungsschritt	G_{gew}	
	B2	B3
Schritt 1: inhomogen, sigmoid	0,007	0,007
Schritt 2: DiLa, SIGA	0,005	0,005
Schritt 3: MaxSumLim-Wins	0,074	0,799
Schritt 4: kontextbasiert	0,241	0,005

Tabelle 5-2: Signifikanzwerte der Gesamtuntersuchung in Bezug auf den vorhergehenden Schritt für die gewichtete Güte G_{gew} (zweiseitiger Wilcoxon-Vorzeichen-Rang-Test, $SN=0,05$, signifikante Unterschiede grau hinterlegt)

Im ersten Schritt, der Anwendung inhomogener Daten, tritt mit $p_{\text{Sig}}=0,007$ (in Tabelle 5-1 und Tabelle 5-2 gelb dargestellt) für beide Datensätze eine deutliche und vor allem signifikante Besserung der Klassifikationsgüte von etwa 3-5% ein. Der Schritt 2, die DiLa-Filterung und SIGA-Parameteroptimierung, verbessert wiederum mit $p_{\text{Sig}}=0,005$ (in Tabelle 5-2 orange dargestellt) bei beiden Datensätzen die Güte signifikant um etwa 3-5%. Schritt 3, die Anwendung der MaxSum-Lim-Wins-Strategie, bewirkt nur noch eine sehr kleine Änderung von unter einem Prozent. Diese Änderung ist im Vergleich zu Schritt 2 mit $p_{\text{Sig}}=0,074$ (B2) und $p_{\text{Sig}}=0,799$ (B3) aus Tabelle 5-2 (rot) nicht signifikant. Die kontextbasierte Klassifikation von Schritt 4 kann die Güte nochmal um 0,5% anheben. Mit einem Signifikanzwert von $p_{\text{Sig}}=0,005$ (in Tabelle 5-2 violett dargestellt) ist diese Verbesserung für den B3-Datensatz signifikant. Die Kombination der Modelle in Schritt 5 bewirkt nochmals eine Steigerung der Güte um 0,6%. Wenn auch nicht alle Verbesserungen zum vorhergehenden Schritt signifikant sind (Tabelle 5-2), im Vergleich zur ursprünglichen Modellbildung sind durch die stetige Verbesserung der Klassifikationsgüte alle Änderungen signifikant (Tabelle 5-1). Infolge der Anwendung der aufgeführten Methoden steigt die gewichtete Güte des resultierenden Klassifikators um fast 10%.

- Insgesamt lässt sich eine stetige Verbesserung der Klassifikationsgüte mit den ausgeführten Schritten erkennen.
- Die Steigerung ist in den ersten Schritten zunächst groß und wird in den letzten Schritten deutlich geringer.

Die Steigerung der Güte über die angewendeten Schritte beschreibt demnach einen nichtlinearen Verlauf. Sie läuft in eine „Sättigung“. Das bedeutet, es wird mit steigender Güte merkbar schwerer, eine Besserung zu erreichen.

Um nun die Robustheit der resultierenden Klassifikatoren durch die angewendeten Schritte analysieren zu können, sollen die einzelnen Gütewerte betrachtet werden. In Abbildung 5-3 sind die Gütewerte GG, DG, GMG und MG in Form von Kastendiagrammen dargestellt. Die Tabelle 5-3 zeigt die zugehörigen Signifikanzwerte aus dem zweiseitigen Wilcoxon-Vorzeichen-Rang-Test in Bezug auf die ursprüngliche Modellbildung und die Tabelle 5-4 die Signifikanzwerte in Bezug auf den vorhergehenden Bearbeitungsschritt. Bei der Betrachtung der einzelnen Gütewerte in Abbildung 5-3 zeigt sich, dass die GG über die angewendeten Methoden, bis auf eine Ausnahme in Schritt 1, nahezu konstant bleibt. Dies lässt sich durch die Signifikanzwerte aus Tabelle 5-3 bestätigen. Nur der Vergleich von Schritt 1 zur ursprünglichen Modellbildung ergibt einen signifikanten Unterschied. Beim Vergleich aller anderen Schritte zur ursprünglichen Modellbildung bleiben die Signifikanzwerte über dem Signifikanzniveau von 0,05. Allerdings kann der DG durch die angewendeten Methoden insgesamt um mehr als 10% angehoben werden. Die Signifikanzwerte aus Tabelle 5-4 zeigen, dass, mit Ausnahme von Schritt 3, bei jedem Schritt eine signifikante Verbesserung des DG erreicht werden kann. Der GMG kann sogar um 40-50% angehoben werden und ist mit $p_{\text{Sig}}=0,005$ (Tabelle 5-3) signifikant besser zur ursprünglichen Modellbildung. Die MG verbessert sich ebenfalls mit $p_{\text{Sig}}=0,005$ (Tabelle 5-3) signifikant um etwa 20-30%.

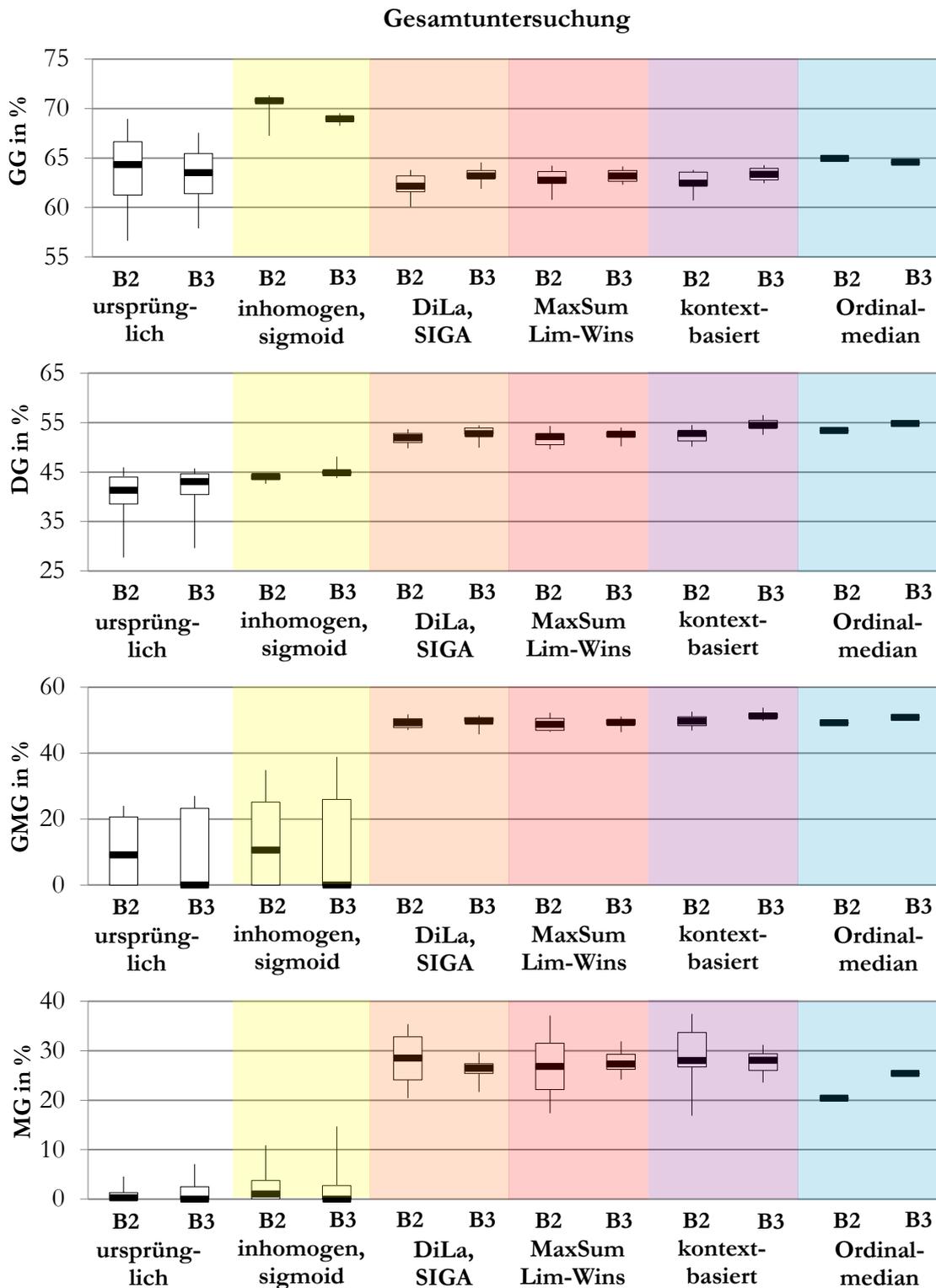


Abbildung 5-3: Kastendiagramme der Gesamtuntersuchung: die Gütewerte GG, DG, GMG und MG (von oben nach unten); ursprüngliche Modellbildung (weiß), Schritt 1: Modellbildung inhomogener Daten und sigmoiden Kernel (gelb), Schritt 2: DiLa-Datenfilterung und SIGA-Parameteroptimierung (orange), Schritt 3: MaxSumLim-Wins Entscheidungsstrategie (rot), Schritt 4: kontextbasierte Klassifikation (violett), Schritt 5: Kombination der Modelle mittels Ordinalmedian (blau) - Dabei zeigt sich vor allem die Stärkung der schlechtesten Klassen (meist die schwach vertretenen Klassen) und damit die Erhöhung der Robustheit des Klassifikators durch Anwendung der vorgestellten Schritte.

Signifikanzwerte p_{sig} der Gesamtuntersuchung im Vergleich zur ursprünglichen Modellbildung	GG		DG		GMG		MG	
	B2	B3	B2	B3	B2	B3	B2	B3
Schritt 1: inhomogen, sigmoid	0,005	0,005	0,047	0,017	0,484	0,600	0,263	0,753
Schritt 2: DiLa, SIGA	0,241	0,878	0,005	0,005	0,005	0,005	0,005	0,005
Schritt 3: MaxSumLim-Wins	0,386	0,959	0,005	0,005	0,005	0,005	0,005	0,005
Schritt 4: kontextbasiert	0,386	0,799	0,005	0,005	0,005	0,005	0,005	0,005

Tabelle 5-3: Signifikanzwerte der Gesamtuntersuchung in Bezug auf die ursprüngliche Modellbildung für die Gütewerte GG, DG, GMG und MG (zweiseitiger Wilcoxon-Vorzeichen-Rang-Test, $SN=0,05$, signifikante Unterschiede grau hinterlegt)

Signifikanzwerte p_{sig} der Gesamtuntersuchung im Vergleich zu dem vorhergehenden Bearbeitungsschritt	GG		DG		GMG		MG	
	B2	B3	B2	B3	B2	B3	B2	B3
Schritt 1: inhomogen, sigmoid	0,005	0,005	0,047	0,017	0,484	0,600	0,263	0,753
Schritt 2: DiLa, SIGA	0,005	0,005	0,005	0,005	0,005	0,005	0,005	0,005
Schritt 3: MaxSumLim-Wins	0,007	0,799	0,959	0,093	0,093	0,059	0,047	0,059
Schritt 4: kontextbasiert	0,017	0,005	0,005	0,005	0,005	0,005	0,128	0,074

Tabelle 5-4: Signifikanzwerte der Gesamtuntersuchung in Bezug auf den vorhergehenden Schritt für die Gütewerte GG, DG, GMG und MG (zweiseitiger Wilcoxon-Vorzeichen-Rang-Test, $SN=0,05$, signifikante Unterschiede grau hinterlegt)

- Die schlecht erkannten Klassen, meist die schwach vertretenen Klassen, können durch die angewendeten Schritte besser klassifiziert werden. Damit steigen die meisten Gütewerte an und der Klassifikator wird deutlich robuster.
- Die Anwendung der abgeänderten Entscheidungsstrategie in Schritt 3 zeigt in der Güte jedoch häufig keine signifikanten Änderungen (in Tabelle 5-4 rot dargestellt).

Der in den Kastendiagrammen demonstrierte Einfluss der aufgeführten Methoden soll nun anhand eines Hypnogramms nochmals verdeutlicht werden. Dazu wurde ein Datensatz von B3 der Proband 351001 aufgrund seines typisch aussehenden, gesunden Hypnogramms ausgewählt. Die klassifizierten Hypnogramme sind in Abbildung 5-4 und Abbildung 5-5 blau dargestellt, in den Abbildungen orange hinterlegt ist die Expertenbewertung. Die ursprüngliche Modellbildung (Abbildung 5-4) wurde der Modellbildung mit Einsatz aller bereits aufgeführten Bearbeitungs- und Optimierungsmethoden (Abbildung 5-5) gegenüber gestellt.

Zunächst wird deutlich, dass beide automatisch klassifizierten Hypnogramme vermehrt Klassenwechsel zeigen. In der ursprünglichen Modellbildung wird die am wenigsten auftretende Klasse Movement gar nicht erkannt. Dies ist, wie in Abschnitt 4.1.1 Wirkung der Gleichverteilung und der untypischen Daten ab Seite 39 erläutert, typisch, wenn die Belehrungsdaten stark ungleichverteilt sind. Auch die Klasse S4, ebenfalls vergleichsweise selten, wird häufig vertauscht (siehe beispielsweise Zeitabschnitt zwischen 150 und 180 Minuten in Abbildung 5-4). Weiterhin fällt auf, dass Vertauschungen nicht nur innerhalb der beiden ähnlichsten Klassen, sondern vermehrt auch andere Vertauschungen auftreten.

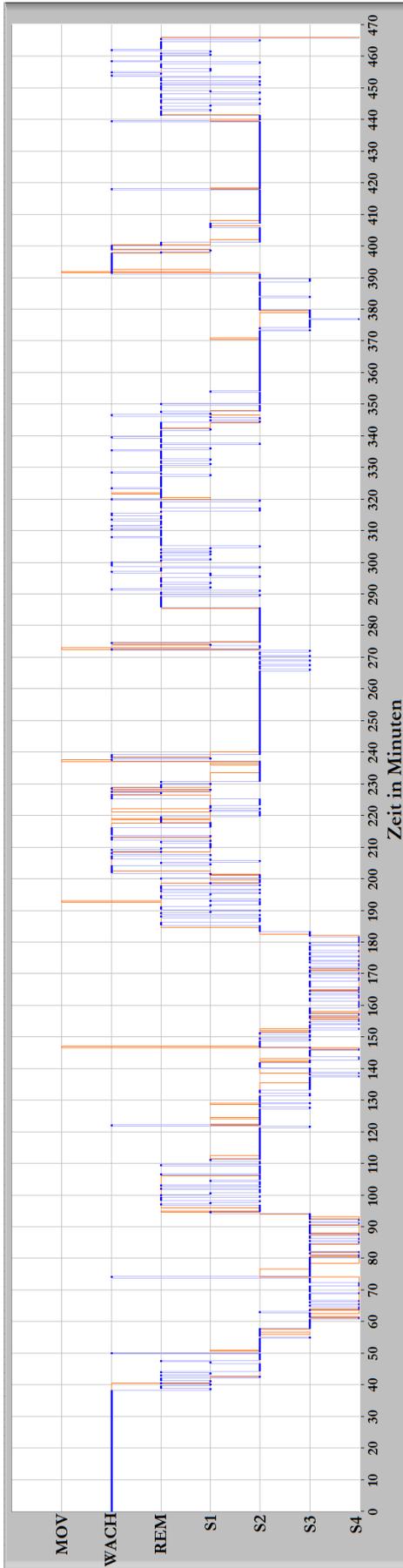


Abbildung 5-4: Klassifiziertes Hypnogramm mittels ursprünglicher Modellbildung (orange) und Expertenhypnogramm (blau) von Proband 351001

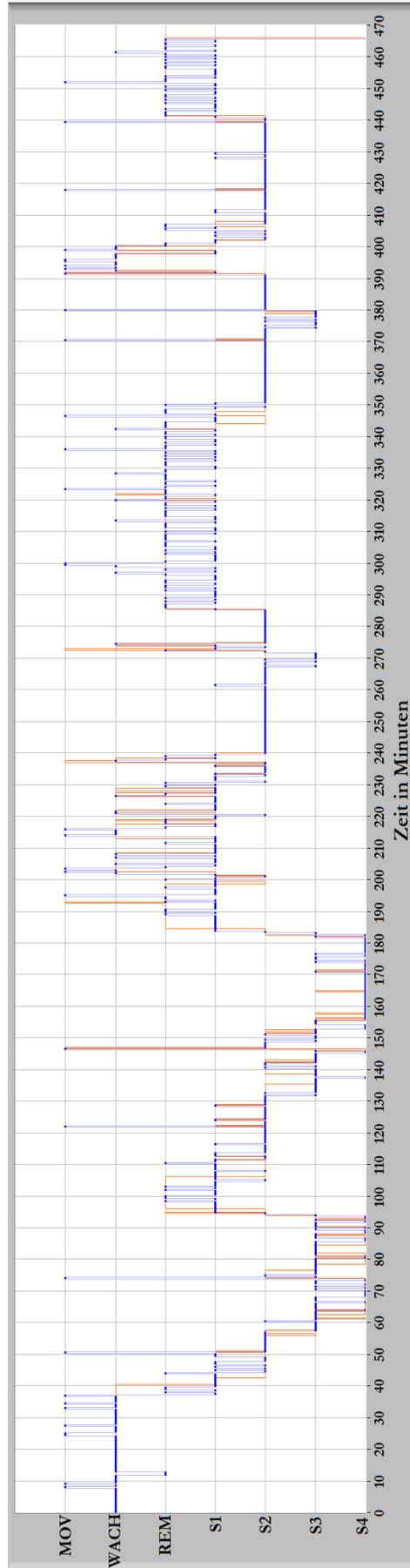


Abbildung 5-5: Klassifiziertes Hypnogramm mittels optimierter Modellbildung (orange) und Expertenhypnogramm (blau) von Proband 351001

Beispielsweise werden im Zeitabschnitt zwischen 290 und 340 Minuten aus Abbildung 5-4 Epochen der Klasse REM den Klassen Wach, S1 und S2 zugeordnet. Es treten vereinzelt auch schwere Vertauschungen auf, wie in Zeitabschnitt zwischen 370 und 380 Minuten. Hier wird die Sollklasse S2 der Tiefschlafklasse S4 zugeordnet.

Die optimierte Modellbildung zeigt im Hypnogramm eine verstärkte Klassifikation der schwach vertretenen Klassen, Movement und S4. Dies ist auf die Anwendung der Gleichverteilung der Belehrungsdaten zurückzuführen. Besonders die Klasse Movement wird im Laufe des Hypnogramms aber auch vermehrt aus allen Klassen fehlklassifiziert, welches auch als Konsequenz der Gleichverteilung zu sehen ist. Auch im Hypnogramm der optimierten Modellbildung sind häufige Wechsel der ermittelten Klasse zu verzeichnen, jedoch beschränken sich die Vertauschungen weitestgehend auf ähnliche Klassen. So wurde im Zeitabschnitt zwischen 290 und 340 Minuten die Klasse REM nicht mit der Klasse S2 vertauscht und nur vereinzelt fand eine Klassifikation von Wach anstatt der gewünschten Klasse REM statt. Schwere Vertauschungen werden durch die Anwendung der Optimierungsmethoden, wie der kontextbasierten Klassifikation, reduziert, sodass im Bereich zwischen 370 und 380 Minuten keine Vertauschungen von S2 mit S4 auftreten.

→ Der Vergleich der beiden Hypnogramme zeigt, wie die optimierte Modellbildung die Klassifikation verändert und auch verbessert. Es werden auch die schwach vertretenen Klassen verstärkt klassifiziert und schwere Vertauschungen reduziert. Dennoch treten vermehrt Klassenwechsel bei der automatischen Klassifikation auf.

Obwohl das prinzipielle Hypnogramm aus der Klassifikation herauszulesen ist, wirken diese Schwankungen wie Rauschen auf einem Signal und machen das Hypnogramm undeutlich. Zur Lösung dieses Problems könnte eine Art Glättung der Klassifikation vorgenommen werden. Eine Glättung des Hypnogramms wird ebenfalls bei der manuellen Bewertung durch den Experten vorgenommen, indem dieser sein Kontextwissen und seine Erfahrung in die Bewertung einbezieht. So könnte der in Abschnitt 4.5.3 Kontextbasierte Klassifikation von Hypnogrammen ab Seite 93 vorgeschlagene Ansatz, die Einbeziehung der vorherigen Zustände und der Übergangswahrscheinlichkeiten, diese Glättung bewirken.

Durch die Verwendung der gleichen Datensätze und Merkmale wie in [Wen05] ist ein direkter Vergleich der Ergebnisse möglich. Dieser Vergleich ist in Form von Kastendiagrammen in Abbildung 5-6 dargestellt. Die Tabelle 5-5 zeigt die zugehörigen Signifikanzwerte aus dem zweiseitigen Wilcoxon-Vorzeichen-Rang-Test. Die Ergebnisse zeigen die Klassifikation der einzelnen Patientendatensätze von B2 und B3, wobei die Klassen S3 und S4 sowie Wach und Movement zusammengefasst wurden. Zunächst ist eine höhere Streuung der meisten betrachteten Kriterien bei der SVM-Klassifikation festzustellen. Dies deutet auf höhere Abweichungen der Klassifikationsgüten für jeden Datensatz bei der SVM-Klassifikation. Im Gegensatz dazu sind die Ergebnisse der neuronalen Netze stabiler. Die Mediane der meisten betrachteten Kriterien sind bei den Netzen höher, besonders für die Klassenklassifikationsraten der Klassen S3/S4 und REM.

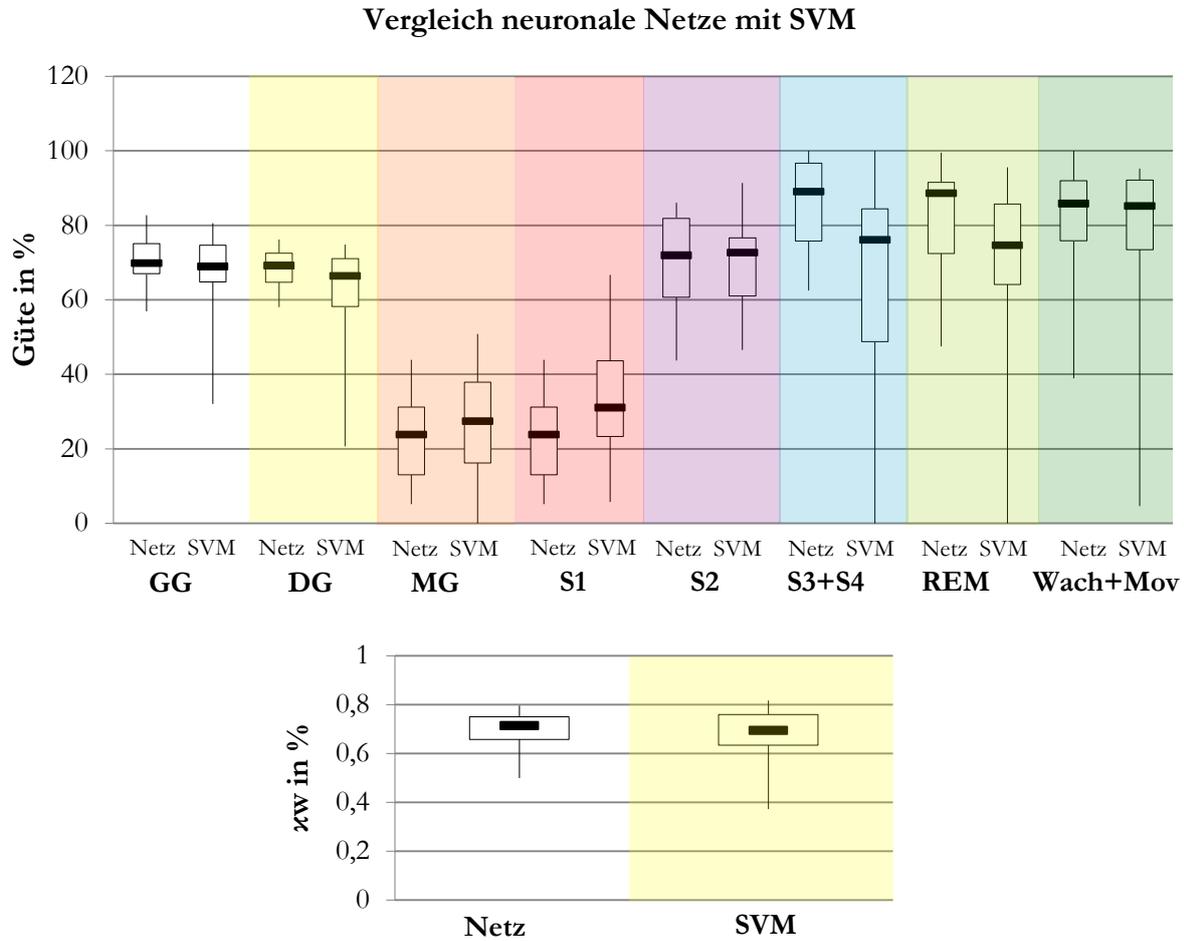


Abbildung 5-6: Kastendiagramme des Vergleichs der neuronalen Netze aus [Wen05] mit der SVM-Klassifikation der Datensätze B2 und B3 bei Zusammenfassung der Klassen S3 und S4 sowie Wach und Movement; Gütwerte (oben): GG (weiß), DG (gelb), MG (orange) sowie die Klassenerkennungsraten von S1 (rot), S2 (violett), S3+S4 (blau), REM (hellgrün) und Wach+Mov (dunkelgrün), Gütwert (unten): gewichteter Kappa-Koeffizient κ_w - Dabei zeigt sich eine bessere Klassifikationsleistung der neuronalen Netze gegenüber der SVM für diese Klassifikationsaufgabe.

Signifikanzwerte p_{Sig} des Vergleichs Netz zu SVM	GG	DG	MG	S1	S2	S3+S4	REM	Wach +Mov	κ_w
	0,0007	0,0001	0,0003	<0,0001	0,0049	<0,0001	0,0014	0,0005	0,0006

Tabelle 5-5: Signifikanzwerte des Vergleichs der neuronalen Netze aus [Wen05] und SVM (zweiseitiger Wilcoxon-Vorzeichen-Rang-Test, SN=0,05, signifikante Unterschiede grau hinterlegt)

Der gewichtete Kappa-Koeffizient, in dem eine Gesamtgüte unter Berücksichtigung schwerer Vertauschungen zu verstehen ist, macht nochmals die bessere Klassifikationsleistung der Netze gegenüber der SVM deutlich. Dieser Gütewert weist ebenfalls für die SVM-Klassifikation eine höhere Streuung auf. Alle Ergebnisse von neuronalen Netzen und SVM unterscheiden sich signifikant, wie Tabelle 5-5 zeigt.

→ Daraus ist zu schließen, dass für die vorliegende Klassifikationsaufgabe trotz aller Datenbearbeitung und Optimierung die SVM nicht an die Ergebnisse der optimierten Populationen von künstlichen neuronalen Netzen [Wen05] heranreicht.

Eventuell kann das SVM-Modell durch weitere Untersuchungen und Maßnahmen noch verbessert werden. Jedoch kann die in der Literatur verbreitete These, die SVM sei ein Ersatz für die neuronalen Netze und würde vergleichbare oder sogar bessere Ergebnisse liefern⁶⁷, für die vorliegenden Daten nicht bestätigt werden.

⁶⁷ "...but are able to serve as a replacement for neural networks that perform as well or better, in a wide variety of fields." [SS01]

6 Einbindung der Support-Vektor-Maschine auf ein ressourcenarmes eingebettetes System

Gegenüber einem Personal Computer (PC) hat ein eingebettetes System geringere Ressourcen. Der Speicher und, besonders bei einer echtzeitfähigen Klassifikation, die Abarbeitungszeit (Taktfrequenz) sind begrenzende Faktoren bei der Implementierung eines Programms auf ein ressourcenarmes eingebettetes System. Der Begriff Echtzeitfähigkeit⁶⁸ im Zusammenhang mit einer Klassifikation wird im Folgenden mehrfach verwendet und beschreibt den Sachverhalt:

$$t_{\text{Abarbeitung}} < t_{\text{Epoche}}, \quad (6.1)$$

wobei $t_{\text{Abarbeitung}}$ die Zeit zur Abarbeitung des Programms darstellt und t_{Epoche} die gewählte Epochenlänge. Für die vorgestellte Anwendung soll die Epochenlänge zehn Sekunden betragen.

Zur Einbindung des SVM-Algorithmus wurde eine Hardware mittlerer Klasse gewählt. Das Evaluation Board LM3S3748 von Stellaris®, dessen Mikrocontroller auf der ARM Cortex™-M3 Architektur basiert, wurde für die Einbindung stellvertretend für diese Klasse verwendet. Dieses Evaluation Board ist in Abbildung 6-1 zu sehen. Die wichtigsten Hardware-Restriktionen des LM3S3748 sind:

	Größe	Einheit
Flash	128	KB
SRAM	64	KB
Max Clock Frequency	50	MHz
ADC Speed	1 Mio.	samples per second

Tabelle 6-1: Hardware-Restriktionen des LM3S3748 Mikrocontrollers [Tex11]

⁶⁸ siehe Begriffsverzeichnis

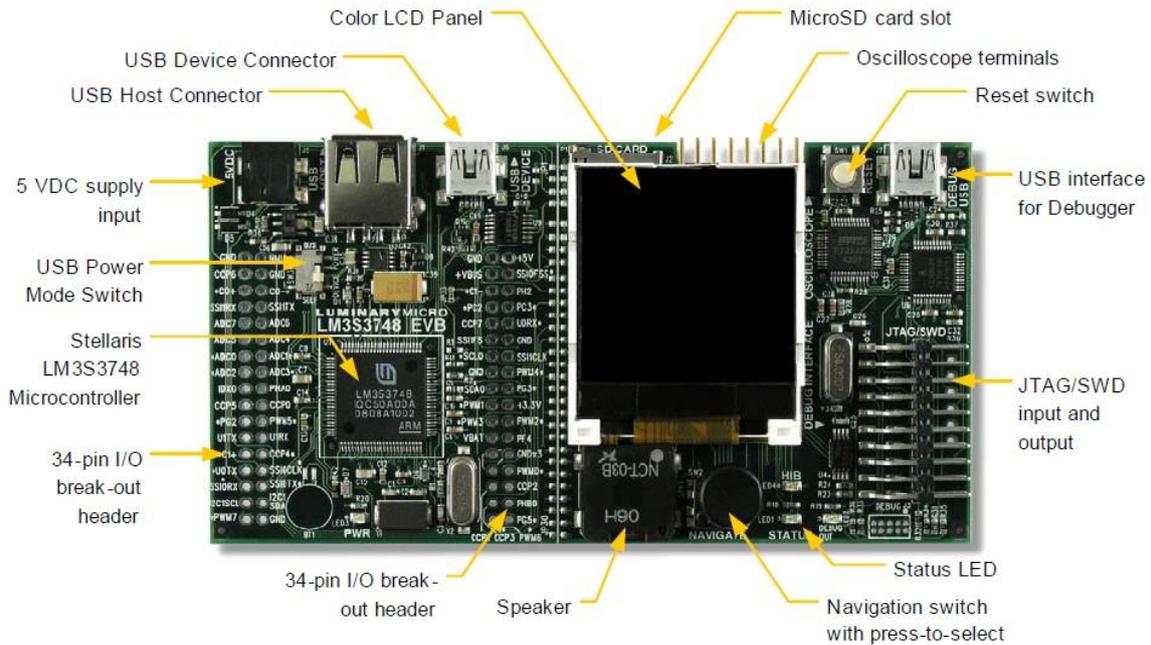


Abbildung 6-1: Fotografie des Stellaris® LM3S3748 Evaluation Board [Tex10]

Ziel ist es, den Klassifikator in einem mobilen Analysegerät einzusetzen. Dadurch reicht es nicht, den Klassifikator alleine auf ein eingebettetes System einzubinden. Ebenso Prozesse wie die Datenakquisition, Merkmalsberechnung und Datenspeicherung müssen bei der Ressourcenabschätzung berücksichtigt werden. Im Hinblick auf die mögliche Anwendung des SVM-Klassifikators in einem mobilen Schlaf-Analysegerät erfolgt die Einbindung des SVM-Algorithmus in das von [Men10] realisierten Software Framework. Dieses mit Free-RTOS (Real Time Operating System) umgesetzte Framework implementiert alle für die Anwendung nötigen Funktionen (Bluetooth-Verbindungsaufbau mit der Elektrode, EEG-Akquisition, Merkmalsextraktion und Schreiben der Roh-EEG-Daten sowie Merkmale und Klassen auf SD-Karte). In Abbildung 6-2 ist der prinzipielle Aufbau des Systems verdeutlicht. Der Mikrocontroller und dessen externe Ein- und Ausgabeelemente sowie Speichermedien bilden die unterste Schicht, die Hardware. Das FreeRTOS-System zusammen mit den von Texas Instruments bereitgestellten Bibliotheken und dem FAT-Dateisystem stellt den Kern der Software dar. Durch die Basisaufgaben zur Benutzerinteraktion und den anwendungsspezifischen Aufgaben zur EEG-Analyse wurde die Funktion eines EEG-Klassifikators verwirklicht. Damit entstand bereits ein erster Prototyp zur automatischen echtzeitfähigen EEG-Klassifikation mit Hilfe einer SVM.

Für die Umsetzung der Software wurde die Programmiersprache C gewählt. Diese ist damit universell einsetzbar. Free-RTOS ist ein frei erhältliches echtzeitfähiges Betriebssystem für präemptives oder kooperatives Multitasking. Die einzelnen Funktionen (engl. Tasks) werden mit Prioritäten versehen und über eine Steuer-Funktion (controlTask) je nach Bedarf und Priorität gestartet und angehalten. Auf dem beschriebenen Prototypen soll eine echtzeitfähige Klassifikation eines Acht-Klassen-Problems mit 53 Merkmalen und einer Epochenlänge von zehn Sekunden realisiert werden. Die Implementierung dieses Problems soll beispielhaft aufzeigen, wie Ressourcen und Modellkomplexität korrelieren.

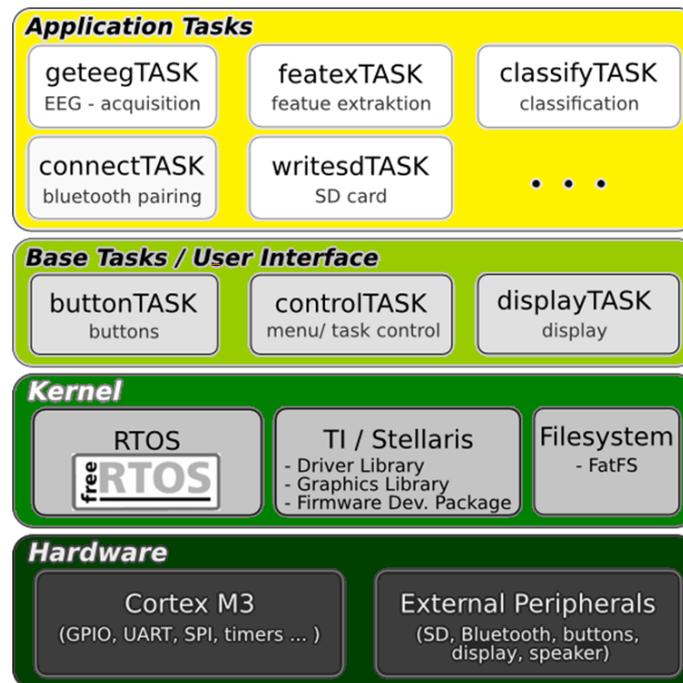


Abbildung 6-2: Funktion und Struktur des mit FreeRTOS eingebundenen Software Frameworks auf dem eingebetteten System [TMBST+11]

Da eine echtzeitfähige Belehrung sehr zeitaufwendig und für die Anwendung nicht zwingend notwendig ist, wurde lediglich der Klassifikationsalgorithmus der SVM implementiert. Zur Implementation der SVM-Klassifikation sind Skalierungs- und Modelldaten aus dem Belehrungsprozess notwendig. Die Skalierungsdatenmenge ist linear abhängig von der Anzahl der Merkmale und beträgt bei 53 Merkmalen etwa 2 Kilobyte (KB). Die Modelldaten hingegen hängen von mehreren Faktoren ab. Da sie aus den Support-Vektoren (SV) bestehen, sind sie ebenfalls von der Anzahl der Merkmale abhängig. Außerdem spielt die Komplexität des Modells, genauer die Anzahl der ermittelten SV, eine wichtige Rolle. Wie in Abschnitt 4.1 Einfluss der Belehrungsdaten ab Seite 39 gezeigt wurde, hat auch die Anzahl und Beschaffenheit der verwendeten Belehrungsdaten Einfluss auf die Komplexität. Des Weiteren bestimmt die Anzahl der Klassen die Anzahl der zu ermittelnden Trennebenen und damit auch die Anzahl der SV. Gerade bei der Einbindung eines Algorithmus auf eine rechenschwache Plattform muss ein Kompromiss zwischen Modellkomplexität und Klassifikationsgüte gefunden werden. Bei einem Acht-Klassen-Problem und 53 Merkmalen beträgt der Speicherbedarf der Modelldaten bei 100 Belehrungs-epochen pro Klasse etwa 500 KB, wobei dieses Modell aufgrund der wenigen Belehrungsdaten und der ursprünglichen Modellbildung bereits sehr einfach gehalten ist. Um Kapazität des internen Speichers, dem Flash, einzusparen, werden die Modelldaten auf der „Secure Digital“ (SD)-Karte abgelegt und nur kurz vor der Verwendung im Algorithmus stückweise ausgelesen. Die zur Berechnung notwendigen Gleitkommawerte wie Merkmale, Koeffizienten oder Skalierungswerte erhielten den „Float“-Datentyp (4 Byte pro Wert) statt des bisher verwendeten „Double“-Datentyps (8 Byte). So konnte die Hälfte an Speicherplatz für solche Werte eingespart werden. Vergleichende Untersuchungen haben gezeigt, dass bei Berechnungen erst die fünfte Stelle nach dem Komma Abweichungen aufwies. Auf das Klassifikationsergebnis hat das keinen Einfluss.

Die Komplexität des Modells erhöht nicht nur den Speicherbedarf der Modelldaten, sondern bedingt Berechnungen mit größeren Feldern, was zum einen den „Static random-access memory“ (SRAM) auslastet und zum anderen die Berechnungszeit erhöht. Da diese beiden Größen, SRAM und Zeit, die begrenzenden Faktoren bei der Implementierung des Algorithmus auf einem eingebetteten System sind, soll in den nächsten Abschnitten gezeigt werden, welche Anforderungen an das System gestellt werden müssen, um die gestellte Aufgabe echtzeitfähig zu realisieren. Alle folgenden Ergebnisse wurden experimentell ermittelt [TMBST⁺11].

6.1 Timing-Anforderungen

Durch die gewünschte Epochenlänge von zehn Sekunden pro Klassifikation werden die Timing-Anforderungen maßgeblich beeinflusst. Um echtzeitfähig klassifizieren zu können, müssen innerhalb von zehn Sekunden folgende Aufgaben von der Software abgearbeitet werden:

- Die nächsten EEG-Daten aufnehmen,
- die Benutzer-Interface-Tasks bearbeiten,
- die Merkmale aus den Roh-EEG-Daten berechnen,
- die Werte skalieren und
- die EEG-Daten klassifizieren.

Zeitlich spielen die ersten zwei Punkte eher eine untergeordnete Rolle. Hauptanteil bilden die Merkmalsextraktion, welche durchschnittlich 0,32 Sekunden Berechnungszeit auf dem gewählten System benötigt, und die Klassifikation, deren durchschnittliche Berechnungszeit in Abbildung 6-3 zu sehen ist. Die Klassifikation (9,62 Sekunden) und die Merkmalsextraktion (0,32 Sekunden) beanspruchen bei acht Klassen bereits durchschnittlich 9,94 Sekunden der 10-Sekunden-Epoche. Weiterhin müssen Sicherheiten für die EEG-Datenakquisition, die Interface-Tasks und Berechnungsschwankungen der Merkmalsextraktion und Klassifikation einbezogen werden.

→ Aus den Daten wird ersichtlich, dass eine echtzeitfähige Acht-Klassen-Klassifikation auf dieser Hardware so nicht möglich ist. Das System ist auf maximal sieben Klassen beschränkt.

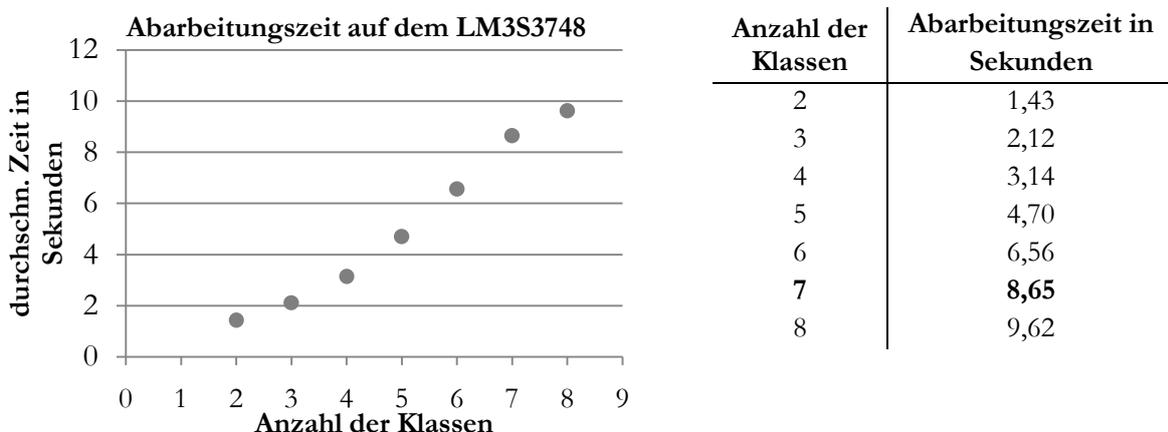


Abbildung 6-3: Diagramm und Wertetabelle der durchschnittlichen Abarbeitungszeiten der SVM-Klassifikation auf dem LM3S3748 in Abhängigkeit der Klassenanzahl - Dabei zeigt sich eine nahezu lineare Abhängigkeit, wodurch die Reduktion der Klassen gut zur Reduktion der Abarbeitungszeit genutzt werden kann.

Den eingesparten Speicherplatz durch das Speichern der Modell-Daten auf SD-Karte wurde mit einer Erhöhung der Abarbeitungszeit erkaufte. Um diesen Faktor so gering wie möglich zu halten, wurden Untersuchungen zur optimalen Anzahl an Bytes, die mit einem Lesevorgang von der SD-Karte gelesen werden, vorgenommen. Dazu wurde die durchschnittliche Abarbeitungszeit einer Acht-Klassen-Klassifikation bestimmt. Aus den Ergebnissen in Abbildung 6-4 wird ein deutliches Minimum bei 16 Bytes sichtbar. Dies lässt sich auf das File Allocation Table (FAT)-System und die dafür genutzte Software-Bibliothek zurückführen.

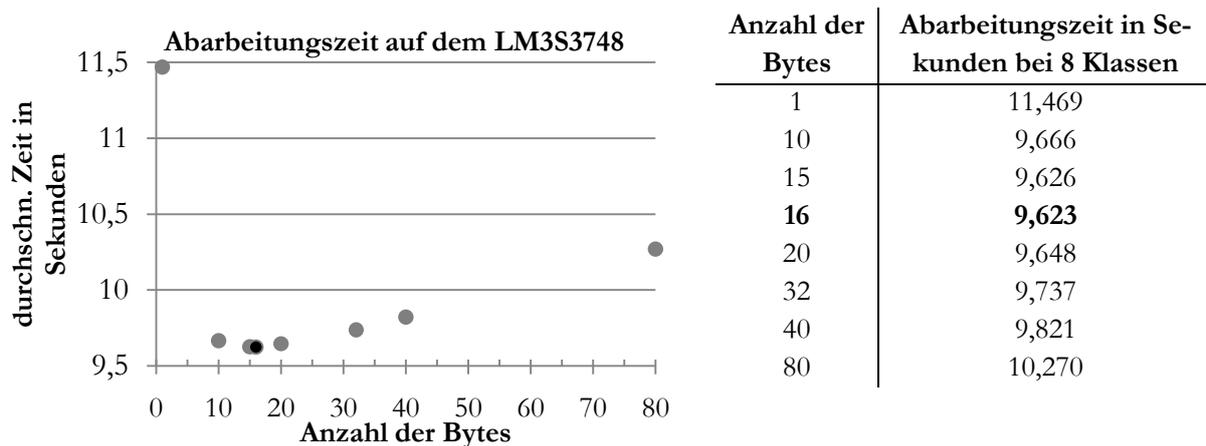


Abbildung 6-4: Diagramm und Wertetabelle der durchschnittlichen Abarbeitungszeiten einer Acht-Klassen-Klassifikation auf dem LM3S3748 in Abhängigkeit der Anzahl der mit einmal ausgelesenen Bytes - Dabei zeigt sich ein Minimum bei 16 Bytes.

Um dennoch eine echtzeitfähige Acht-Klassen-Klassifikation realisieren zu können, gibt es folgende Möglichkeiten:

- Die Steigerung der Hardware-Ressourcen, zum Beispiel den LM3S5791 mit 80MHz Taktfrequenz [Tex09] oder einen anderes leistungsstärkeres System,
- eine Softwareoptimierung,
- die Reduzierung der SVM-Modellkomplexität, bei eventueller Verschlechterung der Klassifikationsgüte,
- die Modell-Datenspeicherung auf einem anderen Speicher mit höherer Zugriffsgeschwindigkeit, zum Beispiel auf ein zusätzliches externes Flashmodul oder
- die Reduzierung der Merkmale.

Welchen Einfluss die Anzahl der Merkmale auf die durchschnittliche Abarbeitungszeit $t_{\text{Abarbeitung}}$ haben, zeigt Abbildung 6-5. Es besteht eine lineare Abhängigkeit der Abarbeitungszeit von der Anzahl der Merkmale. Weiterhin ist erkennbar, dass eine Reduzierung um nur wenige Merkmale zum Erhalt der Echtzeitfähigkeit dieses Klassifikationsmodells mit acht Klassen ausreichen würde.

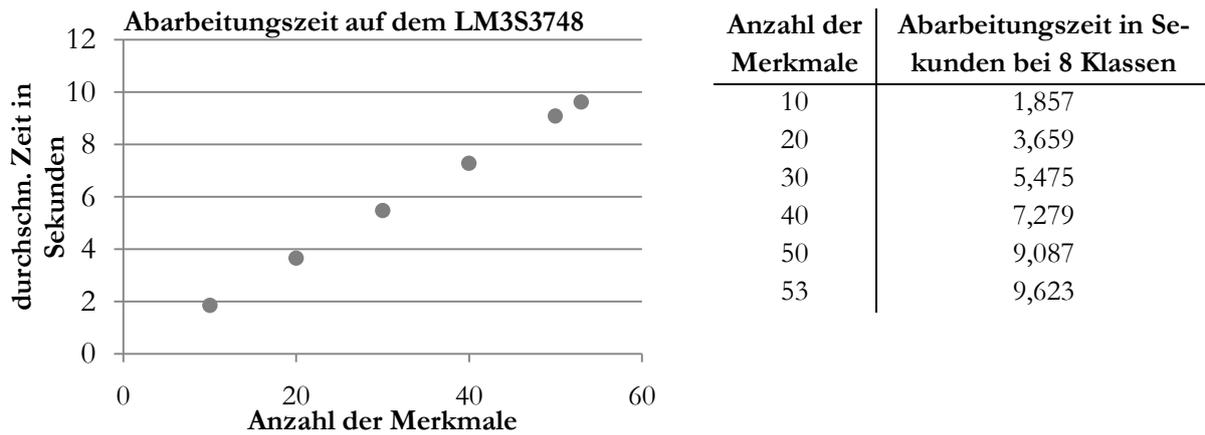


Abbildung 6-5: Diagramm und Wertetabelle der durchschnittlichen Abarbeitungszeiten einer Acht-Klassen-Klassifikation auf dem LM3S3748 in Abhängigkeit der Merkmalsanzahl - Dabei zeigt sich eine lineare Abhängigkeit, wodurch die Reduktion der Merkmale gut zur Reduktion der Abarbeitungszeit genutzt werden kann.

- Bei der Speicherung der Modelldaten auf SD-Karte zeichnet sich für den Lesevorgang eine Datenmenge von 16 Bytes als Optimum heraus.
- Klassenreduktion und Merkmalsreduktion führt direkt zu einer Reduktion der Abarbeitungszeit und ist damit eine gute Möglichkeit, die Echtzeitfähigkeit zu erhalten.

6.2 Speichieranforderungen

Die Speichieranforderungen sind unter anderem von der Anzahl der Klassen und der Anzahl der Merkmale abhängig. Abbildung 6-6 zeigt, welchen SRAM-Bedarf der SVM-Algorithmus bei zwei bis acht Klassen hat. Es besteht eine nichtlineare Abhängigkeit. Bei dem Acht-Klassen-Problem wird ungefähr ein Drittel des SRAM benötigt. Abbildung 6-7 zeigt, welchen SRAM-Bedarf eine Acht-Klassen-Klassifikation in Abhängigkeit der Merkmale hat. Die Abhängigkeit ist, wie auch beim Timing, linear.

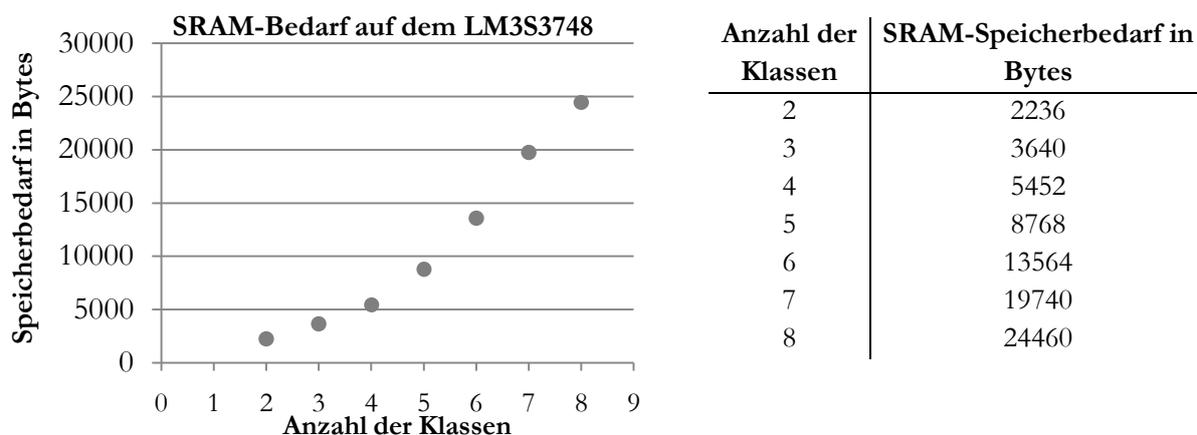


Abbildung 6-6: Diagramm und Wertetabelle des SRAM-Bedarfs des SVM-Algorithmus auf dem LM3S3748 in Abhängigkeit der Klassenanzahl - Dabei zeigt sich eine nichtlineare Abhängigkeit.

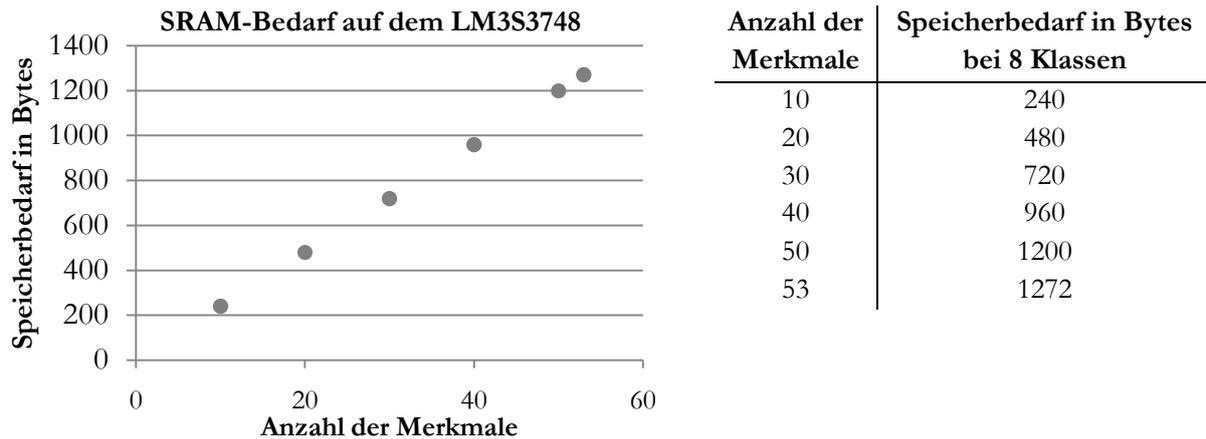


Abbildung 6-7: Diagramm und Wertetabelle des SRAM-Bedarfs der Acht-Klassen-Klassifikation auf dem LM3S3748 in Abhängigkeit der Merkmalsanzahl - Dabei zeigt sich eine lineare Abhängigkeit.

→ Die zur Verfügung stehenden Speicher des SRAM und des Flash sind für die Einbindung und Ausführung des Acht-Klassen-SVM-Algorithmus mit 53 Merkmalen ausreichend, unter der Bedingung, dass die Modell-Daten auf der SD-Karte gespeichert werden.

Anhand der experimentellen Untersuchungen können folgende Schlussfolgerungen getroffen werden:

- Diese sehr einfache Klassifikationsaufgabe mit 100 Belehrungsepochen ist auf der gewählten ressourcenarmen Plattform möglich.
- Auch wenn die Merkmalsreduktion durch die Anwendung der SVM zur Steigerung der Klassifikationsgüte nicht nötig ist (siehe Abschnitt 4.3 Untersuchung der Merkmalsvektoren ab Seite 63), sollten zur Reduzierung der Berechnungszeit und des Speicherbedarfs die Merkmale bei ressourcenarmen Plattformen auf das Nötigste reduziert werden.
- Für die Bewältigung von komplexeren Klassifikationsaufgaben, wie sie zum Beispiel auch durch die Anwendung der Gleichverteilung resultieren, bedarf es einer leistungsstärkeren Plattform.

Eine solche Plattform ist zum Beispiel ein COM (Computer on Modul), oder auch SOM (System on Modul) genannt [Dat16]. Diese Module stellen komplette kleine Computer auf einer einzigen Leiterplatte mit einer Abmessung von etwa 10x10 cm (je nach Typ) dar. Hierbei kann auf verschiedene Hersteller mit verschiedenen Technologien zurückgegriffen werden. Bei einer solchen Implementation müssten jedoch noch die Kosten und der Energiebedarf auf die jeweilige Anwendung abgewogen werden.

7 Zusammenfassung und Ausblick

7.1 Zusammenfassung

Es wurde in dieser Arbeit die automatische Klassifikation von biologischen Daten mit SVM, speziell der Schlafstadienklassifikation anhand von Elektroenzephalogramm (EEG)-Daten, untersucht. Diese untersuchten biologischen Signale:

- zeigen starke inter- und intraindividuelle Ausprägungen,
- sind durch kleine Amplituden extrem störanfällig,
- weisen naturbedingt eine starke Ungleichverteilung der Klassen auf,
- beinhalten ungenaue Daten und
- sind mit abweichenden Expertenbewertungen behaftet.

Insgesamt stellt eine robuste Klassifikation solcher Signale eine anspruchsvolle Aufgabe dar.

Wie in der Abbildung 1-1: Gesamtübersicht des SVM-Generierungs- und Klassifikationsprozesses auf Seite 4 zu sehen, fanden in dieser Arbeit an den verschiedenen Punkten der Belehrung und Klassifikation Untersuchungen statt, um die Methode der SVM besser zu verstehen und die Klassifikationsleistung steigern zu können.

Zunächst wurde unter dem Punkt Belehrungsdatenbearbeitung der Einfluss ungleichverteilter und untypischer Belehrungsdaten auf das Modell der SVM untersucht (Abschnitt 4.1 ab Seite 39). Die Erkenntnisse aus dem synthetischen Drei-Klassen-Problem zeigen, dass stark ungleichverteilte Daten eine Verschiebung der Trennebenen zugunsten der häufig vertretenen Klassen zur Folge hat. Eine Gleichverteilung der Daten oder eine interne Gewichtung ist dadurch bei starken Ungleichverteilungen unbedingt notwendig. Ansonsten würde die häufig vertretene Klasse immer bevorzugt werden. Dadurch steigt jedoch die Modellkomplexität. In diesem Zusammenhang wurde die Gleichverteilungsmethode des SMOTE-Algorithmus [CBHK02] vorgestellt. Untypische Daten, Daten die aufgrund ihrer Merkmalsvektoren etwas abseits der Haupt-Datenmenge liegen, verzerren das Modell und erhöhen dessen Komplexität. Dieser Effekt wird noch verstärkt, wenn diese Daten in eine externe Gleichverteilung mit einbezogen werden. Aus diesem Grund sollten sie aus der Belehrungsdatenmenge gelöscht werden. Dazu wurde der eigens entwickelte

DiLa-Filter zur automatischen Identifikation von untypischen Daten vorgestellt (Abschnitt 4.1.2 ab Seite 50). Dieser neuartige Algorithmus bezieht für die Identifikation sowohl die Dichteinformation eines Vektors als auch seine Lage innerhalb der Klasse ein. Durch die Bearbeitung der Belehrungsdaten mit dem DiLa-Filter, der Parameterbestimmung und der anschließenden Gleichverteilung konnte vor allem die Klassifikation der schlechtesten Klassen (meist die schwach vertretenen Klassen) gestärkt werden und damit die Robustheit des Klassifikators erhöht werden.

Durch Untersuchungen an realen Datensätzen zeigte sich, dass, im Gegensatz zu anderen Klassifikationsmethoden, bei der SVM eine hohe Inhomogenität der Belehrungsdaten zu besseren Klassifikatoren führt. Weiterhin wurden in Abschnitt 4.2 Untersuchung zur Skalierung von Daten ab Seite 60 verschiedene Skalierungsvarianten für die vorliegenden EEG-Daten erprobt, wobei die Methode der Standardisierung die besten Ergebnisse zeigte.

Die Untersuchungen in Abschnitt 4.3 Untersuchung der Merkmalsvektoren ab Seite 63 verdeutlichen, wie sich invalide Merkmale und wie sich sehr viele Merkmale auf die Güte des SVM-Klassifikators auswirken. Die SVM kann im Gegensatz zu vielen anderen Klassifikatoren, wie z. B. den künstlichen neuronalen Netzen [Wen05], mit sehr vielen Merkmalen ohne erheblichen Güteverlust umgehen. Da aber die Anzahl der Merkmale Einfluss auf die Berechnungszeit und den Speicherbedarf haben, ist eine Selektion durchaus sinnvoll. So konnte weiterhin gezeigt werden, dass invalide Merkmale das SVM-Modell negativ beeinflussen. Sie sollten daher aus den Daten entfernt werden.

Unter dem Thema Belehrung der SVM (siehe Abbildung 1-1 auf Seite 4) wurden verschiedene Kernel-Funktionen sowie die SVM-Parameteroptimierung, untersucht. Bei den Untersuchungen zur Kernel-Funktion erwies sich der sigmoide Kernel als der geeignetste Kernel für die angewendeten Daten (Abschnitt 4.4.2 Empirische Untersuchung der Kernel-Funktion ab Seite 87). Für die Parameteroptimierung wurden die evolutionären Algorithmen thematisiert (Abschnitt 4.4.1 ab Seite 66). Speziell für die Parameteroptimierung der SVM zur Schlafklassifikation wurde der SIGA (Special Immigrative multi-criteria Genetic Algorithm) entwickelt. Dies ist ein neuer multikriterieller genetischer Algorithmus, basierend auf dem von [DPAM00] veröffentlichten NSGA-II. Durch die neue genetische Operation, der Immigration, kann die Wahrscheinlichkeit einer einseitigen Entwicklung der Lösungspopulation, d. h. die Wahrscheinlichkeit für lokale Optima, verringert werden. Untersuchungen mit synthetischen und realen Daten zeigten nicht nur den Vorteil der genetischen Algorithmen gegenüber einer Rastersuche, sondern konnten auch die verbesserte Konvergenz des SIGA gegenüber dem NSGA-II nachweisen.

→ Insgesamt konnten in diesen Untersuchungen das Verhalten und damit die Vor- und Nachteile der SVM bestimmt werden. Es stehen bei dieser Methode nur geringe Steuermöglichkeiten für die Komplexität des Modells zur Verfügung. Aus diesem Grund sind die Belehrungsdaten ein zentraler Faktor für die Güte und Komplexität des entstehenden Modells. Eine Analyse und die Bearbeitung der Belehrungsdaten ist somit unabdingbar.

Auch nach der Bildung des Modells können verschiedene Klassifikationsmodifikationen die Klassifikationsgüte beeinflussen. Unter dem Punkt automatische Klassifikation in Abbildung 1-1 auf Seite 4 sind die untersuchten Methoden aufgeführt. Zunächst wurden einige Entscheidungsstrategien (Abschnitt 4.5.1 ab Seite 88) untersucht, die nicht nur das Vorzeichen, sondern auch

den resultierenden Wert der Entscheidungsfunktion einbeziehen. Es konnten bei den Methoden MaxSumLim und MaxSumNorm leichte Verbesserungen der Güte festgestellt werden. Bei der Kombination mehrere Modelle, dem sogenannten „Bagging“ (Abschnitt 4.5.2 ab Seite 91), kann für die Schlafklassifikation eine Ordinalskala der Schlafstadien genutzt werden. Dabei erwies sich die Bildung des Ordinalmedians als geeignetste Methode. Für die Klassifikation eines Hypnogramms sollte die Tatsache benutzt werden, dass bestimmte Stadienübergänge unwahrscheinlich sind. Unter Einbeziehung der vorherigen Schlafstadien bei der hierfür entwickelten kontextbasierten Klassifikation (Abschnitt 4.5.3 ab Seite 93) kann die Klassifikationsgüte gesteigert und schwere Vertauschungen stark reduziert werden. Damit wird die Robustheit des Klassifikator gesteigert.

Die Anwendung aller resultierenden Methoden in Kombination aus Abschnitt 5 ab Seite 97 ergab eine wesentliche Verbesserung der SVM-Klassifikation. Besonders die am schlechtesten erkannten Klassen, welche meist die schwach vertretenen Klassen darstellten, konnten durch die angewendeten Maßnahmen erheblich gestärkt werden. Die Hypnogrammklassifikation bestätigte diese Erkenntnisse. Sie zeigte außerdem, dass schwere Vertauschungen minimiert und die verbleibenden Vertauschungen meist nur innerhalb ähnlicher Klassen stattfanden. Obwohl die resultierenden Hypnogramme vermehrte Klassenwechsel aufwiesen, war der prinzipielle Schlafverlauf gut erkennbar. Ein direkter Vergleich der optimierten SVM-Klassifikatoren mit den optimierten Populationen von künstlichen neuronalen Netzen konnte die These, die SVM-Methode sei ein Ersatz für die neuronalen Netze mit einer vergleichbaren oder sogar besseren Klassifikationsleistung⁶⁹, nicht bestätigen.

Die Anwendung der SVM-Klassifikation auf einer ressourcenarmen Plattform ist, wie in Abschnitt 6 ab Seite 107 nachzulesen, unter Berücksichtigung der vorhandenen Ressourcen möglich. Hierbei sind aufgrund der eingeschränkten Ressourcen die implementierbare Modellkomplexität und die benötigte Klassifikationsgüte abzuwägen. Somit können viele Anwendungen mit Hilfe der SVM realisiert werden. Zum Beispiel entstand im Rahmen dieser Arbeit bereits ein erster Prototyp zur automatischen und mobilen Schlafklassifikation.

→ In Anbetracht der Komplexität dieser Art von Klassifikationsaufgaben durch extrem ungleichverteilte, stark schwankende und mit Störungen behaftete Daten, zeigt sich die SVM nach Optimierung als brauchbarer Klassifikator. Die in der Literatur verbreitete Aussage, die Methode der SVM wäre anderen Klassifikationsmethoden überlegen, konnte an den vorliegenden Daten jedoch nicht bestätigt werden. Die Methode erfordert zur Generierung geeigneter Modelle einen hohen Optimierungsaufwand, besonders da die SVM nur eingeschränkte Möglichkeiten zur Komplexitätssteuerung besitzt und somit die Bearbeitung der Belehrungsdaten zur Modelloptimierung genutzt werden muss. Dennoch schnitten beim Vergleich die in vier Ebenen optimierten Populationen von künstlichen neuronalen Netzen in fast allen Gütekriterien besser ab als die optimierten SVM-Klassifikatoren.

⁶⁹ “...but are able to serve as a replacement for neural networks that perform as well or better, in a wide variety of fields.” [SS01]

→ In dieser Arbeit konnten zum einen die Stärken und Schwächen der SVM-Klassifikation durch experimentelle Untersuchungen belegt und zum anderen ein Vorschlag zur Vorgehensweise der SVM-Optimierung aufgezeigt werden.

Trotz des Versuchs an allen Stellen der SVM-Optimierung (Belehrungsdaten, Belehrung, Klassifikation) anzusetzen, überschreitet der Forschungsbedarf bei weitem den Umfang dieser Arbeit. So konnten bei den einzelnen Untersuchungen oftmals nur einige wenige Varianten erprobt werden, die Umsetzung vieler Ideen musste zudem ausgesetzt werden.

7.2 Ausblick

Es konnte in dieser Arbeit gezeigt werden, dass die Anwendung von verschiedenen Datenbearbeitungsalgorithmen, Skalierungs- und Klassifikationsvarianten sowie Klassifikationserweiterungen erhebliche Modellverbesserungen bewirken können. Durch die Vielzahl der möglichen Optimierungsmöglichkeiten konnten jedoch nur einige wenige für diese Art der Klassifikationsaufgabe erprobt werden.

So existieren noch viele weitere Gleichverteilungs- und Skalierungsprinzipien. Es könnten weitere Abwandlungen des vorgestellten DiLa-Filters untersucht oder komplett andere Algorithmen zur Identifikation untypischer Daten entwickelt werden. Das Thema der Merkmalsselektion konnte in dieser Arbeit nur ansatzweise behandelt werden. Eine gezielte Auswahl der Merkmale speziell zur SVM-Klassifikation der komplexer biologischer Daten kann nicht nur die Klassifikationsgüte erhöhen, sondern auch die Modellkomplexität reduzieren. Durch den nachgewiesenen Einfluss der Belehrungsdaten auf das Modell entstand die Idee eines Algorithmus zur Belehrungsdatenwahl, welcher Belehrungsdaten aus einer Menge auswählt und auf ihre Eignung zur Modellbildung prüft. Die Art der Datenauswahl, zufällig oder gezielt, sowie die Gütebewertung kann hierbei frei variiert werden. Zum Beispiel könnten zur Gütebewertung zwei oder mehrere Validierungsdatensätze eine Anpassung auf nur einen Datensatz vermeiden. Ein solcher Algorithmus, wie beispielhaft in Abbildung B-3 auf Seite 131 dargestellt, könnte Gegenstand weiterer Untersuchungen zur Optimierung der SVM-Klassifikation sein.

Es existieren weiterhin eine Vielzahl von SVM-Arten und Abwandlungen, wie sie ansatzweise in Abschnitt 3.1.9 und Abschnitt 3.1.10 ab Seite 21 vorgestellt wurden. Eine genaue Analyse für biologische Daten anhand der vorliegenden Klassifikationsaufgabe steht noch aus. Künftig könnten weitere Kernel-Funktionen, Methoden der Parameteroptimierung, Entscheidungsstrategien und SVM-Modellkombinationen ebenfalls Gegenstand der Forschung sein. So könnte auch die vorhandene Parameteroptimierung durch den SIGA mittels einer über den Suchraum verteilten Anfangspopulation verbessert werden. Weiterhin würde eine Gewichtung der Ergebnisse bei der SVM-Modellkombination die Klassifikationsgüte eventuell erhöhen. Die automatisch klassifizierten Hypnogramme, wie in Abschnitt 5 in Abbildung 5-4 und Abbildung 5-5 auf Seite 103 zu sehen, weisen häufige Klassenwechsel auf. Durch eine Art Glättung dieser Hypnogramme könnten diese noch deutlicher gemacht werden. Ein vorgeschlagener Ansatz dazu wäre die Einbeziehung der vorherigen klassifizierten Zustände und die Übergangswahrscheinlichkeiten der einzelnen Klassen. Da die Übergangswahrscheinlichkeiten über eine Nacht nicht konstant sind, müssten diese dynamisch bestimmt werden. So würde die Expertenbewertung wohl am besten

nachempfunden werden. Dies zu untersuchen, überschritt den Umfang dieser Arbeit und sollte daher in künftigen Arbeiten abgeklärt werden.

Die Implementierung auf ein ressourcenarmes eingebettetes System, wie in Abschnitt 6 ab Seite 107 zu sehen, ist mit Einschränkungen möglich. Diese Implementierung kann durch Modelloptimierung, Softwareoptimierung und Hardwareanpassung weiter verbessert werden. Jedoch konnte nur die Klassifikation implementiert werden. Die Belehrung auf einem ressourcenarmen System ist nicht ohne erheblichen Qualitätsverlust möglich, da wichtige Hilfsalgorithmen (Datenwahl, Datenlöschung, Parametersuche usw.) zu viele Ressourcen benötigen. Es könnte jedoch eine Art Adaptionroutinen vor jeder Klassifikation entwickelt werden, welche die Anpassung des Modells auf die aktuellen Daten ermöglicht. Dabei könnte zum Beispiel mittels Parameter eine Modellanpassung erreicht werden. Eine weitere Möglichkeit ist die Implementation auf eine Plattform mit ausreichend Ressourcen, wie zum Beispiel einem COM (Computer on Modul), oder auch SOM (System on Module) genannt [Dat16]. Jedoch müssten die Kosten und der Energiebedarf für die jeweilige Anwendung abgewogen werden.

Dies gibt nur einen kleinen Einblick in die noch bestehenden Forschungsfragen und erklärt die anhaltende Flut an Veröffentlichungen auf diesem Gebiet.

Anhang A

Schlafklassifikation mittels EEG-Daten

A1. Verwendeter EEG-Datensatz und dessen Ableitung

Die verwendeten Datensätze wurde bereits in der Arbeit von [Wen05] genutzt, welche diese als Aufzeichnungssatz B1, B2 und B3 bezeichnete. Diese Aufzeichnungen wurden an der Freien Universität Berlin in Zusammenarbeit mit der Fachhochschule Schmalkalden aufgenommen. Der Datensatz B1 besteht aus zehn Probanden, von denen keiner schwer schlafgestört ist. Der Datensatz B2 besteht aus 16 Probanden, unter denen vorwiegend schlafgestörte Probanden sind. Der Datensatz B3 umfasst elf gesunde bzw. leicht schlafgestörte Probanden. Die drei Datensätze sind in Tabelle A-1, Tabelle A-2 und Tabelle A-3 im Detail aufgelistet.

Berlin1								
Probandenmr.	S1	S2	S3	S4	REM	WACH	MOV	Summe
970142PB	262	1120	97	7	559	367	27	2439
970152PB	266	1299	159	103	643	159	39	2668
970372	200	1141	271	89	554	525	37	2817
970482	140	1281	249	61	757	516	18	3022
970502	256	1227	235	289	392	301	51	2751
970522	402	1047	280	82	760	229	49	2849
970542	339	1332	274	30	759	269	58	3061
970553ES	375	1517	230	3	600	362	105	3192
970572SD	277	1550	263	99	579	463	48	3279
970592	343	774	308	246	243	1308	16	3238
Summe	2860	12288	2366	1009	5846	4499	448	29316
Summe in %	9,8	41,9	8,1	3,4	19,9	15,3	1,5	

Tabelle A-1: Auflistung der Epochen vom Datensatz B1 aufgeschlüsselt nach Klasse und Proband [Wen05]

B2								
Probandennr.	S1	S2	S3	S4	REM	WACH	MOV	Summe
970602PB	137	285	62	35	138	67	5	729
970612	105	395	94	40	171	247	4	1056
970622	163	313	45	1	81	437	15	1055
970636	330	409	4	0	106	152	19	1020
970641	166	412	5	0	83	346	14	1026
970642	148	500	16	0	206	179	20	1069
970652	129	253	21	1	93	463	4	964
970662	114	465	3	0	114	285	4	985
970682	209	358	0	0	147	323	10	1047
970692	109	546	32	77	169	93	11	1037
970701	104	546	22	2	194	74	3	945
970702	127	619	13	0	187	26	7	979
970712	107	254	65	5	106	418	5	960
970732	105	336	78	35	150	290	13	1007
970742	105	359	10	0	196	197	5	872
970772	138	439	13	0	242	261	8	1101
Summe	2296	6489	483	196	2383	3858	147	15852
Summe in %	14,5	40,9	3,0	1,2	15,0	24,3	0,9	

Tabelle A-2: Auflistung der Epochen vom Datensatz B2 aufgeschlüsselt nach Klasse und Proband [Wen05]

B3								
Probandennr.	S1	S2	S3	S4	REM	WACH	MOV	Summe
19980	53	632	71	50	218	42	1	1067
121980	87	568	93	65	183	28	2	1026
175990	113	394	59	108	277	17	9	977
267990	86	538	48	103	197	29	13	1014
271980	96	303	47	109	137	140	0	832
351001	75	371	54	91	209	127	5	932
441011	58	387	95	144	186	72	1	943
451001	104	415	90	52	138	142	0	941
1712980	158	506	56	0	174	98	4	996
1931011	97	464	133	31	236	48	1	1010
2631011	109	383	117	83	202	69	1	964
Summe	1036	4961	863	836	2157	812	37	10702
Summe in %	9,7	46,4	8,1	7,8	20,2	7,6	0,3	

Tabelle A-3: Auflistung der Epochen vom Datensatz B3 aufgeschlüsselt nach Klasse und Proband [Wen05]

Die Ableitung der EEG-Daten erfolgte über die typische 23-kanalige Ableitung nach dem 10-20-System. Diese Ableitung zur Schlafklassifikation ist in Abbildung A-1 schematisch dargestellt. Jedoch wurde parallel dazu das EEG 1-kanalig an den Punkten Fp1' und Fp2' mit Fpz als Referenz (in Abbildung A-1 blau) im Hinblick auf eine einfache Handhabbarkeit eines mobilen Gerätes abgenommen. Die Abtastrate betrug dabei 128 Hz mit einer Auflösung von 10 bit. Die Expertenklassifikation konnte somit anhand der gewohnten 23-kanaligen Signale erfolgen, wobei das

1-kanalige Signal zur Konditionierung des automatisch klassifizierenden Algorithmus dient. In einer späteren mobilen Anwendung soll nur noch die einkanalige Ableitung zur Bewertung des Schlafstadiums herangezogen werden, welches mittels einer Stirnelektrode aufgenommen wurde.

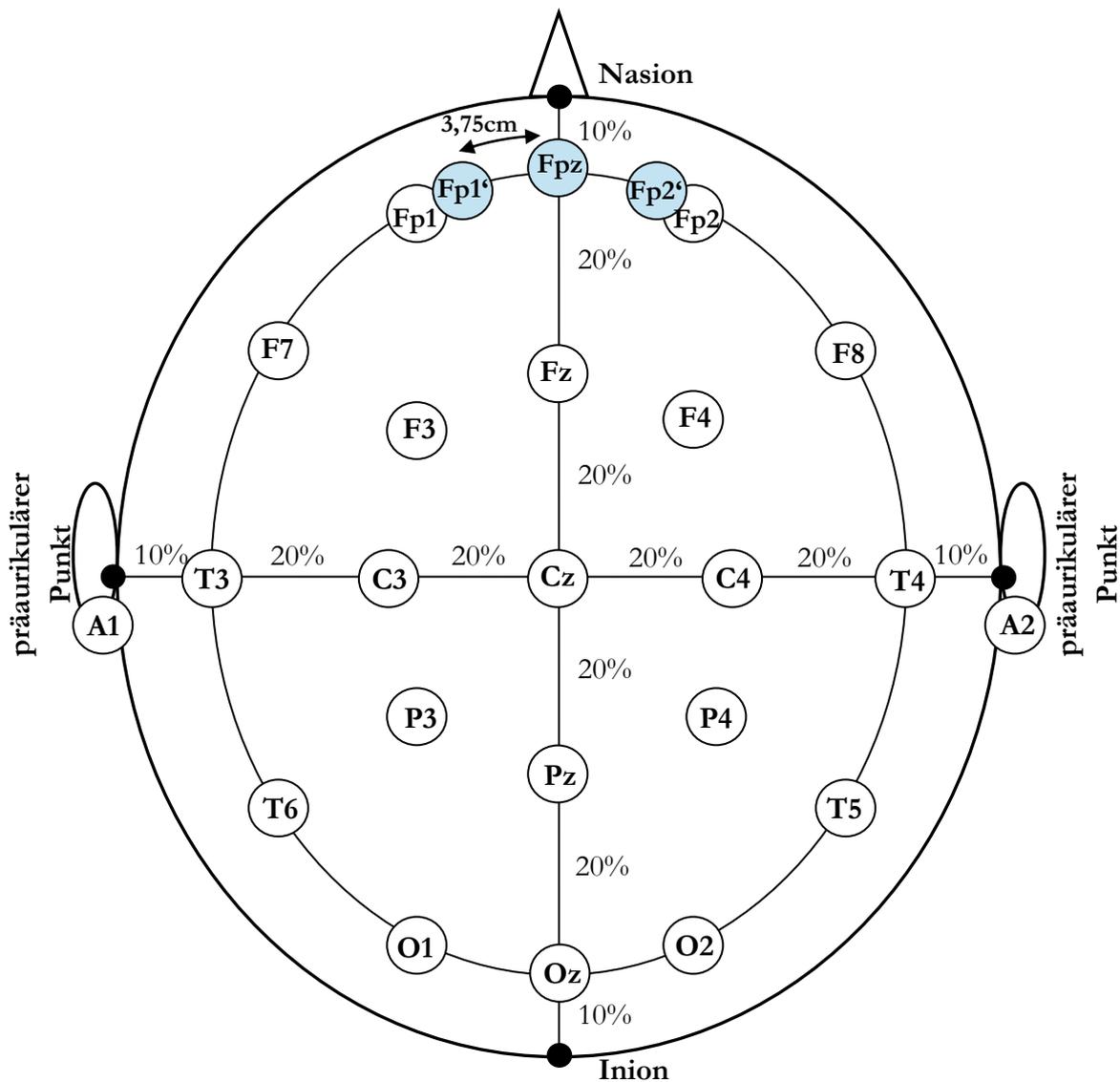


Abbildung A-1: EEG-Elektrodenpositionierung am menschlichen Kopf nach dem 10-20 System [Wen05]

A2. Manuelle Bewertung des EEG

Im Gegensatz zur Expertenbewertung, die hauptsächlich anhand des Zeitsignals erfolgt, soll der automatisch klassifizierende Algorithmus die Zustände mit Hilfe von Merkmalen im Frequenzbereich ermitteln. Diese hier aufgeführten Grundlagen zur manuellen Schlafbewertung wurden bereits in [Wen05] dargelegt. Es sind im Folgenden die typischen Signalverläufe der Zustände im Zeit- und Frequenzbereich dargestellt. Zur Bewertung des Zustandes im Frequenzbereich wird dieser in Frequenzbänder eingeteilt:

- Deltaband im Bereich von 0,5 bis 4 Hertz (Hz),
- Thetaband im Bereich von 4 bis 8 Hz,
- Alphanband im Bereich von 8 bis 13 Hz und
- Betaband im Bereich von 13 bis 30 Hz.

Je nach Leistungsanteil in diesen Bändern kann Rückschluss auf das Schlafstadium gezogen werden.

In Abbildung A-2 sind die Signalverläufe für den Zustand Wach zu sehen, frontal abgeleitet an den Positionen Fp1' und Fp2'. Dieser Zustand ist hauptsächlich von Frequenzen im Alpha- und Betaband oder höher geprägt. Die Amplitude des Zeitsignals ist gering.

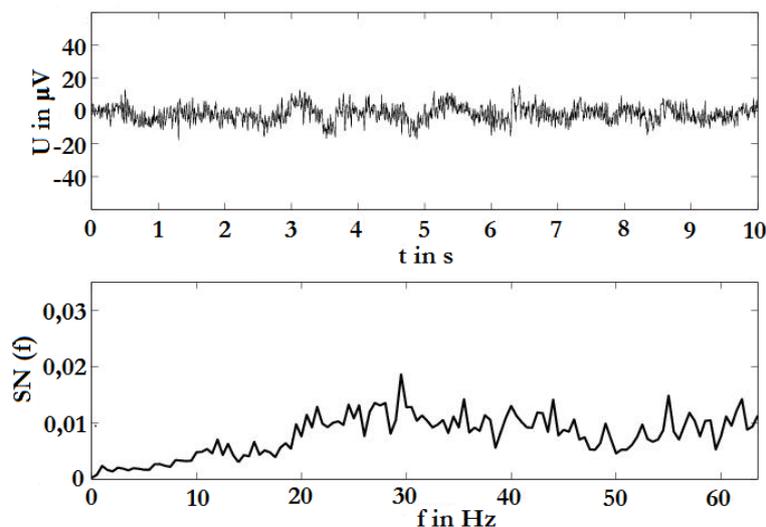


Abbildung A-2: Signalverlauf des EEG-Signals und des zugehörigen Amplitudenspektrums im Wachzustand abgeleitet an den Positionen Fp1' und Fp2' [Wen05]

Das Schlafstadium S2 ist geprägt von Spindeln und K-Komplexen. Die Schlafspindeln, in Abbildung A-3 im Bereich von 2-3 Sekunden zu sehen, sind eine Abfolge von abgrenzbaren Wellen im Bereich von 11 bis 16 Hz für eine Dauer von mehr als 0,5 Sekunden. Die K-Komplexe, in Abbildung A-3 im Bereich von 3 bis 4 Sekunden zu sehen, bestehen aus einer steilen negativen Amplitude, gefolgt von einer flachen positiven für eine Zeitdauer von 0,5 bis 1 Sekunde. Die Leistungsanteile des Frequenzspektrums verlagern sich hin zu den niedrigeren Frequenzen im Alpha- und Thetaband. Die Amplitude des Signals ist gerade durch die K-Komplexe höher als im Wachzustand.

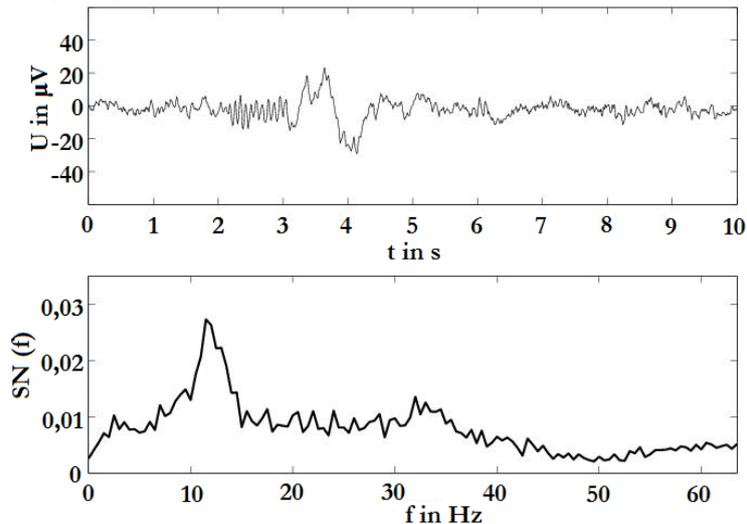


Abbildung A-3: Signalverlauf des EEG-Signals und des zugehörigen Amplitudenspektrums im Zustand S2 abgeleitet an den Positionen Fp1' und Fp2' [Wen05]

Das Zeitsignal und das zugehörige Frequenzspektrum des Zustandes S4, dem Tiefschlaf, ist in Abbildung A-4 zu sehen. Das Frequenzspektrum zeigt nun deutlich hohe Leistungsanteile in den tiefen Frequenzen im Delta- und Thetaband. Die Amplitude des Zeitsignals steigt im Gegenzug.

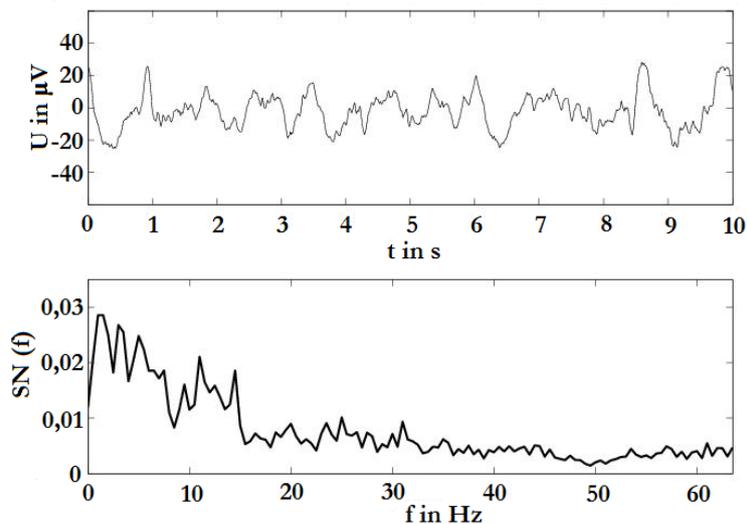


Abbildung A-4: Signalverlauf des EEG-Signals und des zugehörigen Amplitudenspektrums im Zustand S4 abgeleitet an den Positionen Fp1' und Fp2' [Wen05]

Eine gesonderte Rolle nimmt der Zustand REM ein, in Abbildung A-5 zu sehen. Ein typisches Signalmuster für den Zustand REM ist die Sägezahnwelle, welche hier im Bereich von 3,5 bis 5,5 Sekunden im Zeitsignal auftritt. Die Sägezahnwelle hat eine Frequenz von 2 bis 6 Hz und dauert etwa 2 Sekunden. Abgesehen von den Sägezahnwellen kommen bei REM nur Wellen mit niedriger Amplitude und hohen Frequenzen im Betaband vor.

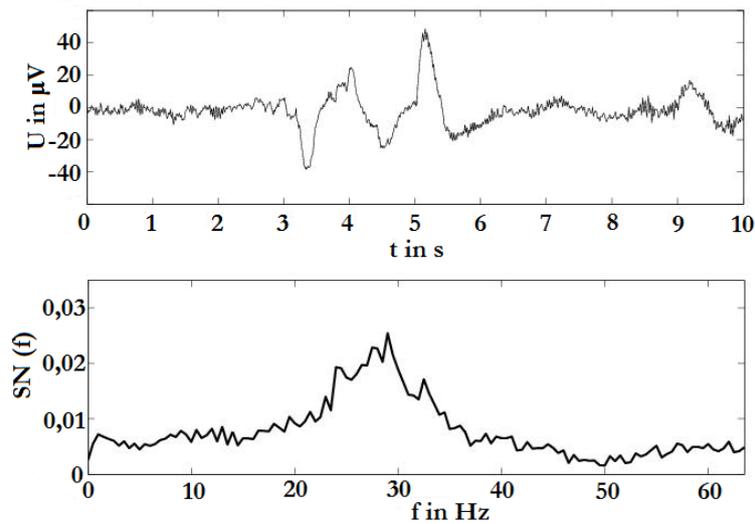


Abbildung A-5: Signalverlauf des EEG-Signals und des zugehörigen Amplitudenspektrums im Zustand REM abgeleitet an den Positionen Fp1' und Fp2' [Wen05]

A3. Merkmale zur automatischen Bewertung des EEG

Die verwendeten Merkmale, wie sie in Tabelle A-4 aufgelistet und beschrieben sind, resultieren aus den frühen Arbeiten von [BSHEB97] und [BSEH97]. Diese wurden in der Arbeit von [Wen05] zur vorliegenden Form weiterentwickelt. Dort können die genauen Berechnungsformeln nachgelesen werden. Da die Relevanz dieser Merkmale bereits umfassend in [Wen05] untersucht wurde, soll dies in der vorliegenden Arbeit nicht weiter Gegenstand sein.

Merkmalsnummer	Bereich	Spektrum	Bezeichnung
1	0Hz-4Hz	normiertes FFT-Amplitudenspektrum	relative Leistung des Amplitudenspektrums
2	4Hz-8Hz		
3	8Hz-12Hz		
4	12Hz-15Hz		
5	15Hz-31Hz		
6	31Hz-63,5Hz		
7	25%		Quantilfrequenz Frequenz bei x% der akkumulierten Leistung
8	50%		
9	75%		
10	0Hz-5Hz		Frequenz der maximalen Amplitude
11	8Hz-15Hz		
12	21Hz-31Hz		Hauptkomponente nach Änderung im Spektrum in absteigender Reihenfolge
13	1		
14	2		
15	3		
16	4		
17	5		
18	6		
19	7		
20	8		
21	9		
22	10		
23	11		
24	12		
25	13		
26	14		
27	15		
28	0,03Hz-0,12Hz	normiertes Hüllkurvenspektrum 0Hz-4Hz Band	relative Leistung des akkumulierten Spektrums
29	0,03Hz-0,25Hz		Frequenz der maximalen Amplitude
30	0,05Hz-0,5Hz		maximale Amplitude
31	0,05Hz-0,5Hz		zweithöchste Amplitude
32	0,05Hz-0,5Hz		Hauptkomponente nach Änderung im Spektrum in absteigender Reihenfolge
33	1	normiertes Hüllkurvenspektrum 4Hz-8Hz Band	Hauptkomponente nach Änderung im Spektrum in absteigender Reihenfolge
34	2		
35	3		
36	4		
37	5		
38	0,1Hz-0,5Hz	normiertes Hüllkurvenspektrum 8Hz-13Hz Band	Frequenz der maximalen Amplitude
39	0,1Hz-0,5Hz		maximale Amplitude
40	1		Hauptkomponente nach Änderung im Spektrum in absteigender Reihenfolge
41	2		
42	3		
43	4		
44	5		
45	0,03Hz-0,3Hz	normiertes Hüllkurvenspektrum 13Hz-63Hz Band	relative Leistung des akkumulierten Spektrums
46	0,03Hz-0,15Hz		Frequenz der maximalen Amplitude
47	0,05Hz-0,5Hz		maximale Amplitude
48	0,05Hz-0,5Hz		zweithöchste Amplitude
49	0,05Hz-0,5Hz		Hauptkomponente nach Änderung im Spektrum in absteigender Reihenfolge
50	1	normiertes Hüllkurvenspektrum 63Hz-130Hz Band	Hauptkomponente nach Änderung im Spektrum in absteigender Reihenfolge
51	2		
52	3		
53	4		
54	5		
55	0,05Hz-0,5Hz	normiertes Hüllkurvenspektrum 130Hz-630Hz Band	Frequenz der maximalen Amplitude
56	0,05Hz-0,5Hz		maximale Amplitude
57	0,05Hz-0,5Hz		zweithöchste Amplitude
58	1	normiertes Hüllkurvenspektrum 630Hz-1300Hz Band	Hauptkomponente nach Änderung im Spektrum in absteigender Reihenfolge
59	2		
60	3		
61	4		
62	5		

Tabelle A-4: Auflistung der verwendeten Merkmale zur Schlafklassifikation [Wen05]

Anhang B

Algorithmen

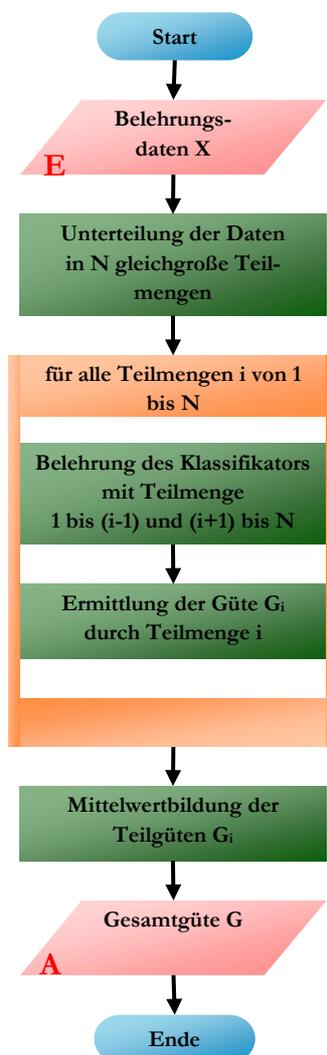


Abbildung B-1: Programmablaufdiagramm der einfachen Kreuzvalidierung

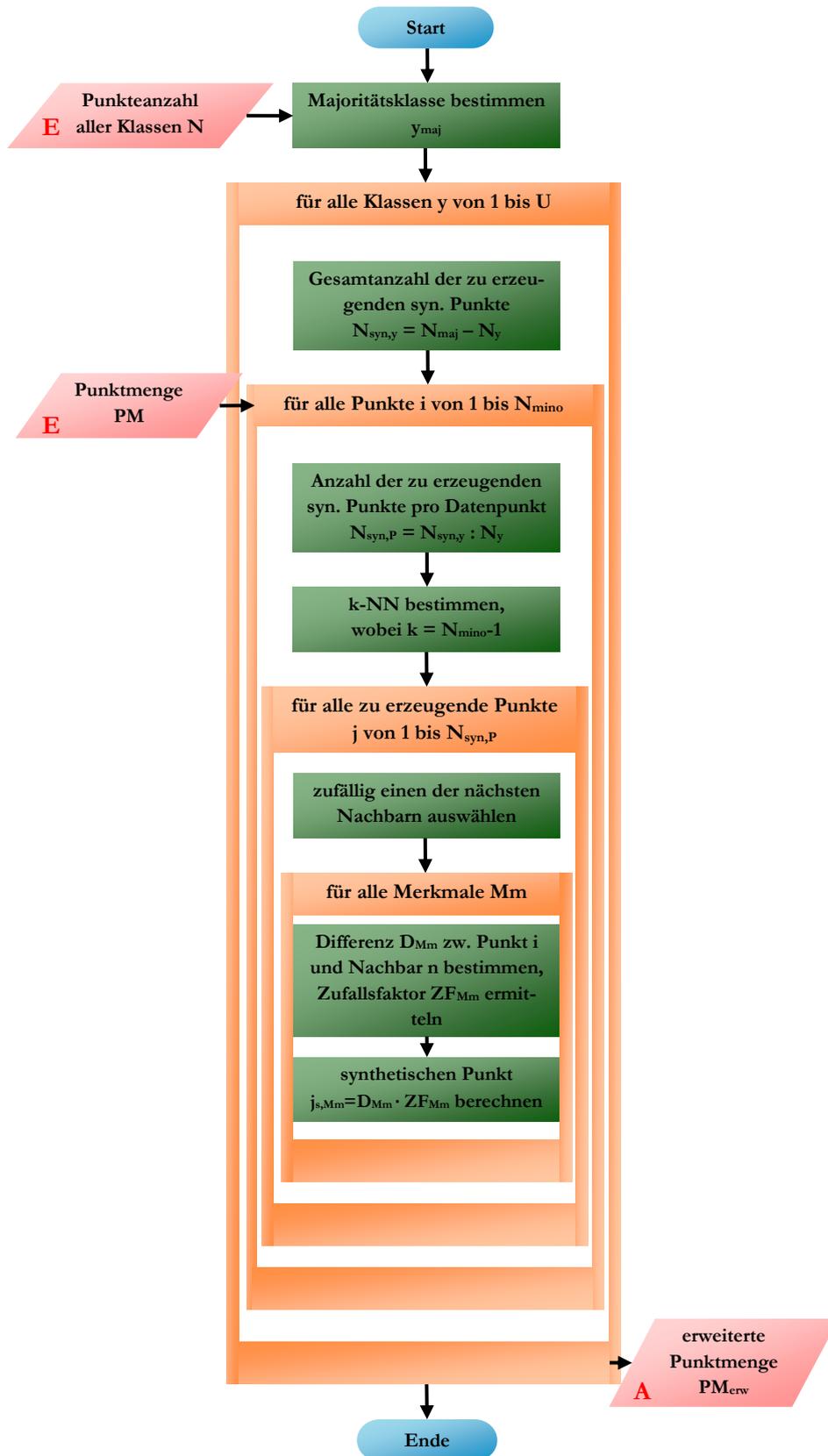


Abbildung B-2: Programmablaufdiagramm des abgepassten SMOTE-Algorithmus

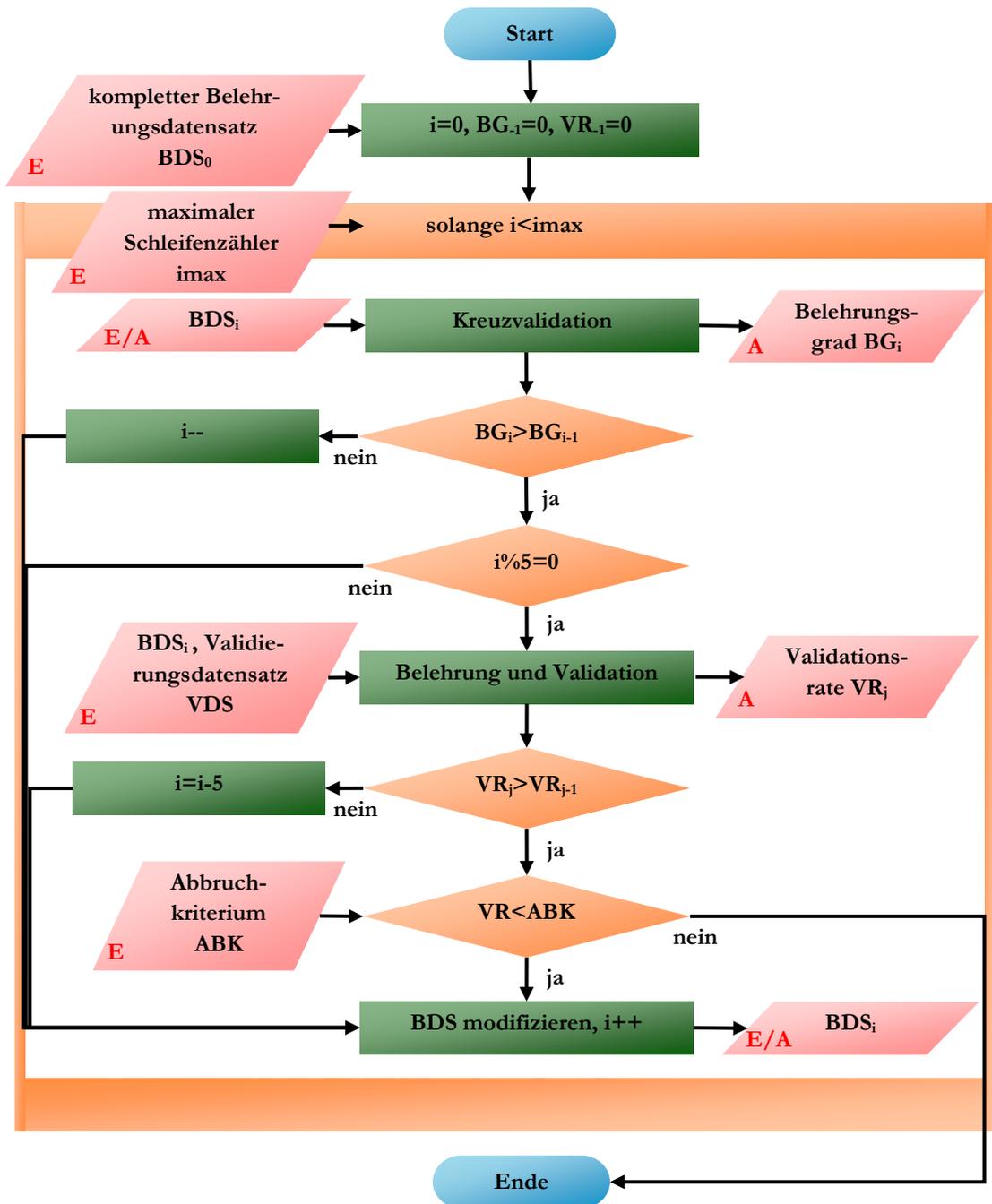


Abbildung B-3: Programmablaufdiagramm des Datenwahlalgorithmus

Anhang C

Algorithmus Parametrierungen

Daten		DiLa-Filterung	
Anz. Klassen	7	C-SVC	
Anz. Merkmale	22	Suchbereich Faktor Dichte	
Skalierung	Standardisierung	Obergrenze	1,5
SVM-Parameter		Untergrenze	0,4
		Schrittweite	0,1
SVM-Art	C-SVC	Suchbereich Faktor Abstand	
SVM-Kernel	RBF	Obergrenze	1,5
Kreuzvalidierungsteile	5	Untergrenze	0,4
C-Untergrenze	-5	Schrittweite	0,1
C-Obergrenze	15	→ 78 Kombinationen	
γ -Untergrenze	-15		
γ -Obergrenze	3		
Auflösung	0,5		

Tabelle C-1: Eingestellte Parameter zur Untersuchung der Datenbearbeitung

Daten		Kreuzvalidierung	
Anz. Klassen	7	C-Untergrenze	-5
Anz. Merkmale	62	C-Obergrenze	15
Skalierung	Standardisierung	γ -Untergrenze	-15
SVM-Parameter		γ -Obergrenze	3
SVM-Art	C-SVC	Auflösung	0,5
SVM-Kernel	RBF	Kreuzvalidierungsteile	5

Tabelle C-2: Eingestellte Parameter zur Untersuchung der Datenhomogenität

Daten		Kreuzvalidierung	
Anz. Klassen	7	C-Untergrenze	-5
Anz. Merkmale	62	C-Obergrenze	15
Skalierung	Min-Max	γ -Untergrenze	-15
SVM Parameter		γ -Obergrenze	3
SVM-Art	C-SVC	Auflösung	0,5
SVM-Kernel	RBF	Kreuzvalidierungsteile	5

Tabelle C-3: Eingestellte Parameter zur Untersuchung der Skalierungsstrategie

Daten		Kreuzvalidierung	
Anz. Klassen	7	C-Untergrenze	-5
Anz. Merkmale	62	C-Obergrenze	15
Skalierung		γ -Untergrenze	-15
SVM Parameter		γ -Obergrenze	3
SVM-Art	C-SVC	Auflösung	0,5
SVM-Kernel	RBF	Kreuzvalidierungsteile	5

Tabelle C-4: Eingestellte Parameter zur Untersuchung der Skalierungsmethode

Daten		Kreuzvalidierung	
Anz. Klassen	7	C-Untergrenze	-5
Anz. Merkmale		C-Obergrenze	15
Skalierung	Standardisierung	γ -Untergrenze	-15
SVM Parameter		γ -Obergrenze	3
SVM-Art	C-SVC	Auflösung	0,5
SVM-Kernel	RBF	Kreuzvalidierungsteile	5

Tabelle C-5: Eingestellte Parameter zur Untersuchung des Merkmalsvektors

Für die Vergleichsstudie der Rastersuche (RS) mit dem genetischen Algorithmen (GA) lief der RS-Algorithmus mit folgenden Parametern:

SVM		Suchbereiche	
SVM-Art	C-SVC	C-Untergrenze	-5
SVM-Kernel	RBF	C-Obergrenze	15
Kreuzvalidierungsteile	5	γ -Untergrenze	-15
		γ -Obergrenze	3
Auflösung	1	(1er Schritte)	
→ 399 Lösungen			

Tabelle C-6: Eingestellte Parameter der Rastersuche für die Vergleichsstudie zu den genetischen Algorithmen

Für das klassische Zwei-Klassen-XOR-Problem ergab die Rastersuche eine Klassifikationsleistung von 96,875% unter anderem mit diesen SVM-Parametern:

SVM-Art	C-SVC
SVM-Kernel	RBF
Kreuzvalidierungsteile	5
C	8192
γ	0,125

Tabelle C-7: Resultierende SVM-Parameter aus der Rastersuche für das klassische Zwei-Klassen-XOR-Problem

Die genetischen Algorithmen liefen zum Vergleich 1 mit der gleichen Auflösung, im gleichen Suchbereich und der gleichen Anzahl an probierten SVM-Parameterkombinationen wie die Rastersuche:

Daten		genetische Parameter	
Anz. Klassen	8	Anz. Individuen pro Generation	31
Anz. Merkmale	53	Anteil der Nachkommen einer Population	35%
Skalierung	Min-Max	Mutationsrate	40%
SVM-Parameter		Mutationsweite	500
SVM-Art	C-SVC	Immigrationsrate (SIGA/NSGA-II)	5%/0%
SVM-Kernel	RBF	eingesetzte Kriterien	123457
C-Untergrenze	-5	Gewichtungsmatrix der VM	-
C-Obergrenze	15	Kriteriengewichtung	52111
γ-Untergrenze	-15	Abbruchkriterium	-
γ-Obergrenze	3	Anz. der Gesamtgenerationen (SIGA/NSGA-II)	16/17
Auflösung	1	Anz. der Gesamtindividuen	399
Kreuzvalidierungsteile	5		

Tabelle C-8: Eingestellte Parameter der genetischen Algorithmen (SIGA/NSGA-II) bei Vergleich 1 der gleichen Auflösung und der gleichen Zeit mit der Rastersuche

Von den 31 Individuen pro Generation waren 10 Nachkommen aus Kreuzung (35%), 12 aus Mutation (40%) und 1 Individuum durch Immigration (5%) beim SIGA bzw. keines (0%) beim NSGA-II. Für eine konstante Anzahl an auszuprobierenden Lösungen von 399 ist das Abbruchkriterium ausgeschaltet und die Generationenanzahl fest bei 15 für den SIGA bzw. 16 für den NSGA-II, da hier ein Individuum weniger pro Generation erzeugt wird. Es werden die Kriterien:

- GG mit 50% Gewichtung,
- DG mit 20% Gewichtung,
- GMG mit 10% Gewichtung,
- MG mit 10% Gewichtung und
- die Vertauschungsmatrix mit 10% Gewichtung verwendet.

Auf eine zusätzliche Gewichtung der Vertauschungsmatrix mittels einer Gewichtungsmatrix wurde verzichtet.

Zum Vergleich 2 mit der gleichen Anzahl an probierten SVM-Parameterkombinationen liefen die genetischen Algorithmen mit folgenden Einstellungen:

Daten		genetische Parameter	
Anz. Klassen	8	Anz. Individuen pro Generation	31
Anz. Merkmale	53	Anteil der Nachkommen einer Population	35%
Skalierung	Min-Max	Mutationsrate	40%
SVM-Parameter		Mutationsweite	500
SVM-Art	C-SVC	Immigrationsrate (SIGA/NSGA-II)	5%/0%
SVM-Kernel	RBF	eingesetzte Kriterien	123457
C-Untergrenze	-5	Gewichtungsmatrix der VM	-
C-Obergrenze	15	Kriteriengewichtung	52111
γ -Untergrenze	-15	Abbruchkriterium	-
γ -Obergrenze	3	Anz. der Gesamtgenerationen (SIGA/NSGA-II)	16/17
Auflösung	100	Anz. der Gesamtindividuen	399
Kreuzvalidierungsteile	5		

Tabelle C-9: Eingestellte Parameter der genetischen Algorithmen (SIGA/NSGA-II) bei Vergleich 2 der gleichen Zeit mit der Rastersuche

Der Vergleich 3 mit der gleichen Auflösung wies diese genetischen Parameter auf:

Daten		genetische Parameter	
Anz. Klassen	8	Anz. Individuen pro Generation	31
Anz. Merkmale	53	Anteil der Nachkommen einer Population	35%
Skalierung	Min-Max	Mutationsrate	40%
SVM-Parameter		Mutationsweite	500
SVM-Art	C-SVC	Immigrationsrate (SIGA/NSGA-II)	5%/0%
SVM-Kernel	RBF	eingesetzte Kriterien	123457
C-Untergrenze	-5	Gewichtungsmatrix der VM	-
C-Obergrenze	15	Kriteriengewichtung	52111
γ -Untergrenze	-15	Abbruchkriterium	>52,75%
γ -Obergrenze	3	Anz. der Gesamtgenerationen	max. 500
Auflösung	1	Anz. der Gesamtindividuen	max. 11531
Kreuzvalidierungsteile	5		

Tabelle C-10: Eingestellte Parameter der genetischen Algorithmen (SIGA/NSGA-II) bei Vergleich 3 der gleichen Auflösung mit der Rastersuche

Daten		Kreuzvalidierung	
Anz. Klassen	7	C-Untergrenze	-5
Anz. Merkmale	62	C-Obergrenze	15
Skalierung	Standardisierung	γ -Untergrenze	-15
SVM-Parameter		γ -Obergrenze	3
SVM-Art	C-SVC	Auflösung	0,5
SVM-Kernel	-	Kreuzvalidierungsteile	5

Tabelle C-11: Eingestellte Parameter zur Untersuchung der Kernel-Funktion

Daten		Kreuzvalidierung	
Anz. Klassen	7	C-Untergrenze	-5
Anz. Merkmale	62	C-Obergrenze	15
Skalierung	Standardisierung	γ -Untergrenze	-15
SVM-Parameter		γ -Obergrenze	3
SVM-Art	C-SVC	Auflösung	0,5
SVM-Kernel	sigmoide	Kreuzvalidierungsteile	5

Tabelle C-12: Eingestellte Parameter zu den Untersuchungen der Entscheidungsstrategie, der Kombination mehrerer SVM und der kontextbasierten Klassifikation

Daten		Kreuzvalidierung	
10 homogene Belehrungsdaten aus B1		C-Untergrenze	-5
Anz. Klassen	7	C-Obergrenze	15
Anz. Merkmale	62	γ -Untergrenze	-15
Skalierung	Min-Max	γ -Obergrenze	3
SVM-Parameter		Auflösung	0,5
SVM-Art	C-SVC	Kreuzvalidierungsteile	5
SVM-Kernel	RBF		
Entscheidungsstrategie	Max-Wins		
kontextbasiert	-		
Bagging	-		

Tabelle C-13: Eingestellte Parameter zur Gesamtuntersuchung für die ursprüngliche Modellbildung

Daten		Kreuzvalidierung	
10 inhomogene Belehrungsdaten aus B1		C-Untergrenze	-5
Anz. Klassen	7	C-Obergrenze	15
Anz. Merkmale	62	γ -Untergrenze	-15
Skalierung	Standardisierung	γ -Obergrenze	3
SVM-Parameter		Auflösung	0,5
SVM-Art	C-SVC	Kreuzvalidierungsteile	5
SVM-Kernel	sigmoid		
Entscheidungsstrategie	Max-Wins		
kontextbasiert	-		
Bagging	-		

Tabelle C-14: Eingestellte Parameter zur Gesamtuntersuchung Schritt 1 (inhomogene Daten, Standardisierung und sigmoider Kernel)

Daten		DiLa-Filterung		genetische Parameter	
10 inhomogene Belehrungsdaten aus B1		C-SVC		Anz. Individuen pro Generation	40
Anz. Klassen	7	Suchbereich Faktor Dichte		Anteil der Nachkommen einer Population	35%
Anz. Merkmale	62	Obergrenze	1,5	Mutationsrate	40%
Skalierung	Standardisierung	Untergrenze	0,4	Mutationsweite	500
SVM-Parameter		Schrittweite	0,1	Immigrationsrate	5%
SVM-Art	C-SVC	Suchbereich Faktor Abstand		eingesetzte Kriterien	1234567
SVM-Kernel	sigmoid	Obergrenze	1,5	Gewichtungsmatrix der VM	ja
Entscheidungsstrategie	Max-Wins	Untergrenze	0,4	Kriteriengewichtung	52111
Kreuzvalidierungsteile	5	Schrittweite	0,1	Abbruchkriterium	>90%
kontextbasiert	-	→ 78 Kombinationen		Anz. der Gesamtgenerationen	max. 50
Bagging	-	GV	SMOTE	Anz. der Gesamtindividuen	max. 2000

Tabelle C-15: Eingestellte Parameter zur Gesamtuntersuchung Schritt 2 (DiLa-Filterung, Gleichverteilung der Daten und SIGA-Parameteroptimierung)

Daten		DiLa-Filterung		genetische Parameter	
10 inhomogene Belehrungsdaten aus B1		C-SVC		Anz. Individuen pro Generation	40
Anz. Klassen	7	Suchbereich Faktor Dichte		Anteil der Nachkommen einer Population	35%
Anz. Merkmale	62	Obergrenze	1,5	Mutationsrate	40%
Skalierung	Standardisierung	Untergrenze	0,4	Mutationsweite	500
SVM-Parameter		Schrittweite	0,1	Immigrationsrate	5%
SVM-Art	C-SVC	Suchbereich Faktor Abstand		eingesetzte Kriterien	1234567
SVM-Kernel	sigmoid	Obergrenze	1,5	Gewichtungsmatrix der VM	ja
Entscheidungsstrategie	MaxSumLim-Wins	Untergrenze	0,4	Kriteriengewichtung	52111
Kreuzvalidierungsteile	5	Schrittweite	0,1	Abbruchkriterium	>90%
kontextbasiert	-	→ 78 Kombinationen		Anz. der Gesamtgenerationen	max. 50
Bagging	-	GV	SMOTE	Anz. der Gesamtindividuen	max. 2000

Tabelle C-16: Eingestellte Parameter zur Gesamtuntersuchung Schritt 3 (Anwendung der Entscheidungsstrategie MaxSumLim-Wins)

Daten		DiLa-Filterung		genetische Parameter	
10 inhomogene Belehrungsdaten aus B1		C-SVC		Anz. Individuen pro Generation	40
Anz. Klassen	7	Suchbereich Faktor Dichte		Anteil der Nachkommen einer Population	35%
Anz. Merkmale	62	Obergrenze	1,5	Mutationsrate	40%
Skalierung	Standardisierung	Untergrenze	0,4	Mutationsweite	500
SVM-Parameter		Schrittweite	0,1	Immigrationsrate	5%
SVM-Art	C-SVC	Suchbereich Faktor Abstand		eingesetzte Kriterien	1234567
SVM-Kernel	sigmoid	Obergrenze	1,5	Gewichtungsmatrix der VM	ja
Entscheidungsstrategie	MaxSumLimWins	Untergrenze	0,4	Kriteriengewichtung	52111
Kreuzvalidierungsteile	5	Schrittweite	0,1	Abbruchkriterium	>90%
kontextbasiert	ja	→ 78 Kombinationen		Anz. der Gesamtgenerationen	max. 50
Bagging	-	GV	SMOTE	Anz. der Gesamtindividuen	max. 2000

Tabelle C-17: Eingestellte Parameter zur Gesamtuntersuchung Schritt 4
(Anwendung der kontextbasierten Klassifikation)

Daten		DiLa-Filterung		genetische Parameter	
10 inhomogene Belehrungsdaten aus B1		C-SVC		Anz. Individuen pro Generation	40
Anz. Klassen	7	Suchbereich Faktor Dichte		Anteil der Nachkommen einer Population	35%
Anz. Merkmale	62	Obergrenze	1,5	Mutationsrate	40%
Skalierung	Standardisierung	Untergrenze	0,4	Mutationsweite	500
SVM-Parameter		Schrittweite	0,1	Immigrationsrate	5%
SVM-Art	C-SVC	Suchbereich Faktor Abstand		eingesetzte Kriterien	1234567
SVM-Kernel	sigmoid	Obergrenze	1,5	Gewichtungsmatrix der VM	ja
Entscheidungsstrategie	MaxSumLimWins	Untergrenze	0,4	Kriteriengewichtung	52111
Kreuzvalidierungsteile	5	Schrittweite	0,1	Abbruchkriterium	>90%
kontextbasiert	ja	→ 78 Kombinationen		Anz. der Gesamtgenerationen	max. 50
Bagging	log. Median	GV	SMOTE	Anz. der Gesamtindividuen	max. 2000

Tabelle C-18: Eingestellte Parameter zur Gesamtuntersuchung Schritt 5
(Anwendung des Ordinalmedians)

Literaturverzeichnis

- [AAS05] AMERICAN ACADEMY OF SLEEP MEDICINE: *ICSD-2 - International classification of sleep disorders, 2nd ed: Diagnostic and coding manual*. Westchester, Illinois: American Academy of Sleep Medicine, 2005
- [ABR64] AIZERMAN, M. A.; BRAVERMAN, E. M.; ROZONOER, L. I.: Theoretical foundations of the potential function method in pattern recognition learning. In: *Automation and Remote Control* 25 (1964), S. 821–837
- [AF14] ABOALAYON, K.; FAEZIPOUR, M.: Multi-Class SVM Based on Sleep Stage Identification Using EEG Signal. In: *Health Innovations and Point-of-Care Technologies Conference HIC*, Seattle, USA, 8.-10. Oktober 2014, S. 181-184
- [AH01] AKSOY, S; HARALICK; R. M.: Feature Normalization and Likelihood-based Similarity Measures for Image Retrieval. In: *Pattern Recognition Letters* 5 (2001), April, Nr. 22, S. 563-582
- [AKJ04] AKBANI, R.; KWEK, S.; JAKOWICZ, N.: Applying support vector machines to imbalanced datasets. In: *Proceedings of European Conference on Machine Learning* (2004), S. 39–50
- [Bau09] BAUMANN, D.: *Frame-Klassifizierung von Sprachsignalen mit Support-Vektor-Maschinen*. Semesterarbeit, Eidgenössische Technische Hochschule Zürich, 2009
- [Bäc96] BÄCK; T.: *Evolutionary Algorithms in Theory and Practice - Evolution Strategies, Evolutionary Programming, Genetic Algorithms*. New York : Oxford University Press, 1996. - ISBN 0-19-509971-0
- [BC00] BENNETT, K. P.; CAMPBELL, C.: Support vector machines: Hype or hallelujah? In: *SIGKDD Explorations* 2 (2000), Nr. 2, S. 1-13
- [BGV92] BOSER, B. E.; GUYON I. M.; VAPNIK V. N.: A training algorithm for optimal margin classifiers. In: *COLT '92 - Proceedings of the Fifth Annual Workshop on Computational Learning Theory*. New York, NY, USA : ACM Press, 1992, S. 144–152

-
- [BLB00] BORTZ, J.; LIENERT, G. A.; BOEHNKE, K.: *Verteilungsfreie Methoden in der Biostatistik*. 2. Aufl. Heidelberg, New York: Springer, 2000
- [BP12] BATUWITA, R.; PALADE, V.: Class imbalance learning methods for support vector machines In: He H.; Ma Y. (Hrsg.): *Imbalanced Learning: Foundations, Algorithms and applications*, Wiley-IEEE Press, 2012
- [Bre96] BREIMAN, L.: Bagging predictors In: *Machine Learning* 24 (1996), S. 123-140
- [BS96] BAUMGART-SCHMITT, R.: *Praktische Anwendung Neuronaler Netze*. Hamburg : LIT Verlag Hamburg, 1996
- [BSEH97] BAUMGART-SCHMITT R., EILERS R. und HERRMANN W. M.: On the Use of Neural Network Techniques to Analyze Sleep EEG Data, Second Communication: Training of Evolutionary Optimized Neural Networks on the Basis of Multiple Subjects Data and the Application of Context Rules According to Rechtschaffen and Kales. In: *Somnologie* 1 (1997), S. 171-183
- [BSHEB97] BAUMGART-SCHMITT R.; HERRMANN, W. M.; EILERS, R.; BES, F.: On the Use of Neural Network Techniques to Analyse Sleep EEG Data, First Communication: Application of Evolutionary and Genetic Algorithms to Reduce the Feature Space and to Develop Classification Rules. In: *Pharmacoelectroencephalography* 36 (1997), S. 194-210
- [BSRW⁺98] BAUMGART-SCHMITT, R.; RÖTHIG, A.; WENZEL, A.; GÖBEL, R.; HERRMANN, W.M.: How Populations of Neural Networks Cooperate in the Sleep Analyzer QUISI. In: *Proceedings of the 6th EUFIT*. Aachen : Verlag Mainz, 1998, S. 1757-1761
- [CBHK02] CHAWLA, N.; BOWYER, K.; HALL, L.; KEGELMEYER, W.: SMOTE: Synthetic Minority Oversampling Technique. In: *Journal of Artificial Intelligence Research* 16 (2002), S. 321-357
- [CCT10] COSTANTIN, G.; CASALI, D.; TODISCO M.: An SVM based Classification Method for EEG Signals. In: *Proceedings of the 14th WSEAS*, Wisconsin, USA, 2010, S. 107-109
- [CL01] CHANG, C.-C.; LIN, C.-J.: Training ν -support vector classifiers: Theory and algorithms. In: *Neural Computation* 13 (2001), S. 2119-2147
- [CL08] CHANG, C.-C.; LIN, C.-J.: *LIBSVM: a Library for Support Vector Machines*. (2008), unter: <http://www.csie.ntu.edu.tw/~cjlin/papers/libsvm.pdf> (abgerufen am 18.06.2016)
- [Cov65] COVER, T. M.: Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. In: *IEEE Transactions on Electronic Computers* 14 (1965), Nr. 3, S. 326–334
- [CS00] CHRISTIANINI, N.; SHAWE-TAYLOR, J.: *An introduction to support vector machines and other kernel-based learning methods.*, Cambridge University Press, 2000

-
- [CS01] CRAMMER, K.; SINGER Y.: On the Algorithmic Implementation of Multiclass Kernel-based Vector Machines. In: *Journal of Machine Learning Research* 2 (2001), S. 265-292
- [CV95] CORTES, C.; VAPNIK, V.: Support-vector networks. In: *Machine Learning* 20 (1995), Nr. 3, S. 273–297
- [Dat16] DATACOM BUCHVERLAG GMBH: Wissens-Portal ITwissen.info (2016), unter: <http://www.itwissen.info/definition/lexikon/computer-on-module-COM-COM-Module.html> (abgerufen am 18.06.2016)
- [DB95] DIETTERICH, T. G.; BAKIRI, G.: Solving Multiclass Learning Problems via Error-Correcing output Codes. In: *Journal of Artificial Intelligence Research* 2 (1995), S. 263-286
- [DPAM00] DEB, K.; PRATAP, A.; ARGARWAL, S.; MEYARIVAN, T.: A Fast and Elitist Multi-Objective Genetic Algorithm: NSGA-II. In: *Proceedings Parallel Problem Solving From Nature VI Conference* (2000), S. 849-858
- [ED11] EBNER, A.; DEUSCHL, G.: *EEG*. Stuttgart: Georg Thieme Verlag KG, 2011
- [EED15] EEDUCATION NET E.K.: FH-Studiengang.de (2015) unter: <http://www.fh-studiengang.de/fachwissen/mathe/lagrangemethode.html> (abgerufen am 18.06.2016)
- [FCL05] FAN, R.-E.; CHEN P.-H.; LIN, C.-J.: Working Set Selection Using Second Order Information for Training Support Vector Machines. In: *Journal of Machine Learning Research* 6 (2005), S. 1889-1918
- [Fis36] FISHER, R. A.: The use of multiple measurements in taxonomic problems. In: *Annals of Eugenics* 7 (1936), S. 111–132
- [Fis07] FISCHER, J.: Support Vecor Machines - Seminar "Statistische Lerntheorie und ihre Anwendung". (2007), Univerität Ulm, unter: http://www.mathematik.uni-ulm.de/stochastik/lehre/ss07/seminar_sl/ausarbeitung_fischer.pdf (abgerufen am 18.06.2016)
- [FM01] FUNG, G.; MANGASARIAN, O. L.: Proximal Support Vector Machine Classifiers. In: *7th ACM SIGKDD international conference on Knowledge discovery and data mining*, San Francisco, 2001, S. 77-86
- [FP02] FORSYTH, D. A.; PONCE, J.: *Computer Vision: A Modern Approach*, Prentice Hall, 2002. - ISBN: 0130851981
- [GB01] GRAF, A.; BORER, S.: Normalization in Support Vector Machines. In: *Pattern Recognition* 2191 (2001), S. 277-282
- [Gn01] GNÖRICH, B.: Höhere Mathematik - Formelsammlung. (2001), unter: <http://www.gnoerich.de/formelsammlung/pdf/fs19082001.pdf> (abgerufen am 18.06.2016)

-
- [Gol16] GOLEM MEDIA GMBH: golem.de (2016), unter: <http://www.golem.de/news/deep-learning-maschinen-die-wie-menschen-lernen-1510-116468-7.html> (abgerufen am 18.06.2016)
- [Gol89] GOLDBERG, D. E.: *Genetic Algorithms in Search Optimization and Machine Learning*. Addison-Wesley Publishing Company Inc., 1989. - ISBN 0-201-15767-5
- [HCL10] HSU, C.-W.; CHANG, C.-C.; LIN C.-J.: *A Practical Guide to Support Vector Classification*. (2010), unter: <http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf> (abgerufen am 18.06.2016)
- [HL02] HSU, C.-W.; LIN, C.-J.: A Comparison of Methods for Multiclass. In: *IEEE Transactions on neural networks* 13 (2002), Nr. 2, S. 415-425
- [Jap00] JAPKOWICZ, N.: The Class Imbalance Problem: Significance and Strategies. In: *Proceedings of the 2000 International Conference on Artificial Intelligence: Special Track on Inductive Learning*, Las Vegas, Nevada, 2000
- [Joa99] JOACHIMS, T.: Transductive inference for text classification using support vector machines. In: *International Conference on Machine Learning (ICML)* (1999), S. 200-209
- [KD12] KOLEY, B.; DEY, D.: An ensemble system for automatic sleep stage classification using single channel EEG signal. In: *Computers in Biology and Medicine* 42 (2012), Dezember, Nr. 12, S. 1186-1195. - ISSN: 0010-4825
- [KS15] KHAN, A.; SYED, N. A.: Image Processing Techniques for Detection of Tumor in Human Brain Using SVM. In: *International Journal of Advanced Research in Computer and Communication Engineering IJARCCCE* 4, April 2015, Nr. 4, S. 541-544. - ISSN 2319-5940
- [KTBSW⁺10] KRAUTWALD, M.; TROMMER, D.; BAUMGART-SCHMITT, R.; WALTHER, C.; MENZ, C.: Parameter optimization for Support Vector Machines by using a multicriteria genetic algorithm for classification of sleep-stages. In: *55th IWK TU Ilmenau*, Ilmenau, Germany, 2010, S. 527-532. - ISSN 978-3-938843-53-6
- [LLGC15] LOKANNAVAR, S.; LAHANE, P.; GANGURDE, A.; CHIDRE, P.: Emotion Recognition Using EEG Signals. In: *International Journal of Advanced Research in Computer and Communication Engineering IJARCCCE* 5 (2015), Nr. 4, S. 54-56. - ISSN 2319-5940
- [LLW04] LEE, Y.; LIN, Y.; WAHBA, G.: Multicategory Support Vector Machines: Theory and Application to the Classification of Microarray Data and Satellite Radiance Data. In: *Journal of the American Statistical Association* (2004), S. 67-81
- [LR15] LOPES, N.; RIBEIRO, B.: *Support Vector Machines (SVMs) Machine Learning for Adaptive Many-Core Machines - A Practical Approach.*, Springer, 2015, Bd. 7, S. 85-105

-
- [MCE15] MONTALVO, A. J.; CASTILLO, J.; ELIAS, D.: EEG Signals Processing Based on Fractal Dimension Features and Classified by Neural Network and Support Vector Machine in Motor Imagery for a BCI. In: *IFMBE Proceedings*, Springer, 2015, Nr. 49, S. 615-618
- [MCS06] MILGRAM, J.; CHERIET, M; SABOURIN, R.: "One Againsts One" or "One Against All": Which One is Better for Handwriting Recognition with SVMs?, In: *Tenth International Workshop on Frontiers in Handwriting Recognition* (2006)
- [Men10] MENZ, C.: *Entwicklung einer Softwarebasis zur Echtzeitanalyse von EEG-Daten auf einer Microcontrollerplattform.*, Forschungsgruppe Neuroinformatik, Fachhochschule Schmalkalden : Forschungsbericht Projektarbeit, 2010
- [MG14] MA, Y.; GUO, G.: *Support Vector Machines.*, Springer, 2014
- [Mik08] MIKUT, R.: *Data Mining in der Medizin und Medizintechnik.* Karlsruhe: Universitätsverlag Karlsruhe, 2008, Nr. 22. - ISSN 978-3-86644-253-5
- [MM92] MITLER, E.; MITLER, M.: *Der Traum vom guten Schlaf.* München: Arcis-Verlag, 1992
- [Mol03a] MOLE, P.: Ockham's Razor cuts both ways - The Uses and Abuses of Simplicity in Scientific Theories. In: *Skeptic 10*, Nr. 1, S. 40 - 47
- [Mol03b] MOLZBERGER, L.: *Multiklassen-SVM.*, Vortrag in der Lehrveranstaltung Muserkennung mit Support-Vektor-Maschinen an der Albert-Ludwigs-Universität Freiburg, 7.2.2003, unter: http://lmb.informatik.uni-freiburg.de/lectures/svm_seminar/svm-multiclass.pdf (abgerufen am 10.03.2011)
- [Mon05] MONIEN, K.: *Support Vektor Maschinen als Analyseinstrument im Marketing.*, Dissertation, Universität Bielefeld, 2005
- [MTSMS15] MATEO, J.; TORRES, A. M.; SANCHEZ-MORLA, E. M.; SANTOS, J. L.: Eye Movement Artefact Suppression Using Volterra Filter for Electroencephalography Signals. In: *Journal of Medical and Biological Engineering*, Springer, 2015. - ISSN 1609-0985
- [Mül05] MÜLLER T. H.: Schlafgestört. (2005), unter: <http://www.schlafgestoert.de/site-51.html> (abgerufen am 18.06.2016)
- [NT10] NGUYEN, M. H.; DE LA TORRE F.: Optimal feature selection for support vector machines. In: *Pattern Recognition* 43 (2010), S. 584-591
- [PG90] POGGIO T.; GIROSI F.: Networks for approximation and learning. In: *Proceedings of the IEEE* 78 (1990), Nr. 9, S. 1481-1497
- [Pla99] PLATT, J. C.: Fast Training of Support Vector Machines Using Sequential Minimal Optimization. In: *Advances in Kernel Methods - Support Vector Learning*, Cambridge, USA: MIT Press, 1999. - ISBN: 0-262-19416-3

- [Rap13] RAPID-I GMBH: Predictive Analytics, Data Mining, Self-service, Open Source - Rapid Miner. (2013), unter: <http://www.rapidminer.com> (abgerufen am 18.06.2016)
- [Rec73] RECHENBERG, I.: *Evolutionsstrategie - Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*. Stuttgart : Fommann-Holzboog, 1973. - ISBN 978-3772803741
- [RK68] RECHTSCHAFFEN A.; KALES A.: *A Manual of Standardized Terminology Techniques and Scoring System for Sleep Stages of Human Subjects*. Washington DC : National Institute of Health Publications 204, US Government Printing Office, 1968
- [Ros58] ROSENBLATT, F.: The perceptron - a probabilistic model for information storage and organization in the brain. In: *Psychological Review* 65 (1958), S. 386-408
- [Sch05] SCHOLZ, P.: *Softwareentwicklung Eingebetteter Systeme: Grundlagen, Modellierung, Qualitätssicherung*. Springer Verlag, 2005
- [Sch13] SCHEER, A.-W.: *Industrie 4.0*. IMC AG, 2013
- [Sch77] SCHWEFEL, H.-P.: *Numerische Optimierung von Computer-Modellen mittels der Evolutionsstrategie: mit einer vergleichenden Einführung in die Hill-Climbing- und Zufallsstrategie*. Basel und Stuttgart : Birkhäuser Verlag, 1977. - ISBN 3-7643-0876-1
- [Sch90] SCHAPIRE, R.: The strength of weak learnability. In: *Machine Learning* 5 (1990), Nr. 2, S. 197-227
- [Sew10a] SEWELL, M.: Support Vector Machines. (2010), unter: <http://www.svms.org/> (abgerufen am 18.06.2016)
- [Sew10b] SEWELL, M.: Support Vector Machines. (2010), unter: <http://www.svms.org/history.html> (abgerufen am 18.06.2016)
- [Smi68] SMITH, F. W.: Pattern classifier design by linear programming. In: *IEEE Transactions on Computers* 17 (1968), Nr. 4, S. 367-372
- [Sri15] SRINIVASAN, M.: SV-M/D: Support Vector Machine-Singular Value Decomposition Based Face Recognition. In: *Advances in Intelligent Informatics*. (2015), Nr. 320, S. 61-76
- [SS01] SCHÖLKOPF, B.; SMOLA, A. J.: *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond.*, MIT Press, 2001. - ISBN: 0262194759
- [SS04] SMOLA, A. J.; SCHÖLKOPF B.: A tutorial on support vector regression. In: *Statistics and Computing* 14., Kluwer Academic Publishers, 2004, S. 199-222

-
- [SSA15] SUYKENS, J. A. K.; Signoretto, M.; Argyriou, A.: *Regularization, Optimization, Kernels, and Support Vector Machines*, CRC Press, 2015. –ISBN: 978-1-4822-4140-2
- [SSWB00] SCHÖLKOPF, B.; SMOLA, A.; WILLIAMSON, R. C.; BARTLETT, P. L.: New support vector algorithms. In: *Neural Computation* 12 (2000), S. 1207–1245
- [Sta11] STATSOFT: Support Vector Machines. (2011), unter: <http://www.statsoft.com/textbook/support-vector-machines/> (abgerufen am 18.06.2016)
- [Suy02] SUYKENS, J.A.K.: *Least Square support vector machines*. Singapore: World Scientific, 2002. - ISBN: 981-238151-1
- [SV99] SUYKENS, J.A.K.; VANDEWALLE, J.: Least squares support vector machine classifiers. In: *Neural Processing Letters* (1999), Nr. 3, S. 203-300
- [Tex09] TEXAS INSTRUMENTS INCORPORATED: TI Website (2009), unter: http://focus.ti.com/en/download/shared/Stellaris_Brochure.pdf (abgerufen am 18.06.2016)
- [Tex10] TEXAS INSTRUMENTS INCORPORATED: TI Website. (2010), unter: <http://focus.ti.com/lit/ug/spmu025a/spmu025a.pdf> (abgerufen am 18.06.2016)
- [Tex11] TEXAS INSTRUMENTS INCORPORATED: TI Website. (2011), unter: <http://focus.ti.com/lit/ds/spms085f/spms085f.pdf> (abgerufen am 5.08.2011)
- [TMBST⁺11] TROMMER, M.; MENZ, C.; BAUMGART-SCHMITT, R.; TROMMER, D.; WALTHER, C.: An online multiclass Support Vector Machine on a Cortex M3-platform. In: *The 6th International Conference on Electrical and Control Technologies ECT-2011*, Kaunas (Litauen): Kaunas University of Technology, 2011, S. 27-30. - ISSN: 1822-5934
- [TSW14] TROMMER, M.; SCHNEIDER, M.; WALTHER, C.: Auswirkungen von ungleichverteilten und ungenauen Belehrungsdaten auf die Klassifikation der Support Vektor Maschine. In: *Tag der Forschung 2014*, Schmalkalden : FHS-prints, 2014, S. 35-52. - ISSN 0949-1767
- [TW16] TROMMER, M.; WENZEL, A.: Automatische Identifikation und Selektion von untypischen Daten. In: *at - Automatisierungstechnik* 64 (2016), Nr. 1, S. 19-28. - ISSN (Print): 0178-2312
- [Vap95] VAPNIK, V.: *The Nature of Statistical Learning Theory.*, Springer Verlag, 1995
- [Vap98] VAPNIK, V.: *Statistical Learning Theory*. New York : Wiley, 1998. - ISBN: 0-471-03003-1
- [VC64] VAPNIK, V.; CHERVONENKIS, A.: A note on one class of perceptrons In: *Automation and Remote Control* 25 (1964)

- [VC74] VAPNIK, V.; CHERVONENKIS, A.: *Teoriya raspoznavaniya obrazov: Statisticheskie problemy obucheniya*. (Russian) [Theory of pattern recognition: Statistical problems of learning] Moscow: Nauka, 1974
- [VC79] VAPNIK, V.; CHERVONENKIS, A.; Stöckel, K.-G.: (dt. Übersetzung) *Theorie der Zeichenerkennung*. Berlin: Akademie-Verlag, 1979
- [VC82] VAPNIK, V.; CHEVONENKIS A.: *Theory of pattern recognition*. New York: Springer-Verlag, 1982
- [VGBV⁺10] VAIČIUKYNAS, E.; GELŽINIS, A.; BAČAUSKIENĖ, M.; VERIKAS, A.; VEGIENĖ, A.: Exploring kernels in SVM-based classification of larynx pathology from human voice. In: *The 5th International Conference on Electrical and Control Technologies*, Kaunas: Kaunas University of Technology, 2010, S. 67-72
- [VL63] VAPNIK, V.; LERNER, A.: Pattern recognition using generalized portrait method. In: *Automation and Remote Control* 24 (1963), S. 774–780
- [Wal12] WALTHER, C.: *Multikriteriell evolutionär optimierte Anpassung von unscharfen Modellen zur Klassifikation und Vorhersage auf der Basis hirnelektrischer Narkose-Potentiale*. Aachen: Shaker Verlag, 2012
- [Wan05] WANG, L.: *Support Vector Machines: Theory and Applications*. Berlin Heidelberg New York: Springer, 2005. - ISBN 978-3-540-24388-5
- [Wen05] WENZEL, A.: *Robuste Klassifikation von EEG-Daten durch Neuronale Netze*. Aachen: Shaker Verlag, 2005
- [WSMM⁺13] WIERING, M.A.; SCHÜTTEN, M.; MILLEA, A.; MEIJSTER, A.; SCHOMAKER L.R.B.: Deep Support Vector Machines for Regression Problems. In: *Proceedings of the International Workshop on Advances in Regularization, Optimization, Kernel Methods, and Support Vector Machines: Theory and Applications* (2013), S. 53-54
- [WW99] WESTON, J.; WATKINS, R.: Support vector machines for multi-class pattern recognition. In: *Proceedings of the 6th European Symposium on Artificial Neural Networks ESANN* (1999)
- [YX07] YE, J.; XIONG, T.: SVM versus Least Squares SVM. In: *Eleventh International Conference on Artificial Intelligence and Statistics (AISTATS)*, San Juan, Puerto Rico: Omnipress, 2007, Nr. 2., S. 644-651
- [ZT07] ZHOU, X.; TUCK, D. P.: MSVM-RFE: extensions of SVM-RFE for multiclass gene selection on DNA microarray data. In: *Bioinformatics* (2007), S. 1106-1114