

Bachelorarbeit

**Anomalie-Erkennung in
Spritzgieß-Prozessdaten**

Philipp-Jan Honysz
September 2017

Gutachter:

Prof. Dr. Katharina Morik

Dipl.-Inf. Hendrik Blom

Technische Universität Dortmund
Fakultät für Informatik
Lehrstuhl für Künstliche Intelligenz
<http://www-ai.cs.uni-dortmund.de>

Inhaltsverzeichnis

1	Einleitung	1
1.1	Zielsetzung	2
1.2	Verwandte Arbeiten	3
1.2.1	S. Haman: Prozessnahes Qualitätsmanagement beim Spritzgießen . .	3
1.2.2	J. Häußler: Eine Qualitätssicherungsstrategie für die Kunststoffverarbeitung auf der Basis künstlicher neuronaler Netzwerke	4
1.2.3	Konklusion	6
1.3	Aufbau der Arbeit	7
2	Methodik	9
3	Vorverarbeitung	15
3.1	Zeitreihe	15
3.1.1	Sequenzierung	16
3.2	Normalisierung	17
3.3	Merkmalsextraktion	18
3.4	Merkmalsselektion	19
3.4.1	Explorative Statistik	20
3.4.2	Algorithmische Verfahren	21
3.5	Repräsentationen	23
3.5.1	SAX	23
3.5.2	Phasenraumtransformation	25
4	Modellierung	27

4.1	Novelty Detection	27
4.1.1	Single Class SVM	28
4.1.2	Wahl der Trainingsdatenmenge	32
4.2	Clusteranalysen	33
4.2.1	Lernproblem	33
4.2.2	Support Vector Clustering	35
4.2.3	DBSCAN	38
4.2.4	Validierung	41
4.2.5	Interpretation	42
4.3	Hochvolumige Daten	43
4.3.1	Core Vector Machine	44
4.3.2	Ball Vector Machine	46
5	Evaluation	47
5.1	Der Spritzgießprozess	47
5.1.1	Formteilqualität	49
5.2	Datenbeschreibung	49
5.2.1	Datensatz eins: Variation der Granulatfeuchtigkeit	50
5.2.2	Datensatz zwei: Wechsel der Einspritzdüse	51
5.3	Vorverarbeitung	52
5.3.1	Ungültige Zyklen	52
5.3.2	Sequenzierung	53
5.3.3	Merkmalsextraktion	53
5.3.4	Merkmalsselektion	54
5.4	Cluster-Optimierungsproblem	57
5.4.1	Notation und Begrifflichkeit	58
5.4.2	Optimierungskriterien	59
5.4.3	Definition des Optimierungsproblems	61
5.4.4	Evolutionäre Optimierung	62
5.4.5	NSGA-II	63

5.4.6	Wahl des Individuums	64
5.4.7	Suchraum des DBSCAN-Algorithmus	65
5.4.8	Bewertung der Approximation	67
5.5	Random Cluster	69
5.5.1	Algorithmus	70
5.5.2	Gewichtung	73
5.5.3	Verknüpfung mehrerer Modelle	74
5.6	Bewertung der Vorhersagegüte	74
5.7	Anomaliebericht	75
6	Experimente	79
6.1	Testumgebung	79
6.2	Qualität der Merkmalsselektion	79
6.2.1	Aufbau	80
6.2.2	Komponentengruppe „Schnecke“	81
6.2.3	Komponentengruppe „Werkzeug“	83
6.2.4	Komponentengruppe „Heizung“	85
6.2.5	Zwischenfazit	87
6.3	Qualität der Approximation an die Paretofront	87
6.3.1	Aufbau	87
6.3.2	Komponentengruppe „Schnecke“	89
6.3.3	Komponentengruppe „Werkzeug“	92
6.3.4	Komponentengruppe „Heizung“	95
6.3.5	Zwischenfazit	97
6.4	Random Cluster	98
6.4.1	Düsensatensatz	99
6.4.2	Feuchtigkeitsdatensatz	101
6.4.3	Strukturierung des Abstimmungsverhaltens	102
7	Fazit und Ausblick	105

Abbildungsverzeichnis	110
Algorithmenverzeichnis	111
Tabellenverzeichnis	113
Literaturverzeichnis	115
Erklärung	119

Kapitel 1

Einleitung

Produzierende Betriebe jeglicher Größe sind daran interessiert, Produkte in großen Mengen bei gleichbleibend guter Qualität mit möglichst geringem Ausschuss zu produzieren. Selbst minimale Störungen, die von außerhalb (als Umwelteinfluss) oder innerhalb (als Teil der produzierenden Maschine) auf das Produktionssystem wirken, können sich untereinander verstärken und beeinflussen, und sogar von erfahrenen Maschinenbedienern möglicherweise nicht identifiziert werden. Diese Störeinflüsse können sich nachhaltig in der Produktqualität manifestieren und stellen damit ein Hindernis in der Erreichung einer reproduzierbaren und qualitätsstabilen Produktion dar. Hinzukommt, dass Inline-Qualitätsmessungen, also Qualitätsmessungen während oder zum Abschluss des laufenden Produktionsprozesses, häufig kosten- oder prozessbedingt nicht möglich sind.

Eine computergestützte Produktionsanlage könnte Abweichungen im Produktionsprozess von einem definierten *Normalzustand* noch während des laufenden Prozesses als so genannte *Anomalie* registrieren und dem Bediener rechtzeitig das Auftreten eines Produktionsfehlers signalisieren.

Ein erster Schritt in Richtung dieser realzeitlichen Anomalieerkennung soll im Rahmen dieser Bachelorarbeit gegangen und eine Methode entwickelt werden, die auf einem statischen Datensatz, der aus Spritzgussprozessen, einem Urformverfahren aus der Kunststofftechnik, gewonnen wurde, Anomalien detektiert und den Maschinenbediener darüber möglichst ausführlich, mit Verweis auf beteiligte Maschinenkomponenten und Prozessphasen, informiert. Dies soll dem erfahrenen Bediener die Möglichkeit geben, möglichst schnell das zugrundeliegende Problem zu identifizieren und zu beheben. So soll ein langfristig, resilienter Produktionsprozess ermöglicht werden, bei dem der Produktionsanlage beziehungsweise dem Bediener ermöglicht wird, aus einem anomalen Zustand in einen normalen Zustand zurückzukehren [13], was wiederum zu Produkten gleichbleibender Qualität führen soll.

1.1 Zielsetzung

Ziel dieser Arbeit ist die Entwicklung eines Offline-Systems zur Anomalieerkennung auf Basis des Spritzguss-Produktionsprozess: Hierfür soll eine Pipeline definiert werden, die die Rohdaten der Maschine etappenweise verarbeitet, um anschließend eine detaillierte Anomalieentscheidung zu berechnen.

Die Anomalieerkennung soll dem erfahrenen Bediener der Maschine dabei bei seiner Arbeit assistieren: Hierzu soll eine entdeckte Anomalie des Produktionsprozesses möglichst detailliert und nachvollziehbar, das heißt mit Verweis auf beteiligte Maschinenkomponenten, Prozessphasen und den Gründen für die Anomalieentscheidung, präsentiert werden. Diese Informationen können vom Bediener dann dazu genutzt werden, um die Situation einzuschätzen und gegebenenfalls die Ursache der Anomalie zu beheben. In der Arbeit muss daher besprochen werden, wie sich ein solcher *Anomaliebericht* realisieren lässt. Insbesondere muss geklärt werden, wie sich eine Anomalie im Spritzgussprozess ausdrückt und durch welche Einflussgrößen sich diese identifizierbar macht. Dies ist notwendig, um die Anwendung von maschinellen Lernmethoden möglichst effektiv zu gestalten. Dabei ist es vom besonderen Interesse, auch unabhängig vom Prozesswissen eines Ingenieurs oder Bedieners, Zusammenhänge zu entdecken, die zu einer Anomalie führen.

Produktionsanlagen markieren anomale Produktionsprozesse dabei nicht, was ein Problem darstellt. Denn damit entfällt die Möglichkeit der Nutzung einer großen Gruppe von Algorithmen innerhalb des maschinellen Lernens, nämlich jene des *überwachten Lernens*, die am meisten untersucht und angewendet werden [40]. Diese setzen das Vorhandensein von Hintergrundinformationen, wie beispielsweise die einfache Information, ob ein Produktionsprozess normal (*negativ*) oder anomal (*positiv*) war, voraus. Diese Informationen werden häufig *Label* genannt. Ziel der Arbeit ist festzustellen, wie ausgewählte maschinelle Lernverfahren aus dem Bereich des *unüberwachten Lernens*, wie beispielsweise das *Clustering* oder die *Novelty Detection*, dazu eingesetzt werden können, um anomale Produktionsprozesse, auch ohne vorhandene Label, zu identifizieren.

Der Fokus der Arbeit liegt dabei auf der Entwicklung eines Systems, das auf einem Stapel aufgezeichneter Daten, dem so genannten *Batch*, operiert. Dabei ist vorgesehen, dass bei Ankunft neuer Daten, das bereits erlernte Modell vollständig verworfen wird und ein neues Modell zur Bewertung des neuen Produktionsprozess erlernt wird. Während dies einen hohen rechnerischen Aufwand nach sich zieht und in der Praxis womöglich wenig praktikabel ist, bringt diese Herangehensweise dennoch einige Vorteile mit sich. So ist es dadurch zum einen möglich, die prinzipielle Eignung maschineller Lernverfahren zur Lösung des vorgestellten Problems zu evaluieren. Zum anderen wird dadurch die Erkennung von so genannten *Drift*-Situationen ausgeklammert, in denen sich die Verteilung der Daten *abrupt* oder *inkrementell*, beispielsweise aufgrund des Austausches eines Sensors oder als

Resultat von Abnutzungserscheinungen ändert [17]. Dabei sollen nicht alle vorgestellten Methoden letztlich evaluiert werden, sondern vielmehr aufgezeigt werden, welche grundsätzlichen Schritte zur Anwendung von maschinellen Lernverfahren notwendig sind oder situationsabhängig notwendig sein können und wie beziehungsweise welche Methoden aus den oben genannten Gruppen bei einer Anomalieerkennung im industriellen Umfeld genutzt werden können.

1.2 Verwandte Arbeiten

In diesem Abschnitt sollen einige, bereits verfolgte Ansätze zur Qualitätssicherung im Bereich des Spritzgusses, unter Verwendung von maschinellen Lernmethoden, zusammengefasst werden. Der Fokus dieser Arbeiten liegt also auf der *Qualität der Produkte* und weniger auf der *Entdeckung von Anomalien im Produktionsprozess*. Nichtsdestotrotz sind diese Beiträge interessant, da die Produktqualität das Resultat des Produktionsprozesses ist und daher damit zusammenhängt.

Für diese Arbeit ist insbesondere interessant, wie die Autoren in ihren Arbeiten die, für das Feststellen der Qualität relevanten, Merkmale selektiert haben. Letztlich ist auch interessant, wo die Autoren zur Prognose der Qualitätsmerkmale des Endprodukts beziehungsweise der Anomalie im Produktionsprozess ansetzen, um zu verstehen, wie beziehungsweise ob das, im vergangenen Abschnitt dargestellte, Problem der direkt von der Produktionsmaschine stammenden, nicht-gelabelten Daten gelöst wird.

1.2.1 S. Haman: Prozessnahes Qualitätsmanagement beim Spritzgießen

Haman verfolgte in seiner Dissertation das Ziel, Qualitätsmerkmale des Spritzgießens zu überwachen, diese zu regeln und hierzu kontinuierlich auf die eingestellten Maschinenparameter Einfluss zu nehmen [21]. Dabei wurde ein gestufter Ansatz verfolgt, bei dem zunächst die Erfassung der Zusammenhänge zwischen Maschinenparametern und den Qualitätsmerkmalen im Vordergrund stand. Auf Basis dieser Erkenntnisse wurden Modelle entwickelt, die im Online-Betrieb, also während der laufenden Produktion, zur Prognose der Produktqualität genutzt wurden. Eine vom Modell prognostizierte Abweichung vom Sollwert wurde dann als Anlass genommen, um in die Parametrisierung der Maschine einzugreifen. Diese Qualitätsregelung soll im Folgenden nicht besprochen und stattdessen die Auswahl der Merkmale und die Modellierung beleuchtet werden.

Der Autor entwickelte ein Regressionsmodell, eine Methodik des *überwachten Lernens*, für die Vorhersage des Formteilgewichts, da diesem eine besondere Bedeutung in der industriellen Produktion beigemessen werde, und unterschied bei der Modellierung zwischen Einstell- und Überwachungsmodell. Das Einstellmodell sollte die Abhängigkeit zwischen

Maschinenparametern und der resultierenden Produktqualität modellieren und Prognosen zum Sollwert des Formteilm Gewichts erlauben. Das Überwachungsmodell hingegen, sollte den Einfluss der Prozessgrößen und Maschinenparameter auf die Produktqualität modellieren und diene dazu, das *tatsächliche* Formteilm Gewicht zu prognostizieren.

Die für die Modellierung benötigten Merkmale wurden dabei in Abhängigkeit des zu entwickelnden Modells bestimmt: So wurde beim Einstellmodell auf die Auswertung von Versuchsreihen mittels statistischer Tests, wie dem F-Test oder t-Test, zurückgegriffen, um die Maschinenparameter zu finden, die auf die Zielgröße, in diesem Falle das Formteilm Gewicht, Einfluss nehmen. Darauf aufbauend, wurde den Merkmalen ein Effektwert zugeordnet, der angibt, in welche Richtung und wie stark jenes Merkmal die Zielgröße, in diesem Falle das Formteilm Gewicht, beeinflusst und die Bedeutung der gefundenen Parameter mit dem dazugehörigen Prozesswissen unterstrichen. Die gefundenen Merkmale wurden dann zur Aufstellung eines linearen und eines nicht-linearen Regressionsmodells verwendet und die resultierenden Modelle bezüglich ihrer Güte verglichen. Die zum Regressionsmodell gehörigen Koeffizienten wurden dabei anhand des Effektwertes bestimmt. Im Ergebnis empfahl der Autor die Nutzung eines nicht-linearen Modells und resümierte, dass sich die Sollwerte für das Formteilm Gewicht anhand der Maschinenparameter prognostizieren lassen.

Das vorgeschlagene Überwachungsmodell hingegen benötigte Prozessgrößen, die zunächst aus den Zeitreihen extrahiert werden müssen. *Haman* benennt hierfür einige, zu untersuchende Prozessgrößen, deren linearer Zusammenhang mit der Formteilm Qualität mittels einer Korrelationsanalyse festgestellt wurde. Es wurden dann sukzessive die am stärksten korrelierten Merkmale in das (lineare) Regressionsmodell aufgenommen, solange sich die Modellqualität nicht verschlechtert hat. Als Ergebnis wird angegeben, dass sich 98% der gesamten Gewichtsstreuungen erklären lassen und sich somit das Modell für die Gewichtsprognose im Online-Betrieb eignet.

1.2.2 J. Häußler: Eine Qualitätssicherungsstrategie für die Kunststoffverarbeitung auf der Basis künstlicher neuronaler Netzwerke

Häußler entwickelte in seiner Dissertation unter anderem ein System zur Prozessüberwachung und Qualitätsprognose, welches vollständig als Online-Verfahren operiert und sich auf die Gegebenheiten der Kunststoffverarbeitung konzentriert [23]. Die für die vorliegende Arbeit interessanten Aspekte der Modellierung wurden dabei in einem abstrakten und einem konkreten Teil besprochen: Dabei wurde zunächst die Vorgehensweise und die Randbedingungen zur Aufstellung und Verwendung eines abstrakten *Prozessmodells* beziehungsweise *Prognosemodells* besprochen, welches die Zusammenhänge zwischen Prozessparametern und den jeweiligen Qualitätsmerkmalen modelliert. Im konkreten Teil wurden dann so genannte *künstliche neuronale Netzwerke* (engl. *artificial neural networks*, kurz:

ANN) vorgestellt, zur Modellbildung untersucht und mit multivariaten, statistischen Modellbildungsverfahren verglichen. Dabei werden in der Arbeit von *Häußler* weitere Aspekte besprochen, wie beispielsweise die *Adaption*, also die Überprüfung des Modells bei veränderten Prozessbedingungen, einem für Online-Verfahren charakteristischem Problem (siehe Abschnitt 1.1), die hier aber vernachlässigt werden sollen, um die Modellierung näher zu betrachten. Im Folgenden soll der Fokus auf dem konkreten Prozessmodell liegen und einige von *Häußler* getätigte Überlegungen zur Strukturierung der Merkmalsmenge benannt werden. Anschließend sollen die erzielten Ergebnisse einer Auswahl der vom Autor durchgeführten Experimente genannt werden.

Der konkrete Teil der Modellierung (s. Kapitel 5, [23]) befasste sich mit der Realisierung des abstrakten Prognosemodells anhand von ANN, wofür zunächst die benötigten Merkmale näher betrachtet wurden. So unterschied *Häußler* zum einen zwischen Maschineneinstellgrößen, also den Maschinenparametern, und so genannten Prozessfolgegrößen, die von einem oder mehreren Maschinenparametern beeinflusst werden und die letztlich die Qualitätsausprägung des Produktes bestimmen. Weiterhin wurde festgestellt, dass auch Änderung eines Maschinenparameters, die mit zeitlichem Verzug auf den Produktionsprozess und exogene Einflüsse, wie beispielsweise der verwendete Rohstoff, modelliert werden müssen. Diese weitergehenden Aspekte sollen nicht weiter besprochen und stattdessen die durchgeführten Versuche vorgestellt werden.

Allen den nun präsentierten Experimenten ist gemein, dass die Datensätze, wie im Experiment benötigt, gelabelt waren und die Merkmale, bis auf eine Auswahl möglicher qualitätsrelevanter Merkmale im, oben genannten, abstrakten Teil (vgl. Kapitel 3.1.2, [23]) bereits vorselektiert waren.

Das erste Experiment umfasste die Klassifikation attributiver Merkmale, also gerade die Identifikation eines Beispiels in *gut* oder *schlecht*. Dieses Experiment wurde auf einem Datensatz mit acht Prozessgrößen durchgeführt. Um das für die Klassifikationsaufgabe notwendige Label *gut/schlecht* zu bestimmen, wurde der Datensatz anhand des Formteilgewichts und dessen Medians in zwei Klassen mit etwa der gleichen Anzahl von Beispielen aufgeteilt. Die eine Menge wurde „willkürlich“ als *gut*, die andere als *schlecht* bezeichnet. Anschließend wurde ein ANN und ein multiples Regressionsmodell anhand einer zufälligen Auswahl von 50% der gesamten Datenmenge trainiert. Nach Abschluss des Trainings konnte das ANN alle Beispiele des Trainingsdatensatzes korrekt klassifizieren, während das Regressionsmodell nur 60% der Beispiele korrekt klassifizieren konnte. Der Testdatensatz wurde vom ANN zu 75% korrekt klassifiziert, während das Regressionsmodell unverändert 60% korrekt klassifiziert. Der Klassifikationsleistung hinsichtlich des gesamten Datensatz lag beim ANN bei 87%, während diese beim Regressionsmodell bei 60% lag.

Das zweite Experiment bestand zum einen aus der Reduzierung der 127 Beispiele und eines 24 Merkmale umfassenden Beispieldatensatzes, um redundante Informationen und zum anderen aus einem Vergleich der attributiven Klassifizierung und einer Mehrklassenklassifikation des Formteilgewichts. Die Reduzierung wurde dabei so vorgenommen, dass im Beispieldatensatz jedes Merkmal mit seinem minimalen und maximalen Wert vertreten ist. Es wurde dann jeweils mit 20% des Beispieldatensatzes trainiert und mit 80% des Datensatzes getestet. Im attributiven Fall konnten, nach Abschluss des Trainings, 100% der Trainingsdaten und 80% der Testdaten korrekt klassifiziert werden. Im Mehrklassenfall wurden 100% der Trainingsdaten und 75% der Testdaten korrekt klassifiziert.

Das dritte Experiment behandelte die Vorhersage des Formteilgewichts und der Formteillänge im Hinblick auf Produktionsunterbrechungen und die Anfahrphase im Spritzgießprozess, die jeweils ein gegenläufiges Verhalten der genannten Attribute provozieren. Hierzu wurde versucht, das Gewicht und die Länge in zwei verschiedenen und in einem einzelnen ANN vorherzusagen, wobei auf die Unterteilung in Trainings- und Testdatensatz zu Gunsten des Vergleichs dieser Ansätze verzichtet wurde. Zur Messung der Modellqualität wurde das Bestimmtheitsmaß herangezogen, welches die Qualität der Anpassung an die Trainingsdaten wiedergibt [6]. Höhere Bestimmtheitsmaße sprechen für eine bessere Anpassung, niedrigere für eine schlechtere Anpassung. Bei einem ANN, das hinsichtlich des Formteilgewichts trainiert wurde, konnte so ein Bestimmtheitsmaß von 83% erreicht werden, während es bei einem linearen, multiplen Regressionsmodell 48% und bei einem quasi-nicht-linearen Modell 68% waren. Für die Formteillänge galt, dass das ANN mit gleicher Güte, wie jenes ANN des Formteilgewichts arbeitete. Für das lineare Regressionsmodell konnte ein Bestimmtheitsmaß von 75% erreicht werden, während für das quasi-nicht-lineare Modell ein Bestimmtheitsmaß von 79% erreicht werden konnte.

1.2.3 Konklusion

Es wurden zwei Verfahren vorgestellt, die sich der Methoden des maschinellen Lernens bedienen, um Aussagen hinsichtlich der Qualität des Endproduktes beim Spritzgießen im laufenden Betrieb treffen zu können. Beide vorgestellten Verfahren lernen Prognosemodelle dabei in einer überwachten Form mit Zielgrößen, die sich auf das Produkt und nicht auf den Prozess beziehen. Die für das Bilden der Vorhersagemodelle benötigten Ausprägungen der Label wurden dabei empirisch erhoben, sei es im Rahmen einer Versuchsplanung (*Haman*) oder der Annahme, dass die Betriebe die Daten im Rahmen einer *statistischen Prozesslenkung* ohnehin erheben (*Häußler*).

In Abgrenzung zu diesen Verfahren sollen in der vorliegenden Abschlussarbeit primär die nicht-gelabelten Rohdaten des Produktionsprozess untersucht und in diesen Abweichungen vom Prozess-Normalzustand entdeckt werden, die dann das Produkt beeinflussen. Es

soll also nicht die Qualität der Produkte in die Modellierung einfließen. Diese Prozessabweichungen sollen sich durch ein Modell hinreichend qualifizierbar machen und so dem erfahrenen Bediener eine rechtzeitige Abstimmung der Problemursache ermöglichen, bevor sich diese am Produkt manifestiert.

Aus Sicht der Bestimmung relevanter Prozessgrößen sind die vorgestellten Arbeiten in jedem Falle hilfreich, um bewährte Merkmale zu erhalten, die für eine prozesswissenbasierte Modellierung verwendet werden können.

1.3 Aufbau der Arbeit

Wie in der Zielsetzung beschrieben wurde, soll in der Arbeit eine Pipeline definiert werden, in der die Rohdaten etappenweise verarbeitet werden. Dieses Konzept soll sich in der Arbeit widerspiegeln. Hierzu werden in Kapitel 2 die anzuwendende Methodik und die zum Erreichen des Ziels erforderlichen Schritte abstrakt skizziert. Anschließend wird in Kapitel 3 die Vorverarbeitung der Daten besprochen, in denen die wichtigen Begriffe der *Zeitreihe* und *Beobachtungen* definiert und Vorverarbeitungsschritte eingeführt werden, die für die erfolgreiche Anwendung von maschinellen Lernverfahren notwendig sind. Die Ergebnisse aus diesem Kapitel werden dann für die Anwendung von maschinellen Lernverfahren zum Zwecke der Anomalieerkennung verwendet. Diese Verfahren werden in Kapitel 4 diskutiert und die wesentlichen Aspekte ihrer theoretischen Arbeitsweise vorgestellt. In Kapitel 5 soll dann der Spritzgussprozess vorgestellt werden. Einige der, in den vergangenen Kapiteln besprochenen, Verfahren sollen geeignet miteinander verknüpft und Experimente entwickelt werden, die die Eignung dieser zur Anomalieerkennung innerhalb des Spritzgießprozesses feststellen. In Kapitel 6 sollen diese Experimente dann mittels realer Daten aus dem Spritzgussprozess durchgeführt und die Resultate besprochen werden. Abschließend wird in Kapitel 7 ein Fazit gezogen und ein Ausblick darauf gegeben, wie die Arbeit geeignet weiterentwickelt werden könnte.

Kapitel 2

Methodik

Ausgehend von den in Abschnitt 1.1 formulierten Zielvorgaben, sollen nun die Grundzüge der auszugestaltenden Methode zur industriellen Anomalieerkennung auf Zeitreihenbasis skizziert werden. Dabei soll insbesondere auf die Voraussetzungen, die zur Anwendung dieser Methode erfüllt sein müssen, die nötigen Vorverarbeitungsschritte und die Modellierung eingegangen werden. Abschließend wird erläutert, wie die gelernten Modelle verwendet werden sollen, um einen Produktionsprozess auf das Vorhandensein von Anomalien zu prüfen und bei Bedarf den entsprechenden Anomaliebericht zu generieren. In diesem Kapitel soll dabei weniger die konkrete Realisierung der einzelnen Schritte besprochen, sondern vielmehr ihre systematische Verknüpfung und ihre Bedeutung für die Anomalieerkennung aufgezeigt werden.

Voraussetzungen Grundsätzlich wird vorausgesetzt, dass die zu untersuchenden Daten der Produktionsanlage in Form einer so genannten Zeitreihe (s. Definition 3.1.2) strukturiert sind, wobei jeder Produktionsprozess eindeutig identifizierbar ist, sodass Daten eines Prozesses von Daten anderer Prozesse getrennt werden können. Außerdem sollen in Abgrenzung zu Verfahren, die auf strömenden Daten operieren, alle Maschinendaten, die für die Anwendung der Methode verwendet werden sollen, vollständig vorliegen und ein wahlfreier Zugriff auf diese ermöglicht werden. Weitere Voraussetzungen ergeben sich durch die Anwendung bestimmter Vorverarbeitungs- und Modellierungsschritte, die hier aus Vereinfachungsgründen nicht aufgeführt, im weiteren Verlauf der Arbeit jedoch besprochen werden.

Vorverarbeitung Im Rahmen der Datenvorverarbeitung wird dann eine Menge von Zeitreihen untersucht, wobei jede Zeitreihe dieser Menge einem Produktionsprozess zugeordnet ist. Dabei wird jeder Produktionsprozess beziehungsweise jede Zeitreihe aus dieser Menge anhand ihrer Teilprozesse zerlegt, sodass jede Gesamtzeitreihe in Teilzeitreihen zer-

fällt (s. Abschnitt 3.1.1). Diese Zerlegung soll dazu führen, dass sich Anomalien, die in Teilzeitreihen entdeckt wurden, leichter auf einzelne Maschinenkomponenten und Prozessphasen zurückführen lassen und so dem Bediener eine nachvollziehbare Begründung für eine gefundene Anomalie präsentiert wird.

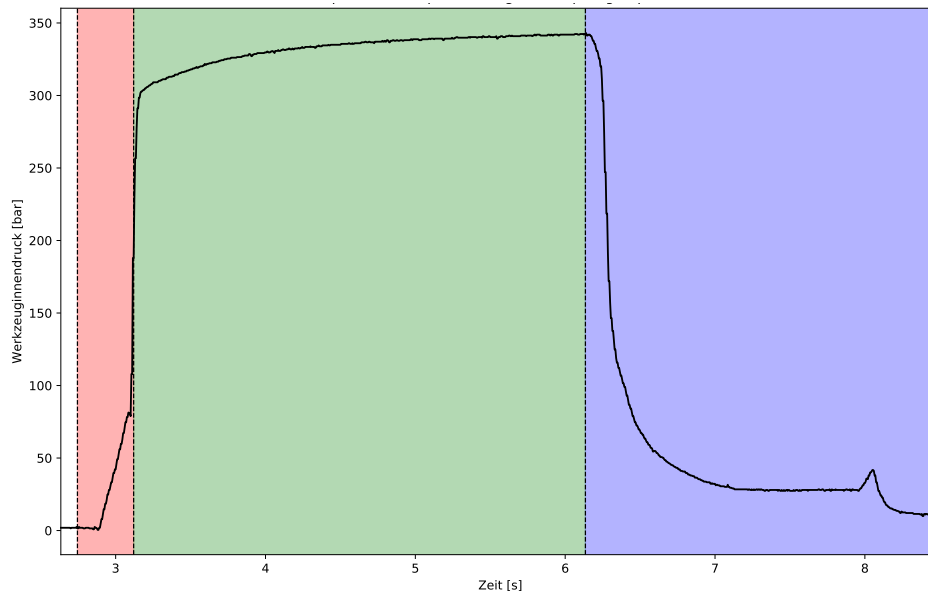


Abbildung 2.1: Ausschnitt des Werkzeuginnendruckverlaufes, einer prozessrelevanten Größe des Spritzgießprozesses, während eines ausgewählten Produktionsprozesses. Die Zeit ist relativ zum Prozessbeginn. Die Trennlinien visualisieren hierbei einen Teil der, von der Maschine gemeldeten, aktuellen Prozessphasen. Der rote Bereich zeigt die Entwicklung des Werkzeuginnendruckes während der Einspritzphase, der grüne Bereich den Werkzeuginnendruck während der Nachdruckphase und der blaue Bereich einen Ausschnitt des Werkzeuginnendruckverlaufes während der Abkühlphase.

Anschließend müssen aus den Teil- und der Gesamtzeitreihe eines Prozesses, neue und für die Anomalieerkennung relevante Merkmale gebildet werden (s. Abschnitt 3.3). Darauf aufbauend werden eine oder mehrere „gute“ Merkmalsmengen für die spätere Modellierung ausgewählt, um möglichst treffende Prognosen zu erhalten (s. Abschnitt 3.4). Dabei soll neben der Bildung (Extraktion) und Auswahl (Selektion) von Merkmalen durch Experten, beziehungsweise mittels Expertenwissen, auch statistisch-algorithmische Ansätze besprochen werden.

Die algorithmischen Ansätze verlangen dabei häufig die Spezifikation eines Labels, welches in diesem Anwendungsfall, in dem keine von der Datenquelle zur Verfügung gestellten Label verfügbar sind, ein Problem ist. Als Lösung wird daher vorgeschlagen, verschiedene Merkmalsmengen zu selektieren und die Label dabei anhand von Maschinenkomponentengruppen zu wählen. So ließe sich beispielsweise bei Betrachtung einer Spritzgießmaschine eine Merkmalsselektion anhand eines Labels durchführen, dass zweifelsfrei zur Komponenten-

tengruppe „Heizung“ gehört, während eine andere Merkmalsselektion sich auf die Gruppe „Werkzeug“ fokussiert (s. auch Abschnitt 5.1). Dadurch ergeben sich „Mengen von Merkmalsmengen“ die unabhängig voneinander zur Modellbildung verwendet werden können und die letztlich auch, aufgrund ihrer Spezialisierung auf bestimmte Maschinenkomponenten, dazu beitragen, eine detaillierte Erklärung zu einer eventuell gefundenen Anomalie zu liefern.

Da in der ursprünglichen von der Maschine gelieferten Zeitreihenrepräsentation Anomalien möglicherweise nicht erkennbar sind, werden Methoden zur Überführung von Zeitreihen in andere Repräsentationen vorgeschlagen, um so, bei Bedarf, zusätzliche Möglichkeiten zur Merkmalsextraktion zu schaffen, falls die Vorhersagequalität auf Basis der Ursprungsrepräsentation unbefriedigend ist (s. Abschnitt 3.5).

Modellierung Wurden die relevanten Merkmale aus den Zeitreihen extrahiert und selektiert, soll auf dieser Basis eine Menge von Modellen gelernt werden, aus dem sich Rückschlüsse auf den Anomaliestatus und die dabei beteiligten Komponenten ziehen lassen. Da sich, wie in der Zielsetzung beschrieben, auf unüberwachte Lernverfahren beschränkt wird, müssen verwendete Basismethoden aus der *Novelty Detection* oder des *Clusterings*, aufgrund der Interpretationsnotwendigkeit des Modellierungsergebnisses, geeignet für den vorliegenden Anwendungsfall adaptiert werden (s. Abschnitt 4.1 und 4.2).

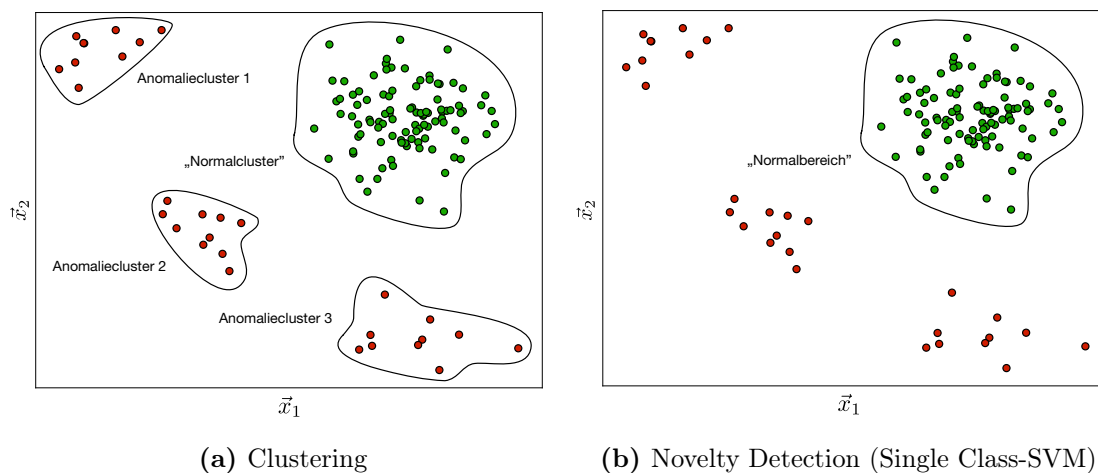


Abbildung 2.2: Exemplarische Illustration der beiden, in der Arbeit vorgestellten Lernaufgaben, zur Identifikation anomaler Produktionsprozesse. Das Clustering gruppiert Punkte anhand eines Distanzbegriffes, sodass sich kleinere Cluster, die weniger Punkte enthalten, von einem großen Cluster abheben, was sich letztlich zur Anomalieerkennung ausnutzen lässt. Die Single Class-SVM hingegen lernt aus einer Menge „guter“ Punkte eine Trennebene zwischen diesen und anomalen Punkten. Es sei darauf hingewiesen, dass eine derart klare Trennung zwischen den anomalen und normalen Punkten nicht immer möglich ist.

Als Ergebnis dieser Adaption wird vorausgesetzt, dass jedes Modell eine binäre Entscheidung hinsichtlich des Anomaliestatus abgeben kann (*Anomalie [ja/nein]*) und für jedes Modell die daran beteiligte Merkmalsmenge bestimmt werden kann. Im Rahmen einer üblichen Modellselektion, soll unter allen möglichen Modellen das für den Anwendungsfall „beste“ Modell anhand gängiger Methoden wie der (evolutionären) Parameteroptimierung und, falls möglich, der so genannten Kreuzvalidierung, bestimmt werden und anschließend in die Bewertung eingehen (s. Abschnitt 5.4 und 5.3.4).

Bewertung Wurde ein Produktionsprozess als anomal bewertet, soll diese Anomalie anschließend bewertet werden. Dabei geht es insbesondere darum, die Anomalie entsprechend zu erklären, wichtige Einflussgrößen zu identifizieren und die Informationen in einem Anomaliebericht für den Bediener der Maschine verständlich aufzubereiten. Während mit der oben beschriebenen Sequenzierung der Zeitreihen und der Fokussierung der Merkmalsselektion auf bestimmte Maschinenkomponenten bereits die Voraussetzungen für eine Anomalieerklärung geschaffen wurden, muss jetzt geklärt werden, welche Merkmale einen besonders hohen Einfluss auf die Anomalieentscheidung genommen haben. Dabei ist es hilfreich, dass einige Lernverfahren, wie beispielsweise *Support Vector Machines* zu der auch die *Single Class-SVM* aus der Novelty Detection gehört, die Wichtigkeit eines Merkmals als Ergebnis der Modellierung enthalten (s. Abbildung 2.3).

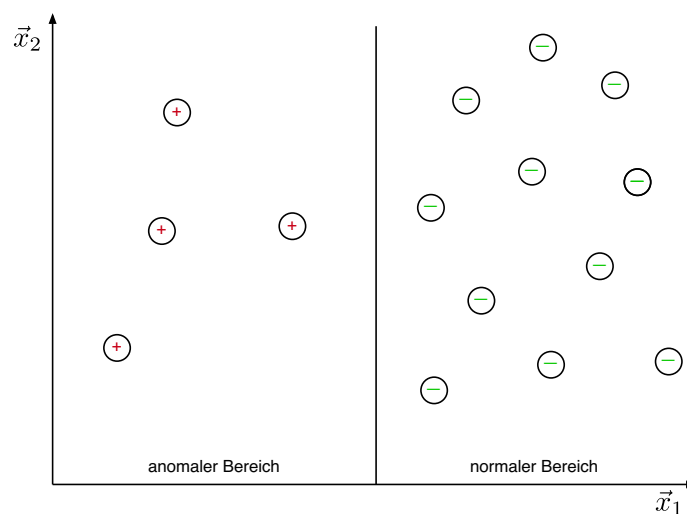


Abbildung 2.3: Exemplarische Illustration einer gelernten Trennebene zwischen normalen und anomalen Punkten im einfachen, zweidimensionalen Fall. Das Merkmal \vec{x}_2 nimmt hier keinen Einfluss auf die Prognose.

Clusterverfahren hingegen liefern üblicherweise keine Informationen zur Wichtigkeit von Merkmalen (s. Definition 4.2.4), sodass diese Informationen auf andere Weise inferiert werden müssen. Dabei ist es beispielsweise möglich, mehrere Clusterungen auf Basis verschie-

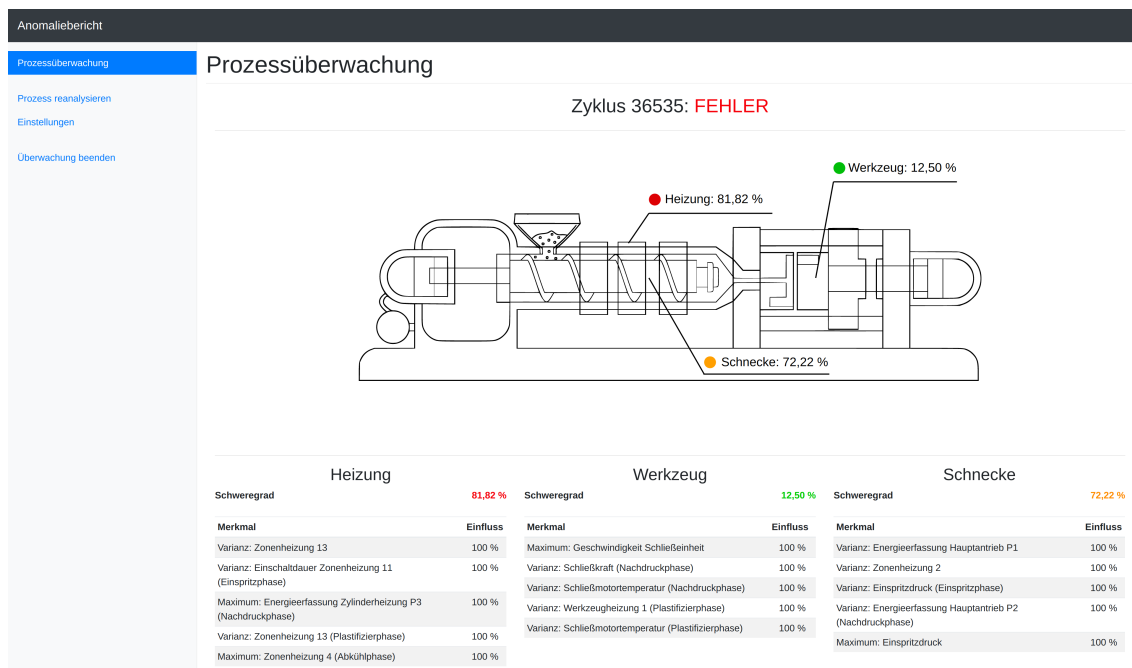


Abbildung 2.4: Konzept einer Schnittstelle zwischen Anomaliedetektor und dem Maschinenbediener in Form eines Anomalieberichts.

dener Merkmalsmengen zu bestimmen, aus diesen eine Anomalieentscheidung zu errechnen und anschließend die Häufigkeit der Merkmale zu ermitteln, die an der Entscheidung beteiligt waren (s. Abschnitt 5.5).

Mit Hilfe dieser Gewichtung ist es dann möglich, dem Bediener zu vermitteln, welche Merkmale bei der Anomalieentscheidung besonders wichtig waren, wobei sich dies dann auch mit der konkreten Merkmalsausprägung des betrachteten Produktionsprozesses verknüpfen lässt.

Kapitel 3

Vorverarbeitung

Dieses Kapitel führt die für das weitere Verständnis der Arbeit elementaren Definition der *Beobachtung* und *Zeitreihe* ein. Des Weiteren werden wichtige Vorverarbeitungsschritte besprochen, die vor der Anwendung maschineller Lernverfahren durchgeführt werden sollten. Da die Präzision der Vorhersagen und damit letztlich der Erfolg bei der Anomalieerkennung in essentieller Weise durch die verwendeten Merkmale bestimmt wird, steht im Fokus dieses Kapitels auch die Frage danach, welche Merkmale aus den Rohdaten extrahiert und welche Merkmale von Relevanz sind und daher selektiert werden sollten. Abschließend sollen zwei alternative Repräsentationen von Zeitreihen vorgestellt werden, die für die Anwendung von maschinellen Lernmethoden und weitergehende Erzeugung neuer Merkmale von Nutzen sein können.

3.1 Zeitreihe

Maschinendaten sind in der Regel in Form einer *Zeitreihe*, also einer zeitabhängigen Abfolge von so genannten *Beobachtungen*, strukturiert, die wie folgt definiert sind:

3.1.1 Definition (Beobachtung). Eine (zeitabhängige) *Beobachtung* ist ein Vektor $\vec{x}_t \in X^d$, wobei $d \in \mathbb{N}$ die Anzahl der Merkmale ist und $t \in \mathbb{N}$ einen Zeitpunkt eindeutig kennzeichnet. Die Ausprägung des i -ten Merkmals kann dann durch $\vec{x}_t[i]$ mit $1 \leq i \leq d$ adressiert werden.

Für den Fall, dass die Beobachtung \vec{x}_t eindimensional ist, also nur ein Merkmal pro Beobachtung betrachtet wird, sei der Zugriff auf die Ausprägung des Merkmals auch anhand der Schreibweise x_t mit $x_t \equiv \vec{x}_t[1]$ gestattet.

Intuitiv gesprochen, bündelt eine *Beobachtung* die Ausprägungen der vorhandenen Merkmale, die so genannte *Merkmalsmenge*, in einem Vektor.

Mit der Definition der Beobachtung kann dann die Zeitreihe definiert werden:

3.1.2 Definition (Zeitreihe). Eine *Zeitreihe* ist eine Menge $Z \subset X^d$ von Beobachtungen, wobei auf Z eine strenge Totalordnung $<_t \subseteq X^d \times X^d$ mittels der Zeit definiert ist, sodass $x_i <_t x_j \Leftrightarrow i < j$ gilt. Die Zeitreihe heißt *multivariat* für $d > 1$, sonst *univariat*.

3.1.1 Sequenzierung

Um dem Bediener den gewünschten, detaillierten Anomaliebericht liefern zu können, muss die Frage geklärt werden, welche Prozessabschnitte beziehungsweise Teilprozesse und welche Maschinenkomponenten die Anomalie verursacht haben. Die Annahme sei nun, dass sich von der Anomalie eines Teilprozesses auf die Anomalie des gesamten Prozesses schließen lässt. Durch die Zerlegung des Gesamtprozesses in Teilprozesse sollen sich Fehlerquellen, wie Maschinenkomponenten, näher eingrenzen lassen, die dann wiederum dem Bediener präsentiert werden können. Diese Informationen können vonseiten des Bedieners dann dazu genutzt werden, um Produktionsfehler genauer zu verorten und rechtzeitig abzustellen. Diese Annahme soll nun mit den Maschinendaten verknüpft werden: Daher wird die betrachtete Zeitreihe eines Prozesses entlang der Teilprozesse in einzelne Zeitreihen „zerschnitten“, was im Folgenden *Sequenzierung* genannt wird.

Um die Zeitreihe anhand der Prozessphasen sequenzieren zu können, wird vorausgesetzt, dass in den Daten, die die Produktionsanlage liefert, die zur Zeit jeweils aktive Produktionsphase anhand so genannter *Triggerattribute* kenntlich gemacht wird. Diese Triggerattribute besitzen den Wertebereich $\{\top, \perp\}$ und nehmen genau dann den Wert \top an, wenn die mit dem Attribut assoziierte Produktionsphase aktiv ist. Damit lässt sich dann die *sequenzierte Zeitreihe* definieren:

3.1.3 Definition (Sequenzierte Zeitreihe). Sei Z eine Zeitreihe und t der Index, mit dem sich die Ausprägung eines, wie oben definierten, Triggerattributs adressieren lässt. Dann ist die anhand des Triggerattributs T sequenzierte Zeitreihe $Z_T \subseteq Z$ wie folgt definiert:

$$Z_T = \{\vec{x}_t \in Z \mid \vec{x}_t[t] = \top\} \quad (3.1)$$

Es sei bemerkt, dass sich die Produktionsphasen auch überlappen oder parallel ablaufen können, sodass für zwei sequenzierte Zeitreihen Z_{T_1} und Z_{T_2} , die anhand zweier verschiedener Triggerattribute sequenziert wurden, diese Mengen nicht notwendigerweise disjunkt sind.

Zusammengefasst ist das Sequenzieren eine elegante Möglichkeit, um eine Zeitreihe zu strukturieren und die Komplexität des gesamten Prozesses aufzuteilen, wodurch sich auch einzelne Produktionsabschnitte näher betrachten lassen. Das Vorgehen ist jedoch nicht für jeden industriellen Produktionsprozess in dieser Einfachheit möglich: So ist die Spritzgusstechnik für ein solches Vorgehen zwar gut geeignet, da die einzelnen Teilprozesse des gesamten Produktionsprozesses in Theorie und Praxis klar definiert sind (vgl. [21], Abb. 3.2). Bei anderen industriellen Produktionstechniken, die beispielsweise die Nutzung einer CNC-Maschine involvieren, ist diese Form der Sequenzierung, aufgrund der hohen Variabilität des Produktionsprozesses, jedoch mit unter Umständen hohen technischen Hürden verbunden.

3.2 Normalisierung

Viele Lernverfahren hängen in essenzieller Weise davon ab, dass die zum Training des zugrundeliegenden Modells verwendeten Daten *normalisiert* werden. Dies ist grundsätzlich notwendig, um Daten miteinander vergleichen zu können [6], aber auch bei der Durchführung von Clusteranalysen, in denen Distanzfunktionen eine wichtige Rolle spielen (s. Definition 4.2.4), von entscheidender Bedeutung. So haben *Keogh* und *Kasetty* gezeigt, wie bereits ein minimaler Versatz von zu clusternden Zeitreihen, unabhängig von der gewählten Distanzfunktion, dazu führen kann, dass die berechnete Clusterung bedeutungslos wird, da die tatsächliche Form der Zeitreihe nicht mehr betrachtet wird [24].

In diesem Abschnitt soll daher ein Verfahren vorgestellt werden, das zum Zwecke der Normalisierung auf die Daten angewendet werden kann, nämlich die *Standardisierung*.

Standardisierung

Sei $B = \{\vec{x}_1, \dots, \vec{x}_n\} \subset X^d$ eine Menge von Beobachtungen und \bar{x}_m das arithmetische Mittel des m -ten Merkmals, dann ergibt sich die standardisierte oder z -transformierte Menge von Beobachtungen durch Berechnen der folgenden Formel,

$$\vec{x}_i[m] = \frac{\vec{x}_i[m] - \bar{x}_m}{\sigma_m}, \quad \forall i \in [1, n], \forall m \in [1, d] \quad (3.2)$$

wobei σ_m der Standardabweichung des m -ten Merkmals entspricht:

$$\sigma_m = \sqrt{\frac{1}{n} \sum_{\forall j \in [1, n]} (\vec{x}_j[m] - \bar{x}_m)^2} \quad (3.3)$$

Eine standardisierte Menge von Beobachtungen besitzt dann für jedes Merkmal ein arithmetisches Mittel von null und eine Stichprobenvarianz von eins [6].

3.3 Merkmalsextraktion

Die Merkmalsextraktion hat das Ziel, aus bestehenden Merkmalen und ihren Ausprägungen neue, abgeleitete Merkmale zu konstruieren, die für die weitere Verarbeitung genutzt werden können. Dieser Schritt ist einer der wichtigsten, um bei der weiteren Verarbeitung und Modellierung Erfolge zu erzielen [20] und sollte daher besonders beachtet werden.

Insbesondere im Bereich der maschinellen Lernverfahren, in denen sich die multivariaten Zeitreihen der Produktionsprozesse nicht als Trainingsdatensatz verwenden lassen, ist es notwendig, die relevanten Informationen aus den Zeitreihen zu *extrahieren* und diese zu einem *Merkmalsvektor*, der im Folgenden definiert wird, zusammenzufassen.

3.3.1 Definition (Merkmalsvektor). Sei $Z \subset X^{d_1}$ eine Zeitreihe und $\vec{m} \in X^{d_2}$ ein Vektor mit Wertausprägungen extrahierter Merkmale aus Z , dann heißt \vec{m} *Merkmalsvektor* der Zeitreihe Z .

Ist $Z^* = \{Z_1, \dots, Z_n\}$ eine Menge von Teilzeitreihen von Z und $M = \{\vec{m}_1, \dots, \vec{m}_n\}$ die zu den Zeitreihen gehörigen Merkmalsvektoren, dann werden diese zu einem einzigen Merkmalsvektor $\vec{m} = (\vec{m}_1, \dots, \vec{m}_n)^T$ konkateniert.

Im weiteren Verlauf der Arbeit wird der Merkmalsvektor aus der multivariaten Zeitreihe *eines* Produktionsprozesses extrahiert, sodass ein Merkmalsvektor genau einem Produktionsprozess zugeordnet ist. Der Merkmalsvektor *beschreibt* den Prozess dann gewissermaßen.

Wie und welche Merkmale extrahiert werden, ist dabei stark vom Anwendungsfall abhängig [20]: Typische Merkmale, die im Kontext von Zeitreihen extrahiert werden können, sind Minima beziehungsweise Maxima, die Stichprobenvarianz oder Mittelwerte (vgl. [21], Kap. 5.2). Aufwändigere Verfahren zur Merkmalsextraktion können beispielsweise die Durchführung einer Hauptkomponentenanalyse involvieren. Für die industrielle Anomalieerkennung ist es grundsätzlich entscheidend, aus den Zeitreihen, die von der Produktionsanlage stammen, gerade die Merkmale zu extrahieren, die eine Trennung zwischen normalen und anomalen Prozessen erlauben.

Ein Beispiel zur Merkmalsextraktion im industriellen Umfeld sei in [29] genannt: Dort wurde ein System zur Qualitätssicherung entwickelt, das Mängel im Bereich der Schweißnähte von miteinander verschweißten Stahlspulen erkennt. Aus der Feststellung, dass sich die Temperaturen und Ströme, die entlang der Schweißnaht entstehen beziehungsweise fließen, auf die Qualität des Endprodukts auswirken, wurde geschlossen, dass die Extraktion vom Mittelwert, Minimum, Maximum und der Varianz der dazugehörigen Merkmale für die Verarbeitung und Modellierung von Nutzen sein werden. Dies ist ein Beispiel für eine Merkmalsextraktion, die auf Basis von Prozesswissen erfolgt ist.

Im Kontext dieser Arbeit sollen zwei Verfahren zur Merkmalsextraktion vorgeschlagen werden: Zum einen könnte, ähnlich zum oben genannten Ansatz, Prozesswissen, auch *Domänenwissen* genannt, unter Verwendung von geeigneter Literatur herangezogen werden, um jene Merkmale zu bestimmen, die einen anomalen Produktionsprozess erkennen lassen und daher aus den sequenzierten Zeitreihe extrahiert werden sollten. Zum anderen könnte versucht werden, eine Menge von statistischen Kennzahlen aus den Zeitreihen zu extrahieren, während eine anschließende Merkmalsselektion die relevanten Merkmale bestimmt.

3.4 Merkmalsselektion

Neben der Extraktion der relevanten Merkmale, ist für den Erfolg der nachfolgenden Lernverfahren ebenfalls entscheidend, gerade die Merkmale auszuwählen, die für den zugrundeliegenden Anwendungsfall von Bedeutung sind, also in diesem Falle jene Merkmale zu wählen, die eine Trennung von normalen und anomalen Prozess erlauben. Dieser Vorverarbeitungsschritt wird *Merkmalsselektion* genannt und ist insbesondere dann wichtig, wenn unabhängig vom Prozesswissen Merkmale extrahiert wurden.

Dieser Schritt ist, neben dem genannten anwendungsspezifischen Aspekt, auch dadurch motiviert, dass hochdimensionale Daten die Verarbeitungsgeschwindigkeit negativ beeinflussen und die durch die Lernverfahren erzeugten Modelle aufgrund zusätzlicher Einflussgrößen schwerer zu verstehen sind [20, 33]. Nicht zu vernachlässigen ist ebenfalls ein Effekt, der als *Fluch der Dimensionalität* bekannt ist: Dieser beschreibt im Wesentlichen ein Problem, welches sich dadurch auszeichnet, dass die relative Distanz zwischen den am entferntesten und den am nächsten gelegenen Punkten in immer höherdimensionaler werdenden Räumen gegen null konvergiert [26], also:

$$\lim_{d \rightarrow \infty} \frac{dist_{max} - dist_{min}}{dist_{min}} \rightarrow 0 \quad (3.4)$$

Dies ist beispielsweise bei Clusteranalysen, die von Distanzfunktionen zur Berechnung einer Clusterung abhängen (s. Definition 4.2.4), ein wichtiger Aspekt, da durch den beschriebenen *Fluch der Dimensionalität* die berechnete Distanz zwischen beliebigen Punkten in hochdimensionalen Räumen an Bedeutung verliert.

Neben einer Auswahl von Merkmalen durch einen Experten beziehungsweise mittels Prozesswissen, ist es auch möglich, sich der Lösung dieser Aufgabe explorativ oder algorithmisch zu nähern. Daher sollen in diesem Abschnitt zunächst ein statistischer Ansatz zur Findung einer geeigneten Menge von Merkmalen genannt und anschließend ein algorithmisches Verfahren zur Merkmalsselektion vorgestellt werden.

3.4.1 Explorative Statistik

Einen ersten Schritt zur Merkmalsselektion bietet die Exploration der vorliegenden Merkmale mittels Methoden der deskriptiven Statistik. Dies empfiehlt sich auch über den Kontext der Merkmalsselektion hinaus, um ein Verständnis für die vorliegenden und zunächst möglicherweise unbekannt Daten zu gewinnen. Methoden der deskriptiven Statistik umfassen unter anderem die Anfertigung von Diagrammen, die Berechnung von statistischen Kennzahlen, wie Lage- oder Streumaßen oder die Bestimmung von empirischen Verteilungsfunktionen [6].

Wurden aus der multivariaten Zeitreihe eine Menge von Merkmalen extrahiert und sollen aus dieser nun Merkmale selektiert werden, ist es denkbar, je extrahiertem Merkmal ein Histogramm anzufertigen. Geht man nun davon aus, dass die zugrundeliegende Zufallsvariable normalverteilt ist und sich daher an den äußeren Intervallen der zugrundeliegenden Dichtefunktion seltene und damit auffällige Beobachtungen wiederfinden, sollte sich diese, wenn auch annähernd, in dem Histogramm widerspiegeln. Ist diese Verteilung hingegen nicht zu erkennen, weil beispielsweise alle Wertegruppen mit gleicher Häufigkeit auftreten, spricht dies dafür, dass das extrahierte Merkmal für die Anomalieerkennung irrelevant sind. Lässt sich im Histogramm hingegen eine Normalverteilung erkennen, sollte dieses extrahierte Merkmal näher in Betracht gezogen werden. In Abbildung 3.1 ist das beschriebene Vorgehen exemplarisch illustriert.

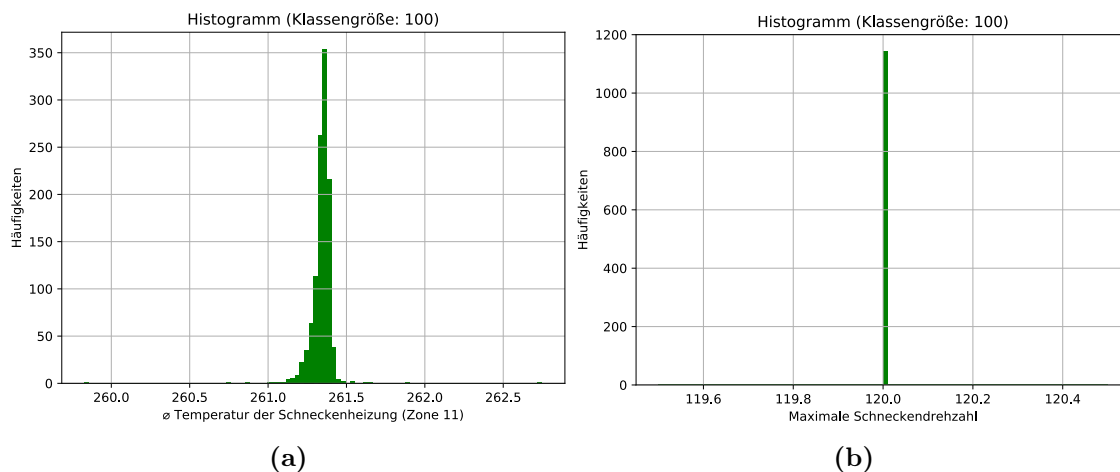


Abbildung 3.1: Histogramme von zwei verschiedenen, aus den vollständigen, unsequenzierten Zeitreihen und über 1154 Produktionsprozessen extrahierten Merkmalen. In Abbildung 3.1a ist ein Histogramm zu sehen, das nahe legt, dass die zugrundeliegenden Daten normalverteilt sind und sich daher dieses Merkmal zu modellieren lohnt. In Abbildung 3.1b hingegen, ist diese Verteilung nicht zu erkennen, was eine Auswahl dieses Merkmals unattraktiv macht.

Dieses Vorgehen eignet sich aber nur, wenn die vorliegende Stichprobe *hinreichend groß* ist und gemäß einer Normalverteilung „viele“ normale und „einige“ anomale Prozesse enthält, was wiederum schwierig festzustellen ist, wenn keine Label vorliegen.

3.4.2 Algorithmische Verfahren

Neben einer Selektion der vorhandenen Merkmale durch einen Experten oder mittels statistischer Annahmen, lässt sich die Merkmalsselektion auch algorithmisch bewerkstelligen. Diese Algorithmen lassen sich hierbei in folgende, übergeordnete Gruppen einordnen [19, 30, 34]:

- *Filter*: Filter bewerten die vorhandenen Merkmale mittels eines statistischen Tests und wählen anschließend die besten Merkmale aus [33]. Verfahren aus dieser Gruppe gelten gegenüber anderen Methoden als schnell und bieten eine allgemeinere Auswahl an Merkmalen, die nicht an das später genutzte maschinelle Lernverfahren angepasst ist [19].
- *Wrapper*: Verwenden ein zugrundeliegendes Lernverfahren, um Merkmalsmengen anhand ihres Einflusses auf die Vorhersage zu bewerten. Das Lernverfahren wird hierbei als „perfekte Blackbox“ angesehen und somit nur die Vorhersagegüte ausgewertet [19]. Da jede zu untersuchende Merkmalsmenge dazu führt, dass das zugrundeliegende maschinelle Lernverfahren erneut trainiert werden muss, werden diese Algorithmen häufig aufgrund ihres „Bruteforce“-Charakters und des hohen Berechnungsaufwandes kritisiert [19].
- *Embeddings*: Arbeiten ähnlich wie die Wrapper-Methoden, jedoch ist die Merkmalsselektion in den Trainingsprozess des Lernverfahrens integriert, wodurch sich einige der oben genannten Nachteile, die bei den Wrappern auftreten, wie beispielsweise das Neulernen bei veränderten Merkmalsmengen, vermeiden lassen [19].

Um von den maschinellen Lernverfahren unabhängig zu bleiben und die Zeit, die für die Merkmalsselektion aufgewendet wird, gering zu halten, soll im Folgenden die *Minimum Redundancy Maximum Relevance*-Merkmalsselektion, ein multivariater Filter, vorgestellt und im weiteren Verlauf der Arbeit verwendet werden.

Minimum Redundancy Maximum Relevance (MRMR)

Die von *Ding* und *Peng* vorgeschlagene Methode zur Merkmalsselektion namens *Minimum Redundancy Maximum Relevance* [12] ist ein multivariater Filter [30], dessen wesentliches

Charakteristikum es ist, jenes Merkmal als nächstes zu einer bereits vorhandenen Merkmalsmenge hinzuzufügen, dass das beste Verhältnis von Relevanz und Redundanz bezüglich dieser Merkmalsmenge aufweist. Dieser Selektionsansatz wird, im Kontrast zur *backward elimination*, als *forward selection* bezeichnet, da im jeden Schritt ein neues Merkmal zu der bereits vorhandenen Merkmalsmenge hinzugefügt wird. Im Folgenden soll die von *Schowe* und *Morik* [34] erstellte Beschreibung des MRMR-Verfahrens wiedergegeben werden, da sich diese auf die für diese Arbeit wesentlichen Aspekte fokussiert. Anschließend soll das Verfahren kurz diskutiert werden.

Um anhand von Relevanz und Redundanz eine Merkmalsselektion durchführen zu können, muss zunächst definiert werden, wie sich diese messen lässt. Grundsätzlich dafür geeignet sind Zusammenhangsmaße, wie beispielsweise der F -Test, die *gegenseitige Information* (engl. *mutual information*) oder der Korrelationskoeffizient von Pearson. Welches Maß letztlich eingesetzt wird, hängt davon ab, welche Merkmalstypen vorliegen: Während für die *gegenseitige Information* ausschließlich ein Zusammenhang nominaler Merkmale ermittelt werden kann, ist es bei der Verwendung des Korrelationskoeffizienten nach Pearson ausschließlich ein Zusammenhang numerischer Merkmale. Speziell bei der Untersuchung industrieller Prozesse werden üblicherweise numerische Merkmale betrachtet, weswegen der Korrelationskoeffizient nach Pearson definiert werden soll:

3.4.1 Definition (Bravais-Pearson-Korrelationskoeffizient [6]). Sei

$X = \{x_1, \dots, x_n\}$ und $Y = \{y_1, \dots, y_n\}$ eine Menge von univariaten Beobachtungen und \bar{x}, \bar{y} die dazugehörigen Durchschnittswerte. Dann definiert sich der Bravais-Pearson-Korrelationskoeffizient (auch: Pearson-Korrelationskoeffizient) r_{xy} wie folgt:

$$r_{xy} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}} \quad (3.5)$$

Da sich dennoch, wie beschrieben, verschiedene Definitionen des Begriffes hier nutzen lassen, wird ein abstrakter Korrelationsoperator $Cor(\cdot, \cdot)$ eingeführt, der, je nach vorliegenden Merkmalstypen, einem passenden Zusammenhangsmaß zugeordnet wird.

Die Merkmalsselektion beruht dann darauf, ausgehend von der gesamten Merkmalsmenge F und einer Menge bereits ausgewählter Merkmale F_j , als nächstes jenes Merkmal auszuwählen, dass das beste Verhältnis von Relevanz und Redundanz aufweist:

$$F_{j+1} = F_j \cup \left\{ \arg \max_{f \in F \setminus F_j} \frac{Cor(f, y)}{\frac{1}{j} \sum_{g \in F_j} Cor(f, g)} \right\} \quad (3.6)$$

Das Label y fungiert dabei als initiales Merkmal und dient dem Algorithmus als Startpunkt.

Die Merkmalsselektion wird abgebrochen, wenn eine im Vorfeld definierte Anzahl von Merkmalen $1 < k < |F|$ gefunden wurde. Das Ergebnis ist dann die Menge F_k .

Das MRMR-Verfahren ist eine einfache Methode um Merkmale, schnell und unabhängig von einem Lernverfahren zu selektieren. Diese gilt jedoch aufgrund ihrer starken Abhängigkeit von einem Korrelationsmaß als *instabil* [34]. Instabilität bedeutet in diesem Falle, dass wertmäßige Änderungen im verwendeten Datensatz dazu führen können, dass andere Merkmale selektiert werden. *Ensemble*-Methoden, die, unter anderem, den vorliegenden Datensatz in geeigneter Weise aufteilen und so die Varianz reduzieren, versuchen dieses Problem zu umgehen, gelten jedoch als langsam. Hierfür wurden jedoch Lösungen, wie das *fast ensemble* von *Schowe* und *Morik* [34], entwickelt. Zu beachten bleibt weiterhin, dass insbesondere bei Untersuchung von industriellen Prozessdaten, in denen hauptsächlich numerische Merkmale betrachtet werden, bei Verwendung des Korrelationskoeffizienten nach Pearson ausschließlich lineare Zusammenhänge zwischen Merkmalen untersucht werden, was sich möglicherweise nicht mit der (Merkmals-)Realität deckt.

3.5 Repräsentationen

Je nach vorliegender Situation, kann die Repräsentation einer Zeitreihe im Ursprungsraum, für die weitere Verarbeitung ungeeignet sein. Dies ist beispielsweise dann der Fall, wenn die ursprüngliche Repräsentation der Zeitreihe die Anwendung eines maschinellen Lernverfahrens verbietet oder aber, wenn durch die Transformierung der Zeitreihe in eine andere Repräsentation „Unterschiede“ zwischen Zeitreihen sichtbar werden, die vorher nicht erkennbar waren. Durch die Überführung der Zeitreihe in eine neue Repräsentation können sich so beispielsweise zusätzliche Möglichkeiten zur Extraktion von Merkmalen ergeben.

Aus diesen Gründen sollen, neben der ursprünglichen, zwei weitere Repräsentationen von Zeitreihen eingeführt werden, nämlich die *Symbolic Aggregate Approximation* (SAX) und die *Phasenraumtransformation*.

3.5.1 SAX

Die von *Lin et al.* eingeführte *Symbolic Aggregate Approximation* (SAX) adressiert die Problematik, dass viele maschinelle Lernverfahren auf diskrete aber nicht auf reellwertige Daten anwendbar sind [27]. Durch die Überführung einer reellwertigen Zeitreihe in eine diskrete Zeichenkette, die so genannte *Symbolisierung*, sollen diese Verfahren anwendbar werden, während sich weitere Vorteile wie eine Reduktion der Dimensionalität (s. Abschnitt 3.4) und die Garantie von unteren Schranken bei der Berechnung von Distanzen zwischen den Zeitreihen im Symbolraum ergeben, die dann wiederum die Geschwindigkeit bei Lernaufgaben, wie dem Clustering, verbessern können.

In diesem Abschnitt sollen die beiden wesentlichen Schritte zur Durchführung einer Symbolisierung einer Zeitreihe mittels SAX wiedergegeben werden, nämlich die Reduzierung der Dimensionalität und die Diskretisierung.

Reduzierung der Dimensionalität

Die *Piecewise Aggregate Approximation* (PAA) zerteilt eine vorliegende univariate Zeitreihe in eine definierte Anzahl von gleichgroßen Intervallen und bildet dann für jene Daten, die in ein Intervall fallen, den Mittelwert, wodurch die Dimensionalität der Daten reduziert wird. Dieses Vorgehen wird wie folgt formalisiert:

Gegeben sei eine univariate Zeitreihe $Z = \{\vec{x}_1, \dots, \vec{x}_n\}$, dann ergibt sich der i -te Vektor \vec{p}_i der, ebenfalls univariaten, PAA-Zeitreihe $Z_p = \{\vec{p}_1, \dots, \vec{p}_w\}$ wie folgt:

$$\vec{p}_i = \left(\begin{array}{c} \frac{w}{n} \sum_{j=\frac{n}{w}(i-1)+1}^{\frac{n}{w}i} x_j \end{array} \right) \quad (3.7)$$

Der Parameter $w \in [1, n]$ bestimmt die Anzahl der Intervalle, in die die Zeitreihe zerlegt werden soll. Für $w = n$ ist das Ergebnis die ursprüngliche Zeitreihe, für $w = 1$ wird der Mittelwert über die gesamte Zeitreihe gebildet.

Diskretisierung

Ausgehend von der von den Autoren getroffenen Annahme, dass normalisierte Zeitreihen normalverteilt sind, soll die univariate Zeitreihe derart symbolisiert werden, dass die Symbole jeweils gleichwahrscheinlich sind. Dazu werden so genannte *Breakpoints* definiert, die definitionsgemäß eine geordnete Liste von Werten $\beta^* = \beta_1, \dots, \beta_{a-1}$ sind, sodass der Flächeninhalt unterhalb einer $\mathcal{N}(\mu = 0, \sigma^2 = 1)$ -Dichtefunktion im Intervall $[\beta_i, \beta_{i+1}]$ gerade $1/a$ entspricht.

Die Diskretisierung und damit die SAX-Repräsentation der Zeitreihe Z ergibt sich dann aus der PAA-Repräsentation $Z_p = \{\vec{p}_1, \dots, \vec{p}_n\}$ und einem Alphabet Σ , auf dessen i -tes Element mittels Σ_i zugegriffen werden kann. Das i -te Element des SAX-Wortes $Z_s = (s_1, \dots, s_n)$ ist dann:

$$s_i = \Sigma_j, \text{ gdw. } \beta_{j-1} \leq \vec{p}_i < \beta_j \quad (3.8)$$

Das SAX-Wort Z_s ist dann das Ergebnis der Symbolisierung.

3.5.2 Phasenraumtransformation

Eine weitere Repräsentation ergibt sich durch die Transformation von Zeitreihen, in den so genannten Phasenraum. Diese Repräsentation kann Unterschiede zwischen Zeitreihen deutlich machen, die vorher nicht erkennbar waren: So konnte *Mierswa* beispielsweise mittels der Durchführung einer Phasenraumtransformation Unterschiede zwischen digitalisierten, als Zeitreihe repräsentierten Musikstücken verschiedener Genres sichtbar machen, die dann wiederum Ansatz zur Extraktion neuer Merkmale war [28]. Eine solche Transformation könnte auch für die Erkennung von Anomalien in Zeitreihen industriellen Ursprungs hilfreich sein, um anomales Verhalten erkennbar zu machen. Daher soll im Folgenden die Phasenraumtransformation vorgestellt werden.

Für eine gegebene, univariate Zeitreihe $Z = \{\vec{x}_1, \dots, \vec{x}_n\}$ ergibt sich die phasentransformierte Zeitreihe $P = \{\vec{p}_1, \dots, \vec{p}_n\}$ durch Bilden neuer Vektoren, deren Komponenten verzögerte Elemente der ursprünglichen Zeitreihe sind [1, 28]:

$$\vec{p}_i = (x_i, x_{i+d}, x_{i+2d}, \dots, x_{i+(m-1)d}) \quad (3.9)$$

Dabei ist d die zeitliche Verzögerung und m die resultierende Dimension des Phasenraums.

Im Rahmen einer Anomalieerkennung bietet die Phasenraumtransformation die Möglichkeit an, Zeitreihen in eine andere Repräsentation zu überführen und so, in Anlehnung an *Mierswa*, zusätzliche Möglichkeiten zur Erkennung von Unterschieden zwischen Zeitreihen zu schaffen. Dies ist insbesondere dann interessant, wenn die Möglichkeiten zur Merkmalsextraktion im ursprünglichen Raum bereits erschöpft und die Anomalieerkennung nur unzufrieden bewerkstelligt wird und somit zusätzliche oder andere Merkmale benötigt werden.

Kapitel 4

Modellierung

Dieses Kapitel knüpft an die Ergebnisse der Vorverarbeitung an und beschäftigt sich mit maschinellen Lernverfahren, die für eine Anomalieerkennung auf Basis des Spritzgussprozesses verwendet werden können. Hierzu werden ausgewählte Lernverfahren diskutiert und anschließend ihre Kernkonzepte vorgestellt. Im Fokus dieses Kapitels stehen dabei insbesondere Methoden des *unüberwachten Lernens*, also Methoden, die auf Daten ohne Label anwendbar sind.

Gemäß dem in der Zielsetzung formulierten Gedanken einer Pipeline, wird in diesem Kapitel vorausgesetzt, dass die im vergangenen Kapitel erhaltenen Merkmalsvektoren der Produktionsprozesse gerade die Ausprägungen jener Merkmale enthalten, die dazu geeignet sind, um zwischen anomalen und normalen Prozessen zu trennen. Dies ist notwendig, um bei der Modellierung mittels maschineller Lernverfahren zum Zwecke einer Anomalieerkennung Erfolge zu erzielen.

Im Folgenden werden Methoden aus dem Bereich der *Novelty Detection*, also der Entdeckung „anderer“ beziehungsweise „neuer“ Beobachtungen, und des Clusterings, also dem systematischen Bündeln von Beobachtungen in Gruppen, vorgestellt und für eine Anomalieerkennung motiviert.

4.1 Novelty Detection

Geht man davon aus, dass für einen Teil der Produktionsprozesse zweifellos festgestellt kann, ob diese anomal sind oder nicht, könnten die Daten jener Prozesse dazu verwenden, um zwischen diesen und „anderen“ Prozessen zu trennen. Dies ist der Ansatz der *Novelty Detection*.

Technisch gesehen, lernen diese Verfahren aus einer beliebigen Menge von Punkten, beziehungsweise in diesem Anwendungsfall, Merkmalsvektoren, dem *Trainingsdatensatz*, eine

Funktion, die für die *meisten* dieser Vektoren ein positives und für das Komplement der Menge ein negatives Ergebnis liefert [32]. Diese Funktion ist dann das *Modell*. Idealerweise würde dieses Modell, wenn es zum Zwecke einer Anomalieerkennung trainiert wurde, dann für einen Produktionsprozess direkt entscheiden, ob dieser bekannt und damit normal oder unbekannt und damit verdächtig ist. Ob diese verdächtigen Prozesse tatsächlich anomal sind, muss sorgfältig geprüft werden, da diese natürlich anomal aber eben auch normal sein könnten. Durch diesen Umstand wird der unüberwachte Charakter dieser Klasse von Methoden nochmals deutlich.

Möchte man die *Novelty Detection* für eine Anomalieerkennung einsetzen, dann ist der Trainingsdatensatz jedoch schwierig zu bestimmen: Aus theoretischer Perspektive wäre die Trainingsdatenmenge, gegeben einem Stapel bereits aufgezeichneter Daten auf den wahlfrei zugegriffen werden kann, gleichzusetzen mit der Menge der Merkmalsvektoren, die zu den sicher normalen Produktionsprozessen gehören. Aus praktischer Sicht fehlen jedoch im Datensatz jene Informationen, um diese Menge zu bestimmen, da die Produktionsanlage anomale Prozesse nicht kennzeichnet, sodass diese fehlerbehaftet approximiert werden muss.

Im Folgenden soll ein Verfahren vorgestellt werden, dass eine, wie oben beschriebene, Entscheidungsfunktion lernen kann, nämlich die *Single Class SVM*. Des Weiteren wird besprochen, wie eine geeignete Trainingsdatenmenge für das Training der Single Class SVM gefunden werden kann.

4.1.1 Single Class SVM

Die *Single Class SVM* wie sie von Schölkopf und Smola in [32] beschrieben wurde, lernt die besagte Entscheidungsfunktion anhand einer linearen Trennebene, die die Trainingsdaten vom Ursprung trennt. Für jeden neu zu bewertenden Punkt wird dann berechnet, auf welche Seite der Ebene dieser fällt, wodurch entschieden wird, ob der Punkt zu der Trainingsdatenmenge gehört oder nicht. Dabei wird diese Ebene derart bestimmt, dass diese die Daten mit maximalem Abstand vom Ursprung trennt: Es wird also ein Optimierungsproblem eingeführt, welches im Folgenden vorgestellt werden soll.

Optimierungsproblem

Sei $T \subset X^d$ ein Trainingsdatensatz und Φ eine Abbildung aus dem Datenraum X in einen höherdimensionalen Raum H , in dem das Skalarprodukt $\langle \cdot, \cdot \rangle$ definiert ist (ein *Prähilber-*

traum). Sei weiterhin $\|\cdot\|$ die euklidische Norm (siehe Definition 4.2.2). Das Optimierungsproblem definiert sich dann wie folgt:

$$\min_{\vec{w} \in H, \xi \in \mathbb{R}^m, \rho \in \mathbb{R}} \frac{1}{2} \|\vec{w}\|^2 + \frac{1}{\nu m} \sum_i \xi_i - \rho \quad (4.1)$$

$$\text{u. d. Nbdg.} \quad \forall i : \langle \vec{w}, \Phi(\vec{x}_i) \rangle \geq \rho - \xi_i, \xi_i \geq 0$$

Durch die Abbildung Φ , auch Merkmalsabbildung (engl. *feature map*) genannt, werden die Vektoren des Trainingsdatensatzes in einem höherdimensionalen Raum, den *Merkmalsraum*, transformiert, wodurch zwar eine lineare Ebene im Merkmalsraum gesucht wird, sich im Ursprungsraum, dem *Datenraum*, jedoch eine nicht-lineare Trennebene ergibt, die die Generalisierbarkeit des Ergebnisses verbessert. Der *Gewichtsvektor* \vec{w} bestimmt dann die Lage der Hyperebene, während die so genannte *Schlupfvariable* ξ_i die individuelle Überschreitung eines Vektors der Ebene angibt, wodurch aus der „harten Trennebene“ eine „weiche Trennebene“ wird. Der Begriff der weichen Trennebene beschreibt dabei die Möglichkeit, dass ein Vektor die Trennebene überschreitet. Der skalare Offsetwert ρ bestimmt, in Abhängigkeit der Norm von \vec{w} , die Verschiebung der Ebene vom Ursprung (s. Abbildung 4.1). Der Parameter $\nu \in (0, 1]$ fungiert als obere Grenze für den Anteil an Ausreißern und untere Grenze für den Anteil an Stützvektoren, die die Trennebene definieren (vgl. Proposition 8.3 in [32]). Nähert sich ν dabei null an, werden keine Ausreißer zugelassen und aus der weichen Trennebene wird eine harte Trennebene.

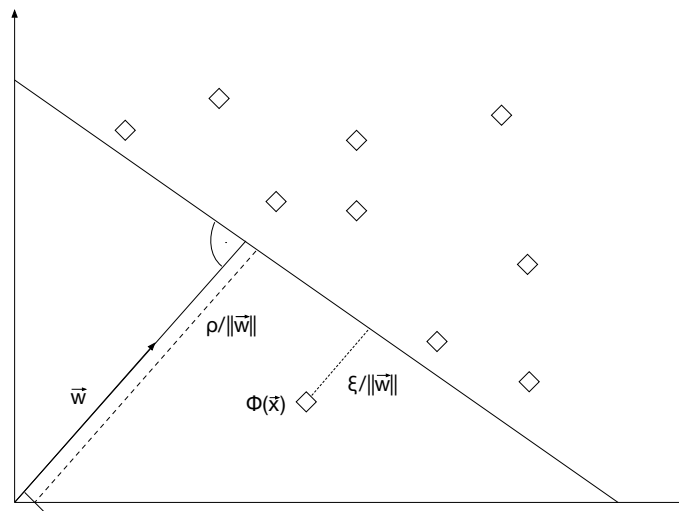


Abbildung 4.1: Illustration des Single Class SVM-Verfahrens [32].

Die Funktion, die für einen (neuen) Punkt entscheidet, auf welcher Seite dieser liegt (die *Entscheidungsfunktion*), wird dann wie folgt definiert:

$$f(\vec{x}) = \text{sgn}(\langle \vec{w}, \Phi(\vec{x}) \rangle - \rho) \quad (4.2)$$

Das Symbol „sgn“ steht für die Signumsfunktion, die dem Argument sein Vorzeichen zuordnet.

Das oben genannte bedingte Optimierungsproblem wird nun durch die Einführung von Lagrange-Multiplikatoren $\alpha_i, \beta_i \geq 0$ in ein äquivalentes, unbedingtes Optimierungsproblem, dem so genannten *primalen Problem*, überführt. Dieses besitzt die folgende Zielfunktion L_P :

$$L_P = \frac{1}{2} \|\vec{w}\|^2 + \frac{1}{\nu m} \sum_i \xi_i - \rho - \sum_i \alpha_i (\vec{w} \cdot \Phi(\vec{x}_i) - \rho + \xi_i) - \sum_i \beta_i \xi_i \quad (4.3)$$

Dieses primale Problem lässt sich auch als *duales Problem*, im Sinne einer Lagrange-Dualität, formulieren und lösen. Die Formulierung des dualen Problems hat für SVM-Probleme einen entscheidenden Vorteil, wie sich gleich zeigen wird.

Prinzipiell liefert die Lösung des *dualen Problems* eine untere Schranke für die optimale Lösung des primalen Problems [4]. Die Differenz zwischen dem optimalen Wert des primalen und dem optimalen Wert des dualen Problems³ wird *Dualitätslücke* (engl. *duality gap*) genannt. Ist diese Differenz größer null, wird von einer *schwachen Dualität*, sonst von einer *starken Dualität* gesprochen.

Dies wird nun dafür verwendet, um die so genannten *Karush-Kuhn-Tucker-Bedingungen* einzuführen, die als notwendige Optimalitätskriterien für Optimierungsprobleme dieser Form dienen:

4.1.1 Definition (Karush-Kuhn-Tucker-Bedingungen [4]). Gegeben sei ein Optimierungsproblem in der Standardform:

$$\begin{array}{lll} \min & f_0(x) & \\ \text{u. d. Nbdg.} & f_i(x) \leq 0, & i = 1, \dots, m \\ & h_i(x) = 0, & i = 1, \dots, p \end{array}$$

Seien die Funktionen f_0, \dots, f_m und h_1, \dots, h_p differenzierbar. Außerdem seien x^* und (λ^*, ν^*) Lösungen für das primale und duale Problem mit einer Dualitätslücke von null, dann gelten die *Karush-Kuhn-Tucker-Bedingungen*:

$$\begin{aligned} f_i(x^*) &\leq 0, & i = 1, \dots, m \\ h_i(x^*) &= 0, & i = 1, \dots, p \\ \lambda_i^* &\geq 0, & i = 1, \dots, m \\ \lambda_i^* f_i(x^*) &= 0, & i = 1, \dots, m \end{aligned} \quad (4.4)$$

$$\nabla f_0(x^*) + \sum_{i=1}^m \lambda_i^* \nabla f_i(x^*) + \sum_{i=1}^p \nu_i^* \nabla h_i(x^*) = 0,$$

Für dieses und folgende SVM-Probleme in dieser Arbeit wird, ohne Beweis, angenommen, dass die starke Dualität gilt und die KKT-Bedingungen anwendbar sind.

Die KKT-Bedingungen fordern also als notwendige Bedingung für die Optimalität einer Lösung, dass die Ableitungen von L_P gleich null sind:

$$\frac{\partial L_P}{\partial \vec{w}} = 0 \Leftrightarrow \vec{w} = \sum_i \alpha_i \Phi(\vec{x}_i) \quad (4.5)$$

$$\frac{\partial L_P}{\partial \xi_i} = 0 \Leftrightarrow \alpha_i = \frac{1}{\nu m} - \beta_i \quad (4.6)$$

$$\frac{\partial L_P}{\partial \rho} = 0 \Leftrightarrow \sum_i \alpha_i = 1 \quad (4.7)$$

Die Gleichungen 4.5 - 4.7 können nun in das primale Problem eingesetzt werden, um so das *duale Problem* zu erhalten:

$$\begin{aligned} \min_{\alpha^* \in \mathbb{R}^m} \quad L_D &= \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j \langle \Phi(\vec{x}_i), \Phi(\vec{x}_j) \rangle \\ \text{u. d. Nbdg.} \quad 0 &\leq \alpha_i \leq \frac{1}{\nu m}, \sum_i \alpha_i = 1 \end{aligned} \quad (4.8)$$

Die im dualen Problem enthaltenen Transformationen der Vektoren in den höherdimensionalen Prähilbertraum gehen nur noch als Operanden innerhalb von Skalarprodukten ein, wodurch sich die Möglichkeit der Nutzung eines *Kernels* ergibt.

4.1.2 Definition (Kernel [31]). Sei X ein Eingaberaum (auch Datenraum), H ein prähilbertscher Merkmalsraum mit dem Skalarprodukt $\langle \cdot, \cdot \rangle$ und $\Phi : X \rightarrow H$ eine Abbildung (*feature map*). Eine Abbildung K heißt *Kernel*, gdw.:

$$K(x, x') = \langle \Phi(x), \Phi(x') \rangle \quad (4.9)$$

Damit lässt sich die Zielfunktion des dualen Problems auch äquivalent schreiben, als:

$$L_D = \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j K(\vec{x}_i, \vec{x}_j) \quad (4.10)$$

Diese Äquivalenz stellt damit einen der wichtigsten Gründe dar, die die Aufstellung und Lösung des dualen Problems motivieren. Anstatt die betrachteten Vektoren *explizit* in einen höherdimensionalen und potentiell unendlichdimensionalen Raum zu überführen, um dort das Skalarprodukt zu berechnen, kann ein Kernel diese Berechnung *implizit* durchführen und ist damit deutlich effizienter [31].

Durch die Nutzung eines Kernels und der Gleichung 4.5, lässt sich die Entscheidungsfunktion aus Gleichung 4.2 auch schreiben als:

$$f_K(\vec{x}) = \text{sgn} \left(\sum_i \alpha_i K(\vec{x}_i, \vec{x}) - \rho \right) \quad (4.11)$$

Es fällt auf, dass sich trotz einer gefundenen Lösung α^* für das duale Problem, die Entscheidungsfunktion für einen beliebigen Punkt \vec{x}_i nicht berechnen lässt, da der Offsetwert ρ unbekannt ist. Mittels der KKT-Bedingungen lassen sich, falls die dualen Variablen $\alpha_i, \beta_i > 0$, die Ungleichungen in der Nebenbedingung des primalen Problems (s. Gleichung 4.1) in Gleichungen überführen, sodass gilt:

$$\rho = \langle \vec{w}, \Phi(\vec{x}_i) \rangle = \sum_j \alpha_j K(\vec{x}_j, \vec{x}_i) \quad (4.12)$$

Damit lässt sich, gegeben einem gelösten dualen Problem, für einen beliebigen Punkt entscheiden, auf welcher Seite der Hyperebene diese liegt.

4.1.2 Wahl der Trainingsdatenmenge

Wie bereits in der Einleitung erwähnt, muss für die *Novelty Detection* zunächst eine Trainingsdatenmenge bestimmt werden. Da für eine Anomalieerkennung normale Prozesse von anomalen Prozessen getrennt werden sollen, wäre die Trainingsmenge, theoretisch gesehen, eben die Menge von Merkmalsvektoren, die zu normalen Prozessen gehören. Aus praktischer Perspektive fehlen Hintergrundinformation, um diese Menge zu bestimmen, da Produktionsmaschinen in der Regel weder anomale noch normale Prozesse kennzeichnen. Eine Trainingsdatenmenge zu finden ist daher eine nicht-triviale Aufgabe, wofür im Folgenden einige Ansätze diskutiert werden sollen.

Zum einen wäre es denkbar, einen Experten zu befragen und diesen die Bewertung *einiger* Zyklen übernehmen zu lassen. Dies kann zum einen lokal, also direkt nach Abschluss des Produktionsprozesses, durch den Maschinenbediener erfolgen. Dies entspricht dann entfernt der Anwendung eines so genannten *Active Learning*-Ansatzes, bei dem der Bediener Einfluss darauf nehmen kann, welche Daten in die Modellierung eingehen [7]. Zum anderen könnte diese manuelle Bewertung durch den Experten auch offline, also anhand aufgezeichneter Daten, erfolgen, sofern die hierfür benötigten Informationen beziehungsweise Merkmale vorhanden sind. Wie viele Produktionsprozesse manuell bewertet werden müssen, hängt dabei letztlich von der Vorhersagequalität ab.

Eine andere Möglichkeit ergibt sich durch die Aufstellung eines *simplen* Systems von Regeln, das für sicher normale Produktionsprozesse immer erfüllt ist. Dieses System könnte insofern *simpel* sein, als dass pro Regel immer nur ein Merkmal betrachtet wird und Zusammenhänge zwischen Merkmalen nicht betrachtet werden. Dadurch ließe sich die gesamte Menge an Merkmalsvektoren danach filtern und das Resultat als Trainingsmenge verwenden. Man ist dann jedoch darauf angewiesen, dass dieses Regelsystem sehr umfassend und eine überproportional hohe Anzahl von falschen Entscheidung ausgeschlossen ist, da ansonsten die Trainingsdatenmenge für eine *Novelty Detection* ungeeignet ist. Dieses Regelsystem ist dabei, aufgrund der beschriebenen Unzulänglichkeiten, nicht als eigen-

ständiger „Anomaliedetektor“ sondern lediglich als Behelf vorgesehen, um eine geeignete Trainingsdatenmenge für die Single Class SVM zu finden.

4.2 Clusteranalysen

Clusteranalysen gehören zu den Lernverfahren des unüberwachten Lernens, die Punkte anhand eines Distanz- oder Ähnlichkeitsbegriffes in Mengen, den sogenannten *Clustern*, gruppieren. Dabei sollen die Punkte eines Clusters möglichst ähnlich sein, während Punkte verschiedener Cluster möglichst unähnlich sind [40].

Die Nutzung von Clusteranalysen zur Anomalieerkennung ist durch die Hypothese motiviert, dass Merkmalsvektoren, die aus den Zeitreihen normaler Produktionsprozessen extrahiert wurden, nicht nur überproportional häufig vertreten sind, sondern auch ähnlicher und damit näher zueinander sind, als jene Vektoren, die aus anomalen Zeitreihen extrahiert wurden. Denkbar ist daher, dass die Clusterverfahren Merkmalsvektoren normaler Prozesse zu einem „großen“ Cluster verhältnismäßig hoher Kardinalität zuzuordnen, während Merkmalsvektor anomaler Prozesse einem oder mehreren „kleinen“ Clustern zugeordnet werden, die sich dann idealerweise auch in Anomalietypen aufteilen lassen (s. Abbildung 2.2a). Dadurch sollen diese zu anomalen Prozessen zugehörigen Merkmalsvektoren als anomal identifizierbar gemacht werden. Voraussetzung dafür ist, dass die Anzahl der Cluster gering gehalten wird, um noch eine Anomalieentscheidung treffen zu können, und diese Cluster möglichst kompakt sind, um sicherzugehen, dass die Vektoren ausreichend ähnlich zueinander sind. Es wird also ein Optimierungsproblem impliziert, dass in Abschnitt 5.4 besprochen wird.

In diesem Kapitel wird zunächst der Begriff der Distanz beziehungsweise Nähe formalisiert, um die Lernaufgabe des Clusterings zu definieren. Anschließend werden zwei Lernverfahren, die eine Clusteranalyse durchführen können, präsentiert und hinsichtlich der Verwendbarkeit zum Zwecke einer Anomalieerkennung bewertet. Zum Abschluss des Kapitels wird diskutiert, wie sich die gefundene Clusterung validieren und für eine Anomalieerkennung interpretieren lässt.

4.2.1 Lernproblem

Algorithmen, die eine Clusteranalyse durchführen, versuchen, wie bereits erwähnt, Punkte anhand eines vorher zu definierenden Distanzbegriffes zu gruppieren, sodass Punkte einer Gruppe (eines Clusters) *näher* zu einander sind, als Punkte der jeweils anderen Cluster. Diese informelle Beschreibung soll nun formalisiert werden, wobei hierzu zunächst der Begriff der Distanzfunktion definiert wird:

4.2.1 Definition (Distanzfunktion [40]). Eine Funktion $dist : X^d \times X^d \rightarrow \mathbb{R}^+$ heißt Distanzfunktion, gdw.:

$$\forall \vec{x} \in X^d : dist(\vec{x}, \vec{x}) = 0 \quad (4.13)$$

$$\forall \vec{x}_1, \vec{x}_2 \in X^d : dist(\vec{x}_1, \vec{x}_2) = dist(\vec{x}_2, \vec{x}_1) \quad (4.14)$$

Wird zusätzlich die Dreiecks-Ungleichung eingehalten, ist die Distanzfunktion eine *Metrik*:

$$\forall \vec{x}, \vec{y}, \vec{z} \in X^d : dist(\vec{x}, \vec{y}) \leq dist(\vec{x}, \vec{z}) + dist(\vec{z}, \vec{y}) \quad (4.15)$$

Als Beispiel sei die euklidische Norm angegeben:

4.2.2 Definition (Euklidische Norm [3]). Sei V ein reeller Prähilbertraum und $\vec{x} \in V$ dann wird durch

$$\|\vec{x}\| = \sqrt{\langle \vec{x}, \vec{x} \rangle} \quad (4.16)$$

eine (euklidische) Norm von V definiert.

Daraus ergibt sich der euklidische Abstand, wie folgt:

4.2.3 Definition (Euklidischer Abstand [22]). Sei V ein reeller Prähilbertraum und $\|\cdot\|$ die euklidische Norm. Seien weiterhin $\vec{x}, \vec{y} \in V$. Dann ist der euklidische Abstand eine Distanzfunktion, die wie folgt definiert ist:

$$d(\vec{x}, \vec{y}) = \|\vec{x} - \vec{y}\| \quad (4.17)$$

Im weiteren Verlauf der Arbeit wird stets der euklidische Abstand als Distanzfunktion für das Berechnen von Clusterungen genutzt.

Ausgehend von dem Begriff der Distanzfunktion kann die Lernaufgabe des Clusterings definiert werden:

4.2.4 Definition (Clustering [40]). Gegeben sei $T \subseteq X^d$ eine Menge von Punkten. Weiterhin sei

$$\text{dist} : X^d \times X^d \rightarrow \mathbb{R}^+ \quad (4.18)$$

eine Distanzfunktion und

$$q : \mathcal{P}(\mathcal{P}(X^d)) \rightarrow \mathbb{R} \quad (4.19)$$

eine Qualitätsfunktion. Finde nun eine Clusterung $\mathcal{C} = \{C_1, \dots, C_k\}$ mit $C_i \subseteq T$ für alle $i = 1, \dots, k$, sodass $q(\mathcal{C})$ maximal wird und $C_i \cap C_j = \emptyset$ für alle $i, j \in \{1, \dots, k\}$ mit $i \neq j$.

Es sei hierbei bemerkt, dass eine Zuordnung aller Elemente von T in einen Cluster explizit nicht erwartet wird. Diese Eigenschaft wird im weiteren Verlauf der Arbeit benötigt, um Punkte, die sich keinem Cluster zuordnen lassen, als *Rauschpunkte* zu klassifizieren.

Algorithmen, die eine Clusterung, wie sie in Definition 4.2.4 definiert wurde, finden, werden im weiteren Verlauf der Arbeit als *Clusterverfahren* bezeichnet.

4.2.2 Support Vector Clustering

Das Support Vector Clustering (SVC) von Ben-Hur *et al.* [2] basiert auf dem Konzept der Support Vector Domain Description, wie sie unter anderem von Tax und Duin beschrieben wurde [36]. Ansatz dieses Verfahrens ist es, die Punktmenge für die eine Clusterung bestimmt werden soll, unter Verwendung einer *feature map* (s. Definition 4.1.2) aus dem Datenraum in einen prähilbertschen Merkmalsraum zu transformieren, in dem eine minimale Sphäre gesucht wird, die die Punkte einschließt. Aus der Sphäre im Merkmalsraum ergeben sich im Datenraum, je nach Parametrisierung, Konturen, aus der die Clusterung bestimmt werden kann.

Das Suchen einer minimalen Sphäre im Merkmalsraum impliziert ein Optimierungsproblem, welches wie folgt definiert wird. Sei R der Radius der Sphäre, \vec{a} ihr Zentrum, $\|\cdot\|$ die euklidische Norm, $\Phi : X \rightarrow H$ eine Abbildung aus dem Datenraum in den Merkmalsraum mit $X \subset \mathbb{R}^d$, $T = \{\vec{x}_1, \dots, \vec{x}_n\} \subset X$ die Punkte, die in der Sphäre eingeschlossen werden sollen, dann wird das Optimierungsproblem wie folgt bestimmt (vgl. auch [36]):

$$\begin{aligned} \min_R \quad & R^2 + C \sum_j \xi_j \\ \text{u. d. Nbdg.} \quad & \forall j : \|\Phi(\vec{x}_j) - \vec{a}\|^2 \leq R^2 + \xi_j, \xi_j \geq 0 \end{aligned} \quad (4.20)$$

Die Schlupfvariable ξ_j gestattet einem Punkt, analog zu Abschnitt 4.1.1, die Überschreitung der Grenzen der Sphäre, wobei C als Bestrafungsparameter, hinsichtlich der Optimierung, eine Abwägung zwischen dem Volumen der Sphäre und der Anzahl der Punkte, die außerhalb der Sphäre liegen, ermöglicht [36].

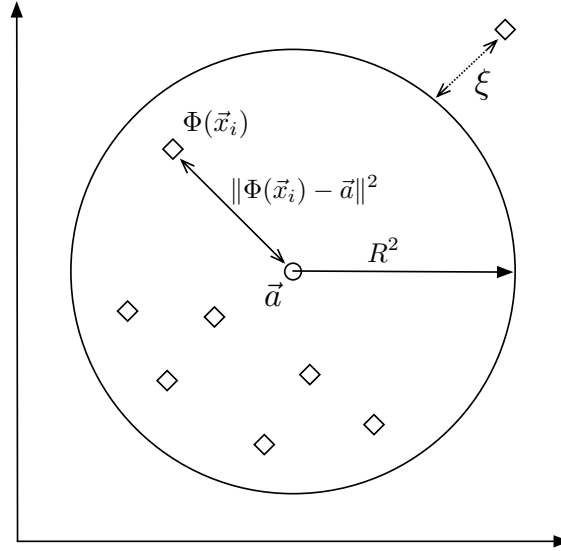


Abbildung 4.2: Visualisierung der Größen im Optimierungsproblem des Support Vector Clusters.

Dieses Optimierungsproblem mit Nebenbedingungen wird durch die Einführung von Lagrange-Multiplikatoren $\beta_i, \mu_i \geq 0$ in die primale Zielfunktion L_P überführt:

$$L_P = R^2 - \sum_j (R^2 + \xi_j - \|\Phi(\vec{x}_j) - \vec{a}\|^2) \beta_j - \sum_j \xi_j \mu_j + C \sum_j \xi_j \quad (4.21)$$

Mit den KKT-Bedingungen (s. Definition 4.1.1) ergibt sich als notwendige Bedingung für eine optimale Lösung, dass die Ableitungen von L_P Null entsprechen müssen:

$$\frac{\partial L_P}{\partial R} = 0 \Leftrightarrow \sum_j \beta_j = 1 \quad (4.22)$$

$$\frac{\partial L_P}{\partial \vec{a}} = 0 \Leftrightarrow \vec{a} = \sum_j \beta_j \Phi(\vec{x}_j) \quad (4.23)$$

$$\frac{\partial L_P}{\partial \xi_j} = 0 \Leftrightarrow \beta_j = C - \mu_j \quad (4.24)$$

Aus den KKT-Bedingungen ergeben sich außerdem folgende notwendige Bedingungen für eine optimale Lösung:

$$\xi_j \mu_j = 0 \quad (4.25)$$

$$(R^2 + \xi_j - \|\Phi(\vec{x}_j) - \vec{a}\|^2) \beta_j = 0 \quad (4.26)$$

Mit Hilfe dieser Bedingungen lassen sich dann, gegeben einem beliebigen Punkt $\Phi(\vec{x}_i)$, drei Fälle unterscheiden, die im Falle einer optimalen Lösung eintreten können:

1. $\Phi(\vec{x}_i)$ liegt außerhalb der Sphäre im Merkmalsraum: Dann gilt $\xi_i > 0$ und aus Gleichung 4.25 folgt $\mu_j = 0$. Aus Gleichung 4.24 folgt dann $\beta_i = C$. Diese Punkte werden *bounded support vectors* (BSV) genannt und stellen damit gewissermaßen die Rauschpunkte dar.
2. $\Phi(\vec{x}_i)$ liegt auf oder innerhalb der Sphäre im Merkmalsraum: Dann gilt $\xi_i = 0$ wegen der Nebenbedingung.
3. Falls $0 < \beta_i < C$ und $\xi_i = 0$ gilt, dann liegt $\Phi(\vec{x}_i)$ wegen Gleichung 4.26 auf der Sphäre im Merkmalsraum.

Dabei sei bemerkt, dass im Falle von $C \geq 1$ keine Punkte außerhalb der Sphäre liegen können, da wegen Gleichung 4.22 $\beta_i \leq 1$ gelten muss. Für $C \geq 1$ folgt aus Gleichung 4.24 dann jedoch $\mu_i > 0$ und mit Gleichung 4.25 gilt dann $\xi_i = 0$, womit kein Punkt außerhalb der Sphäre liegt.

Durch das Einsetzen der Gleichungen 4.22 - 4.24 in die Lagrange-Funktion L_P wird dann, ähnlich wie in Kapitel 4.1.1, das duale Problem L_D eingeführt, wobei $\langle \cdot, \cdot \rangle$ dem Skalarprodukt entspricht:

$$\begin{aligned} \max_{\beta^*} \quad L_D &= \sum_j \langle \Phi(\vec{x}_j), \Phi(\vec{x}_j) \rangle \beta_j - \sum_{i,j} \beta_i \beta_j \langle \Phi(\vec{x}_i), \Phi(\vec{x}_j) \rangle \\ \text{u. d. Nbdg.} \quad &0 \leq \beta_j \leq C, \forall j \end{aligned} \quad (4.27)$$

Diese Repräsentation des vorliegenden Optimierungsproblems erlaubt dann die Nutzung eines Kernels, sodass sich die Zielfunktion des dualen Problems wie folgt schreiben lässt:

$$L_D = \sum_j K(\vec{x}_j, \vec{x}_j) \beta_j - \sum_{i,j} \beta_i \beta_j K(\vec{x}_i, \vec{x}_j) \quad (4.28)$$

Wurde das duale Problem gelöst, muss man sich, um die Clusterung bestimmen zu können, der im Merkmalsraum resultierenden Sphäre widmen. Diese wird letztlich genau durch die auf der Sphäre liegenden Punkte definiert, also:

$$R = \{R(\vec{x}_i) \mid \vec{x}_i \text{ ist ein Stützvektor}\} \quad (4.29)$$

$R(\cdot)$ entspricht hierbei dem Abstand eines Punktes zum Mittelpunkt der Sphäre:

$$R^2(\vec{x}) = \|\Phi(\vec{x}) - \vec{a}\|^2 \quad (4.30)$$

Durch die Berechnung des quadrierten Abstandes eines Punktes zum Mittelpunkt der Sphäre, lässt sich diese Formel, ähnlich zum dualen Problem, äquivalent mit Skalarprodukten und daher mittels Kernen schreiben:

$$R^2(\vec{x}) = K(\vec{x}, \vec{x}) - 2 \sum_j \beta_j K(\vec{x}_j, \vec{x}) + \sum_{i,j} \beta_i \beta_j K(\vec{x}_i, \vec{x}_j) \quad (4.31)$$

Die Konturen im Datenraum, aus denen sich dann anschließend die Clusterung bestimmen lassen, werden dann durch folgende Punkte bestimmt:

$$\{\vec{x} \mid R(\vec{x}) = R\} \quad (4.32)$$

An dieser Stelle ist jedoch nichts über die Clusterzuordnung bekannt. Um diese zu bestimmen, wird wie folgt verfahren: Gegeben zwei Punkten \vec{x}_i und \vec{x}_j die zu verschiedenen Clustern gehören, dann muss die Verbindungslinie, die zwischen diesen beiden Beobachtungen gezogen werden kann, die Sphäre im Merkmalsraum verlassen. Es lässt sich dann eine Adjazenzmatrix A erzeugen, deren Komponenten wie folgt bestimmt werden:

$$A_{ij} = \begin{cases} 1, & \text{wenn, für alle } \vec{y} \text{ auf der Verbindungslinie, } R(\vec{y}) \leq R \\ 0, & \text{sonst} \end{cases} \quad (4.33)$$

Die Cluster lassen sich dann durch die miteinander verbundenen Punkte bestimmen, während BSVs als Rauschpunkte klassifiziert werden.

Das *Support Vector Clustering* bietet den Vorteil, Rauschpunkte zu identifizieren und Cluster beliebiger Form finden zu können. Dies ist wichtig, da wir keine Annahmen über die Struktur der Daten treffen können. Das Lösen des SVM-Problems hat jedoch standardmäßig eine asymptotische, zeitliche Komplexität von $\mathcal{O}(n^3)$ [38]. Die zeitliche Komplexität für das Bestimmen der Cluster beträgt $\mathcal{O}((N - n_{bsv})n_{sv}^2)$, wobei N der Anzahl der Punkte, n_{bsv} der Anzahl der BSV und n_{sv} der Anzahl der Stützvektoren entspricht [2].

4.2.3 DBSCAN

Das DBSCAN-Verfahren (Density Based Spatial Clustering of Applications with Noise) von Ester *et al.* [14] ist ein Verfahren zur dichtebasierten Clusteranalyse. Im *DBSCAN*-Verfahren bilden Punkte einen Cluster, wenn hinreichend andere Punkte in der Nähe liegen und diese in geeigneter Weise miteinander verbunden sind, was letztlich auf den Begriff der *Dichte* führt. Es muss daher neben der für Clusteralgorithmen üblichen Distanzfunktion, ebenso eine Anzahl von Punkten definiert werden, die in der Nähe eines Punktes liegen müssen. Dieses Verständnis des Clusterbegriffes erlaubt es, Punkte auch keinem Cluster zuzuordnen und diese dementsprechend als *Rauschen* aufzufassen. Diese Einführung in den *DBSCAN*-Algorithmus wird nun um die in [14] eingeführte Begrifflichkeit erweitert, die dann letztlich auf den, für dieses Verfahren spezifischen, Clusterbegriff führt.

Wie bereits erwähnt, muss für die Clusterbildung im DBSCAN-Verfahren zunächst ein Begriff für Punkte geprägt werden, die in der Nähe eines anderen Punktes liegen. Dieses Konzept wird von Ester *et al.* als ϵ -Nachbarschaft wie folgt formalisiert:

4.2.5 Definition (ϵ -Nachbarschaft eines Punktes). Sei $dist : X^d \times X^d \rightarrow \mathbb{R}^+$ eine Distanzfunktion, dann ist die ϵ -Nachbarschaft eines Punktes \vec{x} unter der Menge von Punkten $T \subset X^d$, notiert als N_ϵ , definiert durch:

$$N_\epsilon(\vec{x}) = \{\vec{y} \in T \mid dist(\vec{x}, \vec{y}) \leq \epsilon\} \quad (4.34)$$

Die ϵ -Nachbarschaft eines Punktes beschreibt also eine Menge von anderen Punkten, dessen Distanz geringer als ein Schwellwert, nämlich ϵ , ist.

Dies wird nun dazu verwendet, um eine Dichtebeziehung zwischen zwei Punkten zu formulieren.

4.2.6 Definition (Direkte Dichteerreichbarkeit). Ein Punkt \vec{x} heißt ausgehend von einem Punkt \vec{y} *direkt dichteerreichbar* hinsichtlich der Definition von ϵ und $minPts$, gdw. folgende Bedingungen zutreffen:

$$\vec{x} \in N_\epsilon(\vec{y}) \quad (4.35)$$

$$|N_\epsilon(\vec{y})| \geq minPts \quad (4.36)$$

Die *direkte Dichteerreichbarkeit* definiert also eine binäre, nicht-symmetrische Relation zwischen Punkten, die jeweils in ihrer ϵ -Umgebung liegen. Es ist nun wünschenswert, diese Relation zwischen zwei Punkten, insofern zu erweitern, als dass eine beliebige Folge von Punkten zwischen ihnen liegen darf, die aber untereinander direkt dichteerreichbar sind. Dieses Konzept wird von Ester *et al.* als *Dichteerreichbarkeit* bezeichnet:

4.2.7 Definition (Dichteerreichbarkeit). Ein Punkt \vec{x} heißt ausgehend von einem Punkt \vec{y} *dichteerreichbar* gdw. eine Folge von Punkten $(\vec{x}_1, \dots, \vec{x}_n)$ mit $\vec{x}_1 = \vec{x}$ und $\vec{x}_n = \vec{y}$ existiert, sodass \vec{x}_{i+1} ausgehend von \vec{x}_i direkt dichteerreichbar ist.

Die direkte Dichteerreichbarkeit ist aufgrund der *minPts*-Bedingung im Allgemeinen keine symmetrische Relation, was sich für die Definition von Clustern beim DBSCAN-Verfahren als Schwierigkeit erweist. Insbesondere bei Betrachtung von Randpunkten eines Clusters, die weniger Punkte in ihrer ϵ -Umgebung aufweisen, könnte zwar ein Randpunkt \vec{x} von einem anderen Punkt \vec{y} dichteerreichbar sein, während die symmetrische Fall nicht gilt. Um dieses Problem zu lösen, wird der Begriff der *Dichteverbundenheit* definiert:

4.2.8 Definition (Dichteverbundenheit). Ein Punkt \vec{x} heißt ausgehend von einem Punkt \vec{y} *dichteverbunden* gdw. ein Punkt \vec{z} existiert, sodass \vec{x} und \vec{y} ausgehend von \vec{z} dichteerreichbar sind.

Die Dichteverbundenheit verlangt also die Spezifikation eines weiteren Punktes, die zwischen den beiden zu verbindenden Punkten liegt und von *diesem* aus dann die Dichteerreichbarkeit verlangt wird. Dies resultiert in einer symmetrischen Relation, die dann für die Definition des Clusterbegriffs, wie folgt, verwendet wird:

4.2.9 Definition (Cluster (DBSCAN)). Sei T eine Menge von Punkten, dann ist ein Cluster \mathcal{C} eine nicht-leere Teilmenge von T , die folgende Bedingungen erfüllt:

$$\forall \vec{x}, \vec{y} \in T : \text{Wenn } \vec{x} \in \mathcal{C} \text{ und } \vec{y} \text{ ausgehend von } \vec{x} \text{ dichteerreichbar ist, dann ist } \vec{y} \in \mathcal{C} \quad (4.37)$$

$$\forall \vec{x}, \vec{y} \in \mathcal{C} : \vec{x} \text{ ist dichteverbunden mit } \vec{y} \quad (4.38)$$

Aus der sich ergebenden Clusterung kann nun der Begriff des *Rauschens* formalisiert werden, womit alle Definition, die zum Verständnis des DBSCAN-Verfahren benötigt werden, eingeführt sind.

4.2.10 Definition (Rauschen (DBSCAN)). Sei $\mathcal{C} = \{C_1, \dots, C_k\}$ eine Clusterung, dann sind Rauschpunkte alle Punkte, die keinem Cluster zugeordnet wurden, also:

$$N = \{\vec{x} \in T \mid \nexists C \in \mathcal{C} : \vec{x} \in C\} \quad (4.39)$$

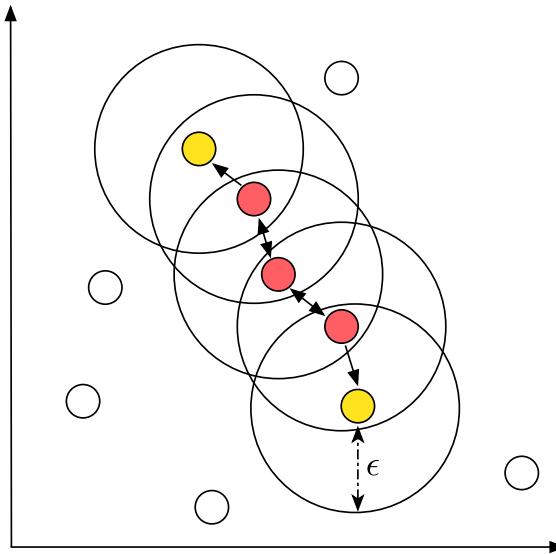


Abbildung 4.3: Illustration des DBSCAN-Verfahrens ($minPts = 2$) anhand einer Menge von zweidimensionalen Punkten und unter Verwendung der euklidischen Distanz. Rauschpunkte sind weiß, Randpunkte des Clusters gelb und Kernpunkte des Clusters rot markiert (eigene Darstellung, angelehnt an [14]).

Das DBSCAN-Verfahren bietet, wie das *Support Vector Clustering*, die Eigenschaft, Cluster beliebiger Form und Rauschpunkte zu identifizieren. DBSCAN ist jedoch nicht in der Lage, Cluster unterschiedlicher Dichte zu erkennen und weist im schlimmsten Fall eine zeitliche Komplexität von $\mathcal{O}(n^2)$ auf [18].

4.2.4 Validierung

Der vorletzte Schritt bei der Durchführung einer Clusteranalyse ist die Clustervalidierung [41]. Die Eigenschaften einer Clusterung werden stark davon beeinflusst, welcher Algorithmus und die dazugehörigen Parameter gesetzt werden [41]. Insbesondere kann ein Clusterverfahren eine Clusterung berechnen, obwohl die gefundene Struktur in den Daten nicht vorhanden ist [41]. Ziel der Clustervalidierung ist es, eine solche, für den Anwendungsfall bedeutungslose, Clusterung zu identifizieren.

Ein erster Schritt zur Clustervalidierung wurde bereits mit der Wahl geeigneter Clusterverfahren gegangen. Durch die gezielte Wahl von Clusterverfahren, die Cluster beliebiger Struktur finden können, werden bestimmte Voraussetzungen, wie beispielsweise das Vorliegen von konvexen Clustern bei Anwendung des Clusterverfahrens *k*-Means, vermieden und so Cluster gefunden, die am ehesten der vorliegenden räumlichen Struktur entsprechen. Da dies jedoch nicht ausreichend sein kann, soll zum einen diskutiert werden, wie und ob

sich eine Verbindung zwischen dem Optimierungsproblem und einer validen Clusterung herstellen lässt und wie eine Clusterung auch grafisch überprüft werden kann.

So werden durch die Lösung des Optimierungsproblems, wie es in der Einleitung angedeutet und in Abschnitt 5.4 besprochen wird, objektive Qualitätskriterien geboten, die sicherstellen, dass die Clusterung insgesamt kompakt und die Cluster in niedriger Zahl vorliegen, was eine notwendige Bedingung für die Validität bei Betrachtung des vorliegenden Anwendungsfalles darstellt.

Eine weitere Möglichkeit zur Clustervalidierung ergibt sich durch die grafische Visualisierung der geclusterten Punkte. Dabei ist wichtig, dass die gewählte Visualisierungsform die Möglichkeit bietet, Punkte entsprechend einzufärben und sich aus der Grafik schließen lässt, welche Punkte, im Sinne der gewählten Distanzfunktion, nah beieinander liegen. Bei Betrachtung der visualisierten Punkte kommt es dann darauf an, subjektiv festzustellen, ob die gefundene Clusterung, in Abhängigkeit der gewählten Merkmale, dem entspricht, was erwartet wurde. So sollten Merkmalsvektoren normaler Produktionsprozesse nahe beieinander liegen und Ausreisser zu anomalen Prozessen gehören.

Für Punkte niedriger Dimensionalität ($d \leq 3$) bietet sich hierfür ein Scatterplot an, bei dem die Komponenten der Punkte als kartesische Koordinaten interpretiert werden und diese punktweise in ein entsprechendes Koordinatensystem eingetragen werden.

Häufig sind die Punkte jedoch deutlich höherer Dimensionalität, wodurch diese Form der Datenvisualisierung nicht genutzt werden kann. In diesem Falle kann beispielsweise auf Techniken wie Parallelkoordinaten oder Spinnennetzdiagramme zurückgegriffen, um die Punkte zu visualisieren. Diese Visualisierungsformen lassen sich jedoch nicht auf Punkte beliebiger Dimensionalität anwenden und wirken bei entsprechend hoher Dimensionalität unübersichtlich oder überladen, was eine Clustervalidierung zusätzlich erschwert. Daher muss für den Fall, dass eine visuelle Clustervalidierung notwendig ist, dafür gesorgt werden, dass die Punkte eine hinreichend geringe Dimensionalität aufweisen.

4.2.5 Interpretation

Nach der Validierung der Clusterung, folgt die Clusterinterpretation [41]. Wie bereits in der Einleitung beschrieben, arbeiten Algorithmen der Clusteranalyse unüberwacht, also ohne jegliches dediziertes Hintergrundwissen. Folglich kann kein Hintergrundwissen bei der Findung der Clusterung einbezogen werden und die gefundenen Cluster müssen durch den Benutzer interpretiert werden.

Eine Möglichkeit zur Clusterinterpretation, die in dieser Arbeit diskutiert werden soll, ist das temporäre Bereitstellen von Labeln. Werden für einen Trainingsdatensatz Informationen über den tatsächlichen Anomaliestatus zur Verfügung gestellt, ließe sich damit

überprüfen, ob die Clusterung, wie in Abschnitt 4.2 gewünscht, alle normalen Prozesse einem Cluster und alle anomalen Prozesse einem, beziehungsweise allen anderen, Clustern zugeordnet hat. Ist diesem Fall ist die Clusterung und die zugrundeliegende Merkmalsmenge ein aussichtsreicher Kandidat, um Prognosen hinsichtlich des Anomaliestatus eines neuen Produktionsprozesses zu treffen.

Sind keine Label verfügbar, können einige Repräsentanten der Cluster ausgewählt und unter Verwendung von Prozesswissen überprüft werden. So sollte ein Merkmalsvektor, der einem kleinen Cluster zugeordnet wurde und damit verdächtig ist, Merkmalsausprägungen aufweisen, die definitiv auf einen anomalen Prozess hinweisen. Unter der Bedingung dass eine optimale Lösung gemäß Abschnitt 5.4 gefunden wurde und die Cluster damit möglichst kompakt sind, könnte von dem Repräsentanten des Clusters auf den gesamten Cluster geschlossen und dieser zu einem Cluster, der anomale Prozesse beinhaltet, erklärt werden. An dieser Stelle wird nochmals deutlich, dass eine gute Auswahl an Merkmalen zur Identifikation anomaler Prozesse unbedingt notwendig ist.

4.3 Hochvolumige Daten

Industrielle Fertigungsprozesse liefern im Allgemeinen in fest definierten Abständen Daten ihrer Sensorik und ihrer Aktorik, wodurch sich in relativ kurzer Zeit eine beträchtliche Anzahl von Beobachtungen akkumuliert. Hinzukommt, dass einige Fertigungsprozesse verhältnismäßig kurz sind und sich so auch schnell Merkmalsvektoren aus den Zeitreihen der Produktionsprozesse ansammeln. Setzt man für den, hier exemplarisch betrachteten, Spritzgießprozess eine Einzelprozessdauer von 30 Sekunden und eine Abtastrate von 200 Hz an, so können während einer „normalen“ Arbeitsschicht von acht Stunden 960 Einzelprozesse betrachtet und etwa 5,7 Millionen Beobachtungen aufgezeichnet werden, wobei die Dimensionalität der Daten noch zu betrachten wäre. Neben der Frage, wie man diese Datenmengen möglichst effizient ablegt, muss man sich unter anderem damit beschäftigen, wie sich Lernverfahren auch auf größere Datenmengen applizieren lassen.

So wurden in den vergangenen Abschnitten zwei Stützvektormethoden präsentiert, die zur *Novelty Detection* und zum *Clustering* eingesetzt werden können. Die Lösung der zugrundeliegenden Optimierungsprobleme weist jedoch standardmäßig eine kubische zeitliche Komplexität und eine quadratische Speicherkomplexität auf [38], was insbesondere bei großen Daten zu einer rechnerisch anspruchsvollen Aufgaben wird. Aus diesem Grund sollen die Grundzüge zweier Verfahren vorgestellt werden, die zur schnellen Approximation einer optimalen Lösung bei Stützvektormethoden genutzt werden können, nämlich die *Core Vector Machine* und die *Ball Vector Machine*.

4.3.1 Core Vector Machine

Die von Tsang *et al.* vorgeschlagene Methode der *Core Vector Machine* (CVM) basiert auf der Idee, eine Äquivalenz zwischen dem so genannten *Minimum Enclosing Ball*-Problem (MEB) und Kernelmethoden, zu denen auch die SV-Methoden gehören, herzustellen und anschließend eine Approximation der MEB-Lösung mittels so genannter *Core Sets* zu finden [38]. Es soll nun, wie in [38] dargelegt, das MEB-Problem und die Approximation des Problems vorgestellt werden, um anschließend die Verbindung zwischen Kernelmethoden und dem MEB-Problem herzustellen und das grundsätzliche Vorgehen der *Core Vector Machine* herausgestellt werden.

Das MEB-Problem ist durch eine Menge von Punkten $T = \{\vec{x}_1, \dots, \vec{x}_n\}$ mit $T \subset \mathbb{R}^d$ definiert, wofür die kleinste, diese Punkte einschließende, Sphäre $\text{MEB}(T)$ gefunden werden soll. Analog zu Abschnitt 4.2.2 definiert sich die Sphäre durch einen Mittelpunkt \vec{c} und einen Radius R , sodass die Sphäre als $B(\vec{c}, R)$ angegeben wird. Anstatt eine exakte Lösung für dieses Problem zu finden, kann auch eine so genannte $(1 + \epsilon)$ -Approximation von $\text{MEB}(T)$ angegeben werden, mit $B(\vec{c}, (1 + \epsilon)R)$. Diese ist genau dann gegeben, wenn $T \subset B(\vec{c}, (1 + \epsilon)R)$ und $R \leq R_{\text{MEB}(T)}$, also wenn der Radius der approximierten Sphäre kleiner oder gleich der optimalen Lösung ist. Es kann nun ein *Core Set* Q der gesamten Punktmenge T für die Approximation angegeben werden, sodass eine $(1 + \epsilon)$ -Erweiterung von $\text{MEB}(Q)$ alle Punkte aus T enthält, also $T \subset B(\vec{c}, (1 + \epsilon)R)$ mit $B(\vec{c}, R) = \text{MEB}(Q)$. Dies soll laut Tsang *et al.* zu einer genaueren und effizienteren Approximation des MEB führen.

Um die Äquivalenz zwischen dem MEB-Problem und Kernelmethoden herzustellen, wird das *Support Vector Domain Description*-Problem mit harter Trennung (*hard margin*) eingeführt. Sei T , wie oben definiert, der Trainingsdatensatz, Φ eine Abbildung aus dem Datenraum in einen höherdimensionalen Prähilbertraum, $\|\cdot\|$ die euklidische Norm, \vec{c} das Zentrum einer Sphäre und R ihr Radius, dann ist das *hard-margin SVDD*-Problem wie folgt definiert:

$$\begin{aligned} \min_{R, \vec{c}} \quad & R^2 \\ \text{u. d. Nbdg.} \quad & \forall i : \|\vec{c} - \Phi(\vec{x}_i)\|^2 \leq R^2 \end{aligned} \tag{4.40}$$

Dieses Problem entspricht dabei dem SVC-Problem in Abschnitt 4.2.2 (siehe auch Abbildung 4.2) mit dem Unterschied, dass dort unter Verwendung von Schlupfvariablen eine weiche Trennung (*soft margin*) ermöglicht wird.

Es lässt sich nun zeigen, dass sich Kernelmethoden als *hard margin*-SVDD-Problem und somit als MEB-Problem schreiben lassen. Grundlage hierfür ist jedoch, dass, gegeben einen Kernel K , die Gleichung $K(\vec{x}, \vec{x}) = \kappa$ für alle \vec{x} erfüllt und κ konstant ist. Tsang *et al.* führen die Umformulierung einer *Single Class SVM* und einer *Two-Class SVM* exempla-

risch durch. Die technischen Details dieser Umformulierungen sollen hier jedoch unberührt bleiben.

Die Idee der CVM ist nun, die Sphäre derart inkrementell zu vergrößern, dass der vom Zentrum entfernteste Vektor in neuen *Core Set* enthalten ist. Seien nun T_t , \vec{c}_t und R_t das Core Set, das Zentrum der Sphäre und der Radius der Sphäre zum Zeitpunkt t , dann ist der CVM-Algorithmus wie folgt gegeben:

Algorithmus 4.1 Core Vector Machine

Eingabe: Trainingsdatensatz $\{\vec{z}_1, \dots, \vec{z}_n\}$

- 1: Initialisiere T_0 , \vec{c}_0 , R_0 und $t = 0$.
 - 2: Terminiere, wenn kein Element des Trainingsdatensatzes mit $\tilde{\Phi}(\vec{z})$ außerhalb der $(1+\epsilon)$ -Sphäre $B(\vec{c}_t, (1+\epsilon)R_t)$ liegt.
 - 3: Finde den von \vec{c}_t entferntesten Punkt mit $\tilde{\Phi}(\vec{z})$ und setze $T_{t+1} = T_t \cup \{\vec{z}\}$
 - 4: Finde den neuen MEB(T_{t+1}) und setze $\vec{c}_{t+1} = \vec{c}_{\text{MEB}(T_{t+1})}$ und $R_{t+1} = R_{\text{MEB}(T_{t+1})}$
 - 5: Inkrementiere t und fahre mit dem zweiten Schritt fort.
-

Wie sich diese Schritte konkret realisieren lassen, ist von Tsang *et al.* beschrieben worden [38]. An dieser Stelle soll jedoch die zeitliche und die Speicherkomplexität genannt werden, wie sie in [38] beschrieben worden sind, um die Vorteile der CVM herauszustellen.

Unter der Annahme, dass ein Lösungsverfahren für quadratische Programme mit einer zeitlichen Komplexität von $\mathcal{O}(n^3)$ und einer Speicherkomplexität von $\mathcal{O}(n^2)$ für jeweils n Punkte genutzt wird, ist die zeitliche Komplexität des CVM-Verfahrens linear in der Anzahl der Punkte und liegt bei $\mathcal{O}(\frac{n}{\epsilon^2} + \frac{1}{\epsilon^4})$. Die wesentliche Erkenntnis, die zu dieser Laufzeitbewertung führt, ist die, dass die Gesamtzahl an Iteration von Algorithmus 4.1 und die finale Kardinalität des *Core Sets* durch $\mathcal{O}(1/\epsilon)$, mit festgelegtem ϵ , beschränkt ist.

Zur Lösung des quadratischen Programms werden nur die Elemente des, in der aktuellen Iteration vorhandenen, *Core Sets* benötigt. Mit der genannten Beschränkung der Iterationen folgt für die Speicherkomplexität $\mathcal{O}(1/\epsilon^2)$.

Wird der so genannte *probabilistic speedup* genutzt, bei dem im Wesentlichen eine zufällige Teilmenge des aktuellen *Core Sets* zur Berechnung des entferntesten Punktes betrachtet wird, lässt sich die zeitliche Komplexität zu $\mathcal{O}(1/\epsilon^8)$ und die Speicherkomplexität zu $\mathcal{O}(1/\epsilon^4)$ benennen und damit die Komplexität unabhängig von der Kardinalität der Beobachtungsmenge ausdrücken.

Zusammengefasst lassen sich zwei wichtige Vorteile bei der Nutzung der CVM benennen: Zum einen wird durch die Nutzung der *Core Sets* der MEB immer nur für eine deutlich kleinere Teilmenge der gesamten Punktmenge berechnet, wodurch die zu lösenden quadratische Programme (QP) eine geringere Komplexität aufweisen. Zum anderen wird durch den inkrementellen Charakter des CVM-Algorithmus, bei dem eine bereits gefundene Lö-

sung aus einer vorherigen Iteration als Startpunkt verwendet wird, schneller eine Lösung seitens des QP-Lösungsverfahrens erreicht. Das CVM-Verfahren ist daher eine hilfreiche Methode, um eine SVM auch mit sehr vielen Beispielen effizient zu trainieren.

4.3.2 Ball Vector Machine

Die *Ball Vector Machine* (BVM) von Tsang *et al.* baut auf dem *Core Vector Machine*-Verfahren (CVM), welches im vergangenen Abschnitt beschrieben wurde, auf [37]. Jedoch wird, in Abgrenzung zu diesem Verfahren, nicht der *Minimum Enclosing Ball* (MEB) sondern ein so genannter *Enclosing Ball* (EB) gesucht. Dieser definiert sich, Tsang *et al.* zu folge, dabei durch eine Sphäre $B(\vec{c}, R)$, die die gesamte Punktmenge $T = \{\vec{x}_1, \dots, \vec{x}_n\}$ einschließt, wobei der Radius des EB R fest definiert und größer oder gleich der optimalen Lösung R^* , also jener des MEB, ist.

Die $(1 + \epsilon)$ -Approximation des EB erfolgt, nach Tsang *et al.*, nun so, dass das Zentrum der Sphäre immer derart verschoben wird, sodass ein Punkt, der außerhalb der Sphäre liegt, nach der Verschiebung des Mittelpunkts gerade die Oberfläche tangiert. Der Algorithmus wird wie folgt skizziert:

Algorithmus 4.2 $(1 + \epsilon)$ -Approximation des *Enclosing Balls*.

Eingabe: Trainingsdatensatz $T = \{\vec{x}_1, \dots, \vec{x}_n\}$, Radius R , Featuremap Φ

- 1: Initialisiere $\vec{c}_0 = \Phi(\vec{x}_1)$ und $t = 0$.
 - 2: Terminiere, wenn kein Element des Trainingsdatensatzes mit $\Phi(\vec{x})$ außerhalb von $B(\vec{c}_t, (1 + \epsilon)R)$ liegt. Ansonsten sei $\Phi(\vec{x}_t)$ ein solcher Punkt.
 - 3: Finde die kleinste Veränderung des Zentrums der Sphäre, sodass $B(\vec{c}_{t+1}, R)$ gerade $\Phi(\vec{x}_t)$ berührt.
 - 4: Inkrementiere t und fahre mit dem zweiten Schritt fort.
-

Bei geeigneter Wahl des Radius, was letztlich die BVM ausmacht, nähert sich das Zentrum des EB jenem des MEB an, wodurch dann die Prozeduren der CVM anwendbar werden [37].

Die zeitliche Komplexität des BVM-Verfahrens liegt bei $\mathcal{O}(1/\epsilon^4)$, während die Speicherkomplexität bei $\mathcal{O}(1/\epsilon^2)$ liegt [37].

Zum zentralen Vorteil dieses Verfahrens gehört, dass kein numerisches Lösungsverfahren erforderlich ist. Zusätzlich ist dieses Verfahren in der Lage, sehr große Datensätze zu verarbeiten und ist allgemein schneller als die Core Vector Machine [37].

Kapitel 5

Evaluation

In diesem Kapitel sollen einige, ausgewählte Methoden des vergangenen Abschnittes in geeigneter Weise miteinander verknüpft und Experimente entwickelt werden, die die Eignung der verwendeten Verfahren zur Anomalieerkennung innerhalb der Spritzgusstechnik feststellen. Dabei soll es hier insbesondere darum gehen, wie die Qualität der Modellierung auch ohne die dazugehörigen *wahren* Werte festgestellt werden kann.

Hierzu wird zunächst der Spritzgießprozess, in dem Anomalien detektiert werden sollen, in seinen Grundzügen vorgestellt. Anschließend werden die Rohdaten, die die betrachtete Spritzgießanlage liefert, beschrieben. Im Hauptteil dieses Kapitels werden, entsprechend einer Pipeline, ausgehend von den Rohdaten, die praktische Realisierung der Anomalieerkennung in Abhängigkeit der vorliegenden Daten und der betrachteten Spritzgießmaschine besprochen. Dazu gehört insbesondere, dass die zu verwendenden, abstrakten Konzepte zur Modellierung und Vorverarbeitung auf das konkrete Problem übertragen und adaptiert werden.

5.1 Der Spritzgießprozess

Im Folgenden soll die von *Haman* erstellte Beschreibung des Spritzgussprozesses reproduziert werden [21]. Der Fokus liegt dabei insbesondere darauf, den Prozess zu strukturieren und die qualitätsrelevanten Phasen herauszustellen. Dies soll im weiteren Verlauf der Arbeit dabei helfen, die Sequenzierung und Merkmalsselektion beziehungsweise -extraktion angemessen zu bewerkstelligen. Durch die Strukturierung des einzelnen Produktionsprozesses sollen außerdem die Komponenten, die an einer Anomalie beteiligt sind, eingegrenzt werden.

Beim Spritzgießen wird im Wesentlichen, wenn auch stark vereinfacht und verkürzt ausgedrückt, geschmolzenes Plastikgranulat, auch schlicht *Schmelze* genannt, in den Hohlraum

eines Werkzeug, die so genannte *Werkzeugkavität*, eingespritzt, um so ein Produkt, das *Formteil*, zu erhalten. Dieses Werkzeug, beziehungsweise seine Kavität, stellt das Negativ dessen dar, was geformt werden soll. Das Formteil kann dann nach einer Abkühlphase entnommen werden. Dieser Prozess, der von Spritzgießern auch als *Zyklus* bezeichnet wird, ist klar strukturiert (vgl. [21], Abb. 3.2) und dauert, bei der in der vorliegenden Arbeit betrachteten Spritzgussmaschine, etwa zwanzig Sekunden. Der Begriff des *Zyklus* ist dabei als Konkretisierung des abstrakten Begriffes des *Prozesses* zu verstehen.

Im Folgenden soll die oberflächliche Einführung in den Spritzgießprozess nun etwas vertieft werden:

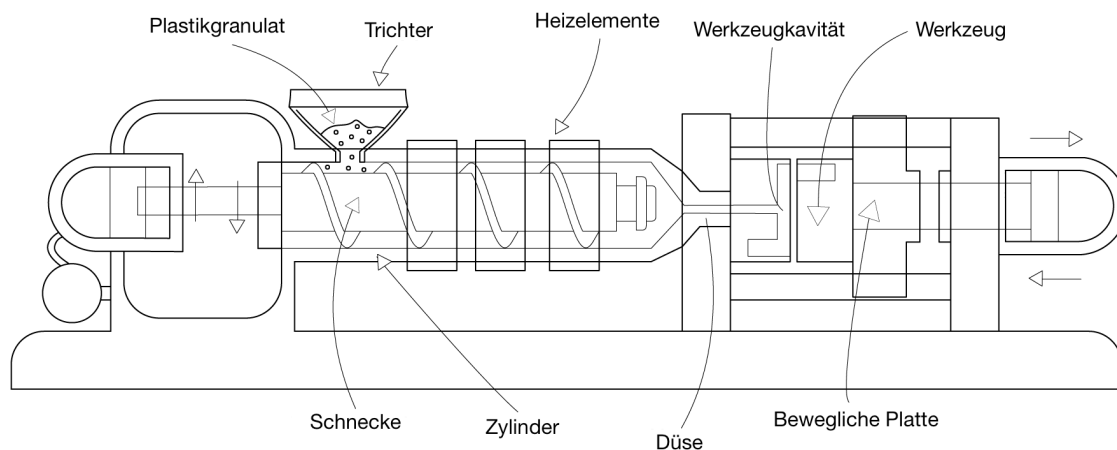


Abbildung 5.1: Skizze einer Spritzgießmaschine [5].

Grundsätzlich kann der so genannte Formteilbildungsprozess in *Plastifizieren* und *Formteilbildung* aufgeteilt werden.

Beim *Plastifizieren* wird das Plastikgranulat aus dem Trichter, mit dem Ziel eine möglichst homogene Schmelze zu erhalten, aus dem festen in den flüssigen Zustand überführt. Die benötigte Wärme wird dem Granulat dabei durch mechanische Reibung innerhalb der Schnecke und mittels angebrachten Heizelementen zugeführt.

Die *Formteilbildung* umfasst dann die Prozessphasen, die für die Qualität des Produktes von Relevanz sind. Dazu gehören die Einspritz-, die Kompressions- und die Nachdruckphase, welche sich bei Betrachtung des Druckes, der im inneren des Werkzeuges herrscht, dem so genannten *Werkzeuginnendruck*, voneinander trennen lassen. In der Einspritzphase wird die Schmelze in die Werkzeugkavität eingespritzt, in der Kompressionsphase wird der Einspritzdruck, mit der die Schmelze in die Kavität gespritzt wurde, auf den niedrigeren Nachdruck gesenkt. Der Zeitpunkt, zu dem diese Umschaltung erfolgt, heißt Umschalt- punkt und ist ein wichtiger Indikator für die Qualität des resultierenden Produktes. Die Nachdruckphase schließt sich an die Kompressionsphase an und hat das Ziel, die Schmelze in der Kavität zu komprimieren. Der Druck, mit dem diese Komprimierung erfolgt, heißt

Nachdruck und wird konstant gehalten. Parallel zum Nachdrücken verläuft der Abkühlvorgang, der solange aufrecht gehalten wird, bis die so genannte *Entformtemperatur* erreicht wird. Die Entformtemperatur stellt die Temperatur dar, bei der das Formteil aus dem Werkzeug entnommen werden kann. Mit dem Erreichen der Entformtemperatur wird der Prozess beendet und ein neuer Zyklus kann beginnen.

5.1.1 Formteilqualität

Nachdem der Spritzgießprozess vorgestellt wurde, sollen nun Eigenschaften benannt werden, die dazu verwendet werden, um die Qualität von Formteilen festzustellen. Dies ist wichtig, um die Anomalieentscheidung des entwickelten Systems mit dem Endprodukt zu verknüpfen. Auch in diesem Abschnitt wird auf die Arbeit von *Haman* zurückgegriffen und die dort benannten Qualitätsmerkmale, eines im Spritzgussverfahren hergestellten Produktes, hier wiedergegeben [21].

Die erste von *Haman* benannte Qualitätsgruppe ist das *Gewicht* des Formteils, welches in der Nachdruckphase des Spritzgießzyklus bestimmt wird. Die zweite Qualitätsgruppe ist die *Maßhaltigkeit* des Produkt in Bezug auf das Werkzeug. Diese wird dabei durch exogene Faktoren, wie der Eigentemperatur oder Wasserhaltigkeit des eingesetzten Plastikgranulats, aber auch endogene Faktoren, wie der Wärmedehnung des Werkzeugs bestimmt. Die dritte Qualitätsgruppe ist die *Oberflächenstruktur* des Formteils, welche unter anderem in der Einspritzphase des Zyklus entschieden wird. Zur vierten Qualitätsgruppe nach *Haman* gehört die *thermische und stoffliche Mischgüte*, wobei die thermische Mischgüte durch ein gleichmäßiges Temperaturprofil der Schmelze entschieden wird. Die fünfte und letzte Qualitätsgruppe ist die Form beziehungsweise die *Morphologie* des Formteils. Diese wird durch die Intensität der Kühlung beeinflusst, die dazu führen kann, dass sich das Formteil verzieht.

5.2 Datenbeschreibung

Zur Durchführung der Experimente stehen zwei Datensätze zur Verfügung, die der gleichen Spritzgießmaschine entstammen, bei denen jedoch kontrolliert bestimmte Produktionsfaktoren variiert wurden. Dazu gehört zum einen die Feuchtigkeit des zugeführten Kunststoffgranulats sowie zum anderen die Düse und die dazugehörigen Düsendurchmesser, durch welche die Schmelze gespritzt wurde. Die Datensätze wurden freundlicherweise von der Forschergruppe um Prof. Dr.-Ing. Johannes Wortberg vom Institut für Produkt Engineering (Konstruktion und Kunststoffmaschinen) an der Universität Duisburg-Essen zur Verfügung gestellt.

Diese beiden Datensätze sollen im Folgenden näher betrachtet werden und insbesondere die Eigenschaften herausgestellt werden, die zur Durchführung der Vorverarbeitungsschritte von Bedeutung sind.

5.2.1 Datensatz eins: Variation der Granulatfeuchtigkeit

Der erste Datensatz besteht aus etwa 5,2 Millionen Beobachtungen die zu 1154 Prozessen, beziehungsweise Zyklen, gehören. Die Spritzgussmaschine hat dabei nach jeweils fünf vergangenen Millisekunden eine Beobachtung erzeugt. Dies entspricht einer Abtastrate von 200 Hertz. Bei 5,2 Millionen Beobachtungen umfasst der Betrachtungszeitraum somit etwa sieben Stunden. Insgesamt sind in dem Datensatz 250 Merkmale vertreten, von denen 123 kontinuierlich und 127 diskret sind und die jeweils endogene, das heißt innere Einflussfaktoren der Maschine, beschreiben. Exogene Faktoren, also von außen auf die Maschine wirkende Einflüsse, wie die Raumtemperatur, Luftfeuchtigkeit oder aber Materialeigenschaften, sind nicht als Merkmal enthalten. Weiterhin wird jeder Zyklus in den Daten mit einer eindeutigen *Zyklusnummer* versehen, die ebenfalls in jeder Beobachtung enthalten ist. Somit kann für jede Beobachtung entschieden werden, zu welchem Zyklus diese gehört. Jeder Beobachtung ist außerdem ein Zeitstempel zugeordnet.

Um die Zeitreihe eines Zyklus sequenzieren zu können, werden die Triggerattribute benötigt, die die betrachtete Spritzgießmaschine zur Verfügung stellt. Diese sind in der folgenden Tabelle aufgelistet:

Attribut	Beschreibung
T_{CycCnt}	Trigger Zyklusnummer
T_{Clp-}	Trigger Schließeinheit schließen
T_{Clp+}	Trigger Schließeinheit öffnen
T_{Inj}	Trigger Einspritzen
T_{Hld}	Trigger Nachdruck
T_{Plast}	Trigger Plastifizieren
T_{Cool}	Trigger Kühlen
T_{ClpCfb}	Trigger Schließkraftaufbau
T_{ClpCfr}	Trigger Schließkraftabbau
T_{Eje+}	Trigger Auswerfer vor
T_{Eje-}	Trigger Auswerfer zurück
T_{CycRun}	Trigger Zyklus läuft
$T_{Dekomp1}$	Trigger Dekompression 1
$T_{Dekomp2}$	Trigger Dekompression 2

Tabelle 5.1: Triggerattribute der betrachteten Spritzgießmaschine.

Über den Datensatz ist weiterhin bekannt, dass zur jeder vollen Stunde die Feuchtigkeit des eingefüllten Plastikgranulat variiert wurde. Da es grundsätzlich interessant ist das Maschinenverhalten unter Variation eines Produktionsparameters zu untersuchen, beziehungsweise das Änderungsverhalten der mit dem Parameter zusammenhängenden Merkmale (z.B. graduell oder abrupt) zu beobachten, soll die Materialfeuchte als Merkmal erfasst werden. Da diese als exogener Faktor jedoch nicht in den Zeitreihen der Maschine signalisiert wird, muss dieses Merkmal anhand der zu jeder Beobachtung gehörenden Zeitstempel extrahiert werden. Hierzu wird jeweils die Stunde, zu der die erste Beobachtung eines Zyklus aufgezeichnet wurde, ausgewertet und diese Stunde dem gesamten Zyklus zugewiesen. Anhand dieser Zuweisungen können die Zyklen hinsichtlich der Granulatfeuchte gruppiert werden. Zu beachten bleibt jedoch, dass sich gerade zu Beginn einer neuen Stunde, noch Restmaterial anderer Feuchte in der Maschine befinden kann, also zwischen den Stunden nicht scharf getrennt wurde.

5.2.2 Datensatz zwei: Wechsel der Einspritzdüse

Der zweite Datensatz umfasst etwa 4,3 Millionen Beobachtungen und 721 Zyklen. Identisch zum ersten Datensatz ist die Abtastrate von 200Hz, sodass sich ein Beobachtungszeitraum von etwa sechs Stunden ergibt. Die Merkmalsmenge ist ebenfalls identisch zu jener im ersten Datensatz, womit auch die Menge der Triggerattribute gleichbleibend ist und ebenfalls nur endogene Merkmale enthalten sind.

Düse	Eintrittsdurchmesser	Austrittsdurchmesser	Gesamtlänge
0	—	—	—
1	6mm	4mm	80mm
2	5mm	3mm	75mm
3	4mm	2mm	70mm

Tabelle 5.2: Versuchsplanung des zweiten Datensatzes: Variation des Düsendurchmessers.

Dieser Datensatz ist dabei durch Wechsel zwischen mehreren Düsen charakterisiert. Aus der gleichen Motivation, wie in Abschnitt 5.2.1, soll die verwendete Düse als Merkmal extrahiert werden. Dabei werden jedoch nicht die Eigenschaften der Düse selber extrahiert, sondern lediglich der Wechsel zwischen den einzelnen Düsen erfasst. Allen Zyklen zwischen zwei Düsenwechseln wird dann die selbe Düsennummer zugeordnet. Um dies technisch zu bewerkstelligen wird die Annahme getroffen, dass der Düsenwechsel mindestens zwanzig Minuten in Anspruch nimmt und ansonsten unterbrechungsfrei produziert wurde. Dann werden paarweise zwei Zyklen untersucht und die zeitliche Differenz zwischen ihnen be-

rechnet. Liegt diese Differenz bei über zwanzig Minuten, wird ein Düsenwechsel unterstellt. Die einzelnen Düsenparameter können der Tabelle 5.2 entnommen werden.

5.3 Vorverarbeitung

5.3.1 Ungültige Zyklen

Zyklen können offensichtliche Mängel aufweisen, die sich beispielsweise durch fehlende Prozessphasen oder Wechsel im Maschinenbetriebsmodus während des Produktionsprozesses bemerkbar machen. Dies kann dazu führen, dass sich benötigte Merkmale nicht extrahieren lassen und so die weitere Verarbeitung innerhalb der Pipeline gestört wird. Daher sollen einige Kriterien definiert werden, die jeweils alle von validen Zyklen erfüllt sein müssen:

1. *Keine Wechsel des Betriebsmodus während des Zyklus:* Die Maschine kann durch den Bediener manuell, halb-automatisch oder automatisch bedient werden. Im manuellen Modus wird jede Produktionsphase einzeln eingeleitet. Im halb-automatischen, beziehungsweise automatischen Modus, werden die Produktionsphasen jeweils automatisch gesteuert, der Bediener hat jedoch im halb-automatischen Modus die Möglichkeit, einen neuen Zyklus manuell anzustoßen. Ein Wechsel des Betriebsmodus während eines laufenden Zyklus sollte zum Ausschluss der dazugehörigen Daten führen. Um diese Eigenschaft für einen Zyklus festzustellen, kann wie folgt verfahren werden: Stellt die Spritzgießmaschine den Betriebsmodus in Form einer univariaten Zeitreihe zu Verfügung, so ist die Standardabweichung dieser Zeitreihe gemäß Gleichung 3.3 für einen validen Zyklus stets null.
2. *Einspritzphase und Nachdruckphase wird eingeleitet und beendet:* Wie in Abschnitt 5.1 beschrieben, sind diese Produktionsphasen essentiell für den Spritzgießprozess. Ein Fehlen dieser Prozessphasen spricht eindeutig für den anomalen Charakter des Zyklus und führt dazu, dass sich benötigte Merkmale innerhalb der dazugehörigen, sequenzierten Zeitreihen nicht extrahieren lassen. Daher werden Zyklen, die diese Eigenschaft nicht erfüllen, von der weiteren Verarbeitung ausgeschlossen. Um dieses Kriterium umzusetzen, werden die univariaten Zeitreihen untersucht, die die Triggerattribute T_{Inj} und T_{Hld} bereitstellen (Triggerzeitreihen) und jeweils paarweise zwei Beobachtungen \vec{x}_i und \vec{x}_{i+1} betrachtet. In jeder der Triggerzeitreihen müssen nun, für beliebige i und j , Beobachtungspaare existieren, sodass $x_i = \perp$ und $x_{i+1} = \top$ (positive Flanke) sowie $x_j = \top$ und $x_{j+1} = \perp$ (negative Flanke).
3. *Es existiert nur eine Einspritz- und Nachdruckphase:* Da im Spritzgießprozess ein festgelegter Ablauf (der Zyklus) existiert und mehrfache Einspritz- beziehungsweise Nachdruckphasen typischerweise nicht vorgesehen sind, werden Zyklen, die diese

aufweisen als *invalide* aussortiert. Um die Menge der validen Zyklen festzustellen, wird dabei die in Punkt zwei genannte Methode adaptiert. Statt jedoch Paare von positiven und negativen Flanken zu suchen, wird jeweils in den Triggerzeitreihen der Triggerattribute T_{Inj} und T_{Hld} die Anzahl der positiven oder negativen Flanken gezählt. Diese müssen jeweils, für den zu untersuchenden Zyklus, bei genau eins liegen, um dieses Kriterium zu erfüllen.

5.3.2 Sequenzierung

In Abschnitt 5.1 wurde der Spritzgießprozess vorgestellt und insbesondere herausgestellt, dass die Einspritz-, die Kompressions-, und die Nachdruckphase die qualitätsrelevanten Phasen eines Zyklus darstellen. Daher sollen die multivariaten Zeitreihen für jeden Zyklus dementsprechend sequenziert werden.

Für die Einspritz- und Nachdruckphase kann die Sequenzierung unmittelbar vorgenommen werden, da hier für Triggerattribute T_{Inj} und T_{Hld} zur Verfügung stehen. Für die Kompressionsphase ist dies jedoch nicht möglich, da kein geeignetes Triggerattribut zur Verfügung steht. Stattdessen müssen benötigte Merkmale algorithmisch extrahiert werden.

Aus der Menge der Beobachtungen innerhalb der Abkühlphase lässt sich, in Verbindung mit der Abtaste, beispielsweise ableiten, wie lang diese gedauert hat. Daher soll die Zeitreihe auch danach sequenziert werden und hierfür der Trigger T_{Cool} genutzt werden.

Weiterhin soll die Möglichkeit genutzt werden, auch Merkmale über die gesamte, multivariate Zeitreihe Z eines Prozesses zu extrahieren, was vor allem für die Extraktion von Durchschnittswerten, wie zum Beispiel der durchschnittlichen Energieaufnahme im Bereich der Heizelemente, interessant ist.

Die für die weitere Verarbeitung innerhalb der Pipeline verwendete Menge der sequenzierten Zeitreihen Z^* lautet damit wie folgt:

$$Z^* = \{Z_{T_{Inj}}, Z_{T_{Hld}}, Z_{T_{Cool}}, Z\} \quad (5.1)$$

5.3.3 Merkmalsextraktion

Aus Z^* müssen nun die für die Anomalieerkennung relevanten Merkmale extrahiert werden. Wie in Abschnitt 3.3 dargestellt, sollen hierfür zwei Verfahren genutzt werden: Zum einen die Extraktion von vordefinierten statistischen Kennzahlen aus der Zeitreihe und zum anderen die Extraktion von Merkmalen mittels Domänenwissen. Die in den folgenden Abschnitten gewonnenen Merkmale, beziehungsweise die Ausprägungen für einen Zyklus, werden dann zu einem einzelnen Merkmalsvektor zusammengefasst, der den Zyklus beschreibt (s. auch Definition 3.3.1).

Extraktion von statistischen Kennzahlen

Im weiteren Verlauf der Arbeit sollen die Varianz, das arithmetischen Mittel und das Minimum, beziehungsweise Maximum, aus sämtlichen kontinuierlichen Merkmalen in den Zeitreihen aus Z^* extrahiert werden (vgl. auch [21], Kap. 5.2), wobei diese Auswahl an zu extrahierenden Merkmalen völlig unabhängig von Domänenwissen erfolgt ist.

Extraktion unter Verwendung von Domänenwissen

Weiterhin wird vorgeschlagen, alle Merkmalsausprägungen am so genannten *Umschaltzeitpunkt* des Zyklus aus den einzelnen Zeitreihen zu extrahieren. Dieser Umschaltzeitpunkt bezeichnet, ohne auf die technischen Details einzugehen, den Zeitpunkt, auf dem vom Einspritzdruck auf den niedrigeren Nachdruck umgeschaltet wird [21]. Schwankt dieser Umschaltzeitpunkt, wirkt sich dies nachteilig auf die Produktqualität aus [21].

Zur Extraktion sämtlicher Merkmalsausprägungen am Umschaltzeitpunkt wird auf die multivariate Zeitreihe Z eines validen Zyklus zurückgegriffen und der Trigger T_{Hld} genutzt. Dieser signalisiert bei einer positiven Flanke das Umschalten und liefert somit den Umschaltzeitpunkt. Sei i nun der Index, mit dem sich die Ausprägung des Triggerattributs T_{Hld} adressieren lässt, dann ergibt sich die Menge der Umschaltzeitpunkte $Z_{Umschalt}$ wie folgt:

$$Z_{Umschalt} = \{\vec{x}_{j+1} \in Z \mid \vec{x}_j[i] = \perp \wedge \vec{x}_{j+1}[i] = \top\} \quad (5.2)$$

Da ein valider Zyklus unterstellt wurde, ist die Menge $Z_{Umschalt}$ einelementig und der in dieser Menge enthaltene Vektor enthält die Ausprägungen der Merkmale am Umschaltzeitpunkt.

5.3.4 Merkmalsselektion

Nun müssen im Rahmen der Merkmalsselektion geeignete Merkmale aus der gesamten Merkmalsmenge gewählt werden, die innerhalb der Extraktion gewonnen wurde. Dafür soll nun das algorithmische MRMR-Merkmalsselektionsverfahren genutzt werden. In diesem Abschnitt soll diskutiert werden, welche initiale Merkmalsmenge dem Algorithmus präsentiert wird und wie die Qualität der Selektion festgestellt werden kann.

Initiale Merkmalsmenge

Werden wie in Abschnitt 5.3.3 beschrieben, aus allen Zeitreihen in Z^* die statistischen Kennzahlen und der Umschaltzeitpunkt extrahiert, so ergibt sich, bei vier Zeitreihen, 123 kontinuierlichen Basismerkmalen, die die Maschine liefert und fünf zu extrahierenden Merkmalen, eine Merkmalsmenge, die 2460 Merkmale umfasst. Statt dem Algorithmus alle die-

se Merkmale zur Auswahl zu präsentieren, könnte eine angemessene Vorselektion dieser Merkmalsmenge, neben der Reduzierung der Laufzeitkomponente, dazu führen, dass sich die (abstrakte) Selektionsqualität, die noch zu konkretisieren ist, verbessert.

Im weiteren Verlauf der Arbeit wird daher eine mit Domänenwissen unterstützte Vorauswahl aller kontinuierlichen Merkmale, die die betrachtete Spritzgießmaschine liefert, vorgeschlagen. Um dies zu bewerkstelligen werden zunächst die Komponenten der betrachteten Spritzgießmaschine geeignet gruppiert. So werden in diesem Anwendungsfall jeweils alle Merkmale, die mit der Heizung, dem Werkzeug oder der Schnecke assoziiert sind, zu einer Gruppe zusammengefasst. Merkmale, die sich keiner der Gruppen zuordnen lassen, werden bei der Vorauswahl entfernt. Die Label die für das MRMR-Verfahren jeweils benötigt werden, werden dabei idealerweise anhand eines qualitätsrelevanten Merkmals aus der jeweiligen Gruppe bestimmt. Durch dieses Vorgehen werden die Merkmale, die auf eine einzelne Komponentengruppe entfallen, reduziert, während gleichzeitig statt einer Merkmalsselektion, mehrere Merkmalsselektionen berechnet werden müssen. Dieses Schema setzt sich dann, aufgrund des Pipelining, bei der Modellierung fort, sodass auch je Komponentengruppe ein Modell erlernt werden muss.

Eine weitere Option sieht vor, dem MRMR-Verfahren alle vorhandenen Merkmale zu präsentieren. Dabei soll der oben eingeführte Gedanke der Komponentengruppierung hier ebenfalls fortgeführt werden und die Merkmale gruppiert werden, was dann letztlich ebenfalls zu separaten Modellen zur Anomalieerkennung pro Gruppe führt. Die Präsentation der gesamten Merkmalsmenge könnte des Weiteren dabei helfen, neue und unbekannte Zusammenhänge zwischen Merkmalen zu entdecken.

Im Rahmen eines Experimentes muss dann festgestellt werden, welche der beiden hier vorgeschlagenen Methoden, je Datensatz und Komponentengruppe, zu einer besseren Selektionsqualität führt und daher präferiert werden sollte.

Qualität der Merkmalsselektion

Werden algorithmische Verfahren, wie beispielsweise die MRMR-Methode, zur Merkmalsselektion eingesetzt, so sollte die resultierende Merkmalsmenge idealerweise das zur Selektion verwendete Label hinreichend gut beschreiben. Dabei ist es wichtig herauszufinden, ob das ausgewählte Selektionsverfahren überhaupt in der Lage ist eine geeignete Merkmalsmenge zu finden und wie viele Merkmale zur Beschreibung des Labels notwendig sind. Speziell im Falle der MRMR-Methode und dem hier betrachteten Anwendungsfall der Spritzgießmaschine, wird zur Merkmalsselektion das Korrelationsmaß nach Pearson verwendet und damit nur lineare Zusammenhänge betrachtet, womit sich zusätzliche die Frage stellt, ob sich dies mit der Realität deckt.

Wie gut das Label durch eine (ausgewählte) Merkmalsmenge beschrieben wird, kann beispielsweise dadurch festgestellt werden, dass ein passendes Modell aus den verfügbaren Daten gelernt wird und konkrete, numerische Werte des Labels prognostiziert werden (eine *Regressionsanalyse*, eine Methodik des *überwachten Lernens*), um anschließend den Vorhersagefehler zu messen. Wird das MRMR-Verfahren eingesetzt und damit, wie oben beschrieben, nur lineare Zusammenhänge betrachtet, ist es daher denkbar auch eine lineare Regression vorzunehmen, welche wie folgt definiert wird:

Gegeben einem vorherzusagenden Merkmal Y (das Label) und den erklärenden Merkmalen $X^t = (X_1, \dots, X_p)$, dann ist das lineare Regressionsmodell von folgender Form [22]:

$$f(X) = \beta_0 + \sum_{j=1}^p X_j \beta_j \quad (5.3)$$

Die Koeffizienten $\beta = (\beta_0, \beta_1, \dots, \beta_p)$ werden dabei in Abhängigkeit eines gelabelten Trainingsdatensatzes $T = \{(\vec{x}_1, y_1), \dots, (\vec{x}_N, y_N)\}$ mit $\vec{x}_i = (x_{i1}, \dots, x_{ip})^T$ derart bestimmt, dass die so genannte *Residuenquadratsumme*, also die Summe der quadratischen Abweichung von Prognose und tatsächlichen Wert, minimiert wird [22]:

$$RSS(\beta) = \sum_{i=1}^N (y_i - f(x_i))^2 = \sum_{i=1}^N (y_i - \beta_0 - \sum_{j=1}^p x_{ij} \beta_j)^2 \quad (5.4)$$

Wurden diese Koeffizienten bestimmt, muss nun, wie oben beschrieben, eine oder mehrere Methoden gefunden werden, um den Fehler des Modells zu quantifizieren. Dafür soll nun ein absolutes und ein relatives Fehlermaß definiert werden:

5.3.1 Definition (Gütemaße in der Regression). Sei $T = \{(\vec{x}_1, y_1), \dots, (\vec{x}_N, y_N)\}$ ein Trainingsdatensatz und $\hat{Y} = \{\hat{y}_1, \dots, \hat{y}_n\}$ eine Menge von dazugehörigen Prognosen. Dann definiert sich der *mittlere absolute Fehler* (engl. *mean absolute error*, MAE) und die *Wurzel der mittleren quadratischen Abweichung* (engl. *root mean squared error*, RMSE) wie folgt:

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |\hat{y}_i - y_i| \quad (5.5)$$

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)^2} \quad (5.6)$$

Histogramme können dabei die Einordnung der erzielten Fehlerwerte unterstützen, insbesondere dann, wenn die wahren, üblichen Werten unbekannt sind.

Beim Trainings eines Regressionsmodells sind so genannte *Überanpassungssituationen* (engl. *overfitting*) unerwünscht: Diese sind dadurch charakterisiert, dass ein Modell eher das Rau-

schen, beziehungsweise die Varianz, eines Trainingsdatensatzes modelliert, anstelle der zugrundeliegenden Wahrscheinlichkeitsverteilung aus der die Daten stammen [39]. Die Kreuzvalidierung ist dabei eine Technik, die einen Trainingsdatensatz in geeigneter Weise aufteilt und dann jeweils Modelle trainiert und testet, um den *wahren Fehler* zu schätzen, wodurch sich Überanpassungssituation vermeiden oder aber zumindest abschwächen lassen. Die m -fache Kreuzvalidierung wird wie folgt definiert:

5.3.2 Definition (m -fache-Kreuzvalidierung [40]). Sei T eine Beispielmenge und L ein Lernverfahren. Dann partitioniere T in m möglichst gleichgroße Teilmengen T_1, \dots, T_m . Erzeuge nun die Hypothesen h_1, \dots, h_m wie folgt:

$$h_i := \text{Ergebnis von } L \text{ auf Basis der Beispielmenge } T/T_i \quad (5.7)$$

Dann schätze den wahren Fehler von L auf T wie folgt:

$$\text{error}_{CV(T_1, \dots, T_m)}(L, T) := \frac{\sum_{i=1, \dots, m} \text{error}_{T_i}(h_i)}{m} \quad (5.8)$$

Mittels der Durchführung einer linearen Regression und der dazugehörigen Ermittlung des Fehlers in Verbindung mit der Kreuzvalidierung, soll kann dann die Qualität der algorithmischen Merkmalsselektion durch das MRMR-Verfahren ermittelt werden.

5.4 Cluster-Optimierungsproblem

Wie bereits in Abschnitt 4.2 angedeutet, sollte die gefundene Clusterung einige Eigenschaften aufweisen, falls diese zum Zwecke der Anomalieerkennung eingesetzt wird: Zum einen sollten die einzelnen Cluster möglichst kompakt sein, um sicherzustellen, dass die Punkte hinsichtlich des definierten Distanzmaßes ausreichend ähnlich zu einander sind. Zum anderen sollte die Anzahl der gefundenen Cluster hinreichend niedrig sein, da es bei einer hohen Anzahl an Clustern unter Umständen nicht mehr möglich ist, Aussagen hinsichtlich der Anomalie des betrachteten Prozesses zu treffen. Außerdem sollte die Anzahl der Rauschpunkte minimiert werden, um eine Anomalieentscheidung für möglichst viele Zyklen treffen zu können und um zu vermeiden, dass der Optimierer zum Erreichen der anderen Ziele, möglichst viele Vektoren als Rauschen klassifiziert. Des Weiteren sollten die Cluster derart gefunden werden, dass sich Elemente verschiedener Cluster möglichst unähnlich sind (s. Abschnitt 4.2). Es ergeben sich also vier Optimierungskriterien, die *multikriteriell* optimiert werden müssen.

In diesem Abschnitt soll zunächst die für multikriterielle Optimierungsprobleme übliche Notation eingeführt und der Begriff der *Dominanz*, beziehungsweise *Paretooptimalität*, definiert werden. Danach sollen die oben genannten Optimierungskriterien definiert wer-

den, um anschließend das Optimierungsproblem zu formalisieren. Abschließend soll die grundsätzliche Arbeitsweise von evolutionären Optimierern eingeführt und der NSGA-II-Algorithmus, der in diesem Zusammenhang eingesetzt werden kann, vorgestellt werden.

5.4.1 Notation und Begrifflichkeit

Ein multikriterielles Optimierungsproblem wird in seiner Normalform wie folgt notiert [9]:

$$\begin{aligned}
 \min / \max \quad & f_m(\vec{x}) & m = 1, \dots, M \\
 \text{u. d. Nbdg.} \quad & g_j(\vec{x}) \geq 0 & j = 1, \dots, J \\
 & h_k(\vec{x}) = 0 & k = 1, \dots, K \\
 & x_i^{(L)} \leq x_i \leq x_i^{(U)} & i = 1, \dots, n
 \end{aligned} \tag{5.9}$$

Die dritte Nebenbedingung gibt hierbei an, dass die i -te Komponente des Lösungsvektors innerhalb der unteren und oberen Grenze für diesen Komponente liegen muss.

Eine *Lösung* $\vec{x} \in S$ mit $S \subset \mathbb{R}^n$ ist dann ein Vektor von n Parametern der Form $\vec{x} = (x_1, \dots, x_n)^T$, wobei S dem Suchraum entspricht, der durch die oben genannten Nebenbedingungen aufgespannt wird. Ein *Punkt* $\vec{z} \in O$ ist im Sinne der multikriteriellen Optimierung dann ein Vektor von Bildern der Zielfunktionen, also $\vec{z} = (f_1(\vec{x}), \dots, f_M(\vec{x}))^T$, wobei $O \subset \mathbb{R}^M$ dem M -dimensionalen Zielraum entspricht [9]. Auf einer Menge von Lösungen lässt sich dann eine (Halb-)Ordnung definieren, um die optimalen Lösungen zu finden:

5.4.1 Definition (Dominanz einer Lösung [9]). Eine Lösung \vec{x}_1 dominiert eine andere Lösung \vec{x}_2 (auch: $\vec{x}_1 \prec \vec{x}_2$), wenn alle folgenden Bedingungen zutreffen:

1. Die Lösung \vec{x}_1 ist nicht schlechter als \vec{x}_2 hinsichtlich aller definierten Ziele.
2. Die Lösung \vec{x}_1 ist echt besser als \vec{x}_2 in mindestens einem Ziel.

Die *Dominanz* stellt dabei eine stärkere Variante der *schwachen ParetoDominanz* dar, die für zwei Lösungen \vec{x}_1 und \vec{x}_2 nur die erste der beiden Bedingungen fordert und mit $\vec{x}_1 \preceq \vec{x}_2$ notiert wird [43].

Mit dem Begriff der *Dominanz* kann dann für eine Menge von Lösungen, beziehungsweise Punkten, das Konzept der *nicht-dominierten Lösungen* etabliert werden, welche all jene Lösungen sind, die von keinen anderen Lösungen in der Menge dominiert werden. Geht man nun davon aus, dass eine Menge von Punkten existiert, die aus dem *gesamten* Suchraum gebildet wurde, lassen sich damit die Begriffe der *pareto-optimalen Punkte* beziehungsweise *pareto-optimalen Lösungen* definieren [9]. Die *wahren pareto-optimalen Lösungen* sind für den vorliegenden Anwendungsfall jedoch unbekannt, weswegen eine möglichst gute Ap-

proximation an diese Lösungen, die beispielsweise durch einen evolutionären Algorithmus gefunden werden könnten, wünschenswert ist.

Für zwei nicht-dominierte Punkte gilt weiterhin, dass die Verbesserung eines Ziels zu Lasten eines anderen Ziels gehen muss [9], woraus folgt, dass es nicht *das* Optimum gibt.

Des Weiteren besteht die Möglichkeit, die zu den nicht-dominierten, beziehungsweise paretooptimalen Lösungen, gehörigen Punkte zu visualisieren. Dabei wird dann von einer *nicht-dominierten Front* beziehungsweise einer *Paretofront* gesprochen.

5.4.2 Optimierungskriterien

Kompaktheit

Das erste Kriterium, nach dem die Clusterung optimiert werden soll, ist die Kompaktheit der Cluster. Die intuitive Vorstellung eines *kompakten* Clusters involviert die Idee, dass Punkte jenes Clusters *nah* beieinander liegen, also ausgehend vom Zentrum des Clusters minimal streuen. Dies ist von der Wahl des jeweiligen Distanzmaßes abhängig. Im Folgenden wird die unkorrigierte Stichprobenvarianz zur Definition der Streuung eines Clusters herangezogen und hierzu leicht abgewandelt:

5.4.2 Definition (Streuung eines Clusters). Sei $T \subset X^d$ eine Menge von Punkten und $C \in \mathcal{P}(T)$ ein Cluster. Dann ist die Streuung von Punkten im Cluster $\sigma_s^2 : \mathcal{P}(T) \rightarrow \mathbb{R}^+$ unter Verwendung einer Distanzfunktion $dist : X^d \times X^d \rightarrow \mathbb{R}^+$ definiert als

$$\sigma_s^2(C) = \frac{1}{|C|} \sum_{\vec{x} \in C} dist(\vec{x}, \vec{c})^2 \quad (5.10)$$

wobei \vec{c} dem Zentroid des Clusters C entspricht, der definiert ist durch

$$\vec{c} = \frac{1}{|C|} \sum_{\vec{x} \in C} \vec{x} \quad (5.11)$$

Der für die Stichprobenvarianz verwendete Mittelwert der Punktmenge wurde durch den Zentroid, also den geometrischen Mittelpunkt des Clusters, ersetzt. Hierzu sei bemerkt, dass dieser Zentroid bei der Verwendung von zentroidbasierten Clusterverfahren freilich nicht berechnet werden muss, sondern nur als Behelf für nicht-zentroidbasierte Verfahren, wie Support Vector Clustering oder DBSCAN, definiert wurde. Weiterhin wurde die Differenz des jeweils betrachteten Datenpunktes zum Mittelwert durch die verwendete Distanzfunktion ersetzt.

Das Konzept der Streuung *eines* Clusters soll nun auf die Gesamtheit der Cluster in einer Clusterung \mathcal{C} übertragen werden:

5.4.3 Definition (Streuung einer Clusterung). Gegeben sei eine Clusterung $\mathcal{C} = \{C_1, \dots, C_n\} \in \mathcal{P}(\mathcal{P}(X^d))$, wie in Definition 4.2.4 definiert. Die Streuung von \mathcal{C} bezeichnet als σ_f^2 sei definiert durch:

$$\sigma_f^2 : \mathcal{P}(\mathcal{P}(X^d)) \rightarrow \mathbb{R}^+, \sigma_f^2(\mathcal{C}) = \frac{1}{|\mathcal{C}|} \sum_{C \in \mathcal{C}} \sigma_s^2(C) \quad (5.12)$$

Dieses Optimalitätskriterium wurde in ähnlicher Weise beispielsweise auch von Fahad *et al.* aufgeschrieben [15].

Abstand zwischen den Clustern

Eng verbunden mit dem Lernproblem des Clusterings, ist der Wunsch, dass Punkte eines Clusters möglichst ähnlich zueinander sind, während diese zu den Punkten der anderen Cluster möglichst unähnlich sind. Der Begriff der *Ähnlichkeit* lässt sich dabei unter bestimmten Bedingungen auch als Distanz auffassen [40], sodass mit den bekannten Distanzfunktion gearbeitet werden kann. Während die erste der beiden Optimierungskriterien bereits in Definition 5.4.3 formuliert worden ist, muss das Optimalitätskriterium des Abstandes zwischen den einzelnen Clustern noch formuliert werden.

5.4.4 Definition (Abstand zwischen den Clustern [15]). Sei $\mathcal{C} = \{C_1, \dots, C_n\} \in \mathcal{P}(\mathcal{P}(X^d))$ eine Clusterung und $\bar{\mathcal{C}} = \{\bar{c}_1, \dots, \bar{c}_n\}$ die dazugehörige Menge der Zentroide, wobei \bar{c}_n der Zentroid des Clusters C_n ist. Sei weiterhin $\|\cdot\|$ die euklidische Norm. Dann ist der Abstand zwischen allen Clustern in \mathcal{C} , wie folgt definiert:

$$\Delta : \mathcal{P}(\mathcal{P}(X^d)) \rightarrow \mathbb{R}^+, \Delta(\mathcal{C}) = \frac{2}{n^2 - n} \sum_{i=1}^n \sum_{j=i+1}^n \|\bar{c}_i - \bar{c}_j\| \quad (5.13)$$

Der Wert Δ nähert sich dabei für immer näher werdende Cluster dem Wert null an [15].

Anzahl der Cluster

Um das Optimierungskriterium der Anzahl der Cluster in Einklang mit der allgemeinen Form für multikriterielle Optimierungsprobleme zu bringen (s. Formalisierung in 5.9), wird

eine Hilfsfunktion c definiert, die jeder Clusterung $\mathcal{C} \in \mathcal{P}(\mathcal{P}(X^d))$ ihre Anzahl der Cluster zuordnet:

$$c : \mathcal{P}(\mathcal{P}(X^d)) \rightarrow \mathbb{N}_{\geq 0}, c(\mathcal{C}) = |\mathcal{C}| \quad (5.14)$$

Anzahl der Rauschpunkte

Im Verhältnis zur Gesamtanzahl der Punkte, sollten Rauschpunkte nur im geringen Maß auftreten und stattdessen eine hohe Anzahl von Punkten einem Cluster zugeordnet werden. Dies ist notwendig, um für möglichst viele Prozesse eine Aussage bezüglich der Anomalieausprägung treffen zu können. Wie in Definition 4.2.10 werden in dieser Arbeit, auch unabhängig vom DBSCAN-Verfahren, alle Punkte, die keinem Cluster zugeordnet werden, als Rauschen identifiziert, sodass das Optimalitätskriterium wie folgt definiert werden kann:

5.4.5 Definition (Anzahl der Rauschpunkte). Sei $\mathcal{C} = \{C_1, \dots, C_n\} \in \mathcal{P}(\mathcal{P}(X^d))$ eine Clusterung und $n : \mathcal{P}(\mathcal{P}(X^d)) \rightarrow \mathcal{P}(X^d)$, die Funktion, die einer Clusterung die Menge der Rauschpunkte gemäß Definition 4.2.10 zuordnet. Die Anzahl der Rauschpunkte ist dann gegeben durch die Funktion c_N :

$$c_N : \mathcal{P}(\mathcal{P}(X^d)) \rightarrow \mathbb{N}_{\geq 0}, c_N(\mathcal{C}) = |n(\mathcal{C})| \quad (5.15)$$

5.4.3 Definition des Optimierungsproblems

Ausgehend von den definierten Optimierungskriterien kann nun ein Optimierungsproblem für das Finden einer optimalen Clusterung formuliert werden. Allerdings kann, gemäß der in Gleichung 5.9 gegebenen Normalform, eine Clusterung nicht direkt als Lösungsvektor genutzt werden, da Lösungen grundsätzlich reellwertige Vektoren darstellen. Daher wird ein zusätzlicher, abstrakter Clusteroperator $Cl(T, \vec{x})$ eingeführt, der unter Verwendung einer Punktmenge $T \subset X^d$ und eines Parametervektors \vec{x} (der Lösung) eine, gemäß Definition 4.2.4, gültige Clusterung berechnet. Weiterhin wird festgelegt, dass dieser Operator deterministisch arbeitet, also für eine zu clusternde Menge und einen Parametervektor stets die gleiche Lösung liefert. Das Optimierungsproblem lässt sich nun, gegeben einem Suchraum S , wie folgt formulieren:

$$\begin{aligned} \min \quad & \sigma_f^2(Cl(T, \vec{x})), c(Cl(T, \vec{x})), c_n(Cl(T, \vec{x})), -\Delta(Cl(T, \vec{x})) \\ \text{u. d. Nbdg.} \quad & \vec{x} \in S \end{aligned} \quad (5.16)$$

Es sei abschließend bemerkt, dass, gemäß Abschnitt 4.2.1, Rauschpunkte nicht in die Berechnung der Optimalitätskriterien eingehen. Dies betrifft jedoch nicht jene Optimierungskriterien, die Rauschpunkte explizit adressieren, wie beispielsweise die *Anzahl der Rauschpunkte*.

5.4.4 Evolutionäre Optimierung

Um das Problem der multikriteriellen Optimierung zu lösen, soll ein evolutionärer Ansatz genutzt werden. Evolutionäre Algorithmen sind zumeist dadurch charakterisiert, dass diese in einer Optimierungsiteration mehr als eine Lösung vorhalten [9], welche zu Beginn häufig zufällig, anhand unterer und oberer Schranken für den zulässigen Wertebereich erzeugt werden. In jeder Optimierungsiteration wird diese Menge von Lösungen dann evaluiert, das heißt auf ihre (evolutionäre) Fitness untersucht und anschließend durch eine Sequenz von Operationen selektiert und variiert. Je nach Implementierung ist es dabei möglich, dass nur die besten Lösungen in den Optimierungsschritt übernommen werden, was als *Elitismus* bezeichnet wird. Die Optimierung wird abgebrochen, wenn spezifizierte Terminierungskriterien, wie beispielsweise das Erreichen einer vordefinierten Anzahl an Optimierungsschritten, erfüllt sind. Eine solche Prozedur ist in Abbildung 5.2 illustriert.

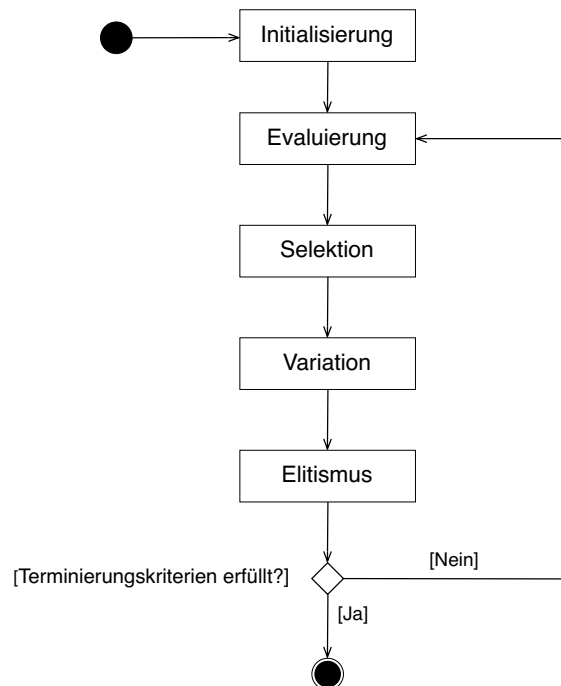


Abbildung 5.2: Eine typische Prozedur zur evolutionären Optimierung (eigene Darstellung, Prozedur nach [9]).

Aufgrund des biologischen Vorbildes dieser Methode, wird die einzelne Lösung als *Individuum*, die Menge von Lösungen in einem Optimierungsschritt als *Population* und der Optimierungsschritt selber als *Generation* bezeichnet. Die *Fitness* eines Individuums bezeichnet dabei die Werteausprägungen der formulierten Optimierungsfunktionen.

Evolutionäre Optimierer bieten den Vorteil, *viele* Lösungen in einem Optimierungsschritt zu probieren, wodurch die Optimierungsschritte einfach parallelisiert werden können. Dies führt auch dazu, dass eine möglichst breite Palette an optimalen Lösungen gefunden wird, was letztlich auch aufgrund der Spezifizierung von mehreren Kriterien notwendig ist [9]. Nachteilig ist jedoch, dass die Optimierung auf heuristischer Basis erfolgt [9], wodurch keine optimalen Lösungen im strengen numerischen Sinne gefunden werden.

5.4.5 NSGA-II

Der *Non Dominated Sorting Genetic Algorithm II*-Ansatz (NSGA-II) von Deb *et al.* [8] bietet im Wesentlichen eine Methode zur Selektion von Individuen und der Übernahme der besten Individuen in die nächste Generation, dem so genannten Elitismus. Die Grundidee des Verfahrens ist dabei, aus der Vereinigungsmenge einer Population P_n und ihren Nachkommen Q_n jeweils der Kardinalität N , im Optimierungsschritt n sukzessive diejenigen Individuen für die Folgepopulation P_{n+1} auswählen, die an den vordersten *nichtdominierten Fronten* liegen. Hierbei wird Wert darauf gelegt, dass ein möglichst breites Spektrum an Punkten gefunden wird, um die Paretofront für einen großen Wertebereich zu finden. Zur Realisation dessen werden zwei Konzepte formuliert: Zum einen das (*Fast*) *Non-Dominated Sorting* und das *Crowding Distance Assignment*, beziehungsweise die *Crowded Comparison*.

(Fast) Non-Dominated Sorting

Gegeben sei eine Population P , also eine Menge von (möglichen) Lösungen für ein spezifiziertes Optimierungsproblem. Das Non-Dominated Sorting hat dann das Ziel, die einzelnen nichtdominierten Fronten zu identifizieren und zu ordnen. Der Algorithmus liefert dann als Resultat eine Menge $F = \{\mathcal{F}_1, \dots, \mathcal{F}_n\} \subset \mathcal{P}(\mathcal{P}(S))$ mit S dem Suchraum, wobei \mathcal{F}_i die Lösungen sind, die zu der i -ten nichtdominierten Front gehören. Da eine Lösung nicht in mehreren nichtdominierten Fronten liegen kann, sind die in F enthaltenen Mengen paarweise disjunkt.

Technisch gesehen werden für jedes $p \in P$ zwei Eigenschaften eingeführt und berechnet: Zum einen eine Anzahl von Lösungen n_p , die p dominieren und eine Menge von Lösungen S_p , die p dominiert. Für jede Lösung der ersten Front \mathcal{F}_1 gilt $n_p = 0$. Für jede Lösung $q \in S_p$ wird anschließend n_q um eins dekrementiert. Fällt dieser Wert auf null, gehört die Lösung q zur zweiten Front und so weiter.

Crowding Distance Assignment

Um ein möglichst breites Spektrum an Lösungen *innerhalb* einer nichtdominierten Front zu erfassen, wird das so genannte *Crowding Distance Assignment* eingeführt. Hierzu wird die Dichte an Lösungen geschätzt, die eine bestimmte Lösung umgeben. Um diese *Crowding Distance* für den Zielvektor \vec{z} eines Individuums aus der Population zu berechnen, werden im Zielraum, an beiden Seiten von \vec{z} Zielvektoren gesucht und die durchschnittliche Distanz für jedes Ziel (das heißt, für jede Dimension) zwischen diesen beiden Punkten berechnet.

Wurden die *Crowding Distances* für eine Population P berechnet, kann auf dieser Population eine Halbordnung \prec_n definiert werden:

$$i \prec_n j \Leftrightarrow (i_{rank} < j_{rank}) \vee ((i_{rank} = j_{rank}) \wedge (i_{distance} > j_{distance})), \quad i, j \in P, i \neq j \quad (5.17)$$

$i_{distance}$ steht hierbei für die *Crowding Distance* und i_{rank} für die Ordnung der nichtdominierten Fronten (1 entspricht der vordersten, 2 der zweit-vorderste und so weiter), auf der das Individuum liegt. Diese Halbordnung besagt, dass jenes Individuum vorgezogen wird, welches, gegeben unterschiedlicher Paretorank, auf der vorderen nicht-dominierten Front liegt. Falls dies nicht der Fall ist, wird jenes Individuum priorisiert, dass eine größere *Crowding Distace* aufweist, also weniger Zielvektoren im Zielraum in der Umgebung besitzt.

Selektion der Individuen

Gegeben ist nun eine Population P_t und eine Nachkommens-Population Q_t , jeweils der Größe N . Diese beiden Menge bilden die Vereinigungsmenge R_t , die anhand des *Non-Domination Sortings* sortiert werden, wodurch sich die oben beschriebene Menge der nichtdominierten Fronten F ergibt. Durch die Vereinigung der beiden Mengen wird sichergestellt, dass die besten Individuen in der nächsten Population P_{t+1} wieder vertreten sind (engl. *elitism*). Es werden dann sukzessive alle Elemente der jeweiligen nicht-dominierten Front \mathcal{F}_i der neuen Population P_{t+1} hinzugefügt, solange $|P_{t+1}| + |\mathcal{F}_i| \leq N$ gilt. Danach werden die Elemente aller nicht-dominierten Fronten in F anhand der oben definierten Halbordnung \prec_n sortiert und die ersten $(N - |P_{t+1}|)$ Elemente der neuen Population P_{t+1} hinzugefügt, wodurch die NSGA-II-Selektion abgeschlossen ist.

5.4.6 Wahl des Individuums

Das Ergebnis einer multikriteriellen Optimierung unter Verwendung des NSGA-II-Verfahrens ist eine Population, also eine Menge von möglichen Lösungen für das spezifizierte Optimierungsproblem, die, je nach Populationsgröße, entweder die gesamte vordere nichtdominierte Front oder aber eine Teilmenge davon enthält. Aufgrund der Tatsache, dass es

sich um das Ergebnis einer multikriteriellen Optimierung handelt, existiert in der Ergebnispopulation keine *einzig*e optimale Lösung. Es ist daher zu klären, welches Individuum letztlich bei dem zugrundeliegende Clusterverfahren zur Anwendung kommt.

Diese Entscheidung soll dabei ausschließlich von den erreichten Fitnesswerten abhängig gemacht werden und mittels diesen eine Auswahlstrategie definiert werden:

1. Da die Clustering zum Treffen einer Anomalieentscheidung genutzt werden soll, ist, gemäß Abschnitt 4.2, eine zu hohe Anzahl von Clustern inakzeptabel. Ebenso ist eine Clustering, die aus lediglich einem Cluster besteht, für den Anwendungsfall nutzlos. Das Intervall, in dem eine akzeptable Anzahl von Clustern liegt, soll daher auf $[2, 5]$ festgelegt werden. Individuen, die dieser Regel nicht genügen, werden aus der Population entfernt.
2. Da möglichst scharf zwischen einer positiven und einer negativen Bewertung eines Zyklus getrennt werden soll, werden nur diejenigen Individuen übernommen, die die geringste Anzahl von Clustern provoziert haben.
3. Unter den verbleibenden Individuen wird dasjenige als Ergebnis der evolutionären Parameteroptimierung zurückgegeben, dass die geringste Anzahl an nicht-zugeordneten Punkten (= Rauschpunkte) aufweist. Dies ist wünschenswert, da wir für möglichst viele Zyklen eine Aussage treffen wollen.

Sollte nach dem Anwenden der Auswahlstrategie mehr als ein Individuum übrig bleiben, ist das Verhalten des Systems undefiniert, das heißt es wird ein beliebiges Individuum ausgewählt, dass der oben genannten Strategie gerecht wird. Falls kein Individuum übrig bleibt, dann sollte das Verfahren, dass die evolutionäre Optimierung eingeleitet hat, eine geeignete Fehlerbehandlungsroutine definieren.

5.4.7 Suchraum des DBSCAN-Algorithmus

Für den DBSCAN-Algorithmus sind, gegeben der euklidischen Distanz als Distanzfunktionen, die Parameter ϵ und $minPts$ zu spezifizieren. Der (abstrakte) Suchraum S_{DBSCAN} , in dem eine optimale DBSCAN-Lösung, gegeben einer Menge von Punkten $T = \{\vec{x}_1, \dots, \vec{x}_n\} \subset \mathbb{R}^d$, gefunden werden soll, ist wie folgt definiert:

$$S_{DBSCAN} = \left\{ (\epsilon, minPts) \mid \epsilon^{(L)} \leq \epsilon \leq \epsilon^{(U)}, minPts^{(L)} \leq minPts \leq minPts^{(U)} \right\} \quad (5.18)$$

Dabei gilt $\epsilon^{(L)}, \epsilon^{(U)} \in \mathbb{R}^+$ und $minPts^{(L)}, minPts^{(U)} \in \mathbb{N}_{>0}$.

Anstelle einer Spezifizierung des Suchraums anhand der Untersuchung der zu clusternden Daten, soll im Folgenden versucht werden, diesen algorithmisch festzulegen.

Es wird nun davon ausgegangen, dass sich die untere und obere Begrenzung von $minPts$ aufgrund der intuitiven Clusterbildung im DBSCAN-Verfahren (s. Abbildung 4.3) und der Tatsache, dass nur natürliche Zahlen als Parameter in Frage kommen, einfacher wählen lässt und daher vom Benutzer vorgegeben werden kann. Dies ist für ϵ schwieriger, da eine Interpretation von Distanzen in höherdimensionalen Räumen erforderlich ist. Im Folgenden soll daher eine algorithmische Unterstützung zur Bestimmung der Suchraumbegrenzung für den ϵ -Parameter vorgestellt werden.

Die untere und obere Begrenzung für ϵ soll nun so gewählt werden, dass im durchschnittlichen Fall für einen Punkt gerade $minPts$ Punkte im Sinne der Distanzfunktion erreichbar sind. Hierzu wird der Algorithmus 5.1 vorgeschlagen, der im Wesentlichen für jeden Punkt die Distanz zu dem $minPts^{(L)}$ -nächsten und dem $minPts^{(U)}$ -nächsten Vektor berechnet und dann ein Mittelwert über alle $minPts^{(L)}$ - beziehungsweise $minPts^{(U)}$ -Distanzen gebildet wird. Diese beiden Mittelwerte ergeben dann die untere und obere Begrenzung des Suchraums für den ϵ -Parameter. Dieses Verfahren kann im Pseudocode wie folgt implementiert werden:

Algorithmus 5.1 Bestimmung des unteren und oberen Begrenzung des ϵ -Parameters im Suchraum des DBSCAN-Algorithmus.

Eingabe: Merkmalsvektorenmenge M , Distanzfunktion $dist$, $minPts^{(L)}$, $minPts^{(U)}$

```

1:  $\forall i, j: Dist[i][j] = \infty$  // Initialisiere Distanzmatrix.
2:  $\forall i: MaxDistancesLowRange[i] = 0$ 
3:  $\forall i: MaxDistancesUpRange[i] = 0$ 
4: for all  $\vec{x}_i \in M$  do
5:   for all  $\vec{x}_j \in M$  do
6:      $Dist[i][j] = dist(i, j)$ 
7:   end for
8: end for
9: for all  $\vec{x}_i \in M$  do
10:   SortedDistances  $\leftarrow$  Sort( $Dist[i]$ ) // Sortiere die Distanzen aufsteigend.
11:   MaxDistancesLowRange[i]  $\leftarrow$  SortedDistances[ $minPts^{(L)}$ ]
12:   MaxDistancesUpRange[i]  $\leftarrow$  SortedDistances[ $minPts^{(U)}$ ]
13: end for
14:  $\epsilon^{(L)} \leftarrow \frac{1}{n} \sum_i MaxDistancesLowRange[i]$ 
15:  $\epsilon^{(U)} \leftarrow \frac{1}{n} \sum_i MaxDistancesUpRange[i]$ 
16: return  $\epsilon^{(L)}, \epsilon^{(U)}$ 

```

Mit dem vom Benutzer zu spezifizierenden Parametern $minPts^{(L)}$ und $minPts^{(U)}$ und dem Ergebnis aus Algorithmus 5.1 ergibt sich ein Suchraum für den DBSCAN-Algorithmus, wie in Menge 5.18 spezifiziert.

5.4.8 Bewertung der Approximation

Nach Abschluss der evolutionären, multikriteriellen Optimierung, wird durch Teile der finalen Population eine Approximation an die *wahre* Paretofront, die *nicht-dominierte Front* (s. Abschnitt 5.4.1) geboten. Werden unabhängig voneinander mehrere nicht-dominierte Fronten gefunden, die beispielsweise durch Nutzung unterschiedlicher Optimierungsverfahren, variierende Parameterwahl oder eine stochastische Komponente zustande gekommen sind, dann sollten Eigenschaften definiert werden, die bessere von schlechteren Approximationen unterscheiden. Diese sollen, neben Metriken, die diese Eigenschaften quantifizieren, im Folgenden vorgestellt werden.

So verweisen *Knowles* und *Corne* auf drei Eigenschaften, die eine Approximation an die Paretofront erfüllen sollte [25]:

1. Die Distanz zwischen der nicht-dominierten Front und der Paretofront sollte minimiert werden.
2. Die Verteilung der Lösungen auf der nicht-dominierten Front sollte möglichst gut (in den meisten Fällen gleichverteilt) sein.
3. Die Ausdehnung der nicht-dominierten Front sollte hinsichtlich aller definierten Ziele maximiert werden.

Während es für die Eigenschaften zwei und drei zumindest denkbar ist, diese ohne Kenntnis der wahren Paretofront zu quantifizieren, ist dies für die erstgenannte Eigenschaft nicht ohne weiteres möglich. Es muss daher geklärt werden, wie sich Verbesserungen oder Verschlechterungen beim Erreichen der ersten Eigenschaft, im Sinne eines relativen Qualitätsmaßes, erfassen lassen.

Im Folgenden sollen nun Metriken präsentiert werden, die die Erfüllung der genannten, drei Eigenschaften für eine nicht-dominierte Front feststellen. Dabei soll keine vollständige, theoretische Analyse der Metriken geboten werden, sondern lediglich ihre Arbeitsweise vorgestellt werden. Für eine tiefere Analyse der Eigenschaften dieser und weiterer Messmethoden sei auf [25, 42, 43] verwiesen.

S-Metrik

Die \mathcal{S} -Metrik nach *Zitzler* wählt zur Bewertung der Approximation den Ansatz, ausgehend von einer Menge an Zielvektoren $A = \{\vec{z}_1, \dots, \vec{x}_n\} \subset O$, die Größe zu messen, die der Raum einnimmt, der von der nicht-dominierten Front dominiert wird [42]. Hierzu wird eine Funktion $\mathcal{S}(A)$ definiert, die das Volumen der Vereinigung aller Polytope $P = \{p_1, \dots, p_n\}$

als Ergebnis hat. Jedes Polytop p_i wird dabei durch die Schnittpunkte der folgenden Hyperebenen bestimmt: Für jede Dimension des Zielraums, wird eine Hyperebene bestimmt, die senkrecht zu dieser Achse ist und durch den Punkt \vec{z}_i führt. Weiterhin wird für jedes Polytop ein Referenzpunkt \vec{z}_{ref} benötigt und für jede Dimension die Hyperebene bestimmt, die senkrecht zu dieser ist und durch diesen Punkt führt. Im zweidimensionalen Fall ergibt sich so für jedes Polytop ein Rechteck, wie in Abbildung 5.3 deutlich wird.

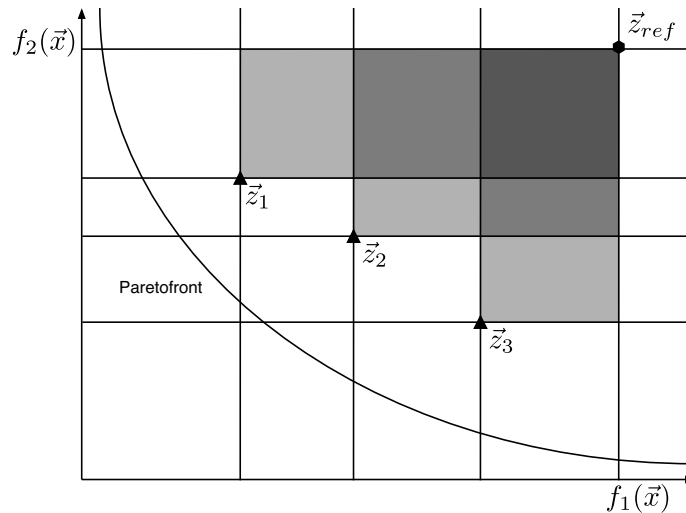


Abbildung 5.3: Beispielfhafte Illustration der Konstruktion der \mathcal{S} -Metrik im zweidimensionalen Zielraum, mit dem Optimierungsziel der Minimierung beider Zielfunktion $f_1(\vec{x})$ und $f_2(\vec{x})$. Die horizontalen und vertikalen Trennlinien visualisieren dabei die Hyperebenen. Die unterschiedlichen gräulichen Schattierung sind für die Berechnung der Metrik irrelevant und unterstreichen lediglich die Zusammensetzung der einzelnen Polytope.

In dieser Arbeit soll die \mathcal{S} -Metrik dabei als relatives Gütemaß interpretiert werden, welches es zu maximieren gilt: So werden nicht-dominierte Fronten präferiert, die eine größere \mathcal{S} -Metrik aufweisen, da hier davon ausgegangen wird, dass diese näher an der *wahren* Paretofront liegen und eine größerer Bereich des betrachteten Raumes dominiert wird. Damit soll sich dann, im Vergleich mit anderen Approximationen, das Erreichen der ersten, oben genannten Eigenschaft messen lassen.

Die Vorteile der \mathcal{S} -Metrik umfassen dabei unter anderem die intuitive Bedeutung des gemessenen Sachverhalts und die Tatsache, dass diese unabhängig von einer Referenzfront, wie beispielsweise der Kenntnis über die wahre Paretofront, operiert [25]. Nachteilig ist, dass man auf die Wahl eines geeigneten Punktes \vec{z}_{ref} angewiesen ist, dieser die Präferenz einer Front gegenüber einer anderen Front beeinflusst und die Wahl des Punktes eher als willkürlich zu beurteilen ist [25]. *Zitzler* verweist weiterhin darauf, dass die Metrik im Falle einer nicht-konvexen Paretofront irreführend sein kann und kommt zu dem Schluss, dass die \mathcal{S} -Metrik nur eine der möglichen Kriterien ist, um die Qualität einer nicht-dominierten Front zu beurteilen.

Messung von Verteilung und Ausdehnung

Eine weitere Möglichkeit zur Messung der Qualität einer nicht-dominierten Front $A = \{\vec{x}_1, \dots, \vec{x}_n\} \subset O$ liegt darin, wie durch das Ziel zwei und drei angeregt, die Verteilung der Punkte auf und die Ausdehnung der Front zu messen. *Zitzler* schlägt hierfür, gegeben einer Norm $\|\cdot\|$, wie der hier genutzten euklidische Norm, und einem *Nachbarschaftsparameter* $\sigma^* > 0$, die Funktionen \mathcal{M}_2^* (Verteilung) und \mathcal{M}_3^* (Ausdehnung) vor [42]:

$$\mathcal{M}_2^* = \frac{1}{|A| - 1} \sum_{\vec{x} \in A} |\{\vec{y} \in A \mid \|\vec{x} - \vec{y}\| > \sigma^*\}| \quad (5.19)$$

$$\mathcal{M}_3^* = \sqrt{\sum_{i=1}^n \max\{\|x_i - y_i\| \mid \vec{x}, \vec{y} \in A\}} \quad (5.20)$$

\mathcal{M}_2^* gibt dabei einen Wert an, der in dem Intervall $[0, |A|]$ liegt und die Anzahl an „ σ^* -Nischen“ wiedergibt, wobei ein höherer Wert bei geeigneter σ^* -Wahl eine bessere Verteilung anzeigt und damit wünschenswerter ist [42]. Die Funktion \mathcal{M}_3^* hingegen verwendet die maximale Ausdehnung der Front hinsichtlich jeder Zieldimension, um die gesamte Ausdehnung zu schätzen [42]. Höhere Werte für die \mathcal{M}_3^* -Metrik werden hier präferiert.

Werden verschiedene Parameterkombinationen miteinander verglichen, die dazu führen, dass die nicht-dominierte Front A durch unterschiedlich viele Punkte definiert wird, dann ist die (absolute) \mathcal{M}_2^* -Metrik nicht geeignet. Es wird daher eine zusätzliche, relative \mathcal{M}_2^* -Metrik eingeführt, die ausnutzt, dass der Wert für \mathcal{M}_2^* in dem Intervall $[0, |A|]$ liegt:

$$\mathcal{M}_2^* (\text{rel}) = \frac{\mathcal{M}_2^*}{|A|} \quad (5.21)$$

Analog zur obigen Beschreibung gilt dann, dass höhere Werte eine bessere Verteilung anzeigen und wünschenswerter sind.

5.5 Random Cluster

Nachdem in den vergangenen Abschnitten die Trainingsdaten festgelegt wurden, die zum Modellieren verwendet werden sollen, und die evolutionäre Optimierung eingeführt wurde, soll in diesem Abschnitt das so genannte *Random Cluster*-Verfahren zur Anomalieerkennung und Modellierung etabliert werden.

Die Idee ist dabei, ausgehend von einer Menge mit Merkmalsvektoren M , die jeweils verschiedenen Produktionsprozessen zugeordnet sind und diese beschreiben, ein Merkmalsselektionsverfahren zu nutzen und so die Dimensionalität der Vektoren zu reduzieren. Anstatt dabei die Dimensionalität auf ein festes k zu reduzieren und so nur *eine* Merkmalsmenge F_k zu betrachten, wird stattdessen ein Intervall $[k^{(L)}, k^{(U)}] \subset \mathbb{N}^+$ betrachtet und die

Dimensionalität auf alle k in dem Intervall reduziert, was letztlich in einer Menge von Merkmalsmengen $F^* = \{F_{k(L)}, \dots, F_{k(U)}\}$ resultiert. Daraus ergibt sich dann statt einem Trainingsdatensatz M_k eine Menge von Trainingsdatensätzen $M^* = \{M_{k(L)}, \dots, M_{k(U)}\}$.

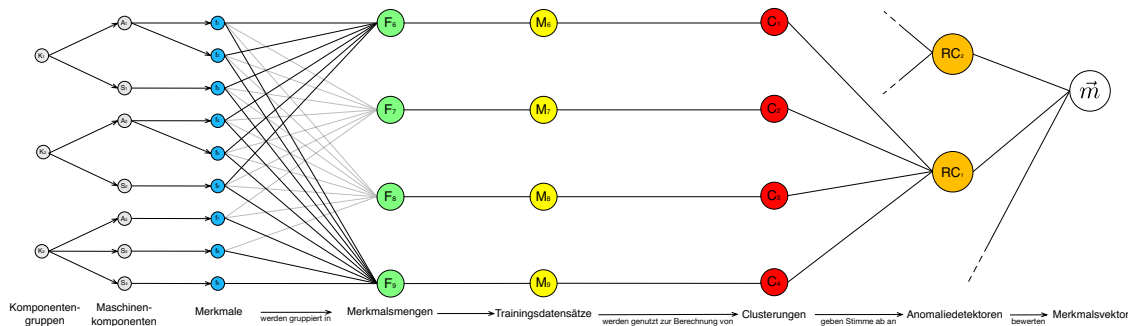


Abbildung 5.4: Visualisierung des *Random Cluster*-Verfahrens in Verbindung mit einer *forward selection*-Merkmalsselektion und der Möglichkeit, mehrere RC-Bewerter einen Merkmalsvektor bewerten zu lassen. Die Maschinenkomponenten (Sensorik und Aktorik), aus denen letztlich die Merkmale resultieren, werden in Komponentengruppen gebündelt.

Für jeden der vorhandenen Datensätze in M^* wird dann eine, nach Abschnitt 5.4 evolutionär optimierte, Clusterung berechnet. Gemäß dem in Abschnitt 4.2 formulierten Gedanken, sollen dabei Merkmalsvektoren, die dem größten Cluster zugeordnet werden, als *normal* und alle anderen Vektoren als *anomal* gelten. Jede Clusterung hat dann die Möglichkeit, ein *Votum* für einen zu bewertenden Prozess hinsichtlich des Anomaliestatus abzugeben. Gibt eine Mehrheit der Clusterungen eine positive Bewertung hinsichtlich des Merkmalsvektors ab, soll dieser Merkmalsvektor mit einem anomalen Prozess assoziiert werden. Falls die gleiche Anzahl von Clusterungen den Merkmalsvektor positiv und negativ bewerten, ist das Verfahren zwischen den beiden Optionen *indifferent*. Ansonsten gilt der Merkmalsvektor und sein Produktionsprozess als normal. Dieses Konzept der Mehrheitsentscheidung existiert in ähnlicher Form auch beim *Random Forest*-Verfahren, welches jedoch üblicherweise überwachte Lernprobleme, wie die Klassifikation oder Regression, löst.

5.5.1 Algorithmus

Das beschriebene Verfahren zur Bildung eines *Random Cluster*-Modells wird dabei in zwei Funktionen implementiert, nämlich `FINDEVERDÄCHTIGEVektoren` und `RANDOMCLUSTER`.

Die Funktion `FINDEVERDÄCHTIGEVektoren` übernimmt dabei die Aufgabe, für ein festes k eine möglichst gute Clusterung anhand einer evolutionären Optimierung (s. Abschnitt 5.4.4) zu bestimmen und die Bewertung sämtlicher eingegebener Vektoren und den dazugehörigen Produktionsprozessen zu übernehmen.

Algorithmus 5.2 RANDOMCLUSTER: Bestimmung verdächtiger Produktionsprozesse.

Eingabe: Merkmalsvektoren $M = \{\vec{m}_1, \dots, \vec{m}_n\} \subset X^d$, Anzahl der Generationen G , Anzahl der Individuen pro Generation μ , $k \in \mathbb{N}^+$, Label y .

```

1: function FINDEVERDÄCHTIGEVektoren( $M, G, \mu, k, y$ )
2:    $M^+ \leftarrow \{\}$  // Initialisiere die Menge der verdächtigen Merkmalsvektoren.
3:    $M_k, F_k \leftarrow$  MERKMALSSELEKTION( $M, k, y$ ) // z.B. MRMR
4:    $P_G \leftarrow$  EVOPARAMETEROPTIMIERUNG( $M_k, G, \mu$ ) //  $G$ -te Population.
5:    $\vec{p}^* \leftarrow$  WÄHLEINDIVIDUUM( $P_G$ ) // nach Abschnitt 5.4.6
6:   if  $\vec{p}^*$  ist undefiniert then
7:     return null, null // Keine verwertbare Clusterung gefunden.
8:   else
9:      $C_k^+ \leftarrow$  FINDEGRÖSSTENCLUSTER( $Cl(M_k, \vec{p}^*)$ )
10:    for all  $\vec{m}_i \in M_k$  do
11:      if  $\vec{m}_i \notin C_k^+$  then
12:         $M^+ \leftarrow M^+ \cup \{\vec{m}_i\}$ 
13:      end if
14:    end for
15:    return  $F_k, M^+$ 
16:  end if
17: end function

```

Der Algorithmus 5.2 führt dabei im Wesentlichen die folgenden Schritte aus: In Zeile 3 wird die Merkmalsselektion durchgeführt, die als Ergebnis die Merkmalsvektoren mit den k selektierten Merkmalen, notiert als M_k , und die selektierte Merkmalsmenge F_k zurückgibt. In Zeile 4 wird die evolutionäre Optimierung nach Abschnitt 5.4 berechnet, die als Ergebnis die Population nach G Operationen liefert. Diese Population enthält dann Lösungen, die als Parameter für das Clustering eingesetzt werden können. Im konkreten Fall des DBSCAN-Verfahrens entspräche dies einer Population von Lösungen, die jeweils den ϵ - und $minPts$ -Parameter definieren (s. auch Abschnitt 5.4.7). In Zeile 5 wird dann ein Individuum beziehungsweise eine Lösung gewählt, mit der dann die Clusterung idealerweise berechnet werden soll. Schlägt diese Auswahl fehl, wird als Ergebnis null zurückgegeben und damit ein Fehler signalisiert. In Zeile 9 wird die Clusterung, mittels des gefundenen Individuums und dem abstrakten Cluster-Operator $Cl(\cdot, \cdot)$ (vgl. Abschnitt 5.4.3), berechnet und dann der größte Cluster, als Teilmenge von M_k , identifiziert. In Zeile 10 - 14 wird dann für jeden Vektor aus M_k entschieden, ob dieser zum größten Cluster gehört oder nicht. Falls nicht, wird dieser positiv bewertet und der Menge der verdächtigen Merkmalsvektoren M^+ hinzugefügt, welche in Zeile 15 als Ergebnis zurückgegeben wird.

Die Funktion RANDOMCLUSTER baut nun auf FINDEVERDÄCHTIGEVektoren auf und aggregiert die abgegebenen Bewertungen beziehungsweise Stimmen, um daraus das Gesamturteil zu bilden.

Algorithmus 5.3 RANDOMCLUSTER: Bestimmung von normalen, anomalen und unentscheidbaren Zyklen.

Eingabe: Merkmalsvektoren $M = \{\vec{m}_1, \dots, \vec{m}_n\} \subset X^d$, Anzahl der Generationen G , Anzahl der Individuen pro Generation μ , $k^{(L)}, k^{(U)} \in \mathbb{N}^+$ mit $k^{(L)} < k^{(U)}$, Label y .

```

1: function RANDOMCLUSTER( $M, G, \mu, k^{(L)}, k^{(U)}, y$ )
2:    $M' \leftarrow$  STANDARDISIERUNG( $M$ ) // nach Abschnitt 3.2
3:   for all  $i \in [1, n]$  do
4:      $V_i^+ \leftarrow \{\}, V_i^- \leftarrow \{\}$  // An positiven oder negativen Entscheidungen beteiligte Merkmalsmengen.
5:   end for
6:   for all  $k \in [k^{(L)}, k^{(U)}]$  do
7:      $F_k, M_k^+ \leftarrow$  FINDEVERDÄCHTIGEVektoren( $M, G, \mu, k, y$ )
8:     if  $F_k$  und  $M_k^+$  nicht null then
9:       for all  $\vec{m}_i \in M'$  do
10:        if  $\vec{m}_i \in M_k^+$  then
11:           $V_i^+ \leftarrow V_i^+ \cup F_k$ 
12:        else
13:           $V_i^- \leftarrow V_i^- \cup F_k$ 
14:        end if
15:      end for
16:    end if
17:  end for
18:   $G \leftarrow \{\}, B \leftarrow \{\}, I \leftarrow \{\}$ 
19:  for all  $\vec{m}_i \in M'$  do
20:    if  $|V_i^+| > |V_i^-|$  then
21:       $B \leftarrow B \cup \{\vec{m}_i\}$  // Die Mehrheit gibt eine positive Bewertung ab.
22:    else if  $|V_i^+| < |V_i^-|$  then
23:       $G \leftarrow G \cup \{\vec{m}_i\}$  // Die Mehrheit gibt eine negative Bewertung ab.
24:    else
25:       $I \leftarrow I \cup \{\vec{m}_i\}$  // Es gibt keine Mehrheit (Indifferenz).
26:    end if
27:  end for
28:  return  $G, B, I$ 
29: end function

```

Dieses Verfahren bietet den Vorteil, viele Merkmalsmengen für das Berechnen von Clusterungen zu verwenden, wodurch die Abhängigkeit von *einer* guten Merkmalsmenge vermieden wird. Nachteilig bei der Verwendung dieses Verfahrens ist der hohe rechnerische Aufwand, der sich im Wesentlichen daraus ergibt, dass für jedes k eine evolutionäre Optimierung durchgeführt wird und folglich auch für jedes Individuum eine Clusterung auf der vollständigen, wenn auch dimensionen-reduzierten, Merkmalsvektorenmenge berechnet werden muss. Dieses Problem könnte dabei umgangen beziehungsweise abgeschwächt werden, indem die Tatsache ausgenutzt wird, dass die Clustermodelle unabhängig von einander gelernt werden. Denn dadurch ergeben sich, zusätzlich zur Nutzung des evolutionären Ansatzes zur Optimierung, weitere Parallelisierungspotenziale.

5.5.2 Gewichtung

Gewichtung eines Merkmals

Wie in Kapitel 2 beschrieben, soll jedes Verfahren, das dazu geeignet ist, den Anomaliestatus eines Merkmalsvektors zu prognostizieren, auch eine Gewichtung der verwendeten Merkmale liefern.

Im *Random Cluster*-Verfahren soll das Gewicht jedes Merkmals anhand seiner Verwendungshäufigkeit bei positiven Entscheidungen bestimmt werden, d.h. dass ein häufig verwendetes Merkmal schwerer wiegt, als ein seltener verwendetes Merkmal. Da in Algorithmus 5.3 ein Votum für oder gegen einen Merkmalsvektor in Form der selektierten Merkmalsmenge registriert wurde, können wir diese unmittelbar zur Berechnung der Merkmalsgewichtung verwenden.

Sei \vec{m}_i nun der zu bewertende Merkmalsvektor und $V_i^+ = \{F_1, \dots, F_k\}$ die Menge der Merkmalsmengen aus Algorithmus 5.3 mit $F_j = \{f_1, f_2, \dots\}$ für alle $j \in [1, k]$, die den Vektor positiv bewertet haben. Dann ergibt sich das, für den Merkmalsvektor \vec{x}_i spezifische, Gewicht eines Merkmals f als W_i^f wie folgt:

$$W_i^f = \frac{|\{F \in V_i^+ \mid f \in F\}|}{|V_i^+|} \quad (5.22)$$

W_i^f ist dann auf das Intervall $[0, 1]$ normiert, wobei $W_i^f = 1$ für einen hohen Einfluss auf die Entscheidungsfindung spricht, während $W_i^f = 0$ keinen Einfluss des Merkmals auf die Anomalieentscheidung anzeigt.

Gewichtung einer Anomalie

Darüber hinaus sollte noch eine Möglichkeit gefunden werden, wie der Schweregrad einer Anomalie bewertet wird, die durch *ein* Random Cluster-Modell ermittelt wurde. Dadurch

soll dem Bediener die Möglichkeit gegeben werden, zu erkennen, wie schwer die von dem jeweiligen Modell gefundene Anomalie wiegt.

In diesem Falle soll eine vom Modell gefundene Anomalie in einem Zyklus schwer wiegen, wenn viele Clustermodelle gegen diesen Zyklus stimmen. Dies lässt sich wie folgt ausdrücken:

Gegeben sei ein zu bewertender Merkmalsvektor \vec{m}_i , die Menge von Merkmalsmengen, die eine positive Stimme abgegeben haben, V_i^+ und die Menge von Merkmalsmengen, die eine negative Stimme abgegeben haben, V_i^- (s. Algorithmus 5.3). Dann wird die Gewichtung des dazugehörigen *Random Cluster*-Modells wie folgt definiert:

$$W_i^+ = \frac{|V_i^+|}{|V_i^+| + |V_i^-|}$$

W_i^+ ist auf den Wert $[0, 1]$ normiert, wobei der Wert eins einen hohen Schweregrad der durch das Modell gefundenen Anomalie ausweist. Der Wert null sagt aus, dass das Modell keine Anomalie gefunden hat und demzufolge auch kein Schweregrad einer Anomalie vorliegt.

5.5.3 Verknüpfung mehrerer Modelle

Die eingeführten *Random Cluster*-Modelle sollen sich, um der in Abschnitt 5.3.4 beschriebenen Merkmalsvorselektion gerecht zu werden, auch im Sinne eines Metamodells verknüpfen lassen (s. Abbildung 5.4). In diesem Falle wird die binäre Information *Anomalie* [ja/nein] jedes einzelnen Detektors verwertet und der Vektor für anomal erklärt, sobald bereits ein einzelner Detektor eine positive Entscheidung abgibt. Das System neigt dann dazu, einen Zyklus für anomal zu erklären und weist dadurch einen tendenziell pessimistischeren Charakter auf.

5.6 Bewertung der Vorhersagegüte

Wurde eine optimale, valide und im Sinne der Anomalieerkennung interpretierbare Clusterung bestimmt oder ein *Single Class SVM*-Modell erlernt und liegen die tatsächlichen Werte in Form von Labeln vor, dann ist es im Sinne einer Selektion der vorliegenden Modelle hilfreich, die Güte der Vorhersage festzustellen. Auch wenn der tatsächlich erhaltene Anomaliebericht deutlich umfangreicher ist, lässt sich dieser in jedem Falle auf die binäre Information *Anomalie* [ja / nein] reduzieren, sodass Gütemaße aus dem Bereich der *Klassifikation*, einem *überwachten* Lernproblem, herangezogen werden können, auch wenn die tatsächliche Klassifikation das Resultat einer Interpretation von Ergebnissen *unüberwachter* Lernmethoden ist. Speziell im Falle einer binären Klassifikation lassen sich dann vier Fälle unterscheiden, die in einer so genannten Konfusionsmatrix aufgetragen werden:

Wahrheit	Vorhersage: Positiv	Vorhersage: Negativ
Positiv	Richtig-positiv (T_P)	Falsch-negativ (F_N)
Negativ	Falsch-positiv (F_P)	Richtig-negativ (T_N)

Tabelle 5.3: Konfusionsmatrix für die binäre Klassifikation [35].

Um die Güte einer Klassifikation messen zu können, stehen verschiedene Messmethoden bereit. Welche letztlich eingesetzt werden, hängt vom Anwendungsfall ab und lässt sich aus den so genannten Invarianzeigenschaften eines Gütemaßes entscheiden [35]. Diese Eigenschaft gibt an, ob sich der Wert eines berechneten Gütemaß gegeben einer Konfusionsmatrix ändert, wenn sich Einträge in eben jener Matrix ändern. Je nach Situation ist dies wünschenswert oder nicht. In der Anomalieerkennung ist es beispielsweise notwendig, gerade die wenigen positiven, also anomalen, Beispiele korrekt zu identifizieren, weswegen ein Gütemaß *nicht* invariant unter einer Änderung von T_P und F_P sein sollte. Ebenfalls sollte ein potenzielles Gütemaß nicht invariant gegenüber dem Austausch von T_P und T_N sein. Diesen Kriterien genügen die *Precision* und der F_1 -Score:

5.6.1 Definition (Gütemaße in der binären Klassifikation). Sei T_P die Anzahl der richtigen-positiv klassifizierten Vektoren, F_P die Anzahl falsch-positiv klassifizierten Vektoren und F_N die Anzahl falsch-negativ klassifizierten Vektoren. Dann definieren sich die Gütemaße *Precision* und F_1 -Score wie folgt:

$$\text{Precision} = \frac{T_P}{T_P + F_P} \quad (5.23)$$

$$F_1 = \frac{2T_P}{2T_P + F_N + F_P} \quad (5.24)$$

Sowohl der F -Score als auch die *Precision* sind auf das Intervall $[0, 1]$ beschränkt, wobei *eins* eine hohe Güte und *null* eine schlechte Güte des betrachteten Modells ausweist.

5.7 Anomaliebericht

Abschließend soll geklärt werden, wie aus einer Menge selektierter Modelle, die für die Anomalieerkennung genutzt wurde, der Anomaliebericht generiert wird, der anschließend dem Bediener präsentiert werden soll. Der Bericht soll dabei, wie in der Zielsetzung bereits beschrieben, dem Bediener bei seiner Arbeit assistieren und hierzu gefundene Anomalien nachvollziehbar und detailliert mit Verweis auf Maschinenkomponenten und Prozessphasen erklären. Neben der technischen Realisierung, wie sie beispielsweise in Kapitel 2 abstrakt und in Abschnitt 5.5.2 konkret besprochen wurde, müssen hierbei auch Aspekte der Interaktion des Systems mit dem Bediener beleuchtet werden (Mensch-Maschine-Interaktion).

Mensch-Maschine-Interaktion

Der Spritzgießprozess ist aufgrund seiner kurzen Einzelprozessdauern von zwanzig bis dreißig Sekunden insbesondere dadurch charakterisiert, dass der Bediener einen angezeigten Anomaliebericht schnell erfassen muss. Dies impliziert im Hinblick auf den Bericht Eigenschaften wie Kompaktheit und Intuitivität, die sich unter dem Begriff der *Mensch-Maschine-Interaktion* bündeln lassen.

Dabei sollen einige Fragen in diesem Zusammenhang exemplarisch beleuchtet werden: So können je nach verwendetem Modell einer Anomalieentscheidung eine relativ hohe Anzahl von Merkmalen zugrunde liegen, die nicht alle dem Bediener präsentiert werden sollten. Dies könnte aus MMI-Perspektive zum einen Überfluss an Informationen führen, der letztlich darauf hinausläuft, dass der Anomaliebericht vollständig ignoriert wird. Stattdessen sollte der Fokus auf der Präsentation der wichtigsten Merkmale und der dazugehörigen Maschinenkomponenten liegen. Hierzu wird die, vom Modell vorgegebene, Gewichtung der Merkmale verwendet und die n wichtigsten Merkmale, in den Bericht einbezogen. Der Parameter n muss in diesem Falle empirisch so bestimmt werden, dass die beschriebene Situation des Informationsüberflusses nicht auftritt, aber der Bediener immer noch ausführlich genug über die Anomalieentscheidung des Systems unterrichtet wird.

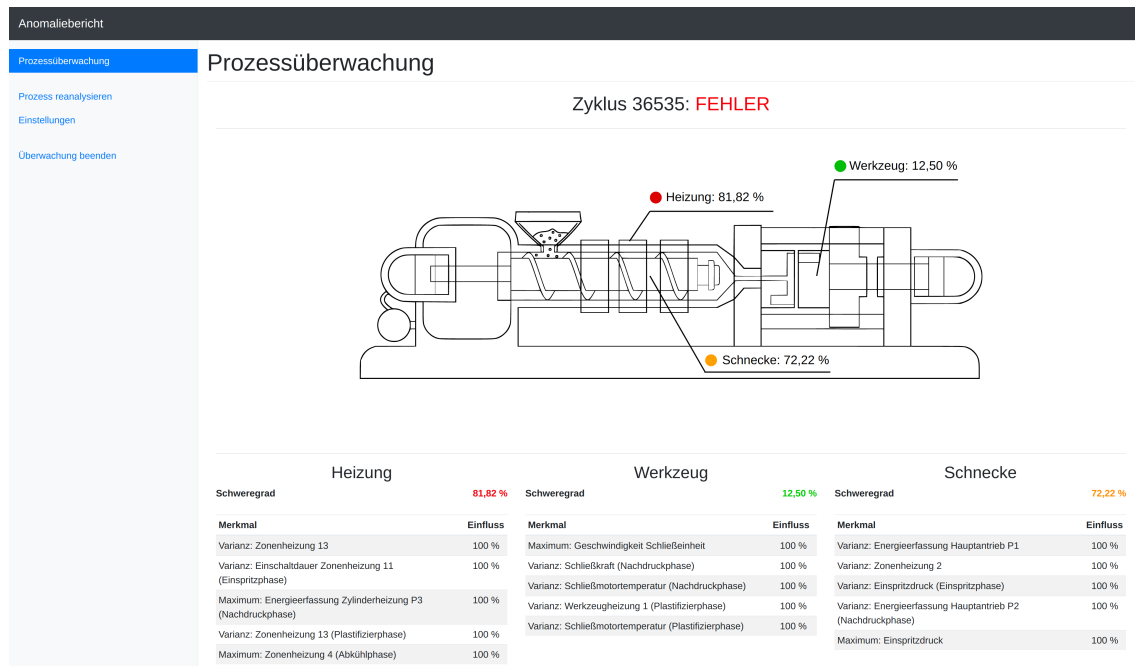


Abbildung 5.5: Konzept einer Schnittstelle zwischen Anomaliedetektor und dem Maschinenbediener in Form eines Anomalieberichts.

Die Gewichtung kann dabei, neben der reinen Filterfunktion, auch dazu verwendet werden, um besonders wichtige Merkmale entsprechend im Bericht hervorzuheben. Denkbar ist

daher, die Gewichte der, bei der Anomalieentscheidung beteiligten, Merkmale ebenfalls in den Anomaliebericht zu übernehmen, wobei eine angemessene Sortierung im Zuge einer intuitiven Darstellung notwendig ist.

Weiterhin wird vorgeschlagen, die oben beschriebene Komponentengruppen und den dazugehörigen Anomalieschweregrad dazu zu nutzen, diese visuell anhand einer schematischen Zeichnung einer Spritzgießmaschine darzulegen und so die Informationsaufnahme zu unterstützen (s. Abbildung 5.5).

Kapitel 6

Experimente

6.1 Testumgebung

Der größte Teil der Experimente wurde auf Basis der Skriptsprache Python, dem *Machine Learning*-Pakets `scikit-learn`¹ und dem EA-Toolkit DEAP [16] entwickelt und durchgeführt, wobei die Python-Version 3.6.1 eingesetzt wurde. Zur Berechnung der MRMR-Merkmalss Selektion wurde die Software *RapidMiner*² in der Version 5.0 zusammen mit dem von *Schowe* entwickelten *Feature Selection Plugin* verwendet (s. auch [33]). Zur Berechnung der \mathcal{S} -Metrik wurde die Implementation genutzt, die durch das Paket PyGMO/PaGMO³ bereitgestellt wird.

Der zur Durchführung der Experimente verwendete Computer besitzt dabei zwei Hauptprozessoren des Typs *Intel(R) Xeon(R) CPU E5-2697 v2 @ 2.70GHz* und 226 Gigabyte Hauptspeicher. Das verwendete Betriebssystem ist Ubuntu 14.04.5 LTS mit einem Linux-Kernel der Version 4.4.16.

6.2 Qualität der Merkmalsselektion

In diesem Abschnitt soll die Qualität der Merkmalsselektion, wie in Abschnitt 5.3.4 beschrieben, festgestellt werden. Hierzu wird zunächst der Aufbau des Experiments im Sinne der spezifische Parameterwahl benannt und anschließend die erzielten Ergebnisse hinsichtlich der einzelnen Komponentengruppen besprochen.

¹<http://scikit-learn.org/stable/> (zuletzt abgerufen am 30.08.2017)

²<https://github.com/rapidminer/rapidminer-5> (zuletzt abgerufen am 30.08.2017)

³<https://github.com/esa/pagmo2>, (zuletzt abgerufen am 30.08.2017)

6.2.1 Aufbau

Wie in Abschnitt 5.3.4 dargelegt, wird eine lineare Regressionsanalyse unter Verwendung des MRMR-Verfahrens eingesetzt, um die Eignung der ausgewählten Merkmale zu Beschreibung des Labels festzustellen. Zur Wahl des Labels wird zwischen den einzelnen Komponentengruppen der Spritzgießmaschine (in diesem Falle: Heizung, Schnecke und Werkzeug) differenziert. Als Label wurden dabei möglichst jene Messgrößen gewählt, die einen Einfluss auf die Qualität des Formteils haben können und dabei ausreichend Varianz aufweisen.

Komponentengruppe	Label (y)
Schnecke	Maximaler Einspritzdruck (über Z)
Werkzeug	Werkzeuginnendruck am Umschaltpunkt
Heizung	Durchschnittlich erfasste Energie in der Zylinderheizung (Phase 2, über Z)

Tabelle 6.1: Wahl des Labels in Abhängigkeit der betrachteten Komponentengruppe.

Für jedes $k \in [3, 40]$, dass die Anzahl einbezogener Merkmale im MRMR-Verfahren parametrisiert, wurde dann eine lineare Regression durchgeführt und der durchschnittliche RMSE und MAE im Rahmen einer zehnfachen ($m = 10$) Kreuzvalidierung gemessen. Dieses Experiment wurde dabei unabhängig von einander auf Basis des Datensatzes eins (im Folgenden *Feuchtigkeitsdatensatz* oder *Feuchtigkeit*) und zwei (im Folgenden *Düsensdatensatz* oder *Düse*) durchgeführt. Zur Merkmalsselektion standen dem MRMR-Verfahren jeweils alle Merkmale zur Verfügung, die nach Abschnitt 5.3.3 (Merkmalsextraktion, statistische Kennzahlen) in Verbindung mit Abschnitt 5.3.2 (Sequenzierung) extrahiert wurden. Die Experimente werden dabei jeweils bei Nutzung und Vernachlässigung einer domänenwissenbasierten Vorselektion durchgeführt. Des Weiteren wurden die beschreibenden Merkmale nach Abschnitt 3.2 standardisiert. Merkmale, die ausschließlich konstante Werte aufweisen, wurden vorher aus dem Datensatz entfernt, da diese keine für die Anomalieerkennung verwertbaren Informationen enthalten.

6.2.2 Komponentengruppe „Schnecke“

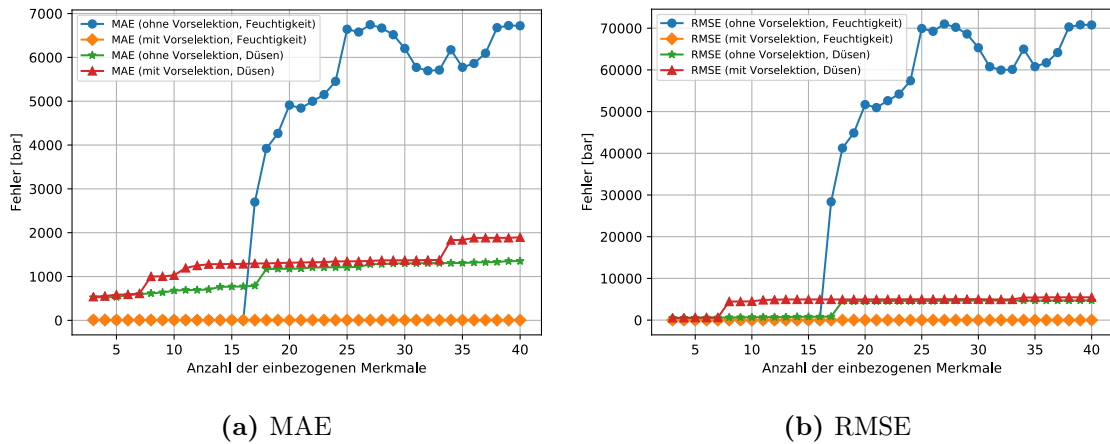


Abbildung 6.1: MAE und RMSE bei einer kreuzvalidierten, linearen Regression mit Fokus auf die Komponentengruppe „Schnecke“ und Differenzierung nach domänenbasierten Vorselektion und verwendetem Datensatz.

Betrachtet man zunächst die Fehlerwerte des Düsendatensatzes, so fällt auf, dass die erreichten MAE-Werte, unabhängig von der Anzahl der einbezogenen Merkmale und der Tatsache, ob eine Vorselektion durchgeführt wurde, inakzeptabel hoch sind. So wird im besten Falle, der bei $k = 3$ und ohne Vorselektion eintritt, ein MAE von etwa 525 bar erreicht, wobei üblicherweise Maximaldrücke im Bereich von etwa 400 bis 600 bar erreicht werden (s. Abbildung 6.2). Die Fehlerkurven für MAE und RMSE weisen dabei einen monoton steigenden Charakter auf, womit auch alle weiteren betrachteten Merkmalskardinalitäten in diesem Datensatz als inakzeptabel zu betrachten sind.

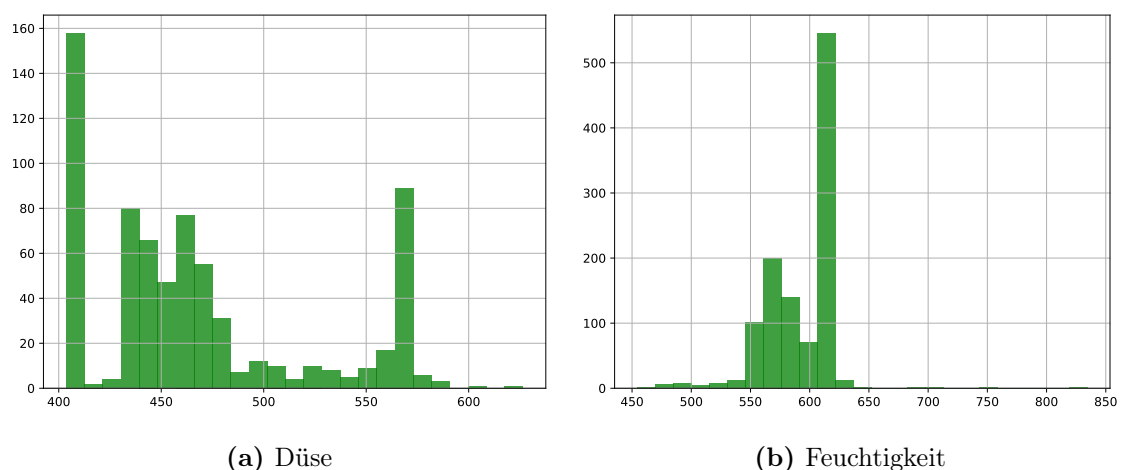
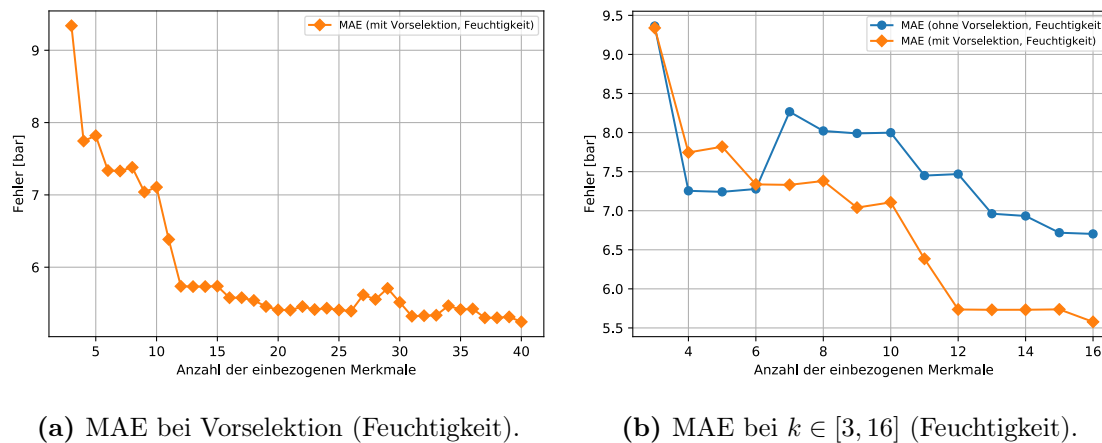


Abbildung 6.2: Histogramm der maximalen Einspritzdrücke (über Z) mit einer Klassengröße von 25.

Bei Betrachtung des Feuchtigkeitsdatensatzes ist zunächst auffällig, dass sich bei nicht-vorgenommener Vorselektion nach einem, im Verhältnis, relativ fehlerarmen Intervall bei $k \in [17, 40]$ sowohl der MAE als auch RMSE sprunghaft verschlechtert. Die in diesen Intervallen zusätzlich aufgenommenen Merkmale scheinen die Modellierung also negativ zu beeinflussen und die Prognosen in einen inakzeptablen Bereich zu rücken. Der Plot suggeriert dabei, dass im Intervall $k \in [3, 16]$ bessere Werte zu erwarten sind. Um dies zu beurteilen, soll die Darstellung auf diesen Bereich eingegrenzt werden.



(a) MAE bei Vorselektion (Feuchtigkeit).

(b) MAE bei $k \in [3, 16]$ (Feuchtigkeit).

Abbildung 6.3: Nähere Betrachtung von Fehlerkurven in der Komponentengruppe „Schnecke“.

Betrachtet man nun das Intervall $k \in [3, 16]$ näher, so sind unter Vernachlässigung der Vorselektion grundsätzlich akzeptable Fehlerwerte feststellbar: Dabei liegt der beste MAE-Wert bei $k = 16$ mit etwa 6,7 bar, während der schlechteste MAE-Wert in diesem Intervall bei $k = 3$ und etwa 9,36 bar erreicht wird. Die Spannweite zwischen dem besten und dem schlechtesten Wert liegt somit bei etwa 2,66 bar. Während grundsätzlich die bessere Konfiguration zu bevorzugen ist, zeigt dies dennoch, dass auch mit wenigen Merkmalen das Label angemessen beschrieben und eine akzeptable Prognose erzielt werden kann.

Anschließend sollen die erzielten MAE-Werte bei Nutzung einer domänenbasierten Vorselektion beleuchtet werden: Dabei ist bei Steigerung der einbezogenen Merkmale grundsätzlich ein fallender Trend hinsichtlich des MAE wahrnehmbar. Der beste MAE-Wert wird dort bei $k = 40$ mit etwa 5,2 bar erreicht, während der schlechteste Wert bei $k = 3$ und einem MAE-Wert von etwa 9,36 bar erreicht wird. Die Spannweite liegt hier bei etwa 4,14 bar, wobei zum Erreichen des besten Wertes bereits 40 Merkmale einbezogen werden müssen.

Für die Komponentengruppe „Schnecke“ lassen sich so zwei sehr unterschiedliche Ergebnisse festhalten: So wurden bei Nutzung des Feuchtigkeitsdatensatzes akzeptable Ergebnisse erzielt, bei denen das gelernte lineare Modell, bei angemessener Parameterwahl, offenbar dazu geeignet ist, die Zielgröße, in diesem Falle den maximalen Einspritzdruck über die ge-

samte Zeitreihe (Z), angemessen zu prognostizieren. Ambitioniertere Modellierungen konnten dabei unter Verwendung einer domänenwissenbasierten Vorselektion die Fehlerwerte in der Regel weiter reduzieren. In diesem Falle scheint die Fehlerrate durch das Einbeziehen weiterer Merkmale nicht negativ beeinflusst zu werden. Für diesen Datensatz scheint die MRMR-Methode also eine angemessene Auswahlstrategie zu bieten. Anders verhalten sich die erzielten Prognosequalitäten bei Nutzung des Düsendatensatzes, bei denen ausschließlich inakzeptable Fehlerwerte erzielt wurden. Hier scheint eine lineare Modellierung der Zusammenhänge und die Nutzung des MRMR-Verfahrens ungeeignet zu sein.

6.2.3 Komponentengruppe „Werkzeug“

Als nächstes werden die Ergebnisse in der Komponentengruppe „Werkzeug“ betrachtet:

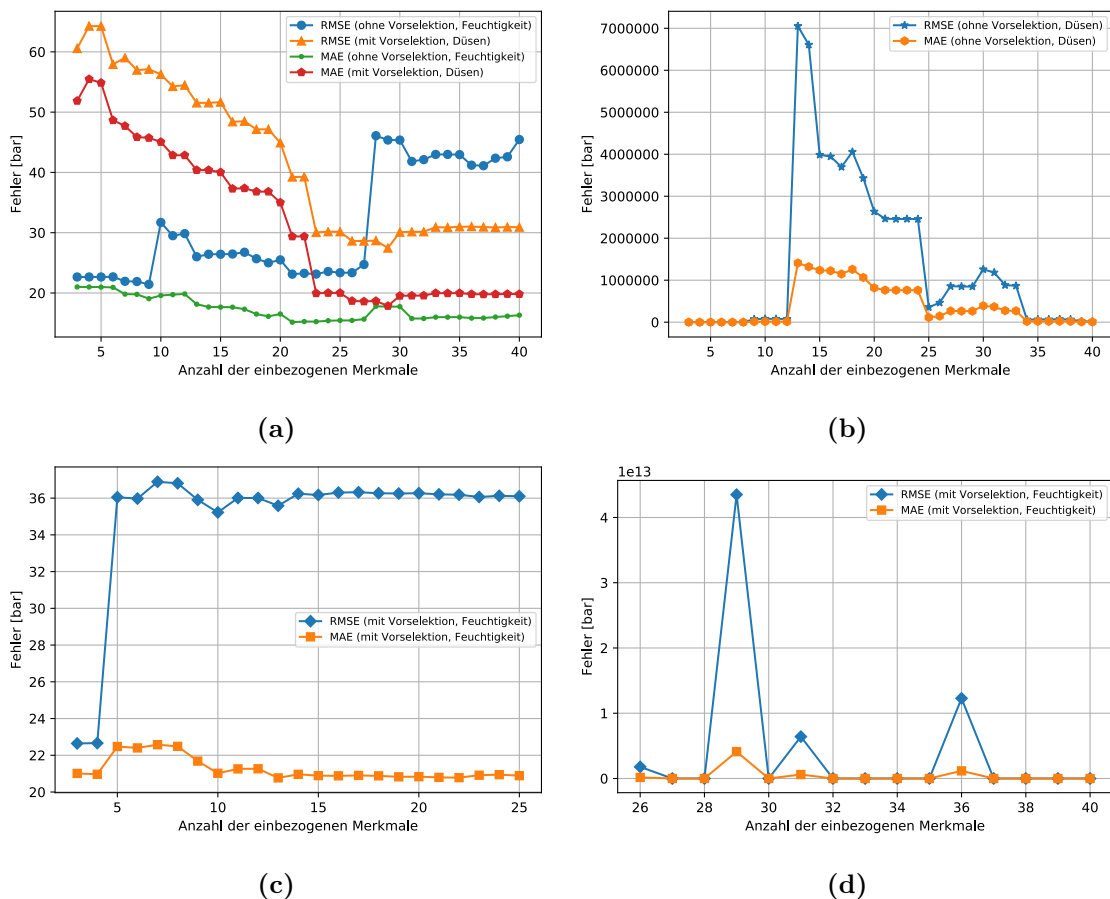


Abbildung 6.4: Plots der erzielten Fehlerwerte (MAE/RMSE) bei Fokussierung auf die Komponentengruppe „Werkzeug“, differenziert nach Feuchtigkeits- und Düsendatensatz.

Nun sollen zunächst wieder die Fehlerwerte des Düsendatensatzes betrachtet werden: Dabei ist auffällig, dass durch die Vorselektion, der erzielte MAE- und RMSE-Wert jeweils gesenkt wird. So liegt der beste erzielte MAE-Wert ohne die Vorselektion bei etwa 86,19 bar ($k =$

3), während der beste MAE-Wert unter Nutzung der Vorselektion bei etwa 17,85 bar ($k = 29$) erreicht wurde. Damit ergibt sich eine MAE-Verbesserung von etwa 68,34 bar, was, angesichts der üblicherweise erreichten Werte im Düsensatz (s. Abbildung 6.5a) als signifikant einzuschätzen ist.

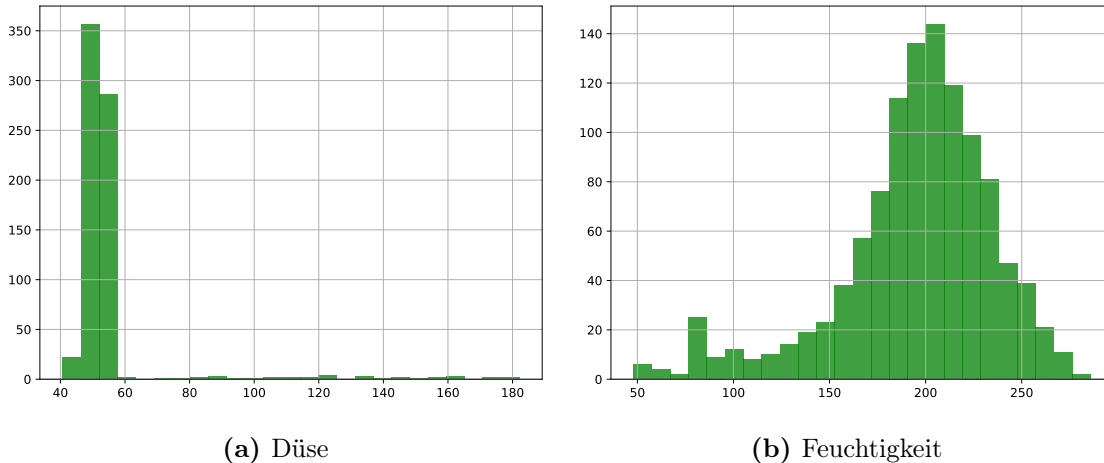


Abbildung 6.5: Histogramm der Werkzeuginnendrucke am Umschaltplatz mit einer Klassengröße von 25.

Es ist nun jedoch schwierig zu beurteilen, ob dieser beste MAE-Wert, für den vorliegenden Anwendungsfall als akzeptabel einzuschätzen ist. So werden im Düsensatz üblicherweise Werkzeuginnendrucke von etwa 50 bis 60 bar erreicht. Wird nun ein MAE-Wert von 17,85 bar erreicht, so muss durch einen Experten eingeschätzt werden, ob dies akzeptabel ist, da die Situation nicht derart eindeutig, wie bei der Komponentengruppe „Schnecke“, ist. Grundsätzlich scheint das lineare Modell und damit auch die MRMR-Merkmalss Selektion für die Komponentengruppe „Werkzeug“ im Düsensatz jedoch nicht völlig ungeeignet zu sein.

Bei dem Feuchtigkeitsdatensatz ergibt sich nun die Situation, dass durch die Vorselektion die Prognosequalität verschlechtert wird: So liegt der beste erreichte MAE-Wert unter Nutzung der Vorselektion bei etwa 20,15 bar ($k = 33$), während der beste MAE-Wert bei Vernachlässigung der Vorselektion bei etwa 15,16 bar ($k = 21$) zu verorten ist. Die Differenz liegt somit bei etwa 4,99 bar, wobei die Verbesserung angesichts der üblicherweise erreichten Innendrucke von 150 bis 250 bar als weniger bedeutsam einzuschätzen ist (s. Abbildung 6.5b). Die Anzahl einbezogener Merkmale beeinflusst die Vorhersagequalität dabei nur schwach: So liegt die Standardabweichung der MAE-Werte unter Nutzung der Vorselektion und Vernachlässigung der Extremwerte bei $k \in \{26, 29, 31, 36\}$ bei etwa $\sigma = 0.65$ bar. Die Standardabweichung der MAE-Werte ohne die Vorselektion liegt bei $\sigma = 1.93$ bar. Grundsätzlich waren im Feuchtigkeitsdatensatz akzeptable Fehlerwerte erreichbar, womit

das lineare Modell und somit auch das MRMR-Verfahren zur Merkmalsselektion zulässig erscheinen.

6.2.4 Komponentengruppe „Heizung“

Abschließend werden die erreichten Ergebnissen in der Komponentengruppe „Heizung“ betrachtet:

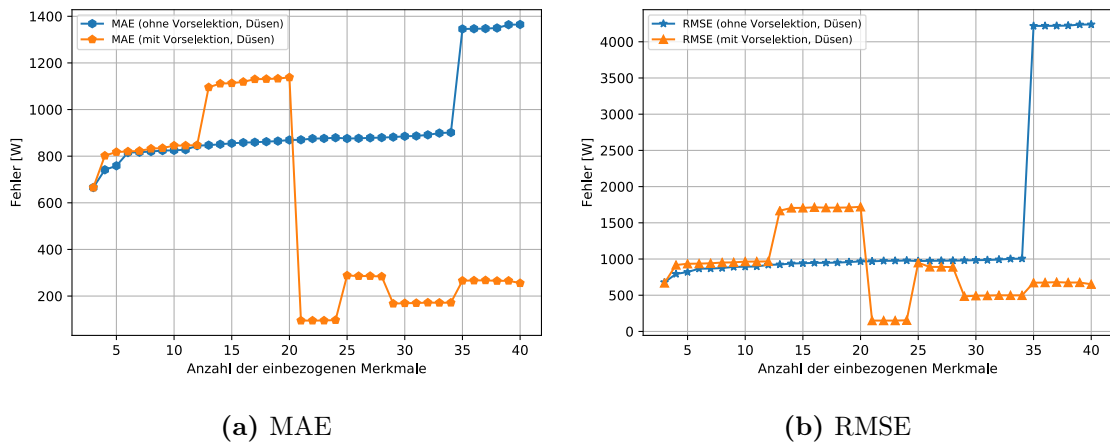


Abbildung 6.6: MAE und RMSE bei einer kreuzvalidierten, linearen Regression mit Fokus auf die Komponentengruppe „Heizung“ bei Verwendung des Düsendatensatzes und Differenzierung nach domänenbasierten Vorselektion.

Zunächst soll wieder mit der Bewertung der Ergebnisse im Düsendatensatz begonnen werden: So weisen die MAE- und RMSE-Fehlerwertkurven unter Vernachlässigung der Vorselektion einen monoton steigenden Charakter auf, wobei Fehlerwerte erreicht wurden, die als inakzeptabel zu beurteilen sind. Hier lag der MAE-Wert bestenfalls bei 656,55 Watt, während üblicherweise Leistungen im Bereich von 500 bis 700 Watt vorliegen. Durch die Nutzung einer Vorselektion konnten hier bessere Fehlerwerte erreicht werden: Denn dadurch werden nach dem Intervall $k \in [3, 20]$, in welchem das Monotonieverhalten und die erreichten Fehlerwerte ähnlich zu denen im vorselektionslosen Fall sind und die Fehler teilweise sogar schlechter wurden, deutlich bessere Werte erreicht: So wird bei $k = 21$ der für diesen Datensatz und Komponentengruppe beste MAE-Wert erreicht, der bei 94,76 Watt liegt. Dies ist bei den üblicherweise erreichten Werten eher als akzeptabel zu beurteilen.

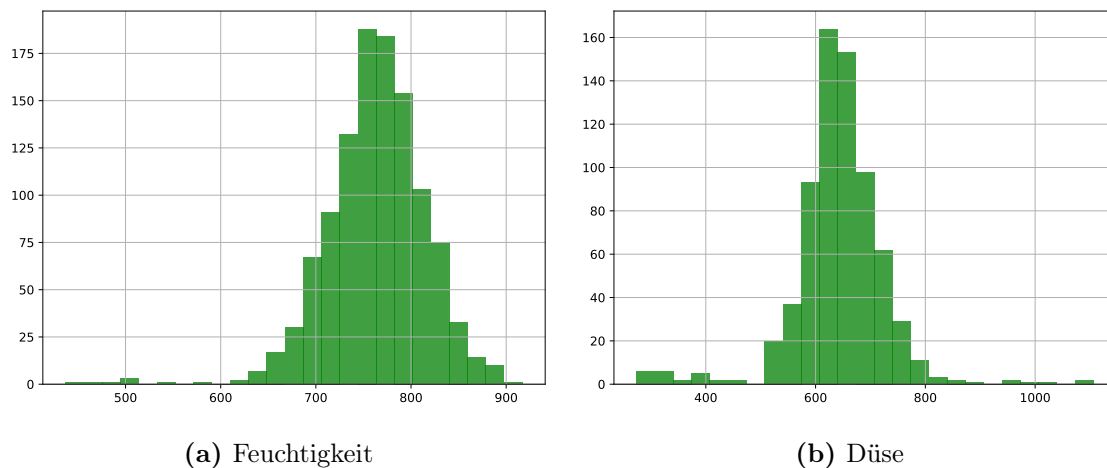


Abbildung 6.7: Histogramm der durchschnittlich erfassten Energie in der Zylinderheizung (Phase 2, über Z) mit einer Klassengröße von 25.

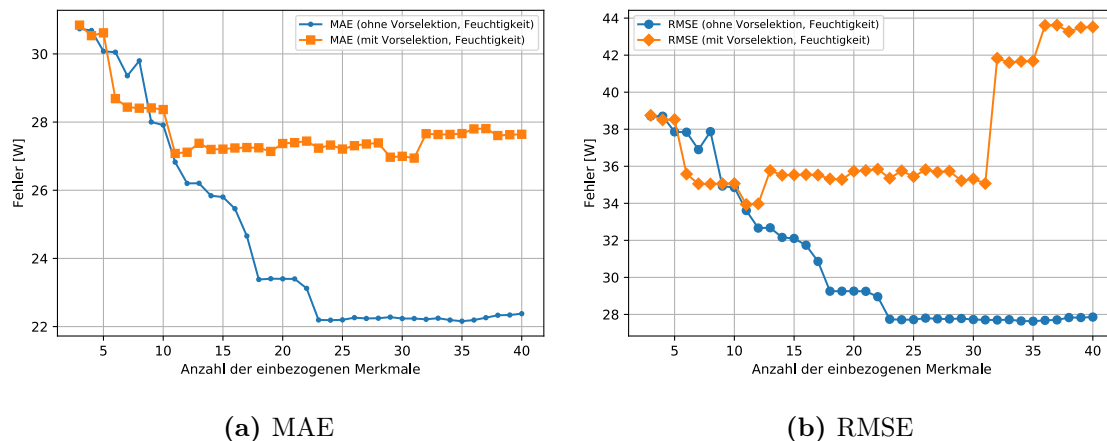


Abbildung 6.8: MAE und RMSE bei einer kreuzvalidierten, linearen Regression mit Fokus auf die Komponentengruppe „Heizung“ bei Verwendung des Feuchtigkeitsdatensatzes und Differenzierung nach domänenbasierten Vorselektion.

Beim Feuchtigkeitsdatensatz lassen sich für diese Komponentengruppe grundsätzlich akzeptable Fehlerwerte feststellen: So wurde im schlechtesten Falle, der bei $k = 3$ ohne Nutzung einer Vorselektion eintrat, ein MAE-Wert von etwa 30,73 Watt festgestellt der im Verhältnis zu einer üblichen erfassten Energie von 700 bis 850 Watt steht. Die Nutzung einer Vorselektion verbessert die Vorhersagequalität dabei nur bedingt: Zwar wurden bei Betrachtung des Intervalls $k \in [3, 7]$ bessere Fehlerwerte gegenüber der ungefilterten Merkmalsmenge erreicht, jedoch konnte die Prognosequalität nur geringfügig gesteigert werden. So liegt die Spannweite zwischen dem besten MAE-Wert unter Verwendung und Vernachlässigung der Vorselektion bei etwa 4,79 Watt, was vermutlich als weniger bedeutsam einzuschätzen ist.

6.2.5 Zwischenfazit

Zum Abschluss des Experiments soll ein Zwischenfazit gezogen werden: Dabei lässt sich zunächst keine eindeutige Präferenz hinsichtlich der domänenwissenbasierten Vorselektion der Merkmalsmenge feststellen. Während die Vorselektion bei der Komponentengruppe „Schnecke“ im Feuchtigkeitsdatensatz die Prognosequalität bedeutsam verbesserte, verschlechterte die Vorauswahl im Düsendatensatz die Vorhersagequalität sogar. Ähnliche Beobachtungen konnten bei den anderen betrachteten Komponentengruppen gemacht werden. Als Fazit bleibt aufgrund der Uneindeutigkeit der Ergebnisse daher festzustellen, dass, bei Verwendung der gleichen Methoden, für jeden weiteren, zu untersuchenden Datensatz eine derartige Untersuchung durchgeführt werden muss, um zu beurteilen, ob eine Vorselektion nützlich ist.

Eine etwas homogenere Situation ergibt sich bei Betrachtung der erreichten Prognosequalitäten insbesondere im Vergleich mit den üblicherweise erreichten *wahren* Werten: Bei Betrachtung der Komponentengruppe „Schnecke“ und des Feuchtigkeitsdatensatzes waren zwar prinzipiell akzeptable Fehlerwerte und damit akzeptable Prognosen möglich, während dies unter Verwendung des Düsendatensatz nicht erreicht werden konnte. Hier ist es möglich, dass sich die Zusammenhänge zwischen den Merkmalen bei den unterschiedlichen Datensätzen in einen nicht-linearen Bereich verschoben haben, weswegen die Selektion und die Modellierung mittels eines linearen Ansatzes nicht mehr zweckmäßig wurde. In den Komponentengruppen „Werkzeug“ und „Heizung“ konnten hingegen durchweg akzeptable Prognosequalitäten erzielt werden, was darauf hindeutet, dass das lineare Modell und damit die MRMR-Merkmalss Selektion grundsätzlich geeignet ist.

6.3 Qualität der Approximation an die Paretofront

Wie in Abschnitt 5.4.8 dargelegt, soll nun in einem Experiment die Qualität der Approximation an die Paretofront gemessen werden. Hierzu wird der Aufbau des Experimentes beschrieben und anschließend die erzielten Ergebnisse besprochen.

6.3.1 Aufbau

Nun soll die Qualität der Approximation an die wahre, wenn auch unbekannte Paretofront gemessen werden. Analog zu den vergangenen Abschnitten werden für die einzelnen Komponentengruppen *Schnecke*, *Werkzeug*, und *Heizung* unabhängig voneinander Modelle gelernt und evolutionär optimiert. Zur Merkmalsselektion wird das MRMR-Verfahren eingesetzt und die initiale Merkmalsmenge anhand der Ergebnisse des vergangenen Ex-

periments festgelegt, wobei jene Konfiguration bevorzugt wurde, bei denen sich bessere MAE-Werte messen ließen:

	Schnecke	Werkzeug	Heizung
Düsensatzensatz	Ungefiltert	Vorselektion	Vorselektion
Feuchtigkeitsdatensatz	Vorselektion	Ungefiltert	Ungefiltert

Tabelle 6.2: Dem MRMR-Verfahren zu Verfügung stehende Merkmalsmengen bei der Messung der Qualität der Approximation an die Paretofront, differenziert nach Komponentengruppe und Datensatz.

Die evolutionäre Parameteroptimierung wird für den DBSCAN-Algorithmus unter Verwendung des NSGA2-Verfahrens durchgeführt. Die Suchraumbeschränkung für den $minPts$ -Parameter wurde für alle durchzuführenden Versuche auf $minPts^{(L)} = 3$ und $minPts^{(U)} = 10$ festgelegt. Die Suchraumbeschränkung für den ϵ -Parameter wurde mit Algorithmus 5.1 ermittelt.

Zur Mutation der Individuen in jeder Generation wurde die DEAP-Implementierung der *polynomielle Mutation* eingesetzt (siehe z.B. [11]). Für das *Crossover* zweier Individuen wurde die DEAP-Implementierung des *Simulated Binary Crossover* (SBX) eingesetzt (s. [10]). Die Wahrscheinlichkeit für die Mutation und das Crossover wurde, wo notwendig, auf 90% festgelegt.

Um herauszufinden, wie sehr sich die in Abschnitt 5.4.8 vorgestellten Qualitätsmaße in Abhängigkeit der Parameterwahl verändern, werden für eine Komponentengruppe und feste $k^{(L)}, k^{(U)}$ jeweils die Parameter im Bereich $\mu = \{100, 200, 300, 400\}$ und $G = \{10, 20, \dots, 100\}$ variiert.

Mit den Ergebnissen aus Abschnitt 6.2 muss nun festgelegt werden, auf welches Intervall $[k^{(L)}, k^{(U)}]$ die jeweiligen Trainingsdatensätze reduziert werden sollen. Die hier gewählte Strategie sieht nun vor, ausgehend von den, nach Komponentengruppen und Datensätzen, erreichten Fehlerwerten, dieses Intervall möglichst so zu wählen, sodass dieses, in Verhältnis zu den anderen erreichten Fehlerwerten, möglichst fehlerarme Regressionen provoziert und dabei gleichzeitig die Breite des Intervalls maximiert wird. So sollen möglichst viele gute Modelle erzeugt und die Anzahl der abgegebenen Stimmen erhöht werden. Das Ergebnis dieser Auswahlstrategie findet sich in Tabelle 6.3.

	Schnecke	Werkzeug	Heizung
Düsensatzensatz	$k \in [3, 17]$	$k \in [23, 40]$	$k \in [21, 40]$
Feuchtigkeitsdatensatz	$k \in [12, 26]$	$k \in [13, 27]$	$k \in [23, 40]$

Tabelle 6.3: Konfiguration der Merkmalsselektion bei der Messung der Qualität der Approximation an die Paretofront, differenziert nach Komponentengruppe und Datensatz.

Der Referenzpunkt, der für die \mathcal{S} -Metrik benötigt wird, soll nun wie folgt bestimmt werden: Sei $\vec{z}_{ref} = (z_1, \dots, z_n) \in O^n$ der zu bestimmende Referenzpunkt im Zielraum und $O^* = \bigcup_{i=1}^n O_i$ die Vereinigung aller Punkte, die die zu vergleichenden nicht-dominierten Fronten $\{O_1, \dots, O_n\}$ im Zielraum definieren, dann bestimmen sich die einzelnen Komponenten des Referenzpunktes wie folgt:

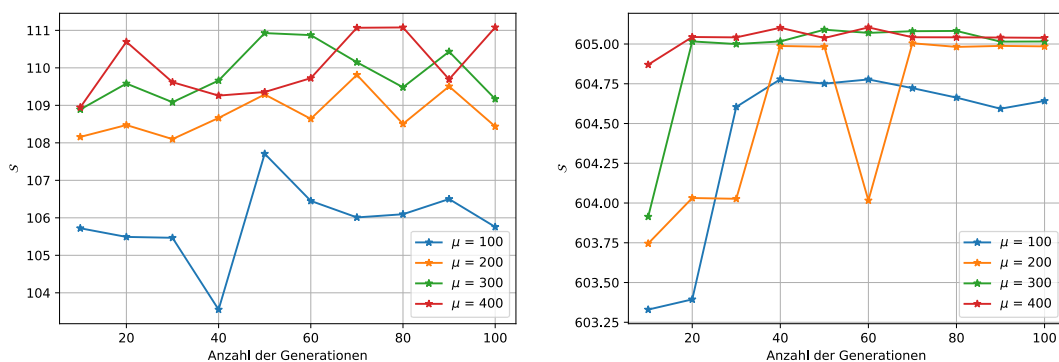
$$z_i = \max_{\forall \vec{x} \in O^*} \vec{x}[i] \quad (6.1)$$

Dabei sollen im Folgenden alle nicht-dominierten Fronten miteinander verglichen werden, die für ein festes k , unabhängig von den parametrisierten μ und G erreicht wurden.

Der σ^* -Parameter, der für die Berechnung der (relativen) \mathcal{M}_2^* -Metrik benötigt wird, wurde standardmäßig auf $\sigma^* = 30$ gesetzt und variiert, sofern ein relativer Vergleich zwischen den Wertereihen nicht mehr möglich ist.

Korreakterweise müssten für eine vollumfängliche Dokumentation der erzielten Ergebnisse jeder Datensatz, jede Komponentengruppe und alle k nach Tabelle 6.3 die jeweils erzielten Messwerte der Metriken besprochen werden. An dieser Stelle soll darauf jedoch verzichtet werden und ausgewählte Ergebnisse der jeweiligen Gruppen exemplarisch besprochen werden. Des Weiteren wird hier ein eher qualitativer Vergleich der erzielten Wertereihen durchgeführt, um festzustellen, in welche Richtung und mit welcher Stärke sich die jeweiligen Gütemaße bei gezielter Variation der Parameter zur evolutionären Optimierung verändern. Eine Analyse der jeweils konkret erzielten Werte soll also nicht durchgeführt werden.

6.3.2 Komponentengruppe „Schnecke“



(a) Düsendatensatz ($k = 3$)

(b) Feuchtigkeitsdatensatz ($k = 12$)

Abbildung 6.9: \mathcal{S} -Metrik in der Komponentengruppe „Schnecke“ im Feuchtigkeits- und Düsen-datensatz.

In Abbildung 6.9 ist die Entwicklung der \mathcal{S} -Metrik bei verschiedenen Werten für μ und G hinsichtlich der Komponentengruppe „Schnecke“ unter Verwendung der beiden Datensätze zu sehen.

Betrachtet man die Werteverläufe im Düsendatensatz so lässt sich rein qualitativ die Beobachtung machen, dass höhere Werte für den μ -Parameter die \mathcal{S} -Metrik überwiegend positiv beeinflussen. Dabei nimmt jedoch die Differenz zu den gemessenen Werten mit dem nächsthöheren μ -Parameter ab: Während sich für $\mu = 100$ und $\mu = 200$ die Wertereihen noch klar voneinander abgrenzen lassen, ist dies bei $\mu = 300$ und $\mu = 400$ nicht mehr in dieser Klarheit möglich. Betrachtet man nun die Entwicklung der Wertereihe in Richtung der berechneten Generation, so lässt sich, mit Ausnahme der Wertereihe für $\mu = 100$ nur eine schwache Varianz feststellen, was darauf schließen lässt, dass für den Düsendatensatz, $k = 3$ und die Komponentengruppe „Schnecke“ eher die Anzahl der Individuen pro Generation (μ) für die Qualität der Approximation entscheidend ist, anstatt der Anzahl berechneter Generationen G . Auffällig ist weiterhin, dass bei $\mu = 300$ und $\mu = 400$ selbst bei einer geringen Anzahl von Generationen, im Vergleich zu den anderen Wertereihen, bereits gute Werte erreicht werden, die bei $\mu = 100$ und $\mu = 200$ selbst nach der Berechnung mehrerer Generationen nicht erreicht werden.

Bei Betrachtung der \mathcal{S} -Metrik für den Feuchtigkeitsdatensatz ergibt sich ein etwas anderes Bild, wobei grundsätzlich ähnliche Aussagen getroffen werden können: So verbesserten sich mit Erhöhung des μ -Parameters die erreichten Werte für die \mathcal{S} -Metrik, während sich die Berechnung zusätzlicher Generationen am ehesten bei Wahl von $\mu = 100$ und $\mu = 200$ zu lohnen scheint. Insbesondere bei $\mu = 200$ wurden mit der Berechnung zusätzlicher Generationen Werte erreicht, die in der Nähe der besten, gemessenen Werte liegen. Ebenfalls ähnlich zum Düsendatensatz ist die Feststellung, dass für $\mu = 300$ und $\mu = 400$ bei einer geringeren Anzahl von berechneten Generationen, bessere Werte für die \mathcal{S} -Metrik erreicht werden konnten, als bei den anderen Konfigurationen.

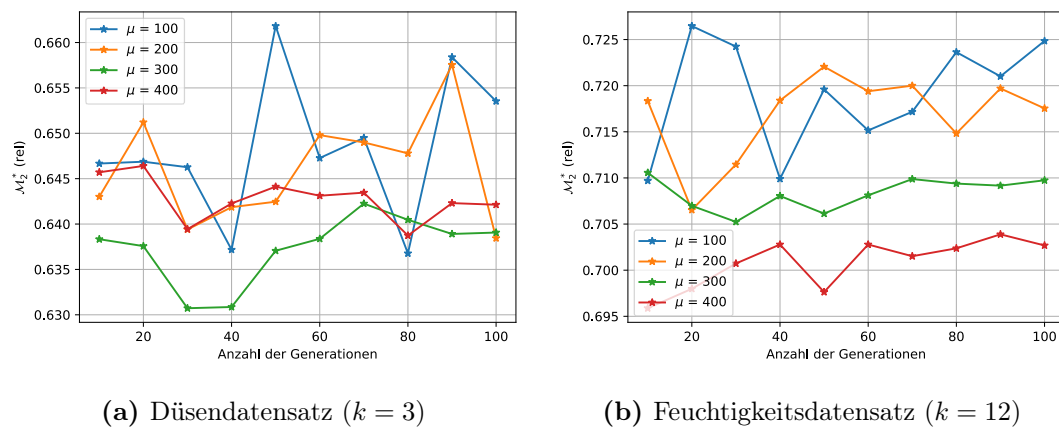
(a) Düsendatensatz ($k = 3$)(b) Feuchtigkeitsdatensatz ($k = 12$)

Abbildung 6.10: Relative \mathcal{M}_2^* -Metrik in der Komponentengruppe „Schnecke“ im Feuchtigkeits- und Düsendatensatz ($\sigma^* = 30$).

Betrachtet werden nun die, in Abbildung 6.10 dargestellten, erzielten Werte bei der relativen \mathcal{M}_2^* -Metrik in der Komponentengruppe „Schnecke“ bei Nutzung des Düsendatensatzes. Hier lässt sich für $\mu \in \{100, 200\}$ bei zusätzlicher Berechnung von weiteren Generationen kein klarer Trend zur Verbesserung oder Verschlechterung feststellen. Dabei ließ sich für diese μ -Parameter jedoch eine bessere, relative \mathcal{M}_2^* -Metrik erreichen, als für $\mu \in \{300, 400\}$. Unter Verwendung von $\mu = 400$ scheint sich die relative \mathcal{M}_2^* -Metrik bei Berechnung weiterer Generationen zu verschlechtern und weist einen schwach fallenden Trend auf, während $\mu = 300$ von der Berechnung zusätzlicher Generationen hinsichtlich der Metrik zu profitieren scheint.

Im Feuchtigkeitsdatensatz sind für $\mu \in \{100, 200\}$ und die relative \mathcal{M}_2^* -Metrik bessere Werte gegenüber $\mu \in \{300, 400\}$ festzustellen. Dabei scheint die Berechnung zusätzlicher Generationen die Werte für diese Metrik zu verbessern. Ein positiver Trend ist auch bei $\mu = 300$ und $\mu = 400$ zu beobachten, wobei diese Wertereihen eine schwächere Veränderung hinsichtlich der betrachteten Werte bei der Berechnung weiterer Generationen aufweisen.

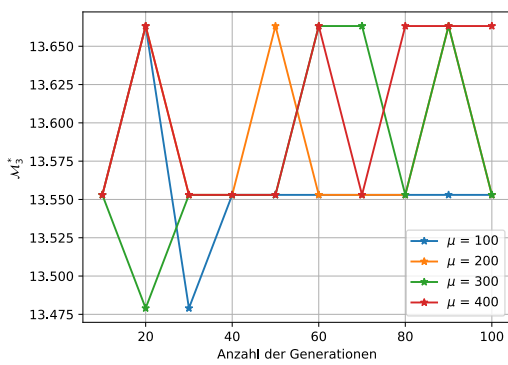
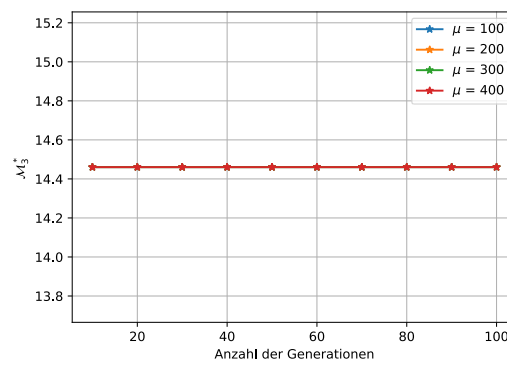
(a) Düsendatensatz ($k = 3$)(b) Feuchtigkeitsdatensatz ($k = 12$)

Abbildung 6.11: \mathcal{M}_3^* -Metrik in der Komponentengruppe „Schnecke“ im Feuchtigkeits- und Düsendatensatz.

Abschließend werden die für die \mathcal{M}_3^* -Metrik erreichten Werte betrachtet: Dabei ergibt sich im Düsendatensatz ein eher diffuses Bild der einzelnen Wertereihen. So lässt sich bei qualitativer Betrachtung der erreichten Werte bei Benutzung des Düsendatensatzes keine eindeutige Präferenz für eine bestimmte Konfiguration aus μ - und G -Parametern feststellen. Eine quantitative Betrachtung der erreichten Werte lässt, soweit dies möglich ist, angesichts der Achsenskalierung, vermuten, dass die einzelnen Konfigurationen nur einen geringen Einfluss auf die Breite der nicht-dominierten Front nehmen.

Bei Betrachtung der Werte für die \mathcal{M}_3^* -Metrik im Feuchtigkeitsdatensatz lässt sich zwischen den einzelnen Konfigurationen kein Unterschied feststellen.

6.3.3 Komponentengruppe „Werkzeug“

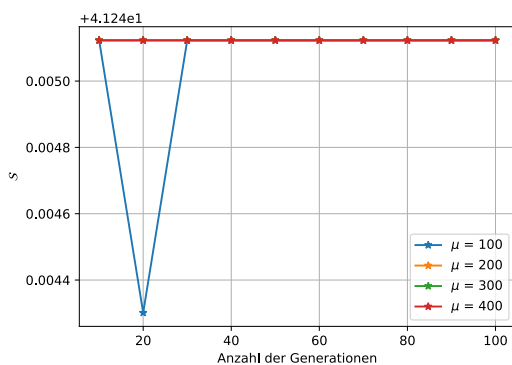
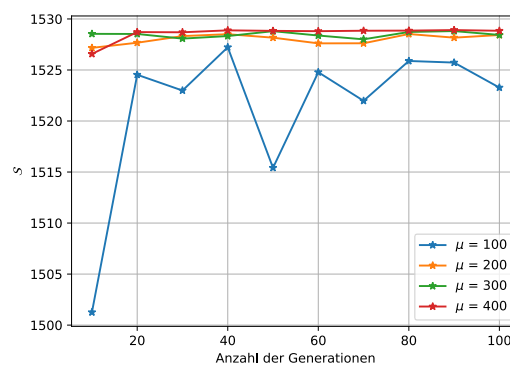
(a) Düsendatensatz ($k = 23$)(b) Feuchtigkeitsdatensatz ($k = 23$)

Abbildung 6.12: \mathcal{S} -Metrik in der Komponentengruppe „Werkzeug“ im Feuchtigkeits- und Düsendatensatz.

Betrachtet werden nun die in Abbildung 6.12 aufgetragenen Werte für die \mathcal{S} -Metrik in der Komponentengruppe „Werkzeug“. So wurden im Düsendatensatz überwiegend qualitativ ähnliche Werte für die \mathcal{S} -Metrik erreicht, wobei sich durch die Variation der Anzahl an Generationen oder die Anzahl an Individuen pro Generation keine größeren Veränderungen feststellen lassen.

Im Feuchtigkeitsdatensatz ergibt sich für $\mu \in \{200, 300, 400\}$ eine relativ unauffällige Situation in Hinblick auf die erreichten Werte und den Werteverlauf der \mathcal{S} -Metrik. Hier scheint bereits ab $\mu = 200$ die Situation erreicht, in dem eine zusätzliche Aufnahme von Individuen in die finale Population, den nicht-dominierten Bereich nicht weiter ausdehnt und sich die \mathcal{S} -Metrik folglich nicht verbessert. Eine zusätzliche Berechnung von Generationen scheint die \mathcal{S} -Metrik nicht über die heuristische Komponente der evolutionären Optimierung hinaus zu beeinflussen. Eine etwas andere Situation ergibt sich bei Betrachtung der Wertereihe für $\mu = 100$: Diese ist stets schlechter als die üblichen Wertereihen. Eine Berechnung von zusätzlichen Generationen scheint die Werte für die \mathcal{S} -Metrik jedoch in Richtung der gemessenen, besten Werten zu rücken.

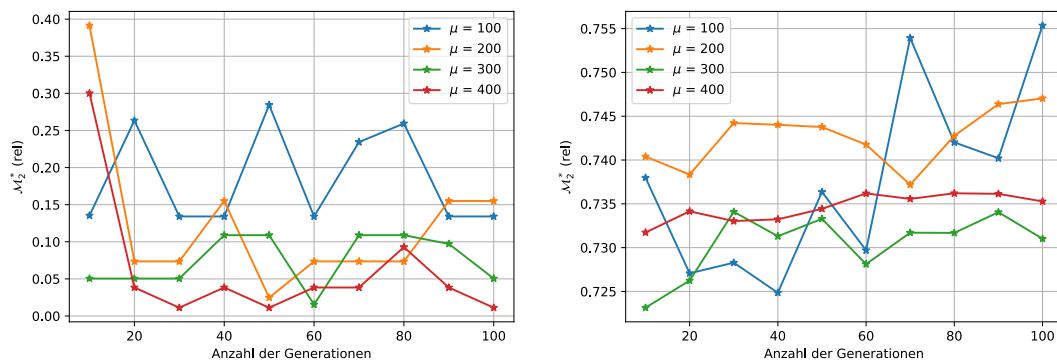
(a) Düsendatensatz ($k = 23, \sigma = 20$)(b) Feuchtigkeitsdatensatz ($k = 23, \sigma = 30$)

Abbildung 6.13: Relative \mathcal{M}_2^* -Metrik in der Komponentengruppe „Werkzeug“ im Feuchtigkeits- und Düsendatensatz.

Nun sollen die erzielten Ergebnisse für die relative \mathcal{M}_2^* -Metrik beleuchtet werden: Hinsichtlich des μ -Parameters lässt sich kein eindeutiger Trend zur Verbesserung oder Verschlechterung bei gezielter Variation des Parameters erkennen. Grundsätzlich ließen sich mit $\mu = 400$ und $\mu = 200$ bei $G = 10$ im Verhältnis zu $\mu \in \{100, 300\}$ gute Ergebnisse erzielen, die jedoch bei Hinzunahme weiterer Generationen nicht beibehalten werden konnten, wobei in dieser Hinsicht kein eindeutiger Trend zu erkennen ist. So lässt sich des Weiteren für $\mu = 100$ und $\mu = 300$ keine eindeutige Aussage dazu treffen, ob weitere Generationen bei der evolutionären Optimierung berechnet werden sollten.

Im Feuchtigkeitsdatensatz lassen sich ähnliche Beobachtungen bei der relativen \mathcal{M}_2^* -Metrik machen, wie bei der Komponentengruppe „Schnecke“: So lassen sich unter Verwendung von $\mu \in \{100, 200\}$ bessere Werte erreichen, als für $\mu \in \{300, 400\}$, insbesondere dann, wenn weitere Generationen berechnet. Die Werteverläufe lassen dabei jeweils alle vermuten, dass durch die Berechnung zusätzlicher Generationen die Werte für die relative \mathcal{M}_2^* -Metrik verbessert werden, wobei $\mu = 100$ die stärksten Unterschiede bei der Metrik hinsichtlich der Berechnung zusätzlicher Generationen aufweist.

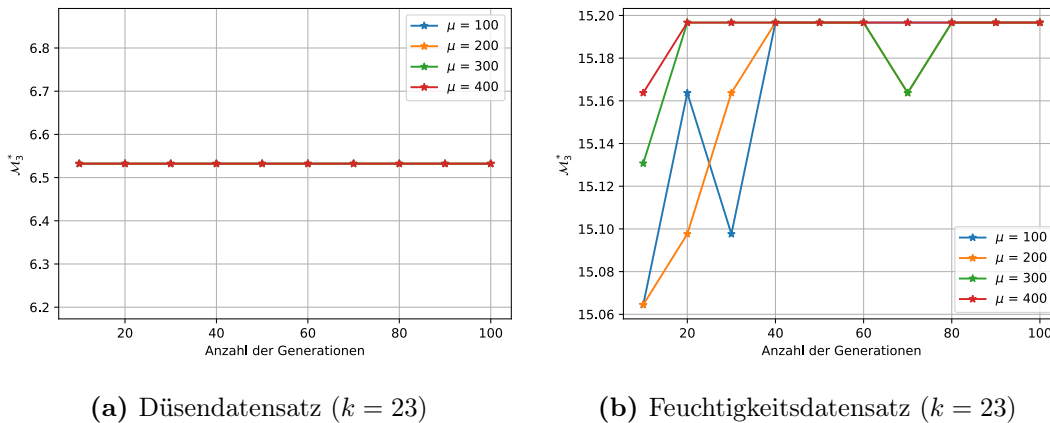
(a) Düsendatensatz ($k = 23$)(b) Feuchtigkeitsdatensatz ($k = 23$)

Abbildung 6.14: \mathcal{M}_3^* -Metrik in der Komponentengruppe „Werkzeug“ im Feuchtigkeits- und Düsendatensatz.

Abschließend werden die Wertereihen für die \mathcal{M}_3^* -Metrik in der Komponentengruppe „Werkzeug“ betrachtet, die in Abbildung 6.14 zu sehen sind. So ergeben sich im Düsendatensatz bei Variation des μ - und G -Parameters keine Änderungen für die \mathcal{M}_3^* -Metrik, was dafür spricht, dass sich die Ausdehnung der nicht-dominierten Front mit der Parameterwahl nicht verbessern (oder verschlechtern) lässt.

Im Feuchtigkeitsdatensatz lässt sich mit Zunahme des G -Parameters für alle Wertereihen ein positiver Trend beobachten. Die Variation des μ -Parameters führt offenbar dazu, dass sich die besseren Werte bereits mit weniger Generationen erreichen lassen. Die Wertebereiche, in denen die Veränderungen stattfinden, lassen jedoch vermuten, dass der tatsächliche Gewinn an Breite der nicht-dominierten Front eher als gering einzustufen ist.

6.3.4 Komponentengruppe „Heizung“

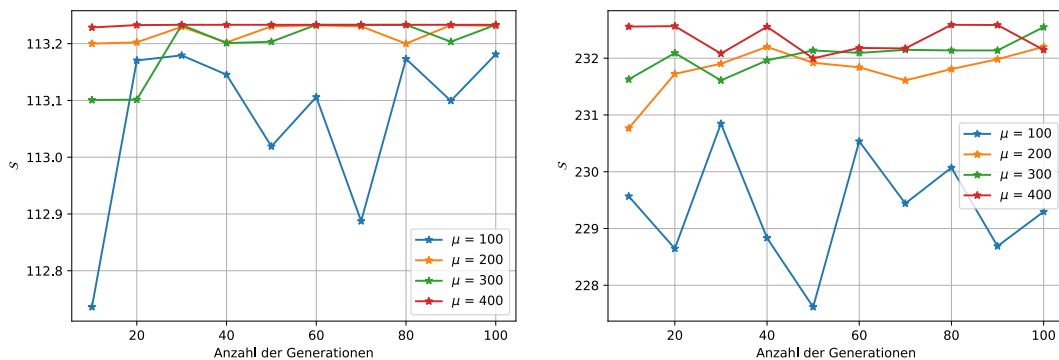
(a) Düsendatensatz ($k = 23$)(b) Feuchtigkeitsdatensatz ($k = 23$)

Abbildung 6.15: \mathcal{S} -Metrik in der Komponentengruppe „Heizung“ im Feuchtigkeits- und Düsendatensatz.

Betrachtet wird nun der Verlauf der Wertereihen der \mathcal{S} -Metrik in der Komponentengruppe „Heizung“ im Feuchtigkeits- und Düsendatensatz, welcher in Abbildung 6.15 zu sehen ist. Hierbei ergeben sich im Düsendatensatz für $\mu \in \{200, 300, 400\}$ bei Variation des G -Parameters nur geringe Veränderungen. Für $\mu = 100$ ergeben sich, $G = 20$ ausgenommen, stets schlechtere Werte gegenüber den Wertereihen für $\mu \in \{200, 300, 400\}$, wobei die Berechnung zusätzlicher Generationen, neben einer anfänglichen Verbesserungen, zwischenzeitlich sogar Verschlechterungen hervorgerufen hat. Eine pauschale Erhöhung der zu berechnenden Generation scheint die \mathcal{S} -Metrik hier also nicht zu verbessern, wobei die Entwicklung für weitere Generationen zu beobachten wäre. Bei Betrachtung aller Wertereihen scheint die Variation des μ -Parameters zu bewirken, dass früher, also bei der Berechnung von weniger Generationen, bessere Werte erreicht werden.

Im Feuchtigkeitsdatensatz ergibt sich bei Betrachtung von $\mu = 100$ ein ähnliches Bild, wie bei dem Düsendatensatz. Die Werte sind stets schlechter als jene, die bei der Nutzung von $\mu \in \{200, 300, 400\}$ erreicht wurden, wobei sich aufgrund des oszillierenden Charakters der Kurve nicht eindeutig beurteilen lässt, ob zusätzlich berechnete Generationen das Ergebnis verbessern oder verschlechtern. Die Werteverläufe der Kurven für $\mu \in \{200, 300, 400\}$ ähneln dabei jenen des Düsendatensatzes. Hier scheinen die ersten vierzig Generationen die Werte für die \mathcal{S} -Metrik am stärksten zu beeinflussen, während zusätzlich berechnete Generationen, die darüber hinausgehen, einen geringeren Einfluss auf die Ausdehnung der nicht-dominierten Fläche nehmen.

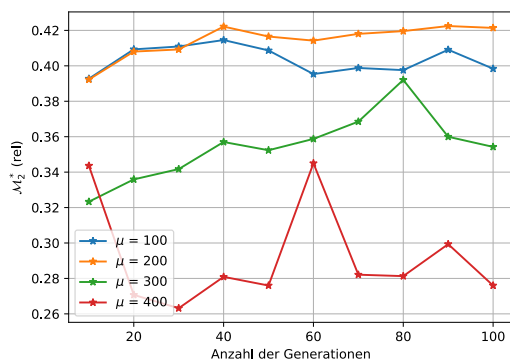
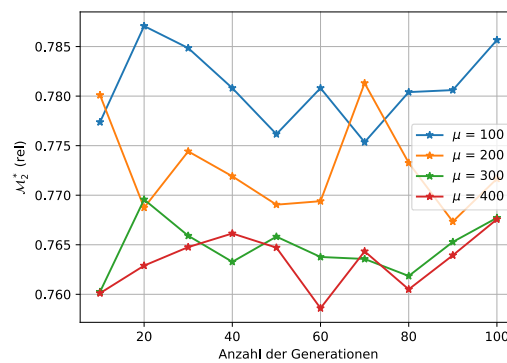
(a) Düsendatensatz ($k = 23$)(b) Feuchtigkeitsdatensatz ($k = 23$)

Abbildung 6.16: Relative \mathcal{M}_2^* -Metrik in der Komponentengruppe „Heizung“ im Feuchtigkeits- und Düsendatensatz ($\sigma^* = 30$).

Hinsichtlich der relativen \mathcal{M}_2^* -Metrik lassen sich einige Beobachtungen der beiden anderen Komponentengruppen wiederholen. So lassen sich im Düsendatensatz erneut für $\mu \in \{100, 200\}$ bessere Werte gegenüber $\mu \in \{300, 400\}$ feststellen. Hinsichtlich der Berechnung zusätzlicher Generationen, lassen sich folgende Beobachtungen machen: Für $\mu \in \{100, 200\}$ werden die Werte der relativen \mathcal{M}_2^* -Metrik nur geringfügig verbessert beziehungsweise verschlechtert. Bei $\mu = 300$ lässt sich ein Trend erkennen, der vermuten lässt, dass sich durch die Hinzunahme weiterer Generationen, die Werte der Metrik verbessern lassen. Bei $\mu = 400$ lässt sich kein eindeutiger Trend erkennen: So finden sich bei $G = 10$ und $G = 60$ die für diese Wertereihe besten Werte vor, während die anderen erfassten Werte auf etwa dem gleichen Niveau verharren.

Im Feuchtigkeitsdatensatz lassen sich für $\mu \in \{100, 200\}$ erneut bessere Werte gegenüber $\mu \in \{300, 400\}$ feststellen. Dabei lässt sich für $\mu \in \{100, 200\}$ eine stärkere Veränderung der Werte beobachten, die jedoch keinen eindeutigen Trend aufweisen. Für $\mu \in \{300, 400\}$ ist diese Varianz der Werte jedoch weniger stark ausgeprägt und ein eindeutiger Trend hinsichtlich der Zunahme weiterer Generationen ist auch hier nicht zu beobachten.

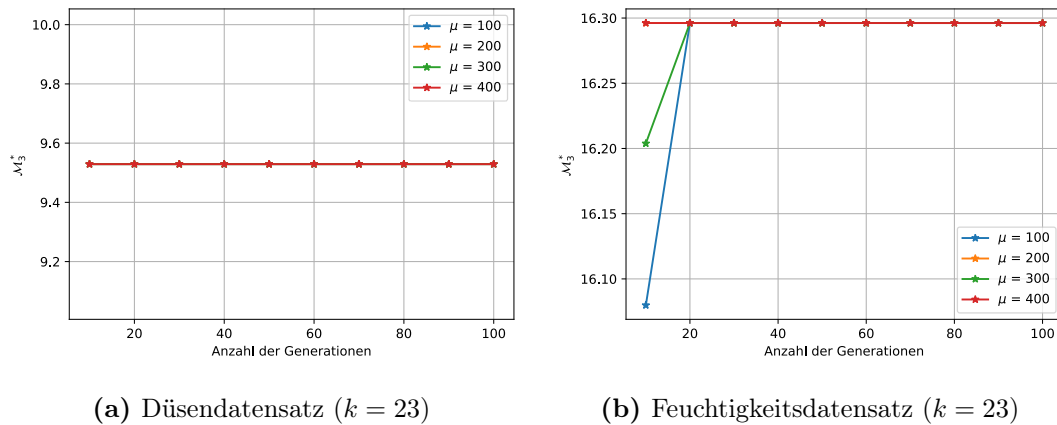


Abbildung 6.17: \mathcal{M}_3^* -Metrik in der Komponentengruppe „Heizung“ im Feuchtigkeits- und Düsendatensatz.

Zum Abschluss werden die Werte der \mathcal{M}_3^* -Metrik in der Komponentengruppe „Heizung“ im Feuchtigkeits- und Düsendatensatz betrachtet, wie sie in Abbildung 6.17 dargestellt sind. Hier ergeben sich für den Düsendatensatz, ähnlich zu Abbildung 6.14a, keine Veränderung bei Variation des μ - und G -Parameters. Daher wird an dieser Stelle der ähnliche Schluss gezogen, dass sich durch die Parameterwahl die Ausdehnung der nicht-dominierten Front nicht beeinflussen lässt.

Im Feuchtigkeitsdatensatz lässt sich dabei eine ähnliche Entwicklung beobachten. Dabei verlaufen die Wertereihen für $\mu \in \{200, 400\}$ konstant hinsichtlich der Anzahl berechneter Generation, während für $\mu \in \{100, 300\}$ diese Werte erst nach zwanzig Generationen erreicht werden. Dabei bewegen sich die Werte jedoch in einem kleinen Intervall, der aus praktischer Perspektive vermutlich keine signifikante Verbesserung hinsichtlich der Ausdehnung der nicht-dominierten Front bewirkt.

6.3.5 Zwischenfazit

Betrachtet man die Werte der \mathcal{S} -Metrik in Abhängigkeit des verwendeten Datensatzes und der Komponentengruppe so konnte grundsätzlich die Beobachtung gemacht werden, dass höhere Werte für den μ -Parameter wichtiger sind, als die Anzahl der zu berechnenden Generation. So konnten meistens bei der Wahl eines hohen μ -Wertes, wie z.B. $\mu = 400$, schon nach wenigen Generationen Werte erreicht werden, die bei der Wahl eines niedrigeren μ -Parameters auch nach hundert Generationen nicht erreicht wurden. Dabei war ebenfalls zu beobachten, dass für $\mu \in \{200, 300, 400\}$ die Verbesserungen nicht mehr so stark ausfielen, wie zwischen $\mu = 100$ und $\mu = 200$, was darauf hindeutet, dass das Verfahren eine Form von Konvergenz erreicht hat. Andere Verfahren, die zur Selektion, Mutation,

oder zum Crossing Over eingesetzt werden, könnten hinsichtlich der \mathcal{S} -Metrik noch bessere Approximationen finden.

Werden nun die Wertereihen für die \mathcal{M}_2^* -Metrik betrachtet, so konnte hier häufig die Beobachtung gemacht werden, dass eine niedrigere Wahl von μ diese Metrik positiv beeinflusst. Dabei ließ sich jedoch nicht immer klar die Frage beantworten, ob zusätzliche Generation berechnet werden sollten. Hier wäre es sinnvoll, für eine größere Anzahl von Generationen die Entwicklung der Wertereihen zu beobachten, um anschließend zu beurteilen, ob eine höhere Wahl des Parameters G sinnvoll ist.

Die Werte der \mathcal{M}_3^* -Metrik entwickelten sich für die meisten Komponentengruppen und Datensätze äußerst unauffällig. Selbst bei der Variierung von μ und G waren nur geringfügige Änderungen bei den Werten der Metrik festzustellen, sodass hier keine Aussage zur empfehlenswerten Wahl des μ - und G -Parameters getroffen werden können. Dabei ist es möglich, dass bereits die maximale Ausdehnung der nicht-dominierten Front gefunden wurde, diese also tatsächlich eine beste Approximation an die Paretofront darstellt.

6.4 Random Cluster

Aufbauend auf die Ergebnisse des zweiten Experiments, sollen nun die Ergebnisse der jeweiligen *Random Cluster*-Modelle bewertet werden. Hierzu werden aus der Menge der Konfigurationen, die im vorherigen Kapitel evaluiert wurden, diejenige ausgewählt, die im Mittel über alle k die beste (höchste) \mathcal{S} -Metrik provoziert haben: Dabei wird davon ausgegangen, dass die nicht-dominierte Front, die eine höhere \mathcal{S} -Metrik aufweist, auch näher an der wahren Paretofront liegt. Zusammen mit Tabelle 6.3 ergibt sich damit die folgenden Konfigurationen für die zu verwendenden Modelle:

	Schnecke	Werkzeug	Heizung
Düsensatz	$G = 100, \mu = 400$	$G = 80, \mu = 400$	$G = 100, \mu = 400$
Feuchtigkeitsdatensatz	$G = 70, \mu = 400$	$G = 80, \mu = 400$	$G = 90, \mu = 400$

Tabelle 6.4: Konfiguration des evolutionären Optimierers für die *Random Cluster*-Modellierung, differenziert nach Komponentengruppe und Datensatz.

Da keine Label zur Beurteilung des *wahren* Anomaliestatus vorliegen, soll an dieser Stelle eine grafische Überprüfung der Plausibilität der erzielten Ergebnisse versucht werden. Hierzu wird im Folgenden der Werkzeuginnendruck am Umschaltzeitpunkt, welcher eine qualitätsrelevante Größe darstellt, in Abhängigkeit der Zyklusnummer, die einen Zyklus eindeutig identifiziert, betrachtet und die Beobachtungen notiert.

Dabei wird sich insbesondere auf die Beobachtungen konzentriert, die an den Übergängen zwischen den einzelnen Düsen beziehungsweise Materialfeuchten gemacht werden können, da sich dort vermutlich, aufgrund der verändernden Prozessparameter, vermehrt anomale Zyklen vorfinden.

6.4.1 Düsendatensatz

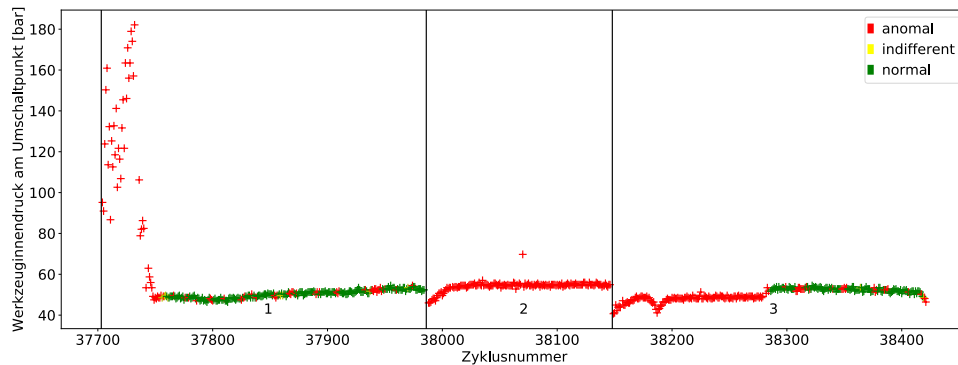


Abbildung 6.18: Bewertung der Zyklen im Düsendatensatz durch das „Schnecken“-Modell.

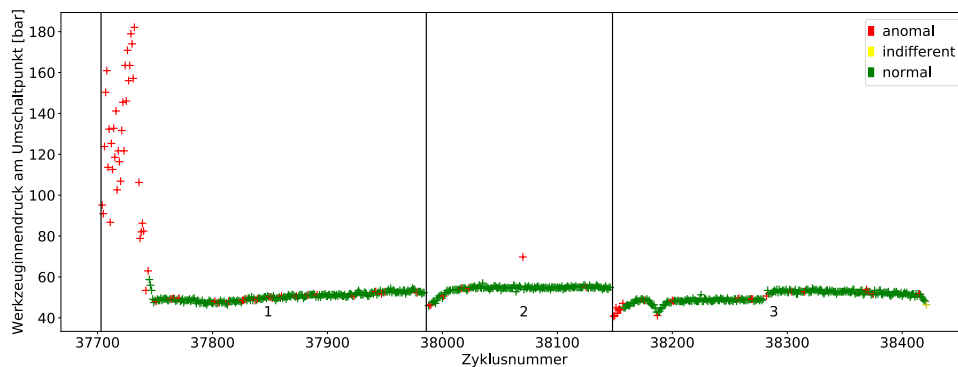


Abbildung 6.19: Bewertung der Zyklen im Düsendatensatz durch das „Werkzeug“-Modell.

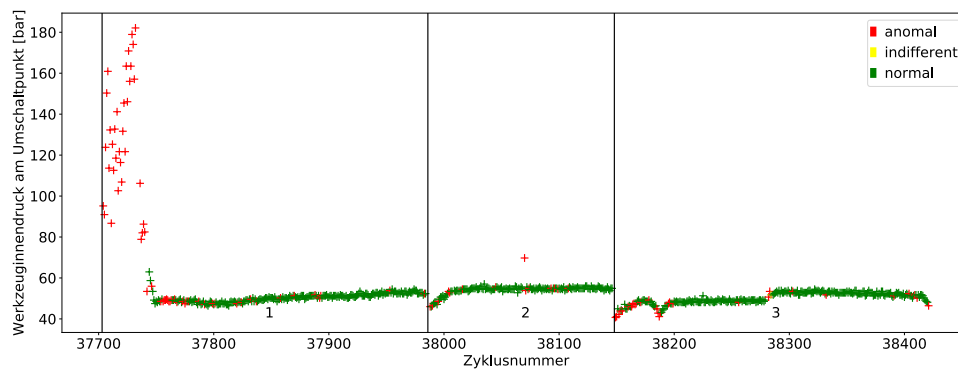


Abbildung 6.20: Bewertung der Zyklen im Düsendatensatz durch das „Heizung“-Modell.

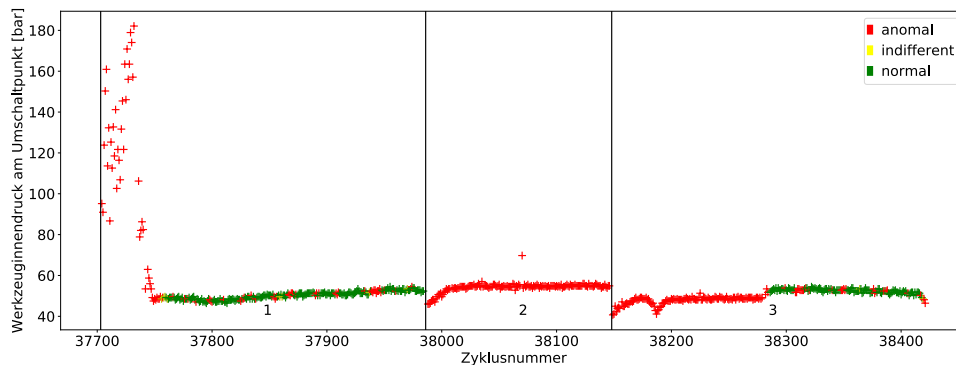


Abbildung 6.21: Bewertung der Zyklen im Düsendatensatz bei Verknüpfung aller Teilmodelle zu einem Metamodell gemäß Abschnitt 5.5.3. Die Trennlinien symbolisieren jeweils den Wechsel der Düse.

In den Abbildungen 6.18 - 6.21 sind die Bewertungen der Zyklen im Düsendatensatz durch die einzelnen Modelle zu sehen. Hierbei ist zu sehen, dass die Modelle in der Lage sind, die anfängliche Sequenz von etwa vierzig Zyklen nach dem initialen Düsentausch als anomal zu registrieren, was an dieser Stelle als plausibel beurteilt wird. Hier ist es beispielsweise möglich, dass der Bediener die Maschinenparameter erst noch korrekt einstellen muss und daher diese Zyklen anomal sind. Hierfür spricht auch der Werkzeuginnendruck, der sich als Ausreißer erkenntlich macht. Interessant ist weiterhin auch, dass im Schneckenmodell nach der beschriebenen Sequenz eine geringe Anzahl an Zyklen folgt, für die das Modell indifferent ist, bevor dann anschließend Zyklen folgen, die weitestgehend normal sind. Hier scheint sich also mit der Anzahl an Zyklen die Prozessqualität mit graduellem Charakter zu verbessern. Ebenso als plausibel zu bewerten, ist die Erkennung des Ausreißers in der zweiten Düse durch die einzelnen Modelle.

Ähnliche Beobachtungen können auch bei den anderen Düsenwechseln im Werkzeug- und Heizungmodell gemacht werden, wobei die Anzahl der als anomal klassifizierten Zyklen deutlich kleiner ist und nicht die gesamte Sequenz durchgehend als anomal registriert wird. Dabei müsste nun mit Domänenwissen beurteilt werden, ob dies der Realität entspricht oder lediglich ein Artefakt des Modellierungsverfahrens ist.

Interessant ist weiterhin auch, dass das Schneckenmodell, nach der Bewertung der ersten Düse, in der die Bewertungsstruktur im Wesentlichen derer des Werkzeug- und Heizungmodell ähnelt, mit dem Wechsel zur zweiten Düse alle Zyklen als anomal bewertet bevor inmitten der dritten Düse die Bewertung wieder heterogener wird. Hier müsste ebenfalls mit Domänenwissen und anhand der Werteausprägungen der konkret selektierten Merkmale geklärt werden, wieso diese Zyklen vollständig als anomal eingestuft werden. Die Entscheidungen durch das Schneckenmodell scheinen zumindest nicht vollständig abwegig zu sein, da diese Bewertungsstruktur genau mit dem Wechsel der Düse deutlich wird.

6.4.2 Feuchtigkeitsdatensatz

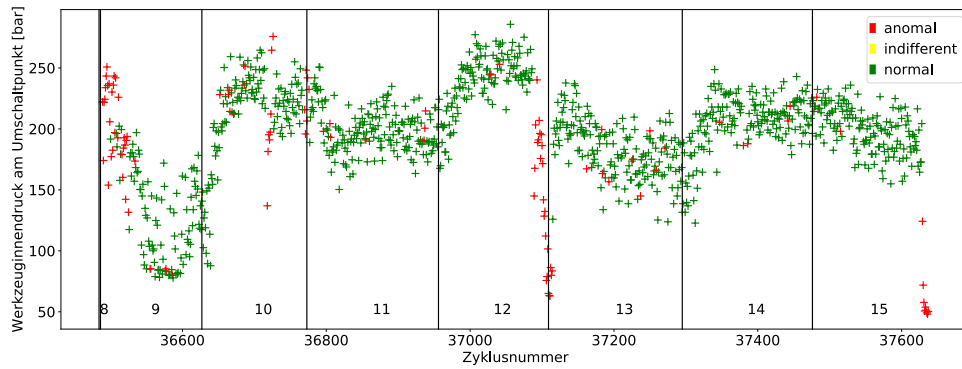


Abbildung 6.22: Bewertung der Zyklen im Feuchtigkeitsdatensatz durch das „Schnecken“-Modell.

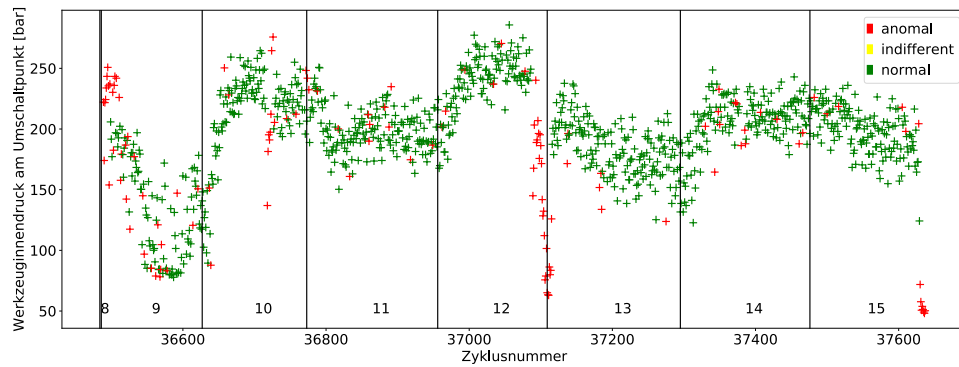


Abbildung 6.23: Bewertung der Zyklen im Feuchtigkeitsdatensatz durch das „Werkzeug“-Modell.

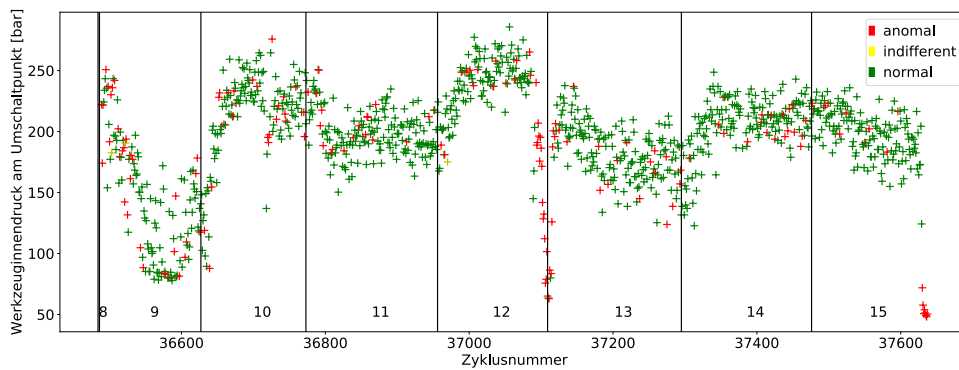


Abbildung 6.24: Bewertung der Zyklen im Feuchtigkeitsdatensatz durch das „Heizung“-Modell.

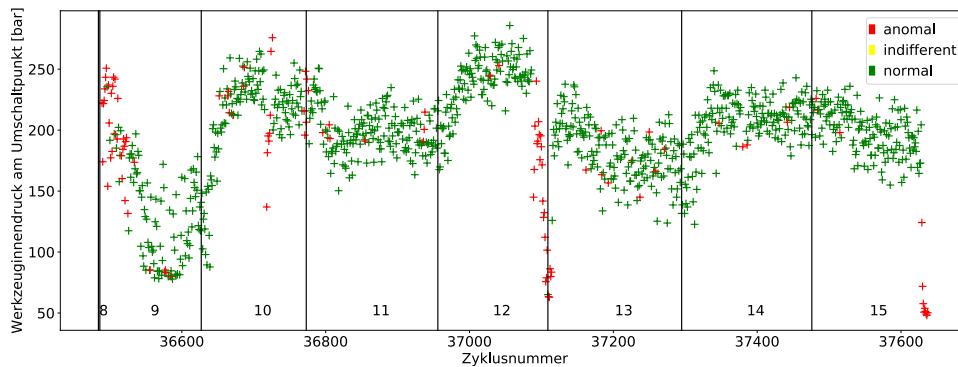


Abbildung 6.25: Bewertung der Zyklen im Feuchtigkeitsdatensatz bei Verknüpfung aller Teilm-
odelle zu einem Metamodell gemäß Abschnitt 5.5.3. Die Trennlinien symbolisieren jeweils den
Wechsel des verwendeten Plastikgranulats.

Nun sollen die *Random Cluster*-Modelle im Fechtigkeitsdatensatz beleuchtet werden, die
in den Abbildungen 6.22 - 6.25 visualisiert sind.

Bei Betrachtung der Bewertungsstruktur bei den einzelnen Übergänge zwischen den Feuch-
tigkeitsgraden des zugeführten Plastikgranulats lassen sich keine großen Unterschiede zwi-
schen den einzelnen Modellen erkennen. Auffällig ist erneut die plausible Häufung von
anormalen Zyklen zu Beginn der neunten Stunde und beim Übergang zwischen der zwölf-
ten und dreizehnten Stunde. Plausibel erscheinen weiterhin die anomalen Zyklen zum Ab-
schluss der fünfzehnten Stunde, bei denen die gemessenen Werkzeuginnendrucke sich auch
als Ausreißer erkennbar machen.

Neben den eben besprochenen Übergängen zwischen den einzelnen Stunden, lässt sich,
gegenüber den vergangenen Zyklen, bei den anderen Übergängen keine eindeutige Verän-
derungen in der Bewertungsstruktur feststellen. Hier lässt sich also, wie bei den Zyklen, die
innerhalb einer Stunde aufgezeichnet wurden, keine eindeutige Aussage zur Plausibilität
der Bewertung durch die einzelnen Modelle machen.

6.4.3 Strukturierung des Abstimmungsverhaltens

Nun soll die Struktur der abgegebenen Stimmen für einen Zyklus näher untersucht werden.
Hierfür soll das *Random Cluster*-Verfahren kurz zusammengefasst werden. Im Zuge der Mo-
dellierung lässt sich idealerweise für alle $k \in [k^{(L)}, k^{(U)}]$ eine Clusterung berechnen, die an-
schließend einen Merkmalsvektor \vec{m}_i positiv oder negativ bewerten. Nun ließe sich für jeden
Merkmalsvektor dieses Abstimmungsverhalten in einem Vektor $\vec{v}_i = (v_{k^{(L)}}, \dots, v_{k^{(U)}})^T \in$
 $\{0, 1\}^{k^{(U)} - k^{(L)}}$ erfassen, wobei jedes Element des Vektors v_k entweder „1“ anzeigt, falls
das k -te Modell eine positive Entscheidung abgibt, und „0“ falls nicht. Eine Menge zu
untersuchenden Merkmalsvektoren $M = \{\vec{m}_1, \dots, \vec{m}_n\}$ lässt sich so auf eine Menge von
„Abstimmungsvektoren“ $V = \{\vec{v}_1, \dots, \vec{v}_n\}$ abbilden, die jeweils das Abstimmungsverhalten

repräsentieren. Die Idee sei nun, die Menge V erneut einem Clustering zu unterziehen, um so Gruppen zu erfassen, die ein ähnliches Abstimmungsverhalten aufweisen.

In den Abbildungen 6.26 und 6.27 ist eine solche Clustering exemplarisch zu sehen:

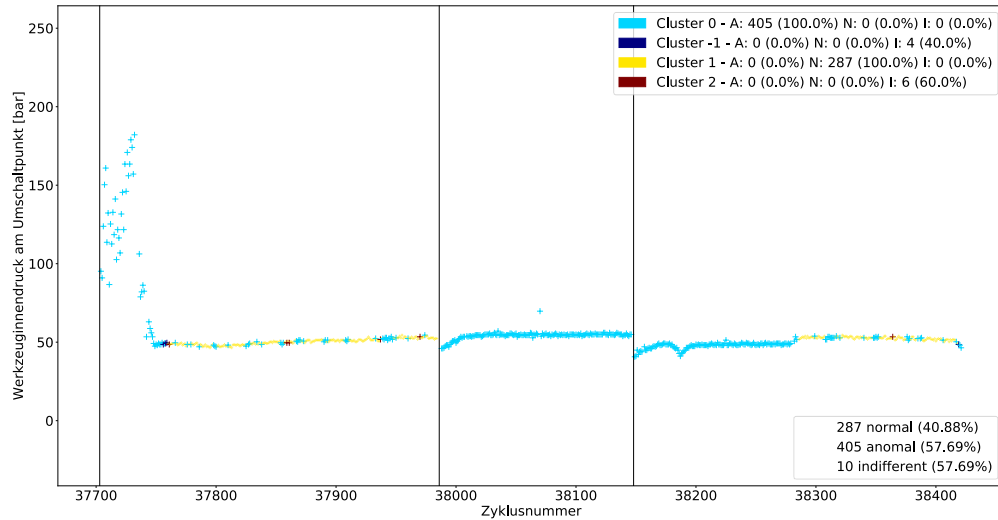


Abbildung 6.26: Clustering des Abstimmungsverhalten im Düsendatensatz bei Nutzung des Schneckenmodells. Die Punkte normaler Zyklen werden kleiner und mit 50% Transparenz dargestellt. Der Cluster -1 steht für den Cluster, in dem alle Rauschpunkte eingeordnet werden.

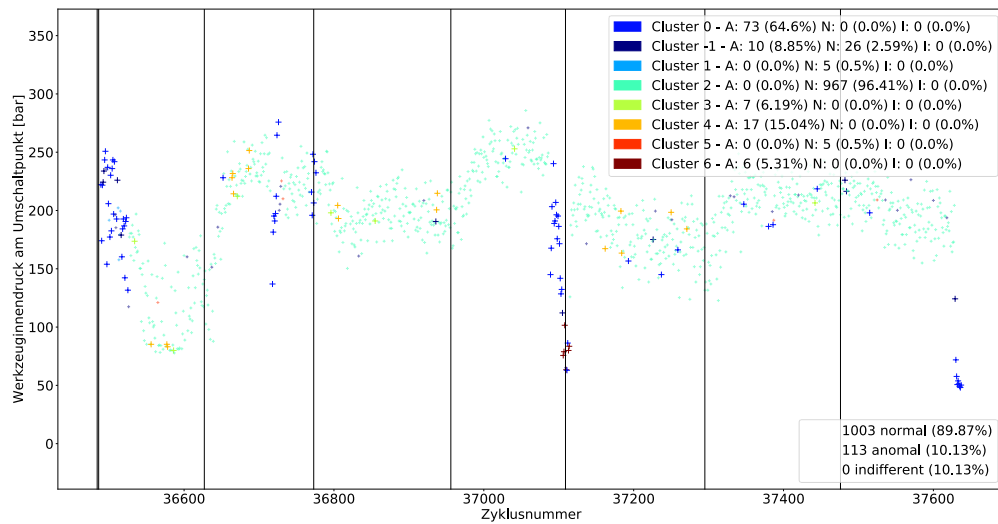


Abbildung 6.27: Clustering des Abstimmungsverhalten im Feuchtigkeitsdatensatz bei Nutzung des Schneckenmodells. Die Punkte normaler Zyklen werden kleiner und mit 50% Transparenz dargestellt. Der Cluster -1 steht für den Cluster, in dem alle Rauschpunkte eingeordnet werden.

Betrachtet man zunächst die Abbildung 6.26 so lässt sich dort ein recht einheitliches Bild wahrnehmen. Alle Zyklen, die anomal bewertet wurden, wurden einem Cluster zugeordnet.

Für die Zyklen, die normal bewertet wurden, lässt sich die gleiche Beobachtung machen. Lediglich die Zyklen, bei denen das Modell indifferent zwischen normal und anomal war, werden unterschiedlichen Clustern zugeordnet, wobei ein Cluster davon der Rauschcluster ist. An dieser Stelle lassen sich also für normale und anomale Zyklen keine wesentlichen Unterschiede im Abstimmungsverhalten feststellen.

Betrachtet man nun die Abbildung 6.27 so lässt sich hier ein deutlich heterogeneres Bild wahrnehmen. Zwar werden in alle „echten“ Cluster, also dem Rauschcluster ausgenommen, ausschließlich Zyklen beziehungsweise ihre Abstimmungsvektoren zugeordnet, die entweder anomal oder normal sind, jedoch ergibt sich, im Gegensatz zu Abbildung 6.26, nicht ein einzelner Cluster, in dem anomale oder normale Zyklen zugeordnet sind. So lassen sich hier vier Anomaliecluster und drei Normalcluster aufzählen. Diese Anomaliecluster könnten nun ein Ansatzpunkt sein, um zusätzliche Erkenntnisse aus der Modellierung zu gewinnen. So könnte eine separate Offline-Bewertung durch einen Experten Aufschluss darüber geben, ob sich durch diese hier exemplarisch durchgeführte Strukturierung des Abstimmungsverhalten, beispielsweise zusätzliche Anomalietypen identifizieren lassen (s. Abbildung 2.2a).

Kapitel 7

Fazit und Ausblick

In der vorliegenden Bachelorarbeit wurde zum Zwecke der Anomalieerkennung in industriellen Fertigungsprozessen eine Pipeline definiert, die ungelabelte Daten produzierender Maschinen etappenweise verarbeitet. Dazu wurde zunächst die systematische Verknüpfung der einzelnen Komponenten innerhalb der Pipeline als Methode vorgestellt. Anschließend wurden mögliche Vorverarbeitungsschritte präsentiert, die immer notwendig sind und, abhängig von dem betrachteten Anwendungsfall, notwendig sein können. Danach wurden geeignete maschinelle Lernverfahren und Lernprobleme präsentiert, die für eine Anwendung zur industriellen Anomalieerkennung in Frage kommen und die zusätzliche Modellierungsschritte aufgezeigt, wie die Bestimmung von geeigneten Trainingsdaten oder die Clustervalidierung beziehungsweise -interpretation, die insbesondere bei der Nutzung unüberwachter Lernverfahren häufig notwendig sind. Im Evaluationskapitel wurde dann der konkrete Anwendungsfall vorgestellt und ausgewählte Methoden aus den vergangenen Kapiteln miteinander verknüpft, um eine Anomalieerkennung bei ungelabelten Daten, die aus Spritzgießprozessen gewonnen wurden, zu ermöglichen. In diesem Zusammenhang wurde auch das *Random Cluster*-Verfahren vorgestellt, welches mehrere Clustermodelle verknüpft, um eine Anomalieentscheidung auf Basis mehrerer Merkmalsmengen zu treffen. Auf Grundlage des *Random Cluster*-Modellierungsverfahrens wurden anschließend Möglichkeiten präsentiert, um aus einem oder mehreren Modellen einen Anomaliebericht zu generieren, der einem Maschinenbediener möglichst detailliert aufzeigt, warum das System einen spezifischen Produktionsprozess für anomal hält. Des Weiteren wurden Experimente entwickelt und Metriken vorgestellt, die aufzeigen sollen, wie gut das verwendete Verfahren zur Merkmalsselektion, in Abhängigkeit einer domänenwissenbasierten Vorselektion, arbeitet und wie die Qualität der nicht-dominierten Front durch die Parameterwahl in der (evolutionären) Optimierung beeinflusst wird. Hinsichtlich der Eignung des MRMR-Verfahrens wurden dabei keine eindeutigen Ergebnisse erzielt, die anzeigen, ob eine Vorselektion oder die Nutzung dieses linearen Verfahrens zur Merkmalsselektion immer zu präferieren sind,

wobei teilweise akzeptable Ergebnisse erzielt wurden. Das Experiment zur evolutionären Optimierung lieferte grundsätzlich die Erkenntnis, dass eine höhere Anzahl an Individuen pro Generation gegenüber der Anzahl der berechneten Generation zu bevorzugen ist. Das abschließende Experiment betrachtete dann die Bewertung der einzelnen Zyklen in Abhängigkeit des Werkzeuginnendruckes am Umschaltpunkt, einer qualitätsrelevanten Größe beim Spritzgießen. Hier wurden plausible Anomaliebewertungen erreicht, wobei eine genaue Messungen zur Prognosequalität aufgrund fehlender Label nicht möglich war. Abschließend wurde aufgezeigt, wie das Abstimmungsverhalten im *Random Cluster*-Modell dazu genutzt werden könnte, um verschiedene Anomalietypen zu identifizieren.

Interessant wäre es nun, die prognostizierten Anomaliebewertungen mit den *wahren* Werten zu vergleichen, um ein Maß für die Prognosequalität zu erhalten. Hier wäre es insbesondere interessant, neben der einfachen Information, ob ein Zyklus anomal oder normal ist, auch die detaillierte Information darüber zu erhalten, welche Prozessphasen und Maschinenkomponenten jeweils betroffen waren, um auch den Anomaliebericht zu verifizieren. Des Weiteren wäre es sicher auch interessant, neben den modellierten endogenen Einflüssen in der Maschine, auch exogene und prozessrelevante Merkmale, wie der verwendete Rohstoff oder bestimmte Umwelteinflüsse (vgl. [23], Abschnitt 3.1.2), in die Modellierung einfließen zu lassen, um die Prognosequalität potenziell zu verbessern. Da sich, wie oben beschrieben, nicht immer eine gute Merkmalsmenge identifizieren ließ, soll die Möglichkeit erwähnt werden, in Zukunft ein nicht-lineares Merkmalsselektionsverfahren zu nutzen. Eine tiefere Betrachtung der selektieren Merkmalsmenge könnte bei einem nicht-linearen Selektionsverfahren, wie bereits in Abschnitt 5.3.4 angedeutet, des Weiteren dabei behilflich sein, neue Zusammenhänge zwischen Merkmalen, auch unabhängig von Domänenwissen, zu entdecken. Außerdem wäre es interessant zu sehen, wie die entwickelten Methoden aus dem Offline-Betrieb in den Online-Betrieb überführt werden könnte, wobei hier insbesondere das Problem der hohen Laufzeit bei der *Random Cluster*-Modellierung gelöst werden muss. Hier könnten sich aufgrund des hohen Parallelisierungspotenzials GPGPU-Lösungen (engl. *general purpose computation on graphics processing units*) anbieten.

Abbildungsverzeichnis

2.1	Ausschnitt des Werkzeuginnendruckverlaufes, einer prozessrelevanten Größe des Spritzgießprozesses, während eines ausgewählten Produktionsprozesses. Die Zeit ist relativ zum Prozessbeginn. Die Trennlinien visualisieren hierbei einen Teil der, von der Maschine gemeldeten, aktuellen Prozessphasen. Der rote Bereich zeigt die Entwicklung des Werkzeuginnendruckes während der Einspritzphase, der grüne Bereich den Werkzeuginnendruck während der Nachdruckphase und der blaue Bereich einen Ausschnitt des Werkzeuginnendruckverlaufes während der Abkühlphase.	10
2.2	Exemplarische Illustration der beiden, in der Arbeit vorgestellten Lernaufgaben, zur Identifikation anomaler Produktionsprozesse. Das Clustering gruppiert Punkte anhand eines Distanzbegriffes, sodass sich kleinere Cluster, die weniger Punkte enthalten, von einem großen Cluster abheben, was sich letztlich zur Anomalieerkennung ausnutzen lässt. Die Single Class-SVM hingegen lernt aus einer Menge „guter“ Punkte eine Trennebene zwischen diesen und anomalen Punkten. Es sei darauf hingewiesen, dass eine derart klare Trennung zwischen den anomalen und normalen Punkten nicht immer möglich ist.	11
2.3	Exemplarische Illustration einer gelernten Trennebene zwischen normalen und anomalen Punkten im einfachen, zweidimensionalen Fall. Das Merkmal \vec{x}_2 nimmt hier keinen Einfluss auf die Prognose.	12
2.4	Konzept einer Schnittstelle zwischen Anomaliedetektor und dem Maschinenbediener in Form eines Anomalieberichts.	13

3.1	Histogramme von zwei verschiedenen, aus den vollständigen, unsequenzierten Zeitreihen und über 1154 Produktionsprozessen extrahierten Merkmalen. In Abbildung 3.1a ist ein Histogramm zu sehen, das nahe legt, dass die zugrundeliegenden Daten normalverteilt sind und sich daher dieses Merkmal zu modellieren lohnt. In Abbildung 3.1b hingegen, ist diese Verteilung nicht zu erkennen, was eine Auswahl dieses Merkmals unattraktiv macht.	20
4.1	Illustration des Single Class SVM-Verfahrens [32].	29
4.2	Visualisierung der Größen im Optimierungsproblem des Support Vector Clusterings.	36
4.3	Illustration des DBSCAN-Verfahrens ($minPts = 2$) anhand einer Menge von zweidimensionalen Punkten und unter Verwendung der euklidischen Distanz. Rauschpunkte sind weiß, Randpunkte des Clusters gelb und Kernpunkte des Clusters rot markiert (eigene Darstellung, angelehnt an [14]).	41
5.1	Skizze einer Spritzgießmaschine [5].	48
5.2	Eine typische Prozedur zur evolutionären Optimierung (eigene Darstellung, Prozedur nach [9]).	62
5.3	Beispielhafte Illustration der Konstruktion der \mathcal{S} -Metrik im zweidimensionalen Zielraum, mit dem Optimierungsziel der Minimierung beider Zielfunktion $f_1(\vec{x})$ und $f_2(\vec{x})$. Die horizontalen und vertikalen Trennlinien visualisieren dabei die Hyperebenen. Die unterschiedlichen gräulichen Schattierung sind für die Berechnung der Metrik irrelevant und unterstreichen lediglich die Zusammensetzung der einzelnen Polytope.	68
5.4	Visualisierung des <i>Random Cluster</i> -Verfahrens in Verbindung mit einer <i>forward selection</i> -Merkmalsselektion und der Möglichkeit, mehrere RC-Bewerter einen Merkmalsvektor bewerten zu lassen. Die Maschinenkomponenten (Sensorik und Aktorik), aus denen letztlich die Merkmale resultieren, werden in Komponentengruppen gebündelt.	70
5.5	Konzept einer Schnittstelle zwischen Anomaliedetektor und dem Maschinenbediener in Form eines Anomalieberichts.	76
6.1	MAE und RMSE bei einer kreuzvalidierten, linearen Regression mit Fokus auf die Komponentengruppe „Schnecke“ und Differenzierung nach domänenbasierten Vorselektion und verwendetem Datensatz.	81
6.2	Histogramm der maximalen Einspritzdrücke (über Z) mit einer Klassengröße von 25.	81

6.3 Nähere Betrachtung von Fehlerkurven in der Komponentengruppe „Schnecke“. 82

6.4 Plots der erzielten Fehlerwerte (MAE/RMSE) bei Fokussierung auf die Komponentengruppe „Werkzeug“, differenziert nach Feuchtigkeits- und Düsendatensatz. 83

6.5 Histogramm der Werkzeuginnendrucke am Umschaltpunkt mit einer Klassengröße von 25. 84

6.6 MAE und RMSE bei einer kreuzvalidierten, linearen Regression mit Fokus auf die Komponentengruppe „Heizung“ bei Verwendung des Düsendatensatzes und Differenzierung nach domänenbasierten Vorselektion. 85

6.7 Histogramm der durchschnittlich erfassten Energie in der Zylinderheizung (Phase 2, über Z) mit einer Klassengröße von 25. 86

6.8 MAE und RMSE bei einer kreuzvalidierten, linearen Regression mit Fokus auf die Komponentengruppe „Heizung“ bei Verwendung des Feuchtigkeitsdatensatzes und Differenzierung nach domänenbasierten Vorselektion. 86

6.9 \mathcal{S} -Metrik in der Komponentengruppe „Schnecke“ im Feuchtigkeits- und Düsendatensatz. 89

6.10 Relative \mathcal{M}_2^* -Metrik in der Komponentengruppe „Schnecke“ im Feuchtigkeits- und Düsendatensatz ($\sigma^* = 30$). 91

6.11 \mathcal{M}_3^* -Metrik in der Komponentengruppe „Schnecke“ im Feuchtigkeits- und Düsendatensatz. 92

6.12 \mathcal{S} -Metrik in der Komponentengruppe „Werkzeug“ im Feuchtigkeits- und Düsendatensatz. 92

6.13 Relative \mathcal{M}_2^* -Metrik in der Komponentengruppe „Werkzeug“ im Feuchtigkeits- und Düsendatensatz. 93

6.14 \mathcal{M}_3^* -Metrik in der Komponentengruppe „Werkzeug“ im Feuchtigkeits- und Düsendatensatz. 94

6.15 \mathcal{S} -Metrik in der Komponentengruppe „Heizung“ im Feuchtigkeits- und Düsendatensatz. 95

6.16 Relative \mathcal{M}_2^* -Metrik in der Komponentengruppe „Heizung“ im Feuchtigkeits- und Düsendatensatz ($\sigma^* = 30$). 96

6.17 \mathcal{M}_3^* -Metrik in der Komponentengruppe „Heizung“ im Feuchtigkeits- und Düsendatensatz. 97

6.18 Bewertung der Zyklen im Düsendatensatz durch das „Schnecken“-Modell. . . 99

6.19 Bewertung der Zyklen im Düsendatensatz durch das „Werkzeug“-Modell. . . 99

6.20	Bewertung der Zyklen im Düsendatensatz durch das „Heizung“-Modell.	99
6.21	Bewertung der Zyklen im Düsendatensatz bei Verknüpfung aller Teilmodelle zu einem Metamodell gemäß Abschnitt 5.5.3. Die Trennlinien symbolisieren jeweils den Wechsel der Düse.	100
6.22	Bewertung der Zyklen im Feuchtigkeitsdatensatz durch das „Schnecken“-Modell.	101
6.23	Bewertung der Zyklen im Feuchtigkeitsdatensatz durch das „Werkzeug“-Modell.	101
6.24	Bewertung der Zyklen im Feuchtigkeitsdatensatz durch das „Heizung“-Modell.	101
6.25	Bewertung der Zyklen im Feuchtigkeitsdatensatz bei Verknüpfung aller Teilmodelle zu einem Metamodell gemäß Abschnitt 5.5.3. Die Trennlinien symbolisieren jeweils den Wechsel des verwendeten Plastikgranulats.	102
6.26	Clustering des Abstimmungsverhalten im Düsendatensatz bei Nutzung des Schneckenmodells. Die Punkte normaler Zyklen werden kleiner und mit 50% Transparenz dargestellt. Der Cluster -1 steht für den Cluster, in dem alle Rauschpunkte eingeordnet werden.	103
6.27	Clustering des Abstimmungsverhalten im Feuchtigkeitsdatensatz bei Nutzung des Schneckenmodells. Die Punkte normaler Zyklen werden kleiner und mit 50% Transparenz dargestellt. Der Cluster -1 steht für den Cluster, in dem alle Rauschpunkte eingeordnet werden.	103

Algorithmenverzeichnis

4.1	Core Vector Machine	45
4.2	$(1 + \epsilon)$ -Approximation des <i>Enclosing Balls</i>	46
5.1	Bestimmung des unteren und oberen Begrenzung des ϵ -Parameters im Suchraum des DBSCAN-Algorithmus.	66
5.2	RANDOMCLUSTER: Bestimmung verdächtiger Produktionsprozesse.	71
5.3	RANDOMCLUSTER: Bestimmung von normalen, anomalen und unentscheidbaren Zyklen.	72

Tabellenverzeichnis

5.1	Triggerattribute der betrachteten Spritzgießmaschine.	50
5.2	Versuchsplanung des zweiten Datensatzes: Variation des Düsendurchmessers.	51
5.3	Konfusionsmatrix für die binäre Klassifikation [35].	75
6.1	Wahl des Labels in Abhängigkeit der betrachteten Komponentengruppe. . .	80
6.2	Dem MRMR-Verfahren zu Verfügung stehende Merkmalsmengen bei der Messung der Qualität der Approximation an die Paretofront, differenziert nach Komponentengruppe und Datensatz.	88
6.3	Konfiguration der Merkmalsselektion bei der Messung der Qualität der Ap- proximation an die Paretofront, differenziert nach Komponentengruppe und Datensatz.	88
6.4	Konfiguration des evolutionären Optimierers für die <i>Random Cluster</i> -Modellierung, differenziert nach Komponentengruppe und Datensatz.	98

Literaturverzeichnis

- [1] BAUER, MARCUS, URSULA GATHER und MICHAEL IMHOFF: *The Identification of Multiple Outliers in Online Monitoring Data*. Technical Report 29/1999, Sonderforschungsbereich 475, Universitat Dortmund, 1999.
- [2] BEN-HUR, ASA, DAVID HORN, HAVA T. SIEGELMANN und VLADIMIR VAPNIK: *Support Vector Clustering*. Journal Of Machine Learning Research, 2:125–137, 2001.
- [3] BEUTELSPACHER, ALBRECHT: *Lineare Algebra: Eine Einführung in die Wissenschaft der Vektoren, Abbildungen und Matrizen*, Kapitel Skalarprodukte, Seiten 295–335. Springer Fachmedien Wiesbaden, Wiesbaden, 2014.
- [4] BOYD, STEPHEN und LIEVEN VANDENBERGHE: *Convex Optimization*. Cambridge University Press, 2004.
- [5] BRENDAN ROCKEY, UNIVERSITY OF ALBERTA INDUSTRIAL DESIGN: *Injection Molding Machine how to work injection moulding*. (https://commons.wikimedia.org/wiki/File:Injection_moulding.png, abgerufen am 17.08.2017), lizenziert unter CC BY 3.0 (<https://creativecommons.org/licenses/by/3.0/deed.en>, abgerufen am 17.08.2017). Von mir in Kopf- und Fußzeilen beschnitten und ins Deutsche übersetzt.
- [6] BURKSCHAT, MARCO, ERHARD CRAMER und UDO KAMPS: *Beschreibende Statistik: Grundlegende Methoden der Datenanalyse*. Springer Spektrum, 2012.
- [7] COHN, DAVID: *Active Learning*. In: SAMMUT, CLAUDE und GEOFFREY I. WEBB (Herausgeber): *Encyclopedia of Machine Learning*, Seiten 10–14. Springer US, Boston, MA, 2010.
- [8] DEB, K., A. PRATAP, S. AGARWAL und T. MEYARIVAN: *A fast and elitist multiobjective genetic algorithm: NSGA-II*. IEEE Transactions on Evolutionary Computation, 6(2):182–197, Apr 2002.
- [9] DEB, KALYANMOY: *Introduction to Evolutionary Multiobjective Optimization*, Seiten 59–96. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

- [10] DEB, KALYANMOY und RAM BHUSHAN AGRAWAL: *Simulated Binary Crossover for Continuous Search Space*. Complex Systems, 9(2), 1995.
- [11] DEB, KALYANMOY und DEBAYAN DEB: *Analysing Mutation Schemes for Real-parameter Genetic Algorithms*. Int. J. Artif. Intell. Soft Comput., 4(1):1–28, Februar 2014.
- [12] DING, CHRIS H. Q. und HANCHUAN PENG: *Minimum Redundancy Feature Selection from Microarray Gene Expression Data*. In: *2nd IEEE Computer Society Bioinformatics Conference (CSB 2003), 11-14 August 2003, Stanford, CA, USA*, Band 0, Seiten 523–529, Los Alamitos, CA, USA, 2003. IEEE Computer Society.
- [13] DINH, LINH T.T., HANS PASMEN, XIAODAN GAO und M. SAM MANNAN: *Resilience engineering of industrial processes: Principles and contributing factors*. Journal of Loss Prevention in the Process Industries, 25(2):233 – 241, 2012.
- [14] ESTER, MARTIN, HANS-PETER KRIEGEL, JÖRG SANDER und XIAOWEI XU: *A density-based algorithm for discovering clusters in large spatial databases with noise*. Seiten 226–231. AAAI Press, 1996.
- [15] FAHAD, A., N. ALSHATRI, Z. TARI, A. ALAMRI, I. KHALIL, A. Y. ZOMAYA, S. FOUFOU und A. BOURAS: *A Survey of Clustering Algorithms for Big Data: Taxonomy and Empirical Analysis*. IEEE Transactions on Emerging Topics in Computing, 2(3):267–279, Sept 2014.
- [16] FORTIN, FÉLIX-ANTOINE, FRANÇOIS-MICHEL DE RAINVILLE, MARC-ANDRÉ GARDNER, MARC PARIZEAU und CHRISTIAN GAGNÉ: *DEAP: Evolutionary Algorithms Made Easy*. Journal of Machine Learning Research, 13:2171–2175, jul 2012.
- [17] GAMA, JOÃO, INDRÉ ŽLIOBAITĖ, ALBERT BIFET, MYKOLA PECHENIZKIY und ABDELHAMID BOUCHACHIA: *A Survey on Concept Drift Adaptation*. ACM Comput. Surv., 46(4):44:1–44:37, März 2014.
- [18] GAN, JUNHAO und YUFEI TAO: *DBSCAN Revisited: Mis-Claim, Un-Fixability, and Approximation*. In: *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data, SIGMOD '15*, Seiten 519–530, New York, NY, USA, 2015. ACM.
- [19] GUYON, ISABELLE und ANDRÉ ELISSEEFF: *An introduction to variable and feature selection*. Journal of machine learning research, 3(Mar):1157–1182, 2003.
- [20] GUYON, ISABELLE und ANDRÉ ELISSEEFF: *An Introduction to Feature Extraction*, Seiten 1–25. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.

- [21] HAMAN, SOROMO: *Prozessnahes Qualitätsmanagement beim Spritzgießen*. Doktorarbeit, Fakultät für Elektrotechnik und Informationstechnik, Technischen Universität Chemnitz, 2003.
- [22] HASTIE, TREVOR, ROBERT TIBSHIRANI und ROBERT FRIEDMAN: *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer Series in Statistics. Springer, Berlin, 2008.
- [23] HÄUSSLER, JÖRG: *Eine Qualitätssicherungsstrategie für die Kunststoffverarbeitung auf der Basis künstlicher Neuronaler Netzwerke*. Doktorarbeit, Universität Gesamthochschule Essen - Fachbereich 12 - Maschinentechnik, 1994.
- [24] KEOGH, EAMONN und SHRUTI KASETTY: *On the Need for Time Series Data Mining Benchmarks: A Survey and Empirical Demonstration*. In: *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '02*, Seiten 102–111, New York, NY, USA, 2002. ACM.
- [25] KNOWLES, J. und D. CORNE: *On metrics for comparing nondominated sets*. In: *Evolutionary Computation, 2002. CEC '02. Proceedings of the 2002 Congress on*, Band 1, Seiten 711–716, May 2002.
- [26] KRIEGEL, HANS-PETER, PEER KRÖGER und ARTHUR ZIMEK: *Clustering High-dimensional Data: A Survey on Subspace Clustering, Pattern-based Clustering, and Correlation Clustering*. *ACM Trans. Knowl. Discov. Data*, 3(1):1:1–1:58, März 2009.
- [27] LIN, JESSICA, EAMONN KEOGH, STEFANO LONARDI und BILL CHIU: *A Symbolic Representation of Time Series, with Implications for Streaming Algorithms*. In: *Proceedings of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery, DMKD '03*, Seiten 2–11, New York, NY, USA, 2003. ACM.
- [28] MIERSWA, INGO: *Beatles vs. Bach: Merkmalsextraktion im Phasenraum von Audiodaten*. In: *LLWA 03 - Tagungsband der GI-Workshop-Woche Lernen - Lehren - Wissen - Adaptivität*, 2003.
- [29] MOLLEDA, J., J. C. GRANDA, R. USAMENTIAGA, D. F. GARCÍA und D. LAUREN-SON: *A quality inspection system for resistance seam welds in endless production of steel coils using anomaly detection techniques*. In: *2012 IEEE Industry Applications Society Annual Meeting*, Seiten 1–8, Oct 2012.
- [30] SAEYS, YVAN, IÑAKI INZA und PEDRO LARRAÑAGA: *A review of feature selection techniques in bioinformatics*. *Bioinformatics*, 23(19):2507, 2007.
- [31] SCHÖLKOPF, BERNHARD und ALEXANDER J. SMOLA: *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*, Kapitel Kernels. MIT Press, Cambridge, MA, USA, 2001.

- [32] SCHÖLKOPF, BERNHARD und ALEXANDER J. SMOLA: *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*, Kapitel Single-Class Problems: Quantile Estimation and Novelty Detection, Seiten 227–250. MIT Press, 2001.
- [33] SCHOWE, BENJAMIN: *Feature Selection for high-dimensional data in RapidMiner*. In: FISCHER, SIMON und INGO MIERSWA (Herausgeber): *Proceedings of the 2nd Rapid-Miner Community Meeting And Conference (RCOMM 2011)*, Aachen, 2011. Shaker Verlag.
- [34] SCHOWE, BENJAMIN und KATHARINA MORIK: *Fast-Ensembles of Minimum Redundancy Feature Selection*. In: OKUN, OLEG, GIORGIO VALENTINI und MATTEO RE (Herausgeber): *Workshop at ECML PKDD on Supervised and Unsupervised Ensemble Methods and their Applications - SUEMA 2010*, 2010.
- [35] SOKOLOVA, MARINA und GUY LAPALME: *A systematic analysis of performance measures for classification tasks*. *Information Processing & Management*, 45(4):427 – 437, 2009.
- [36] TAX, DAVID M. J. und ROBERT P. W. DUIN: *Support Vector Domain Description*. *Pattern Recogn. Lett.*, 20(11-13):1191–1199, November 1999.
- [37] TSANG, IVOR W., ANDRAS KOCSOR und JAMES T. KWOK: *Simpler core vector machines with enclosing balls*. In: *Proceedings of the 24th international conference on Machine Learning (ICML)*, Seiten 911–918, New York, NY, USA, 2007. ACM.
- [38] TSANG, IVOR W., JAMES T. KWOK und PAK-MING CHEUNG: *Core Vector Machines: Fast SVM Training on Very Large Data Sets*. *J. Mach. Learn. Res.*, 6(1):363 – 392, 12 2005.
- [39] WEBB, GEOFFREY I.: *Overfitting*. In: SAMMUT, CLAUDE und GEOFFREY I. WEBB (Herausgeber): *Encyclopedia of Machine Learning*, Seiten 744–744. Springer US, Boston, MA, 2010.
- [40] WROBEL, STEFAN, THORSTEN JOACHIMS und KATHARINA MORIK: *Maschinelles Lernen und Data Mining*. In: *Handbuch der Künstlichen Intelligenz*, Seiten 405–472. de Gruyter, 2013.
- [41] XU, RUI und D. WUNSCH: *Survey of clustering algorithms*. *IEEE Transactions on Neural Networks*, 16(3):645–678, May 2005.
- [42] ZITZLER, E.: *Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications*. Doktorarbeit, ETH Zurich, Switzerland, 1999.

- [43] ZITZLER, ECKART, JOSHUA KNOWLES und LOTHAR THIELE: *Quality Assessment of Pareto Set Approximations*, Seiten 373–404. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

Hiermit versichere ich, dass ich die vorliegende Arbeit selbstständig verfasst habe und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet sowie Zitate kenntlich gemacht habe.

Dortmund, den 7. September 2017

Philipp-Jan Honysz

