
Laufzeitüberwachung in selbst-organisierenden Industriesystemen

Marie Kiermeier

Dissertation
an der Fakultät für Mathematik, Informatik und Statistik
der Ludwig-Maximilians-Universität München

vorgelegt von
Marie Kiermeier

Tag der Einreichung: 09. Mai 2019

Laufzeitüberwachung in selbst-organisierenden Industriesystemen

Marie Kiermeier

Dissertation
an der Fakultät für Mathematik, Informatik und Statistik
der Ludwig-Maximilians-Universität München

vorgelegt von
Marie Kiermeier

1. Berichterstatter:	Prof. Dr. Claudia Linnhoff-Popien
2. Berichterstatter:	Prof. Dr. Stefan Fischer
Tag der Einreichung:	09. Mai 2019
Tag der Disputation:	22. Oktober 2019

Eidesstattliche Versicherung

(siehe Promotionsordnung vom 12.07.11, § 8, Abs. 2 Pkt. 5)

Hiermit erkläre ich an Eides statt, dass die Dissertation von mir selbstständig, ohne unerlaubte Beihilfe angefertigt ist.

München, 25.11.2019 Marie Kiermeier

Zusammenfassung

Die vorliegende Arbeit liefert Lösungsansätze für die Laufzeitüberwachung eines Selbst-organisierenden industriellen Systems (SOIS). Dazu wird ein Konzept zur Erkennung und Analyse von Anomalien in raumbezogenen Daten mit Hilfe von Substrukturverzeichnissen vorgestellt. Dieser Ansatz ermöglicht es, den aufgrund der Autonomie des Systems enorm großen Zustandsraum nachzubilden und damit Änderungen im Verhalten einzelner Systemteilnehmer oder auch Änderungen mit globalen Auswirkungen zuverlässig zu detektieren.

Im Rahmen dieser Arbeit wurde dieses Konzept zunächst für Bewegungsdaten von Objekten mit konstanter Geschwindigkeit, die sich in SOIS mit Fließbandarchitektur ergeben, umgesetzt und umfänglich mit Hilfe von entsprechenden Simulationsmodellen von SOIS evaluiert.

In einem weiteren Schritt wurde die Skalierbarkeit der Substrukturextraktion und die Onlinefähigkeit des Rekonstruktionsprozesses diskutiert. Für beide Fälle konnte eine Modullösung erarbeitet werden, die eine effiziente technische Realisierung des vorgestellten Ansatzes ermöglicht.

Schließlich wurde im Rahmen dieser Arbeit die Übertragbarkeit des vorgestellten Konzepts auf weitere, komplexere raumbezogene Daten, die in einem SOIS verarbeitet werden, gezeigt. Dafür wurde das Konzept sowohl für Bewegungsdaten von Objekten mit variabler Geschwindigkeit, als auch für zustandsbeschreibende räumliche Schichtmodelle umgesetzt. Die umfänglichen Evaluierungen der beiden Verfahren bestätigen, dass auch in diesem Fall das vorgestellte Konzept das Problem des enorm großen Zustandsraums bei der Erkennung und Analyse von Anomalien in SOIS mit Hilfe von Substrukturverzeichnissen löst.

Insgesamt liefert die vorliegende Arbeit damit einen wichtigen Beitrag zum Thema Qualitätssicherung in SOIS, da die vorgestellten Lösungsansätze zur Laufzeitüberwachung als Bestandteil eines Qualitätssicherungsprozesses eingesetzt werden können.

Abstract

This work presents solutions for runtime monitoring in self-organizing industrial systems (SOIS). Therefore, a concept for detecting and analysing anomalies in spatial data by using subpattern dictionaries is presented. This approach allows to model the enormous state space caused by the autonomy of such systems, and, by doing this, to detect reliably both, changes regarding the behaviour of single system components, but also changes of global impact.

The presented concept was implemented first for moving data of objects with constant velocity, which can be found in SOIS with conveyors, and evaluated extensively based on suitable simulations of such SOIS.

Furthermore, as part of this work, the scalability of the subpattern extraction and the online capability of the reconstruction process is discussed. In both cases modular solutions for an efficient realisation of the presented approach were developed.

Finally, the transferability of the presented concept to other, more complex spatial data, which have to be processed in a SOIS, is shown. For doing this, the concept was implemented both, for moving data of objects with variable velocity, but also state describing spatial layer models. The evaluation of both methods confirms that also in this case the presented concept solves the problem of the enormous state spaces when detecting and analysing anomalies in SOIS by using subpattern dictionaries.

In summary, this work makes an important contribution for quality assurance in SOIS, as the presented solutions can be used for runtime monitoring as parts of the overall quality assurance process.

Inhaltsverzeichnis

1	Einführung	1
1.1	Motivation	1
1.2	Beiträge der Arbeit	6
1.3	Vorveröffentlichungen	7
1.4	Aufbau der Arbeit	10
2	Grundlagen	11
2.1	Selbst-organisierende Industriesysteme	11
2.1.1	Maschinelles Lernen	11
2.1.2	Online Planen und Entscheiden	15
2.1.3	Raumbezogene Daten in SOIS	15
2.2	Qualitätssicherung in Industriesystemen	17
2.2.1	V-Modell	17
2.2.2	Validierung und Verifikation in adaptiven Systemen	18
2.3	Laufzeitüberwachung und Anomalieerkennung	19
2.4	Zusammenfassung	21
3	Laufzeitüberwachung mittels Substrukturverzeichnissen	23
3.1	Herausforderungen und Lösungsansatz	23
3.2	Verwandte Arbeiten	24
3.3	Simulationsmodell	26
3.4	Substrukturextraktion	26
3.5	Substrukturverzeichnis	28
3.6	Erkennung und Analyse von Anomalien	31
3.6.1	Einzeltrajektorien	31
3.6.2	Globale Ereignisse	34
3.6.3	Schleifen	37
3.6.4	Lastenverteilung	39
3.6.5	Neuartigkeit	44
3.7	Zusammenfassung	47
4	Effiziente Umsetzung der Laufzeitüberwachung	49
4.1	Skalierbarkeit der Substrukturextraktion	49
4.1.1	Substrukturextraktion mittels SEQUITUR	50
4.1.2	Simulationsmodelle	52
4.1.3	Evaluation	54
4.1.4	Erkennung und Analyse von Anomalien	59

4.1.5	Zusammenfassung	62
4.2	Onlinefähigkeit des Rekonstruktionsprozesses	64
4.2.1	Musterabgleich in Trajektorien­daten	65
4.2.2	Verwandte Arbeiten	67
4.2.3	Inverser online Substrukturabgleich	68
4.2.4	Evaluation	72
4.2.5	Zusammenfassung	77
4.3	Zusammenfassung	77
5	Erweiterungen zur Laufzeitüberwachung komplexerer Daten­strukturen	79
5.1	Laufzeitüberwachung von Objekten mit variabler Geschwindigkeit .	80
5.1.1	Simulationsmodell	80
5.1.2	Erweiterung des Substrukturverzeichnis­es	81
5.1.3	Erkennung und Analyse von Anomalien	82
5.1.4	Zusammenfassung	95
5.2	Laufzeitüberwachung von Zustands­beschreibungen	96
5.2.1	Verwandte Arbeiten	97
5.2.2	Räumliche Schichtmodelle	98
5.2.3	Substrukturextraktion	100
5.2.4	Erkennung und Analyse von Anomalien	104
5.2.5	Zusammenfassung	118
5.3	Zusammenfassung	119
6	Zusammenfassung und Ausblick	121
6.1	Zusammenfassung	121
6.2	Ausblick	123
	Literaturverzeichnis	125

Danksagung

An dieser Stelle möchte ich mich bei allen Personen bedanken, die mich bei der Erstellung dieser Dissertation unterstützt haben.

Hervorzuheben ist dabei Frau Prof. Dr. Claudia Linnhoff-Popien, die mir nicht nur als Doktormutter das nötige Vertrauen entgegen gebracht hat, sondern mir darüber hinaus in der Zeit als wissenschaftliche Mitarbeiterin an ihrem Lehrstuhl viele weitere wertvolle Erfahrungen ermöglicht hat.

Desweiteren möchte ich Herrn Prof. Dr. Stefan Fischer als meinen zweiten Berichterstatter und Herrn Prof. Dr. Christian Böhm als Prüfungskommissionvorsitzenden danken.

Ebenso danke ich allen Kollegen am Lehrstuhl für Mobile und Verteilte Systeme der LMU München und bei der Siemens AG für die jeweils entgegengebrachte Unterstützung.

Der größte Dank geht schließlich an meine Familie, die immer für mich da ist.

1 Einführung

Im Jahr 2011 griffen Henning Kagermann, Wolf-Dieter Lukas und Wolfgang Wahlster zum ersten Mal in der öffentlichen Diskussion den Begriff „Industrie 4.0“ auf [Kage 11]. Dabei bezogen sie sich auf die Geschichte der Industrialisierung, die bisher durch drei revolutionäre Entwicklungen geprägt wurde (siehe Abbildung 1.1). Hierfür gilt die Entwicklung der Dampfmaschine Ende des 18. Jahrhunderts als Auslöser der ersten industriellen Revolution. Durch die Einführung mechanischer Produktionsanlagen entstand damit neben dem Agrarsektor zum ersten Mal eine alternative Möglichkeit der wirtschaftlichen Wertschöpfung. Anfang des 20. Jahrhunderts erlebte die industrielle Fertigung dann durch die Verfügbarkeit von elektrischer Energie eine weitere Revolution. Damit war nun eine arbeitsteilige Massenproduktion möglich, wie sie bis heute mit der Fließbandmontage in der Autoproduktion assoziiert wird. Die Entwicklung der ersten Speicherprogrammierbaren Steuerung (SPS) markiert Anfang der 1970-er schließlich den Beginn der dritten industriellen Revolution. Durch den Einsatz von Elektronik und IT konnten die Produktionsprozesse weiter automatisiert werden, was eine variantenreichere Serienproduktion ermöglichte.

Mit „Industrie 4.0“ formulierten Kagermann, Lukas und Wahlster nun 2011 die Vision, dass durch die weiter wachsende Leistung von Computersystemen, sowie die zunehmende Vernetzung derselben in Zukunft eine neue Stufe der Industrialisierung erreicht werden kann. Dies wäre die vierte industrielle Revolution, in der Produkte noch stärker individualisiert würden. Entsprechend werden seitdem unter dem Label „Industrie 4.0“ fortwährend Lösungen erarbeitet, die eine noch flexiblere industrielle Fertigung ermöglichen sollen.

Diese digitale Transformation der industriellen Produktion ist auch Motivation für die in der vorliegenden Arbeit diskutierten Problemstellungen (siehe Abschnitt 1.1). Die konkreten Beiträge der Arbeit dazu werden in Abschnitt 1.2 ausgeführt, bevor in Abschnitt 1.3 auf entsprechende Vorveröffentlichungen und in Abschnitt 1.4 auf die Struktur der Arbeit eingegangen wird.

1.1 Motivation

Wie in Abbildung 1.1 am rechten Bildrand angedeutet, steigt mit der Zeit die Komplexität der industriell zu bearbeitenden Probleme. Ging es zunächst darum mechanische Vorrichtungen zu entwickeln, die einen bisher manuell ausgeführten Handgriff nachbildeten und mit Hilfe von Wasser- und Dampfkraft ausführten, wurden ab der zweiten industriellen Revolution die Prozessfol-

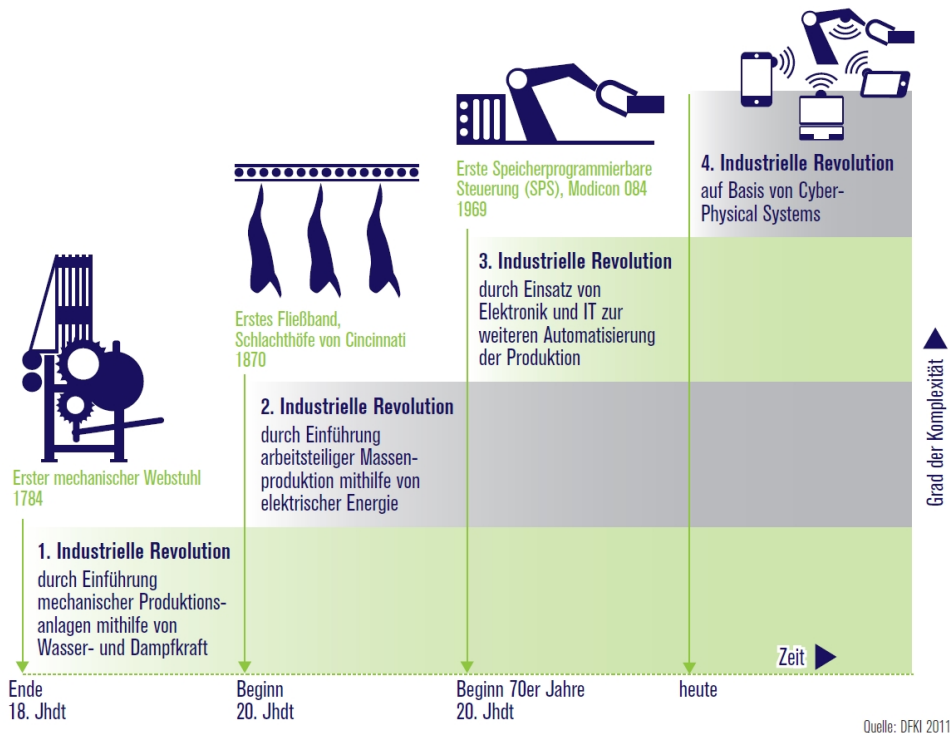


Abbildung 1.1: Darstellung der vier Industrierevolutionen [Kage 12].

gen stets komplexer, die automatisiert ausgeführt wurden. Diese Entwicklung begann 1870 mit dem ersten Fließband in einem Schlachthof und zeigt sich heute in hoch-automatisierten Produktionsanlagen, wie sie aus der Automobilbranche bekannt sind. Dank des Einsatzes von Elektroniksteuerungen und IT können heutzutage auch komplexere Prozesse, die zum Beispiel einen gewissen Variantenreichtum in der Produktion vorsehen, umgesetzt werden. Diese Flexibilität in der Produktion soll nun in der vierten industriellen Revolution noch weiter ausgebaut werden. Das Ziel sind stark individualisierte Produkte in kleinen Stückzahlen (bis hin zu Losgröße 1), wobei der Preis auf dem Niveau bisheriger Massenproduktionen verbleiben soll. Eine wichtige Voraussetzung dafür ist, dass nicht mehr nur vordefinierte Prozessfolgen automatisch ausgeführt werden, sondern das System Entscheidungen autonom und dezentral treffen und ausführen kann [Baue 17]. Für die technische Umsetzung sollen dabei so genannte *cyber-physische Systeme* (CPS), engl. „cyber-physical systems“, zum Einsatz kommen. Diese Systeme sind mit einer eingebetteten Software, engl. „embedded software“, ausgestattet, die folgende Anforderungen anwendungsspezifisch umsetzen soll [arca 11]:

- physische Daten mittels Sensoren aufnehmen und mit Hilfe von Aktoren, engl. „actuators“, in physikalische Prozesse eingreifen;
- aufgezeichnete Daten evaluieren und abspeichern; aktiv oder reaktiv mit der physischen und digitalen Welt kommunizieren;

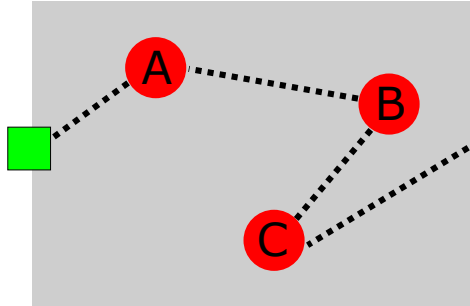
- CPS untereinander, aber auch mit globalen Netzwerken via digitaler Kommunikationsverfahren (drahtlos, engl. „wireless“, und/oder kabelgebunden, engl. „wired“; lokal und/oder global) verbinden;
- global verfügbare Daten und Dienste verwenden;
- eine Menge an dezidierten multimodalen Mensch-Maschinen-Schnittstellen, engl. „human-machine interfaces“, zur Verfügung stellen.

Mit Hilfe dieser CPS können nun intelligente Fabriken, engl. „Smart Factories“, gebaut werden [Baue 17]. Das Ziel dabei ist, dass es keine von Ingenieuren vorgegebene Prozessabläufe mehr gibt, sondern erst zur Laufzeit entschieden wird, welche Aktionen als nächstes ausgeführt werden. Damit erreicht man ein durchwegs neues Ausmaß an Flexibilität. Bisher sind die Lösungen der hochautomatisierten Fertigungsanlagen durch ihren **modellbasierten** Ansatz limitiert, der auf jeden technischen Prozess und dessen Dynamik gesondert ausgelegt ist, weswegen er damit auch den Grenzen der Abstraktion unterliegt [Voge 17].

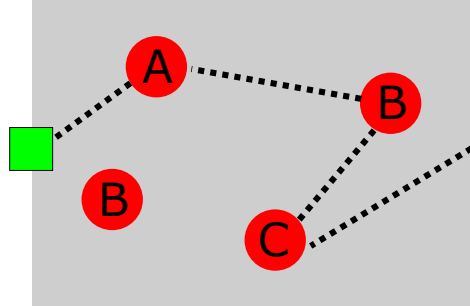
Im Gegensatz dazu erlaubt der Einsatz von CPS **informationsgetriebene** Lösungen. Dazu werden aus den Daten, die von den CPS entweder selbst gesammelt werden oder über andere Kanäle zur Verfügung stehen, zunächst mittels Maschinellen Lernen, engl. „machine learning“, (siehe Abschnitt 2.1.1) generelle Strukturen extrahiert, die dann auf verschiedene Domänen adaptiert werden können [Voge 17]. Damit sind nun auch stark individualisierte Produktspezifikationen möglich, da dies lediglich die Adaption des generellen Datenmodells auf eine andere Domäne, aber nicht, wie bisher, eine Anpassung des gesamten Prozessmodells erfordert.

Zur Illustration sind in Abbildung 1.2 verschiedene Situation in einer Produktionsanlage skizziert, anhand derer sich der Unterschied zwischen einer automatisierten Fertigung wie sie die dritte Industrielle Revolution ermöglichte und einer autonomen, die die Vision von „Industrie 4.0“ umsetzt, aufzeigen lässt. Das graue Rechteck entspricht dabei der Grundfläche der Fabrik. Innerhalb dieser Fabrik sind Maschinen platziert (rote Kreise), wobei jede Maschine einen bestimmten Bearbeitungsschritt ausführen kann (hier mit A, B oder C beschrieben). Auf der linken Seite der Fabrik werden die Werkteilerohlinge (grünes, bzw. blaues Quadrat) angeliefert, die nun entsprechend ihrer jeweiligen Produktspezifikation Maschinen in bestimmter Reihenfolge anfahren müssen, damit schließlich das gewünschte Produkt auf der rechten Seite der Fabrik abgenommen werden kann.

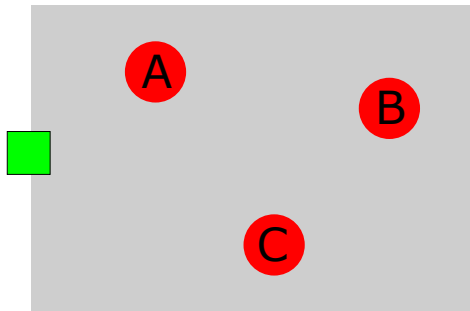
Wie bereits erwähnt, sind in der automatisierten Fertigung die möglichen Lösungen in einem Prozessmodell, das bei der Planung der Fabrik gemäß der technischen Prozesse einmalig festgeschrieben wurde, vorgegeben. Entsprechend sind in den Fabriken in Abbildung 1.2a und Abbildung 1.2b Verbindungsstrecken, die die Werkteile auf dem Weg durch die Fabrik nutzen können, mit



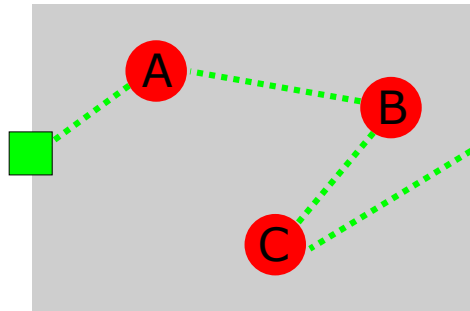
(a) Ausgangssituation des automatisierten Fertigungsprozesses.



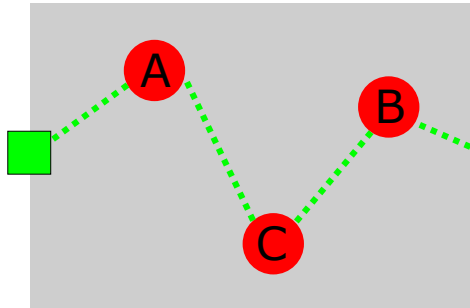
(b) Geänderte Ressourcenverfügbarkeit im automatisierten Fertigungsprozess.



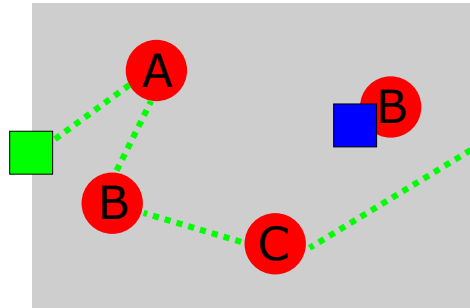
(c) Ausgangssituation des autonomen Fertigungsprozesses.



(d) Beispiel für Prozessverlauf im autonomen Fertigungsprozess.



(e) Beispiel für Prozessverlauf im autonomen Fertigungsprozess.



(f) Prozessverlauf bei geänderter Ressourcenverfügbarkeit im autonomen Fertigungsprozess.

Abbildung 1.2: Gegenüberstellung automatisierter und autonomer Fertigungsprozesse.

schwarz gestrichelten Linien eingezeichnet. Offensichtlich wurde bei der Planung der Fabrik davon ausgegangen, dass die Werkteile sich nur auf der Verbindungslinie A–B–C bewegen müssen, um ihre Aufgaben zu erfüllen. Zwar kann diese Serie relativ flexibel ausgeführt werden (z.B. A–B–A–B–C–B–C), allerdings stößt diese Flexibilität bei einer neuen Produktionsspezifikation wie A–C–B an ihre Grenzen. Statt auf direktem Weg von Maschine A zu Maschine C zu fahren, müsste entsprechend der vorgesehenen Bewegungsprofile ein Umweg über Maschine B genommen werden. Dies kostet zusätzliche Ressourcen und behindert unnötigerweise die Ausführungen der anderen Werkteile.

Auch bei Änderungen bezüglich der Ressourcenverfügbarkeit zeigen sich die Einschränkungen der modellbasierten Automatisierung. In Abbildung 1.2b ist beispielhaft eine weitere Maschine vom Typ B hinzugefügt worden. Nachdem diese zusätzliche Ressource allerdings nicht in der Planung der Fabrik vorgesehen war, kann diese erneut nicht ohne weiteres in den Prozessfluss integriert werden. Eine Entlastung der alternativen Maschine vom Typ B findet also nicht statt. Stattdessen müssten zunächst, wie auch bei neuen Produktspezifikationen, entsprechende Anpassungen am Prozessmodell vorgenommen werden. Dies ist allerdings sehr aufwendig, da der gesamte Entwicklungsprozess (siehe Abschnitt 2.2.1) nochmals durchgeführt werden müsste. Dies erfordert einen enormen personellen Aufwand und würde auch mit einem Produktionsstopp einhergehen, was insgesamt erhebliche Kosten verursacht.

In einer informationsgetriebenen autonomen Fertigungsanlage wären derartige manuelle Anpassungen bei Änderungen der Produktspezifikationen oder Ressourcen nicht nötig, da es überhaupt keine fest vorgegebenen Lösungen gibt (siehe Abbildung 1.2c). Stattdessen werden, wie bereits erwähnt, zur Laufzeit geeignete Lösungen ausfindig gemacht, indem die zuvor gelernten, generellen Strukturen auf die aktuelle Domäne angewendet werden. Das heißt, das System kann mit verschiedenen Produktspezifikationen umgehen, ohne dass diese bei der Planung der Fabrik bekannt gewesen sein mussten.

Diese Flexibilität bezüglich der Produktspezifikation ist in Abbildung 1.2d und Abbildung 1.2e durch die grün gestrichelten Linien dargestellt, die damit den individuellen Weg der Werkteile durch die Fabrik symbolisieren sollen. Auch bezüglich sich ändernder Ressourcen kann in einem autonomen Fertigungsprozess der Produktionsablauf flexibel angepasst werden. Zur Illustration wurde in der Fabrik in Abbildung 1.2f wieder eine zusätzliche Maschine vom Typ B platziert. Diese wird nun unmittelbar beim Finden einer geeigneter Lösung berücksichtigt. Das heißt, sollte die ältere der beiden Maschinen durch ein weiteres Werkteil bereits belegt sein (siehe blaues Quadrat) oder aus anderen Gründen nicht mehr zur Verfügung stehen, kann das grüne Werkteil flexibel auf die neue Maschine ausweichen.

Im Falle von autonomen Fertigungsprozessen kann so problemlos sowohl auf neue Produktspezifikationen, als auch auf Änderungen bezüglich der Ressourcenverfügbarkeit reagiert werden, ohne dass manuell von außen eingegriffen werden muss. Damit können nun stark individualisierte Produkte in niedri-

ger Stückzahl zu marktgerechten Preisen produziert werden, da die mit einer expliziten Modellanpassung anfallenden Personalkosten beziehungsweise Gewinneinbußen durch Produktionsstopps wegfallen.

Insgesamt gehen Unternehmen in der Fertigungstechnik davon aus, dass sie durch Digitalisierung ihrer Produktionsanlagen bereits bis 2025 eine Produktivitätssteigerung von 6.3% – 9.8% des Gesamtumsatzes erzielen können [Siem 17].

Aus der großen Vision „Industrie 4.0“ ergeben sich schließlich auf technologischer Ebene viele interessante Teilfragestellungen. Neben der einfachen Anwendung von bereits bekannten Verfahren des Maschinellen Lernens zum Beispiel zur vorausschauenden Wartung von Maschinen, engl. „predictive maintenance“, die jetzt schon eingesetzt werden können, geht es auch um komplexe Lernverfahren für die Koordination autonomer Agenten, die bisher allerdings noch nicht im großen Stil in der industriellen Fertigung eingesetzt werden. Ein Grund dafür ist, dass noch nicht ausreichend Mechanismen zur Absicherung derartiger autonomer Systeme verfügbar sind. Gerade im industriellen Umfeld ist eine zuverlässige Qualitätssicherung allerdings unabdingbar. Das heißt, hier ergibt sich die Fragestellung, wie das Verhalten eines autonom agierenden Systems objektiv bewertet werden kann, um eben sicherzustellen, dass es zu keinem Fehlverhalten kommt. Zwar gibt es bereits vereinzelt Teillösungen, allerdings werden allumfassende Lösungen auf diesem Gebiet noch relativ offen diskutiert [Schn 17, Burt 17, Koop 18]. In diese Diskussion ordnet sich die Fragestellung der vorliegenden Arbeit ein, deren genaue Beiträge im folgenden Abschnitt ausführlich ausgeführt werden.

1.2 Beiträge der Arbeit

Wie bereits erwähnt, ordnet sich die vorliegende Arbeit in die Diskussion zur Qualitätssicherung Autonomer Systeme ein. Sowohl Burton et al. [Burt 17], als auch Koopmann und Wagner [Koop 18] führen in ihren Beiträgen aus, wie ein gesamtheitlicher Prozess aussehen könnte, der ein fehlerfreies Verhalten derartiger Systeme sicherstellt. Dabei sprechen sie ganz offen an, dass dafür nicht nur auf Prozessebene Anpassungen nötig sind (siehe Abschnitt 2.2.2), sondern gerade auch auf technologischer Ebene neue Lösungen benötigt werden, die der geänderten Systemlogik gerecht werden. Entsprechend sind die Beiträge dieser Arbeit als ebensolche technologischen Teillösungen zu bewerten, die als Bestandteil eines Qualitätssicherungsprozesses eingesetzt werden können.

Diese Teillösungen wurden im Rahmen dieser Arbeit ausgehend von einem Anwendungsfall in einer industriellen Produktionsanlage aufgesetzt. Im Speziellen geht es dabei um selbst-organisierende Industriesysteme (SOIS), die die Vision der „intelligenten Fabrik“ technologisch umsetzen soll. Der Begriff der Selbst-Organisation ist damit eine wissenschaftlich begründete Konkretisierung der „intelligenten Fabrik“ [Seru 05], die zudem den technologischen Aspekt der Autonomie verdeutlicht (siehe Abschnitt 2.1).

Grundsätzlich geht es aber weiterhin um die Fragestellung, wie das Verhalten eines solchen autonom agierenden Systems objektiv bewertet werden kann, um mögliches Fehlverhalten zu erkennen. Dazu liefert die vorliegende Arbeit einen Beitrag im Bereich der Laufzeitüberwachung derartiger Systeme. Im Speziellen sind die vorgestellten Lösungen Verfahren zur Erkennung und Analyse von Anomalien, die es ermöglichen, Daten, die zur Laufzeit sensorisch erfasst werden, auszuwerten, um so das Verhalten des Systems bewerten zu können. Der Fokus der vorliegenden Arbeit liegt dabei auf der Auswertung raumbezogener Daten, wie Bewegungsdaten der sich in der Fabrik befindenden Objekte oder räumlicher Beschreibungen des Gesamtsystemzustands.

Damit ermöglicht das in dieser Arbeit vorgelegte Konzept, dass erkannt wird, wenn sich ein Objekt anormal bewegt (siehe Abschnitt 3.6.1). Zudem können durch eine gesamtheitliche Auswertung der Einzelbewegungsdaten globale anormale Ereignisse wie der Ausfall einer Maschine detektiert werden (siehe Abschnitt 3.6.2). Daneben ermöglicht das vorgestellte Verfahren die objektive Bewertung weiterer für die Qualitätssicherung relevante Aspekte wie das Festfahren in Endlosschleifen (siehe Abschnitt 3.6.3), die Lastenverteilung im System (siehe Abschnitt 3.6.4) oder die Neuartigkeit von Prozessabläufen (siehe Abschnitt 3.6.5).

Neben der Kozeptionierung eines ebensolchen Verfahrens zur Erkennung und Analyse von Anomalien in einem derartigem Umfeld, liefert die vorliegende Arbeit zusätzlich Lösungen zur Skalierbarkeit und Onlinefähigkeit des vorgestellten Ansatzes. Dabei geht es darum, sicherzustellen, dass das Verfahren auch in der praktischen Anwendung, sowohl was die Effizienz (siehe Abschnitt 4.1), als auch die Echtzeitfähigkeit (siehe Abschnitt 4.2) angeht, überzeugt. Schließlich wird gezeigt, dass das vorgestellte Konzept nicht nur für die Auswertung einfacher Bewegungsdaten zum Einsatz kommen kann, sondern auch für komplexere Datenstrukturen mit räumlichen Bezug umgesetzt werden kann, um eine objektive Bewertung derselben vorzunehmen (siehe Abschnitt 5.1 und Abschnitt 5.2).

Insgesamt liefert diese Arbeit also einen Beitrag zur Diskussion zum Thema Qualitätssicherung in autonomen Systemen, wobei insbesondere auf den Anwendungsfall in einer sich selbst-organisierenden industriellen Produktionsanlage eingegangen wird. Die vorgestellten Lösungen dienen der Systemüberwachung zur Laufzeit und bilden damit einen wichtigen Bestandteil des Qualitätssicherungsprozesses (siehe Abschnitt 2.2).

1.3 Vorveröffentlichungen

Die beschriebenen Lösungsansätze wurden bereits auf internationalen Konferenzen oder als Journalbeitrag publiziert. Im folgenden Abschnitt werden daher die entsprechenden Publikationen und die beteiligten Autoren aufgeführt. Zusätzlich wird in den späteren Kapiteln nochmals ein Verweis auf die relevanten vorveröffentlichten Arbeiten zu finden sein.

Bei allen Publikationen stand Prof. Dr. Claudia Linnhoff-Popien als Doktor-mutter beratend zur Seite, im Besonderen bei Publikationen, bei denen sie auch als Koautorin geführt wird. Die weitere Aufteilung auf die beteiligten Autoren und ihren jeweiligen Beitrag zur Publikation sind im Folgenden im Detail aufgeschlüsselt.

M. Kiermeier, M. Werner, C. Linnhoff-Popien, H. Sauer und J. Wieghardt: Anomaly Detection in Self-Organizing Industrial Systems Using Pathlets, IEEE 18th International Conference on Industrial Technology (ICIT), 2017 [Kier 17c] In dieser Arbeit wird ein Konzept zur Erkennung und Analyse von Anomalien in raumbezogenen Daten eines SOIS vorgestellt. Die Annahme dahinter ist, dass sich trotz des hohen Freiheitsgrades des Systems wiederkehrende Substrukturen im Systemverhalten extrahieren lassen. Durch Kombination dieser Substrukturen kann dann das autonome Verhalten des Systems modelliert und evaluiert werden.

Dr. Martin Werner war für diese Arbeit dahingehend Ideengeber, als dass er das Kompressionsverfahren für Trajektorien *Pathlet Learning* als Verfahren zur Extraktion der wiederkehrenden Substrukturen vorgeschlagen hat. Bei der Analyse der Anforderungen und Definition der Evaluationsszenarien standen Horst Sauer und Dr. Jan Wieghardt von der Siemens AG München beratend zur Seite. Implementierung und Evaluierung des Ansatzes wurden vom Autor der vorliegenden Arbeit vorgenommen, wobei Dr. Martin Werner bei der Implementierung beratend zur Seite stand. Diese vorveröffentlichten Inhalte fließen vor allem in Kapitel 3 ein.

M. Kiermeier, H. Sauer und J. Wieghardt, Monitoring Self-Organizing Industrial Systems Using Sub-Trajectory Dictionaries, IEEE 15th International Conference on Industrial Informatics (INDIN), 2017 [Kier 17b] Diese Arbeit stellt weitere Anwendungsmöglichkeiten des in [Kier 17c] vorgestellten Ansatzes zur Anomalieerkennung vor. Insgesamt werden drei weitere Anomaliearten analysiert und anhand entsprechender Szenarien evaluiert.

Die Idee zu diesen Erweiterungen, sowie deren Implementierung stammen vom Autor der vorliegenden Arbeit, wobei Horst Sauer und Dr. Jan Wieghardt wieder als Kritikgeber und Diskussionspartner zur Verfügung standen. Die Ergebnisse dieser Arbeit sind im Abschnitt 3.6 wiederzufinden.

M. Kiermeier, M. Werner, H. Sauer und J. Wieghardt, Building Scalable Models for Anomaly Detection in Self-Organizing Industrial Systems, IEEE 15th International Conference on Industrial Informatics (INDIN), 2017 [Kier 17d] In dieser Arbeit wird ein alternatives Verfahren zur Extraktion wiederkehrender Substrukturen für das in [Kier 17c] vorgestellte Konzept zur Anomalieerkennung präsentiert. Durch die Verwendung von SEQUITUR, einem Grammatikinduktionsverfahren, wird der Extraktionsprozess auch für hochauflösende raumbezogene Daten skalierbar.

Die Idee dazu stammt vom Autor der vorliegenden Arbeit, der auch die Implementierung und Evaluation übernommen hat. Dr. Martin Werner, Horst Sauer und Dr. Jan Wieghardt standen bei dieser Arbeit als Diskussionspartner zur Verfügung. Dieser Beitrag fließt hauptsächlich in Abschnitt 4.1 ein.

M. Kiermeier, Online Reverse Subpattern Matching for Reproduction of Trajectories from Sub-Paths, 17th International Conference on Computational Science and Its Applications (ICCSA), 2017 [Kier 17a] In dieser Veröffentlichung wird eine Datenstruktur vorgestellt, die eine effiziente Rekonstruktion von raumbezogenen Daten aus Substrukturen entsprechend des Anomalieerkennungsansatzes aus [Kier 17c] zur Laufzeit ermöglicht. Dazu werden die Einträge aus den Substrukturverzeichnissen über so genannte Kantenlisten indiziert, worüber sehr effizient Kandidaten- und übereinstimmende Substrukturen identifiziert werden können.

Das Konzept dahinter, sowie Implementierung und Evaluation des Ansatzes sind vom Autor der vorliegenden Arbeit. Der Inhalt dieses Beitrags findet sich in Abschnitt 4.2 wieder.

M. Kiermeier, T. Phan, H. Sauer und J. Wieghardt, Monitoring Autonomous Agents in Self-Organizing Industrial Systems, IEEE 16th International Conference on Industrial Informatics (INDIN), 2017 [Kier 18c]

Diese Arbeit ist einer Erweiterung des Anomalieerkennungsansatzes aus [Kier 17c] auf Bewegungsdaten von Objekten mit variabler Geschwindigkeit. Dazu wird das Substrukturverzeichnis um Zeitdauermodelle ergänzt, sodass für jeden Eintrag auch ein Modell für dessen zeitliches Verhalten vorliegt. Zusätzlich werden Bewegungsparameter definiert, um Anomalien im Bewegungsverhalten der sich autonom bewegenden Objekte zu detektieren und analysieren. Diese Erweiterungen wurden vom Autor der vorliegenden Arbeit entwickelt und umgesetzt. Die Daten zur Evaluation des Ansatzes stammen aus einer Simulation von Thomy Phan, der darüber hinaus mit seinem Wissen über Autonome Systeme ein wichtiger Diskussionspartner war. Horst Sauer und Dr. Jan Wieghardt standen zusätzlich für hilfreiche Anregungen zur Verfügung. Diese Veröffentlichung fließt vor allem in Abschnitt 5.1 ein.

M. Kiermeier, S. Feld, T. Phan und C. Linnhoff-Popien, Anomaly Detection in Spatial Layer Models of Autonomous Agents, 18th International Conference on Intelligent Data Engineering and Automated Learning (IDEAL), 2018 [Kier 18b]

In dieser Arbeit wird eine Umsetzung des Konzeptes aus [Kier 17c] für räumliche Schichtmodelle, die zur Zustandsbeschreibung von Agentensystemen verwendet werden, vorgestellt. Dazu werden die Schichtmodelle mittels rekursiver Koordinaten-Bisektion, engl. „recursive coordinate bisection“ (RCB), zerteilt, um anschließend gemäß des Konzepts aus [Kier 17c] ein Verzeichnis von Substrukturen für die Erkennung und Analyse von Anomalien zu erhalten.

Die Idee, Umsetzung und Evaluation dieses Ansatzes kommen vom Autor der vorliegenden Arbeit. Sebastian Feld und Thomy Phan waren insbesondere bei der Aufbereitung der Ergebnisse wichtige Kritikgeber. Der Inhalt dieser Arbeit findet sich in Abschnitt 5.2 wieder.

1.4 Aufbau der Arbeit

Die vorliegende Arbeit unterteilt sich in insgesamt sechs Kapitel. Kapitel 1 folgt das Grundlagenkapitel 2. In diesem werden relevante Grundlagen zum Thema selbst-organisierende Industriesysteme (siehe Abschnitt 2.1), Qualitätssicherung in Industriesystemen (siehe Abschnitt 2.2), sowie Laufzeitüberwachung und Anomalieerkennung (siehe Abschnitt 2.3) erläutert.

In Kapitel 3 wird das Konzept zur Laufzeitüberwachung mittels Substrukturverzeichnissen eingeführt und eine Umsetzung desselben für Bewegungsdaten von Objekten mit konstanter Geschwindigkeit evaluiert (siehe Abschnitt 3.6). Die Themen Skalierbarkeit und Onlinefähigkeit des Verfahrens werden anschließend in Kapitel 4 ausgeführt. Dafür wird in Abschnitt 4.1 ein alternatives Verfahren zur Substrukturextraktion vorgestellt und dessen Skalierbarkeit evaluiert, bevor in Abschnitt 4.2 eine geeignete Datenstruktur für einen effizienten Rekonstruktionsprozess zur Laufzeit präsentiert wird.

Den dritten inhaltliche Schwerpunkt dieser Arbeit bildet Kapitel 5, in dem Erweiterungen zur Laufzeitüberwachung komplexerer raumbezogener Datenstrukturen in SOIS vorgestellt werden. Dazu wird das Konzept aus Kapitel 3 für Bewegungsdaten von Objekten mit variabler Geschwindigkeit (siehe Abschnitt 5.1), sowie für Zustandsbeschreibungsdaten des Gesamtsystems (siehe Abschnitt 5.2) umgesetzt und evaluiert.

Die Arbeit schließt in Kapitel 6 mit einer Zusammenfassung und einem Ausblick ab.

2 Grundlagen

In diesem Kapitel werden Grundlagen erläutert, wie technische Ansätze zur Realisierung von SOIS und Ansätze auf Prozessebene zur Qualitätssicherung in Industriesystemen. Daneben wird auch das grundsätzliche Vorgehen zur Laufzeitüberwachung und Anomalieerkennung erläutert.

2.1 Selbst-organisierende Industriesysteme

In der vorliegenden Arbeit ist ein SOIS ein Agentensystem, dessen Agenten autonom agieren [Pana 05]. Das heißt, die Agenten – im Fall eines SOIS wären das entweder mobile Werkteile oder Roboter – treffen eigenständig zur Laufzeit Entscheidungen. Dafür kommen sowohl Verfahren des *Maschinellen Lernens* (siehe Abschnitt 2.1.1), als auch das Prinzip des *Online Planens und Entscheidens* (siehe Abschnitt 2.1.2) zum Einsatz.

In beiden Fällen wird eine enorme Menge an Daten ausgewertet. Die in der vorliegenden Arbeit vorgestellten Lösungsansätze beziehen sich allerdings ausschließlich auf die raumbezogenen Daten, die zur Koordination der mobilen Werkteile beziehungsweise Roboter in diesem Zusammenhang in einem SOIS erfasst werden (siehe Abschnitt 2.1.3). Im Speziellen ermöglichen die vorgestellten Ansätze, diese Daten zur Laufzeit zu analysieren und potentielle Anomalien zu detektieren, um darüber die Qualität der Erzeugnisse im SOIS sicherzustellen (siehe Abschnitt 2.2 und Abschnitt 2.3).

2.1.1 Maschinelles Lernen

Analog zum menschlichen Prozess des Lernens, geht es beim Maschinellen Lernen darum, aus Erfahrungen zu lernen. Das grundsätzliche Vorgehen dafür ist in Abbildung 2.1 skizziert. Zunächst wird eine Menge an Beobachtungen ausgewertet, um die den Daten zugrundeliegende Struktur zu erkennen. Diese generelle Struktur wird in Form eines Modells hinterlegt. Schließlich kann das Wissen abgefragt werden, indem das Modell auf eine bestimmte Domäne angewendet wird. Im Konkreten werden dann noch abhängig von der Form der vorhandenen Daten drei Arten des Maschinellen Lernens unterschieden [Russ 10]:

- **Überwachtes Lernen**, engl. „supervised learning“: Sind die zur Verfügung stehenden Beobachtungen Wertepaare der Form (*Eingabewert, Ausgabewert*), kann eine Funktion erlernt werden, mit der auch unbekannte

Eingabewerte auf einen entsprechenden Ausgabewert abgebildet werden können. Man unterscheidet die Vorhersage diskreter (**Klassifikation**) oder kontinuierlicher (**Regression**) Werte.

- **Unüberwachtes Lernen**, engl. „unsupervised learning“: Stehen keine Wertepaare, sondern lediglich eine ungeordnete Menge an Daten zur Verfügung, wird darin ganz generell nach Zusammenhängen oder Mustern gesucht. Dazu können, wie zum Beispiel beim **Clustering**, die Datenpunkte in Gruppen unterteilt werden, sodass entsprechend eines Distanzmaßes ähnliche Beobachtungen derselben Gruppe zugeordnet werden und damit von anderen getrennt werden, denen sie nicht so ähnlich sind.
- **Bestärkendes Lernen**, engl. „reinforcement learning“: Beim bestärkenden Lernen sind zunächst keine Beobachtungen vorhanden. Stattdessen werden die Daten situativ erzeugt, indem ein Agent mit seiner Umgebung interagiert. Jede Aktion des Agenten wird dabei entsprechen einer *Bewertungsfunktion*, engl. „reward function“, bewertet. Damit entstehen Beispielausprägungen für gute bzw. schlechte Handlungen. Diese können nun gemäß überwachtem Lernen verallgemeinert werden. Das dabei entstehende Modell wird dann als *Strategie*, engl. „policy“, bezeichnet, da daraus Handlungsempfehlungen für den Agenten abgeleitet werden können.

Für das Online Planen und Entscheiden in SOIS sind letztendlich das überwachte und bestärkende Lernen relevant (siehe Abschnitt 2.1.2). Im Speziellen werden dabei in den meisten Fällen Neuronale Netze für das überwachte Lernen genutzt, um dem Grad der Komplexität der zu lernenden Modelle gerecht zu werden. Entsprechend wird im Folgenden auf Neuronale Netze und ihre wichtigsten Erweiterungen eingegangen, soweit dies für das weitere Verständnis der folgenden Arbeit notwendig ist.

Neuronale Netze Grundsätzlich besteht ein Neuronales Netz, engl. „neural network“, aus einer Menge von Neuronen, engl. „neurons“, die schichtenweise angeordnet sind. Ein entsprechendes Beispiel eines neuronalen Netzes ist in Abbildung 2.2a zu sehen. In diesem Fall gibt es drei Eingabeneuronen x_1 , x_2 und x_3 auf der Eingabeschicht (Layer 1), engl. „input layer“, gefolgt von einer versteckten Schicht (Layer 2), engl. „hidden layer“, mit vier Neuronen und schließlich eine Ausgabeschicht (Layer 3), engl. „output layer“, mit zwei Neuronen. Die Neuronen benachbarter Schichten sind dabei über Kanten miteinander verbunden.

Wie bereits erwähnt, ist das Ziel beim überwachten Lernen aus Beispielausprägungen der Form (*Eingabewert*, *Ausgabewert*) ein allgemeines Modell zu lernen, um damit auch Ausgabewerte für noch unbekannte Eingabewerte vorherzusagen. Dazu wird bei Neuronalen Netzen nach dem Prinzip der Rückführung, engl. „backpropagation“, für jede Kante ein Gewicht w und für jedes

Neuron einer versteckten Schicht ein Bias b gelernt, sodass nach sukzessiver Ausführung der Aktivierungsfunktionen σ in den Neuronen, die den jeweilige Eingabewert mit den gelernten Gewichten und Bias verrechnen, sich am Ende der gesuchte Ausgabewert ergibt.

Im Beispiel in Abbildung 2.2a sieht die Aktivierungsfunktion σ die Addition des Bias mit der gewichteten Summe der Eingabewerte vor ($b_3 + x_1 * w_{13} + x_2 * w_{23} + x_3 * w_{33}$). Das Ergebnis z_{23} wird über die ausgehenden Kanten an die Neuronen der nächsten Schicht weitergegeben. Dieses Prozedere wird fortgeführt, bis die Ausgabewerte auf der letzten Schicht abgegriffen werden können. Die Anzahl der verwendeten Neuronen, sowie die Anordnung in den Schichten kann beliebig variieren beziehungsweise muss entsprechend der Anwendung gewählt werden. Allerdings spricht man bei einer Architektur mit mehr als zwei oder drei Schichten von so genannten tiefen Neuronalen Netzen, engl. „deep neural networks“.

Darüber hinaus gibt es noch die faltenden Neuronalen Netze, engl. „convolutional neural networks“ (CNN). Diese werden insbesondere bei der Bildverarbeitung sehr erfolgreich eingesetzt [Kriz 12], da sie eine Möglichkeit bieten, auch 2- bzw. 3-dimensionale Eingaben zu verarbeiten. Dazu kommen sogenannte faltende Schichten, engl. "convolutional layer", Max-zusammenfassende Schichten, engl. „max-pooling layer“ und glättende Schichten, engl. „flattening layer“ zum Einsatz.

Deren Funktionsweise ist in Abbildung 2.2b für eine 2-dimensionale Beispiel-eingabe skizziert. Für die faltende Schicht werden – erneut entsprechend des Rückführungsprinzips – Filtermasken gelernt, die mit dem Eingabebild verrechnet werden, indem diese über die Bildfläche verschoben werden. Dabei entstehen abhängig von der Anzahl und Größe der Masken mehrere kleinere Bilder. Zusätzlich können Max-zusammenfassende Schichten verwendet werden, um durch weitere Kompression der Informationen generelle Strukturen erkennen zu können. Dabei wird jede Teilfläche auf ihren maximalen Wert reduziert. Diese beiden Schichtarten können den Anwendungsanforderungen entsprechend implementiert und angeordnete werden.

Im letzten Schritt werden dann die reduzierten Bilder schließlich durch eine glättende Schicht wieder in Form eines 1-dimensionalen Vektors gebracht, der entsprechend zu einem Ausgabewert ausgewertet werden kann.

Der große Vorteil dieser faltenden Neuronalen Netze ist, dass im Lernprozess die räumlichen Zusammenhänge explizit bewahrt werden. Da zudem, wie bereits erwähnt, dieses Verfahren nicht nur für 2-dimensionale Bilder, sondern auch für 3-dimensionale Eingaben genutzt werden kann, werden diese CNNs damit auch für Lernaufgaben mit 3-dimensionalen Zustandsbeschreibungsdaten (siehe Abschnitt 2.1.3.3) als Eingabewert genutzt.

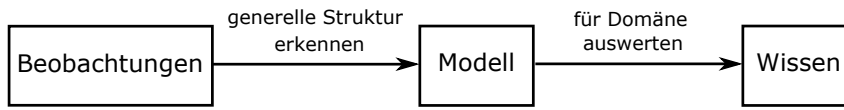
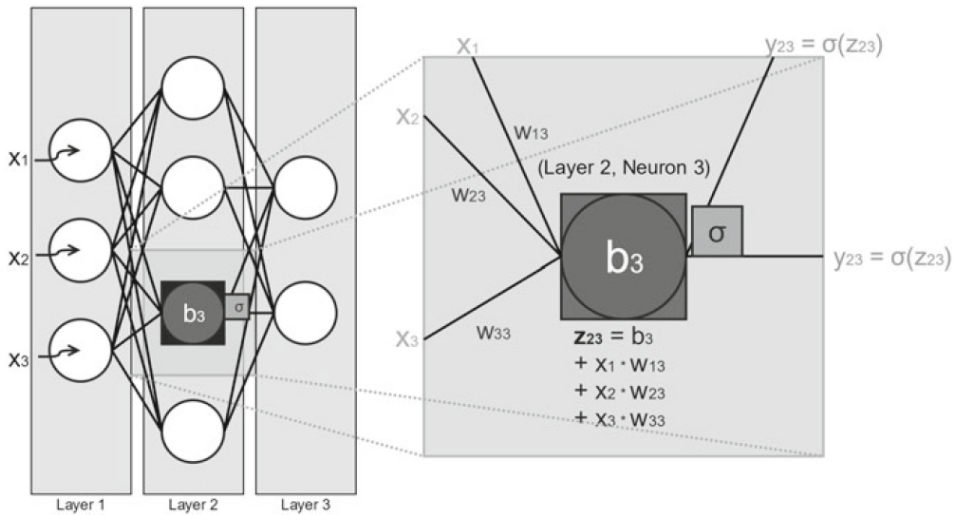
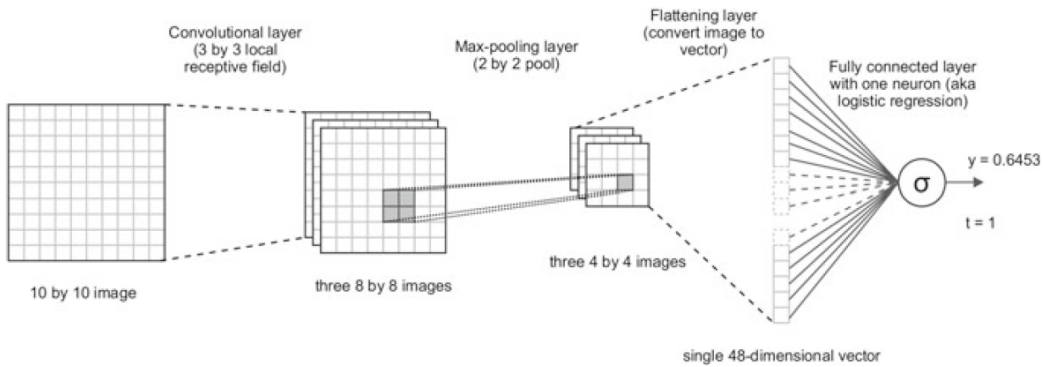


Abbildung 2.1: Vorgehen bei Maschinellern Lernen.



(a) Funktionsweise eines Neuronalen Netzes [Skan 18].



(b) Funktionsweise eines faltenden Neuronalen Netzes [Skan 18].

Abbildung 2.2: Grundlagen Neuronaler Netze.

2.1.2 Online Planen und Entscheiden

Das Vorgehen beim Online Planen und Entscheiden in einem SOIS soll im Folgenden anhand einer Beispielarchitektur von Gabor et al. für CPS erläutert werden [Gabo 16]. Diese Architektur ist generisch für Intelligente Cyberphysische Systeme entwickelt worden und damit auch für SOIS einsetzbar. Eine vereinfachte Darstellung der Architektur ist in Abbildung 2.3 zu sehen. Die drei Hauptbestandteile dieser Systemstruktur sind die physikalische Welt, ein kognitives System und ein Simulationsmodell. Für den Online Planungs- und Entscheidungsprozess wird eine Repräsentation der physikalischen Welt in Form von Sensordaten an das kognitive System übertragen. Dieses trifft eine Auswahl an möglichen Aktionen (Kandidatenauswahl), um Pläne (Sequenz an Aktionen) zu erstellen, die in einem Simulationsmodell ausgeführt werden. Dabei wird versucht die Auswirkungen der Kombination der einzelnen Aktionen auf ihre Umwelt möglichst konkret vorherzusagen. Das kognitive System kann dann darauf basierend die entsprechend einer globalen Zielvorgabe beste Aktion auswählen. Diese Aktion wird schließlich in der physikalischen Welt in die Tat umgesetzt.

Auf ein SOIS mit mobilen Werkteilen übertragen, bedeutet das, dass die Ereignisse der physikalischen Welt hinlänglich mit Sensoren erfasst werden müssen (aktuelle Position der Werkteile, Position und aktuelle Verfügbarkeit der Maschinen, etc). Das kognitive System ist dann das Ergebnis eines Lernprozesses. Für das autonome Routing der mobilen Werkteile könnte zum Beispiel über bestärkendes Lernen, ein entsprechendes Neuronales Netz trainiert werden (siehe Abschnitt 2.1.1). Das Ergebnis dieses Lernprozesses (*Strategie*) kann als kognitives System eingesetzt werden, das eine Auswahl an entsprechend des trainierten Modells besten Aktionen erzeugt.

Damit lässt sich über das Prinzip des Online Planens und Entscheidens die Vision der „intelligenten Fabrik“ technologisch als SOIS umsetzen. Das System trifft informationsgetrieben selbstständig Entscheidungen und bietet damit die nötige Flexibilität, sich den gegebenen Anforderungen zu jeder Zeit dynamisch anzupassen. Zu beachten ist hier allerdings, dass je umfangreicher die Umgebungsänderungen ausfallen können, auch die Anzahl an möglichen Zuständen zunimmt, die vom System bewältigt werden müssen. Im Falle von Systemen mit entsprechend hohen Freiheitsgraden spricht man daher auch von einer regelrechten Explosion des Zustandsraumes, engl. „state space explosion“, [Eber 15]. Diese Zustandsraumexplosion ist eine der zentralen Herausforderung für die Qualitätssicherung in SOIS (siehe Abschnitt 2.2.2).

2.1.3 Raumbezogene Daten in SOIS

Wie in Abschnitt 2.1.2 erläutert, sind Sensordaten, die Ereignisse aus der Umwelt erfassen, die Grundlage für die Umsetzung von SOIS. Die in der vorliegenden Arbeit vorgestellten Lösungsansätze beziehen sich ausschließlich auf raumbezogene Daten als Untermenge aller Daten, die in einem SOIS erfasst werden,

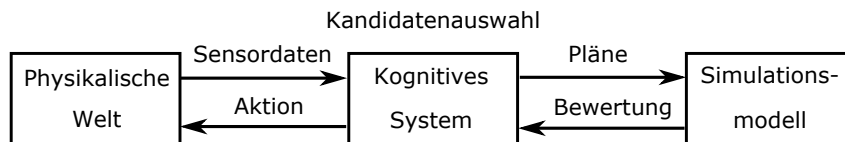


Abbildung 2.3: Beispielarchitektur für Online Planen und Entscheiden (adaptiert von [Kier 18a]).

da diese zur Koordination der mobilen Werkteile beziehungsweise Roboter verwendet werden und damit eine zentrale Bedeutung für die Qualitätssicherung in einem SOIS haben. Im Folgenden werden die drei für diese Arbeit relevanten Arten von raumbezogenen Daten in SOIS eingeführt und ihre jeweiligen Eigenschaften erläutert.

2.1.3.1 Bewegungsdaten von Objekten mit konstanter Geschwindigkeit

Bewegungsdaten, auch (räumliche) Trajektorien genannt, sind Pfade von sich bewegenden Objekten. Diese werden als eine Serie von chronologisch geordneten Punkten $p_1 \rightarrow p_2 \rightarrow \dots \rightarrow p_n$ dargestellt, wobei jeder Punkt p durch seine raumbezogenen Koordinaten x und y , sowie einen Zeitstempel t definiert ist: $p = (x, y, t)$ [Zhen 11b].

In SOIS werden derartige raumbezogene Daten zum Beispiel durch entsprechende Sensoren an mobilen Werkteilen oder Robotern erzeugt. Im Falle von $|t_i - t_{i+1}| = const.$, bewegen sich die Objekte mit konstanter Geschwindigkeit. Diese Eigenschaft trifft man in SOIS an, die eine Fließbandarchitektur haben, das heißt, die Werkteile über Montagebänder von einer Maschine zur nächsten transportiert werden. Eine wichtige Eigenschaft hierbei ist, dass damit die Anzahl an möglichen Pfaden, die ein Werkstück erzeugen kann, damit auch automatisch auf die Kombination der vorhandenen Fließbandabschnitte beschränkt ist.

2.1.3.2 Bewegungsdaten von Objekten mit variabler Geschwindigkeit

Im Fall, dass $|t_i - t_{i+1}| \in \mathbb{R}$, bewegen sich die Objekte offensichtlich mit variabler Geschwindigkeit. Diese Art von raumbezogenen Daten ergibt sich in einem SOIS beim Einsatz von mobilen Robotern, die die Werkteile von Maschine zu Maschine transportieren und dabei nicht auf vorgegebene Transportstraßen festgelegt sind. Dadurch erhöht sich natürlich die Anzahl möglicher Pfade außerordentlich. Allerdings macht man sich dadurch unabhängig von den statisch vorgegeben Verbindungen zwischen Maschinen, möglichen Rückstaus auf den Montagebändern oder deren möglichen temporären Ausfällen.

2.1.3.3 Zustandsbeschreibungsdaten des Gesamtsystems

Wie in Abschnitt 2.1.2 beschrieben wird beim bestärkenden Lernen aus Beispielausprägungen für (*Zustand*, *Aktion*, *Bewertung*) eine *Strategie* gelernt, die

in SOIS als kognitives System eingesetzt werden kann. Eine große Herausforderung ist dabei, eine geeignete Form zur Darstellung des Systemzustands zu finden. Es hat sich gezeigt, dass hier räumliche Schichtmodelle, engl. „spatial layer models“ (SLM), sinnvoll sind. Zum einen erhalten sie die räumliche Struktur. Zum anderen lässt sich diese Datenform gut von faltenden Neuronalen Netzen verarbeiten. Ein räumliches Schichtenmodell besteht dabei aus unterschiedlichen Ebenen, wobei jede Ebene für ein bestimmtes Merkmal dessen räumliche Lage darstellt (siehe Abschnitt 5.2.2).

2.2 Qualitätssicherung in Industriesystemen

Bei der Qualitätssicherung in Industriesystemen geht es darum, dass das System allen Anforderungen gerecht wird und dies auch belegt werden kann. Entsprechend hat sich der Begriff der Validierung und Verifikation (V&V) etabliert, deren offiziellen Definitionen nach [IEEE610 90] lauten:

Validation: *The process of evaluating a system or component during or at the end of the development process to determine whether it satisfies specific requirements.* [deutsch: Der Prozess zur Evaluierung eines Systems oder eines Bausteins während oder am Ende eines Entwicklungsprozesses, um zu entscheiden, ob die spezifizierten Anforderungen erfüllt werden.]

Verification: *The process of evaluating a system or component to determine whether the products of a given development phase satisfy the conditions imposed at the start of the phase.* [deutsch: Der Prozess zur Evaluierung eines Systems oder eines Bausteins, um zu entscheiden, ob die Erzeugnisse einer bestimmten Entwicklungsphase die zuvor an diese Phase gestellten Bedingungen erfüllen.]

2.2.1 V-Modell

Das V-Modell ist ein weitverbreitetes Vorgehensmodell zur Systementwicklung (siehe Abbildung 2.4), wobei die linke Seite den Entwicklungsprozess und die rechte Seite entsprechende V&V-Maßnahmen zur Qualitätssicherung zeigen. Sowie die Entwicklungsanforderungen mit jeder Phase weiter konkretisiert und modularisiert werden, werden auf der V&V-Seite die entstandenen Bausteine sukzessive in umgekehrter Reihenfolge getestet und verifiziert. Am Ende kann ein vollständig evaluiertes System in Betrieb genommen werden.

Voraussetzung für dieses Vorgehen ist das Vorhandensein von entsprechenden Anforderungsspezifikationen. Ausgehend von diesen Anforderungen wird der Problemraum des Systems festgesetzt, sodass darauf basierend der Lösungsraum definiert und in eine Architektur gegossen werden kann. Das Design verfeinert schließlich diese, sodass die eigentliche Umsetzung des Systems starten kann.

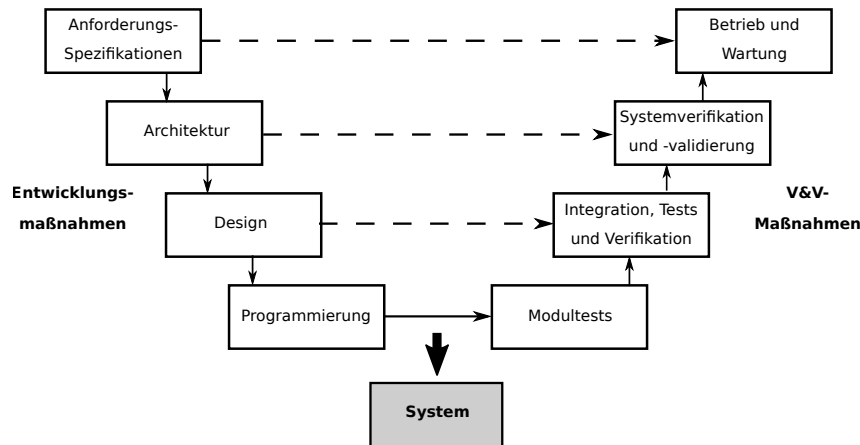


Abbildung 2.4: V-Modell (adaptiert von [Tamu 13]).

Dadurch, dass sowohl Problem- als auch Lösungsraum des Systems bereits zur Entwicklungszeit definiert werden, können diese auf der V&V-Seite entsprechend verifiziert werden. Dabei kommen entweder dynamische oder statische Verifikationsverfahren zum Einsatz.

Bei der **dynamischen** Verifikation (auch: Testen) wird der erzeugte Programmcode ausgeführt. Dazu werden Testfälle definiert, um zu überprüfen, ob das Ausführen des Programms in der jeweiligen Situation den gewünschten Effekt hat. Dieser Ansatz hat zwei Nachteile. Zum einen kann damit keine 100%-ige Abwesenheit von Fehlern bewiesen werden, da man in den seltensten Fällen alle möglichen Testfälle definieren und ausführen kann. Zum anderen muss ein so genanntes *Testorakel* vorhanden sein, um zu wissen, was der „gewünschte Effekt“ ist. Darauf basierend kann nun entschieden werden, ob mit dem erzielten Ergebnis der Test als bestanden gilt oder nicht.

Im Gegensatz dazu wird beim **statischen** Verifizieren (auch: Analyse) unabhängig vom produzierten Programmcode die Korrektheit des Systems überprüft. Dazu werden Systembeschreibungen in Form eines Modells formal gegen erforderliche Spezifikationen verifiziert. Dabei kommen häufig formale Methoden zur Beweisführung zum Einsatz. Voraussetzung für diese Art der Verifikation ist allerdings, dass sich das System entsprechend modellieren lässt, was mit steigender Komplexität des Systems immer weniger praktikabel wird.

2.2.2 Validierung und Verifikation in adaptiven Systemen

In adaptiven Systemen, wie einem SOIS, entstehen durch den hohen Freiheitsgrad des Systems ganz neue Herausforderungen für die Qualitätssicherung (siehe Abschnitt 2.1.2). Für dynamische Verifikationsverfahren bedeutet das zum Beispiel, dass die Auswahl von Testfällen noch kritischer wird, da der Anspruch alles „Durchzutesten“ durch eben jenen hohen Freiheitsgrad noch mehr in weite Ferne rückt.

Hier ist die zentrale Herausforderung, geeignete Testfälle zu generieren, die

trotz des großen Lösungsraums die relevanten Situationen abdecken. Ähnlich problematisch ist in derartig adaptiven Systemen das Vorhalten eines Testorakels, da es unrealistisch ist, für alle möglichen Systemzustände einen entsprechenden Referenzwert vorzuhalten, gegen den die Testergebnisse geprüft werden können

Ähnliche Schwierigkeiten warten bei der statischen Verifikation auf, da das Systemverhalten eines adaptiven Systems noch schwieriger zu formalisieren ist, als von nicht adaptiven. Dazu kommt, dass in adaptiven Systemen die Anforderungsspezifikationen an das System nicht im erforderlichen Detailgrad vorliegen, beziehungsweise man eben bewusst den Lösungsraum noch nicht zur Entwicklungszeit so strikt festsetzt, um dem System die Flexibilität zu erhalten.

Damit ist das im vorherigen Abschnitt beschriebene Vorgehen zur Qualitätssicherung nicht mehr hinreichend [Koop 16]. Zur Design- und Entwicklungszeit liegen nicht alle erforderlichen Informationen vor. Stattdessen muss in solchen Fällen der V&V-Prozess mit den entsprechenden Verfahren auch auf die Laufzeit, das heißt, die Betriebszeit des Systems, ausgeweitet werden [De L 13, Trit 16]. In der Literatur gibt es dazu die Überlegung, dass man im Speziellen statische Verifikationsverfahren entwickeln könnte, die eine formale Konformitätsprüfung des Systems zur Laufzeit ermöglichen [Tamu 13, Eder 13, Eber 17]. Die große Herausforderung dabei bleibt allerdings, dass dafür exakte Systemmodelle benötigt werden.

Im Rahmen dieser Arbeit wird eine Alternative vorgestellt, bei der ein Systemmodell *aus Beobachtungen* erstellt wird und damit „normales“ Verhalten beschreibt. Dieser Ansatz bietet natürlich keine vollständige Verifikation des Systems, ist aber eine Möglichkeit das Verhalten des Systems trotz des enorm großen Freiheitsgrades zur Laufzeit zu bewerten. Auf technischer Ebene entspricht dieses Vorgehen dem Prinzip der Laufzeitüberwachung und Anomalieerkennung (siehe Abschnitt 2.3).

2.3 Laufzeitüberwachung und Anomalieerkennung

Neben den zuvor beschriebenen V&V-Maßnahmen zur Systemvalidierung zur Entwicklungszeit, gibt es in Industriesystemen die Laufzeitüberwachung, engl. „online monitoring“, um die Qualität auch während des Betriebs weiterhin sicherzustellen [Venk 03a, Venk 03b, Venk 03c]. Dabei kommen Anomalieerkennungsverfahren zum Einsatz.

Die Idee von Anomalieerkennung, engl. „anomaly detection“, oder auch Ausreißerererkennung, engl. „outlier detection“, oder Neuartigkeitserkennung, engl. „novelty detection“, ist, zu erkennen, wenn neue (Test-)Daten in welcher Form auch immer von den Daten abweichen, die als Training zur Verfügung standen. In den meisten Fällen stehen dazu nur Daten zur Verfügung, die das

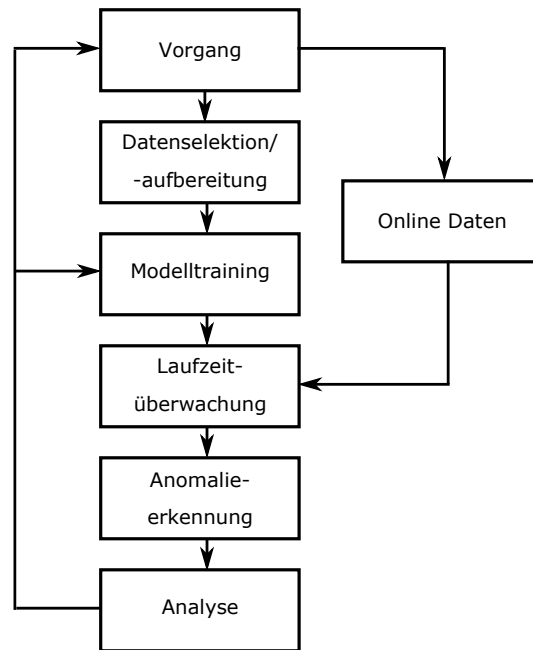


Abbildung 2.5: Grundlagen der Laufzeitüberwachung und Anomalieerkennung in SOIS.

„normale“ Verhalten zeigen und nicht solche, die auch anormales Verhalten charakterisieren. Daher handelt es sich bei Anomalieerkennung auch nicht um ein klassisches Klassifizierungsproblem, da es eben nur eine Klasse („normales“ Verhalten) gibt. Stattdessen muss ein Modell erstellt werden, das das normale Verhalten repräsentiert. Daten, die nicht diesem Modell entsprechen, werden als Anomalien deklariert. [Pime 14]

Bestehende Verfahren zur Laufzeitüberwachung gehen dabei allerdings von einem nicht adaptiven System aus, das zum Beispiel entsprechend des V-Modells (siehe Abschnitt 2.2.1) bereits validiert wurde. Dadurch können die dazu verwendeten Anomaliedetektionsverfahren auch davon ausgehen, dass es zur Entwicklungszeit festgelegte Prozessabläufe und damit ein klar definiertes „normales“ Verhalten gibt, das beobachtet und entsprechend modelliert werden kann. Wie in Abschnitt 2.1.2 erläutert, liegen in adaptiven Systemen, wie einem SOIS, allerdings keine solchen festgelegten Prozessabläufe vor. Ganz im Gegenteil, das „normale“ Verhalten in einem SOIS kann so viele verschiedene Ausprägungen annehmen, dass es sich nicht ohne weiteres modellieren lässt. In der vorliegenden Arbeit werden daher Lösungsansätze vorgestellt, wie dieses Problem der Zustandsraumexplosion für die Anomalieerkennung gelöst werden kann (siehe Kapitel 3).

In Abbildung 2.5 ist zur Übersicht das Vorgehen für die Laufzeitüberwachung und Anomalieerkennung in SOIS dargestellt. Die Grundstruktur für eine derartige datenbasierte Prozessüberwachung wurde von [Ge 13] adaptiert, allerdings auf die für die vorliegende Arbeit relevanten Schritte reduziert und um SOIS-

spezifische Vorgänge ergänzt.

Grundsätzlich werden zunächst Daten ausgewählt, die zur Überwachung des Vorgangs (*Prozess*) geeignet sind. Oft müssen diese erst aufbereitet werden, bevor das eigentliche Modell trainiert werden kann. Dieses Modell bildet die Grundlage für die Laufzeitüberwachung. Dazu werden die Daten aus dem laufenden Betrieb mit Hilfe des Modells überprüft und als „normal“ oder anormal klassifiziert. Im Falle einer Anomalie folgt ein Analyseschritt, um Rückschlüsse für das Modell und den Prozess zu ziehen.

Dieser Schritt unterscheidet sich von nicht adaptiven Systemen, da in adaptiven Systemen eine Anomalie nicht automatisch ein Fehler ist, sondern unter Umständen lediglich die Folge einer ansonsten korrekten Anpassung des Systemverhaltens. In diesem Fall muss das Modell zur Anomalieerkennung angepasst werden, sodass das neue Verhalten nicht mehr als Anomalie klassifiziert wird. Sollte es sich tatsächlich um ein Fehlverhalten des Systems handeln, müssen Änderungen am Prozess selbst vorgenommen werden, damit solche Fehler nicht mehr vorkommen.

2.4 Zusammenfassung

In diesem Kapitel wurden die methodischen Grundlagen zusammengefasst, die für die weiteren Inhalte der vorliegenden Arbeit relevant sind. Dazu wurde zunächst die technische Umsetzung einer „intelligenten Fabrik“ als SOIS erläutert (siehe Abschnitt 2.1). Durch den Einsatz von Verfahren des Maschinellen Lernens (siehe Abschnitt 2.1.1), sowie dem Prinzip des Online Planens und Entscheidens (siehe Abschnitt 2.1.2) ist es möglich, ein System zu entwickeln, das informationsgesteuert Entscheidungen treffen kann und damit die Anforderung der Autonomie umsetzt. Des Weiteren wurden in Abschnitt 2.1.3 drei Typen von raumbezogenen Daten vorgestellt, die im Zusammenhang der Umsetzung von SOIS verarbeitet werden und deren Analyse Gegenstand der vorliegenden Arbeit ist. Es handelt sich dabei zum einen um klassische Bewegungsdaten, die die Positionen von mobilen Werkteilen oder Robotern in einem SOIS nachzeichnen, wobei hier zwischen Objekten mit konstanter oder variabler Geschwindigkeit unterschieden werden muss (siehe Abschnitt 2.1.3.1 und Abschnitt 2.1.3.2). Zum anderen wurden Zustandsbeschreibungsdaten als weitere Daten mit Raumbezug eingeführt, die in Form eines räumlichen Schichtenmodells für verschiedene Merkmale deren räumliche Lage darstellt (siehe Abschnitt 2.1.3.3).

Neben diesen technischen Grundlagen zur Realisierung eines SOIS wurde auch auf die Qualitätssicherung in Industriesystemen und entsprechende Verfahren auf Prozessebene eingegangen (siehe Abschnitt 2.2). Dafür wurde zunächst das V-Modell erläutert, das ein weitverbreitetes Verfahrensmodell zur Softwareentwicklung ist (siehe Abschnitt 2.2.1) und dessen Unzulänglichkeiten bei der Qualitätssicherung in adaptiven Systemen wie SOIS diskutiert (siehe Abschnitt 2.2.2). Sowohl der hohe Freiheitsgrad des Systems, als auch die Tatsa-

che, dass nicht alle Informationen bei der Planung des Systems vorliegen, sind beim statischen und dynamischen Verifizieren in seiner bisherigen Form nicht vorgesehen. Stattdessen werden neue Verfahren benötigt, deren Anwendung auf die Laufzeit ausgeweitet werden.

Eine Möglichkeit einer derartige Bewertung des Systemverhaltens zur Laufzeit ist das Prinzip der Laufzeitüberwachung und Anomalieerkennung, dessen grundsätzliches Vorgehen daher abschließend in Abschnitt 2.3 erläutert wurde. Damit wurde in diesem Kapitel in alle für die vorliegende Arbeit relevanten Themen eingeführt.

3 Laufzeitüberwachung mittels Substrukturverzeichnissen

Die große Herausforderung für die Qualitätssicherung in SOIS ist der enorm große Zustandsraum. Im folgenden Kapitel wird daher nun ein Konzept vorgestellt, das trotz dieses hohen Freiheitsgrads des Systems eine Erkennung und Analyse von Anomalien in raumbezogenen Daten eines SOIS ermöglicht. Das dabei für die Anomalieerkennung erstellte Modell kann als Systemmodell zur Verifikation des Systemverhaltens zur Laufzeit verwendet werden.

Das vorgestellte Konzept und dessen Anwendung auf raumbezogene Daten eines SOIS werden im Folgenden am Beispiel von Bewegungsdaten von Objekten mit konstanter Geschwindigkeit eingeführt.

Die Ergebnisse wurden bereits in [Kier 17c] und [Kier 17b] veröffentlicht. Umsetzung und Evaluation stammen in beiden Fällen vom Autor der vorliegenden Arbeit.

3.1 Herausforderungen und Lösungsansatz

Die große Herausforderung bei der Anomalieerkennung in SOIS ist die Zustandsraumexplosion (siehe Abschnitt 2.3). Diese soll im Folgenden am Beispiel des Routings von Werkstücken in einem SOIS erläutert werden.

In nicht adaptiven Industriesystemen wird zur Entwicklungszeit für jedes Werkteil festgelegt, welchen Weg es durch die Fabrik nehmen muss und welche Maschinen dabei angefahren werden, damit das gewünschte Produkt herauskommt. Damit hat man für jedes Produkt einen Referenzpfad, gegen den man zur Laufzeit die eingehenden Online Daten abgleichen und dabei Anomalien detektieren kann.

In SOIS dagegen entscheidet das System zur Laufzeit entsprechen der aktuellen Gegebenheiten, welche Route das Werkteil nehmen soll. Bei einer Fließbandarchitektur, bei der die Werkteile über Montagebänder von einer Maschine zur nächsten transportiert werden, bedeutet das, dass die Routen zwar an das durch die Montagebänder vorgegebene Wegesystem gebunden sind, aber an Kreuzungen Entscheidungen bezüglich des weiteren Pfades getroffen werden können.

Damit besteht der Zustandsraum beim Routing von Werkteilen aus möglichen Kombinationen der vorhandenen Fließbandabschnitte. Im Gegensatz zu nicht adaptiven Industriesystemen, bei denen sich der Zustandsraum auf die wenigen vordefinierten Pfade für jedes Produkt beschränkt, ist es daher bereits für

kleine SOIS äußerst ineffizient, für alle Produkte alle möglichen Referenzpfade zu hinterlegen und zur Laufzeit Vergleiche anzustellen, um Anomalien zu detektieren.

Stattdessen ist die Idee des hier vorgestellten Ansatzes, die Dimension des Zustandsraum zu reduzieren, indem nicht ganze Referenzpfade für ein Produkt, sondern Teilpfade hinterlegt werden, aus denen die möglichen Routen der Werkteile zusammengesetzt werden können. Dazu wird das System beobachtet, um wiederkehrende Substrukturen zu erkennen und entsprechend zu extrahieren. Wichtig ist hierbei, dass es sich nicht um primitive Substrukturen, wie die räumliche Anordnung der Fließbandabschnitte handelt, sondern um Teilpfade, die das Normalverhalten des Systems beschreiben. Das heißt, wenn auf Grund der vom System abzuarbeitenden Aufgaben ein Fließbandabschnitt nur in eine bestimmte Richtung genutzt wird, wird das durch eine entsprechende Substruktur festgehalten. Genauso würden bestimmte Maschinenabfolgen, die sich aus produktionstechnischen Gründen ergeben, durch entsprechende Substrukturen repräsentiert werden.

Letztendlich wird der Zustandsraum auf das beobachtete Normalverhalten des Systems reduziert. Routen, die sich nicht aus diesen produktionsspezifischen Teilpfaden zusammensetzen lassen, werden schließlich als Anomalien klassifiziert, da sie nicht im reduzierten Zustandsraum („Normalität“) enthalten sind. Insgesamt ermöglicht dieser Ansatz damit, den hohen Freiheitsgrad des Systems durch das Zusammensetzen entsprechender Substrukturen zu modellieren.

Das genaue Vorgehen zur Erkennung und Analyse von Anomalien in SOIS mit Hilfe von Substrukturverzeichnissen ist in Abbildung 3.1 zu sehen, wobei die einzelnen Schritte den entsprechenden Phasen der Laufzeitüberwachung aus Abschnitt 2.3 zugeordnet sind (siehe Abbildung 2.5). So entspricht die Extraktion der Substrukturen aus den Trainingsdaten dem Modelltraining. Basierend auf dem resultierenden Substrukturverzeichnis werden im Rahmen der Laufzeitüberwachung die Testdaten aus den Substrukturen des Modells rekonstruiert. Dabei anfallende Informationen, wie zum Beispiel der Grad der Rekonstruierbarkeit, werden dabei zur Anomalieerkennung und Analyse der detektieren Ausreißer verwendet.

3.2 Verwandte Arbeiten

Bei der Erkennung von Anomalien in Bewegungsdaten lassen sich grundsätzliche zwei Ansätze unterscheiden: Clustering- oder Klassifikationsverfahren. Beim Clustering sind Anomalien als Ausreißer definiert, die sich beim Clustern der Trainingsdaten nicht einem Cluster zuordnen lassen. Das heißt, wenn eine Beobachtung signifikant von allen anderen abweicht, wird diese als anomal klassifiziert. Dafür werden die Trajektorien entweder in ihrer Gänze (z.B. [Zhan 11]) oder als Teiltrajektorien (z.B. [Zhen 11a, Yuan 11]) miteinander verglichen und geclustert.

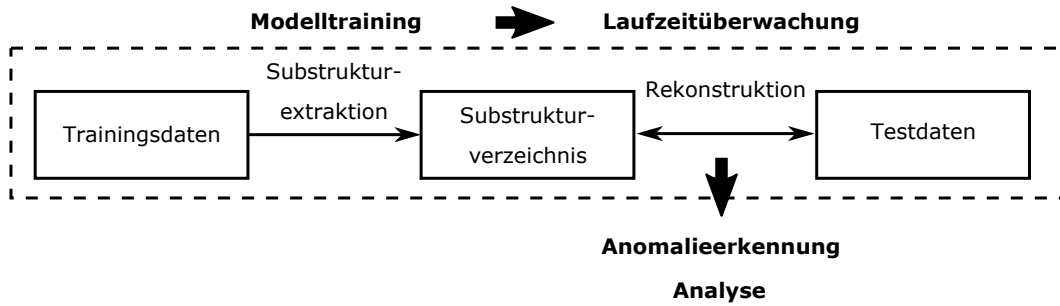


Abbildung 3.1: Übersicht Erkennung und Analyse von Anomalien in SOIS mittels Substrukturverzeichnissen.

Im Gegensatz dazu wird bei der Anomalieerkennung mittels Klassifikation zunächst – ausgehend von den vorhandenen Trainingsdaten – ein Modell erstellt, das das normale Verhalten abbildet. Damit sind Anomalien dann als Beobachtungen definiert, die von diesem Modell abweichen (z.B. [Pnue 07, Abe 06, Liao 10, Sill 08]). Auch wenn die klassifikationsbasierten Techniken durch diesen vorgeschalteten Lernschritt aufwändiger als die clusteringbasierten Verfahren sind, bieten die klassifikationsbasierten Methoden bei bestimmten Anwendungen deutliche Vorteile.

Der hier vorgestellte Ansatz ist ebenfalls ein Klassifikationsverfahren. Allerdings unterscheidet er sich in der Art des gelernten Modells von bestehenden Ansätzen. So wird das Normalverhalten in dem hier vorgestellten Verfahren nicht durch Beispiele für normale Trajektorien definiert, sondern über die in der Trainingsphase extrahierten wiederkehrenden Teilpfade. Das Kombinieren derselben zu ganzen Trajektorien ermöglicht schließlich die Modellierung des enorm großen Zustandsraum in einem SOIS.

Darüber hinaus können zwei Arten von Anomalien unterschieden werden, die bei der Betrachtung von Bewegungsdaten erkannt werden können: anormale Einzeltrajektorien (z.B. [Liu 14, Zhan 11, Lee 08, Chen 11, Ge 11, Zhan 12]) und anormale globale Ereignisse (z.B. [Liu 11, Chaw 12, Pan 13, Pang 11, Pang 13, Barr 11, Zhen 11a]). Bei ersterem geht es darum, einzelne Trajektorien oder Teiltrajektorien zu identifizieren, die nicht konform mit dem „normalen“ Modell sind. Im zweiten Fall dagegen werden Trajektorien dazu genutzt, Abnormalitäten in der zugrundeliegenden Welt zu erkennen. Beispiel für solche globalen Ereignisse wären Verkehrstaus, Straßensperrungen, o.ä..

Der in dieser Arbeit vorgestellte Ansatz ist diesbezüglich ein Hybridverfahren, mit dem sowohl anormale Einzeltrajektorien, als auch anormale globale Ereignisse detektieren werden können. Dies wird durch die Modellierung mittels der Substrukturen und die Informationen, die bei der Rekonstruktion der Testdaten anfallen, möglich.

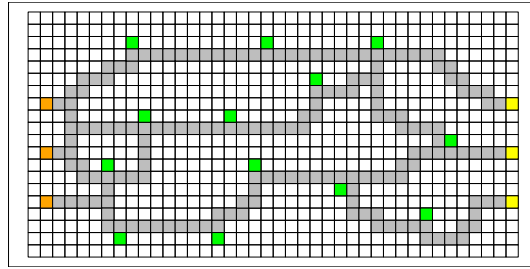


Abbildung 3.2: Simulationsmodell für ein SOIS mit drei Startpunkten (orange), drei Endpunkten (gelb) und 13 Produktionsstationen (grün), die durch Fließbänder (grau) verbunden sind.

3.3 Simulationsmodell

Zur Simulation von Routen mobiler Werkteile in einem SOIS wurde eine kleine Beispielfabrik gestaltet (siehe Abbildung 3.2). Die Grundfläche ist unterteilt in gleichgroße Rasterzellen. Dadurch können die räumlichen Informationen wie die Positionen der Werkteile und Maschinen entweder durch die x- und y-Koordinaten der Rasterzellen oder durch eindeutige Identifikationsnummern (durch Durchnummerieren aller Rasterzellen) beschrieben werden.

Im Modell gibt es:

- Startpunkte, an denen die Werkteile in die Fabrik kommen,
- Produktionsstationen, an denen die Werkteile bearbeitet werden können,
- Endpunkte, an denen die Werkteile die Fabrik verlassen und
- Fließbandelemente, über die die Werkteile von einer Produktionsstation zur nächsten gelangen.

Jede dieser Komponenten ist genau einer Rasterzelle zugeordnet und die Bewegung eines Werkteils ist auf dessen unmittelbar benachbarte Rasterzellen (rechts/links/unten/oben) beschränkt.

Zur Erzeugung von Trainings- und Testdatensätze werden schließlich Pfade von einem zufällig ausgewählten Startpunkt über eine Menge zufällig ausgewählter Maschinen zu einem zufälligen Endpunkt erzeugt, wobei stets der kürzeste Pfad über die Fließbandelemente gewählt wird. Damit erhält man eine Menge verschiedenartiger Ausführungspfade, die so auch in einem SOIS zu erwarten sind.

3.4 Substrukturextraktion

Zur Extraktion der Teilpfade, die das Substrukturverzeichnis bilden, wird *Pathlet Learning*, verwendet. Pathlet Learning wurde ursprünglich von Chen et al. zur Kompression von Trajektoriendaten eingeführt [Chen 13]. Ziel dabei ist

es, aus einer Menge an Trajektorien eine möglichst kleine Menge an Teilpfaden zu extrahieren, sodass damit weiterhin alle Trajektorien rekonstruierbar sind. Chen et al. formulieren dieses Problem als bedingtes Optimierungsproblem:

1. Die Größe des Pathletverzeichnis, das heißt, die Anzahl der Pathlets, die zur Rekonstruktion der Trajektorien benötigt werden, soll minimiert werden.
2. Die Anzahl der Pathlets, die zur Rekonstruktion einer Trajektorie benötigt werden, soll minimiert werden.

Die erste Bedingung garantiert die *Kompaktheit* des Pathletverzeichnis, wobei die zweite die *Effizienz* der Rekonstruktion sicherstellt.

Gegeben nun eine Menge T von n Trajektorien, sei \bar{P} die Menge aller Subtrajektorien der Trajektorien von T . Sei weiterhin $P(t)$ die Menge aller Subtrajektorien der Trajektorie t . Dann ist das gesuchte Pathletverzeichnis die Untermenge $P \subseteq \bar{P}$, die folgende Gleichung optimiert:

$$\min \sum_{p \in \bar{P}} x_p + \lambda \sum_{t \in T} \sum_{p \in P(t)} x_{t,p}, \quad (3.1)$$

so dass $x_{t,p} \leq x_p, \quad \forall p \in P(t), t \in T$ und

$$\sum_{p \in t, e \in p} x_{t,p} = 1, \quad \forall e \in t, t \in T,$$

mit $x_p \in \{0, 1\}$, wobei $x_p = 1$, falls $x_p \in P$, das heißt, das Pathlet p im Verzeichnis enthalten ist, und $x_p = 0$ sonst. Die zweite binäre Integervariable $x_{t,p} \in \{0, 1\}$ ist 1, wenn das Pathlet p zur Rekonstruktion von Trajektorie t benötigt wird, und 0 sonst. Schließlich reguliert der Parameter λ den Zielkonflikt zwischen *Kompaktheit* und *Effizienz*.

Statt das Optimierungsproblem entsprechend Gleichung 3.1 zu lösen, formulieren Chen et al. eine untere Grenze des Problems, die eine effiziente Approximation der Lösung liefert:

$$\min_{x_{t,p} \in \{0,1\}} \sum_{p \in P(t)} \left(\lambda + \frac{1}{|T(p)|} \right) x_{t,p}, \quad (3.2)$$

so dass $\sum_{e \in p, p \in P(t)} x_{t,p} = 1, \quad \forall e \in t,$

wobei $T(p)$ die Menge an Trajektorien $\{t_i\}$ ($i = \{1, 2, \dots, m\}$) ist mit Pathlet $p \in P(t_i)$, das heißt, p eine Teiltrajektorie von t_i ist.

Diese Gleichung kann nun unabhängig für jede Trajektorie gelöst werden. Das Lösen dieser Gleichung entspricht dabei der Suche nach einer Zerlegung für jede Trajektorie, sodass sowohl die Größe des Pathletverzeichnis, als auch die Anzahl an Pathlets, die zur Rekonstruktion der Trajektorien benötigt werden, minimiert werden. Um diese „optimale“ Zerlegung für jede Trajektorie zu

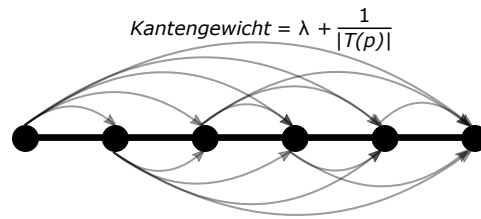


Abbildung 3.3: Trajektorienrepräsentation als gewichteter Graph.

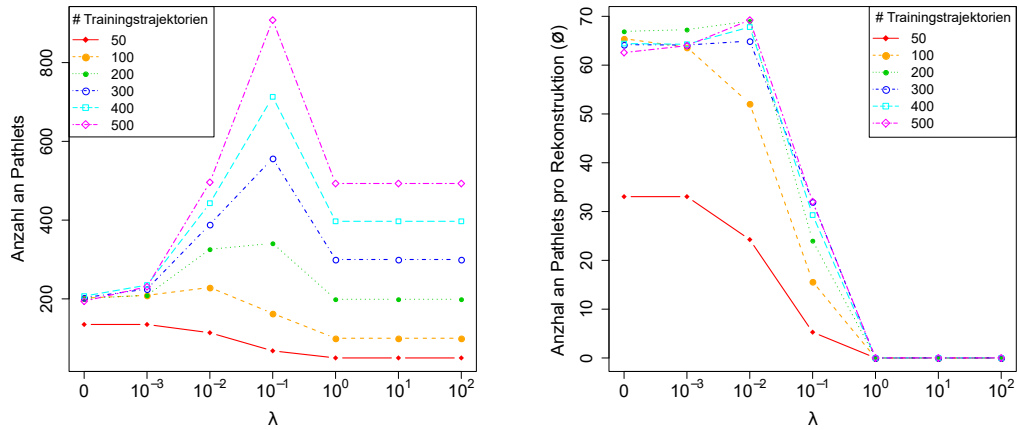
erhalten, wird das Problem auf das Kürzeste-Pfad-Problem übertragen. Dafür wird jede Trajektorie in einen gerichteten gewichteten Grafen transformiert (siehe Abbildung 3.3): die Punkte der Trajektorie entsprechen den Knoten, die Kanten repräsentieren alle möglichen Teiltrajektorien und die Gewichtung ist gegeben durch den Koeffizienten aus Gleichung 3.2. Der kürzeste Pfad kann dann durch dynamische Programmierungsansätze, engl. „dynamic programming“, wie zum Beispiel den Algorithmus von Dijkstra [Dijk 59] berechnet werden.

3.5 Substrukturverzeichnis

Bei der Erstellung eines Substrukturverzeichnisses geht es darum, einen Kompromiss zwischen *Kompaktheit* des Pathletverzeichnisses und *Effizienz* des Rekonstruktionsprozesses zu finden. Dieser Zielkonflikt kann über den Parameter λ gesteuert werden. Der Effekt von λ kann in Abbildung 3.4 gesehen werden, wofür der Pathlet-Extraktionsalgorithmus auf verschiedene Datensätze angewendet wurde, die entsprechend des Simulationsmodells aus Abschnitt 3.3 generiert wurden.

Man kann sehen, dass unabhängig von der Größe des Trainingsdatensatzes ($\#\text{Trainingstrajektorien} = \{50, 100, 200, 300, 400, 500\}$), die Anzahl der Pathlets im Verzeichnis mit steigendem λ ebenfalls steigt (siehe Abbildung 3.4a), ohingegen die über 1000 Testtrajektorien gemittelte durchschnittliche Anzahl an Pathlets, die zur Rekonstruktion einer Trajektorie benötigt wird, mit steigendem λ sinkt (siehe Abbildung 3.4b). Dieser Effekt steht im Einklang mit Gleichung 3.1, da mit größerem λ das zweite Optimierungskriterium dominant wird und so die Anzahl an Pathlets zur Rekonstruktion einer Trajektorie optimiert wird, das heißt sinkt.

Weiterhin ist interessant, dass für $\lambda \geq 1$ ($= 10^0$) die Anzahl an gelernten Pathlets gleich der Anzahl an Trajektorien im Trainingsdatensatz ist. Das passiert, wenn die Trajektorien nicht zerlegt werden, sondern in voller Gänze im Pathletverzeichnis abgelegt werden. Ganz allgemein führen größere λ -Werte zu größeren Kantengewichten der Teiltrajektorien. Daher ist es bei der Suche nach dem kürzesten Pfad durch den Grafen günstiger weniger, aber dafür längere Kanten, oder, wie in dem gerade erwähnten Extremfall, nur die Kante vom ersten zum letzten Knoten zu nutzen. Entsprechend sinkt die durchschnittli-



(a) Anzahl an gelernten Pathlets in Abhängigkeit von λ . (b) Durchschnittliche Anzahl an Pathlets, die zur Rekonstruktion der Testdaten benötigt werden.

Abbildung 3.4: Evaluation von λ .

che Anzahl von Pathlets, die zur Rekonstruktion der Testtrajektorien benötigt werden, auf nahezu 0 für $\lambda \geq 1$ ($= 10^0$), da das Pathletverzeichnis nur noch aus den Trajektorien des Trainingsdatensatzes und damit keinen brauchbaren Pathlets besteht.

Neben der *Kompaktheit* des Pathletverzeichnisses und der *Effizienz* der Rekonstruktion von Trajektorien ist die *Universalität* der Pathlets ein weiterer wichtiger Aspekt, der bei der Nutzung von Pathlets zur Anomalieerkennung berücksichtigt werden muss. Die *Universalität* stellt sicher, dass auch Trajektorien, die nicht im Trainingsdatensatz enthalten sind, aber durch denselben Prozess erzeugt wurden, rekonstruiert werden können.

In Abbildung 3.5 ist die *durchschnittliche Überdeckungsrate* für verschiedene Parametereinstellungen ($\#$ Trainingstrajektorien = $\{50, 100, 200, 300, 400, 500\}$ mit jeweils $\lambda = \{0, 10^i\}$ ($i = \{-3, -2, -1, 0, 1, 2\}$)) dargestellt, die anzeigt, inwieweit Trajektorien eines Testdatensatzes mit dem gegebenen Pathletverzeichnis rekonstruiert werden können.

Zusätzlich kann die Größe des Pathletverzeichnisses variiert werden. Dafür werden, wie von Chen et al. vorgeschlagen, die Pathlets des Verzeichnisses entsprechend der Anzahl an Trajektorien des Trainingsdatensatzes, in denen sie mindestens einmal enthalten sind, geordnet. Nach diesem Bewertungsschema kann die Größe des Pathletverzeichnisses variiert werden, indem eine bestimmte Menge der am besten eingestuftten Pathlets, das heißt, die mit der größten Vorkommensrate im Trainingsdatensatz, genutzt wird.

Wie man in Abbildung 3.5 sehen kann, steigt in allen Fällen die durchschnittliche Überdeckungsrate superlinear mit der Anzahl der verwendeten Pathlets. Allerdings ist auch interessant zu sehen, dass es keine Parametereinstellung für $\#$ Trainingstrajektorien = 50 gibt, sodass eine durchschnittliche Überdeckungsrate von mehr als 90% erreicht werden kann. Das zeigt, in diesem Fall

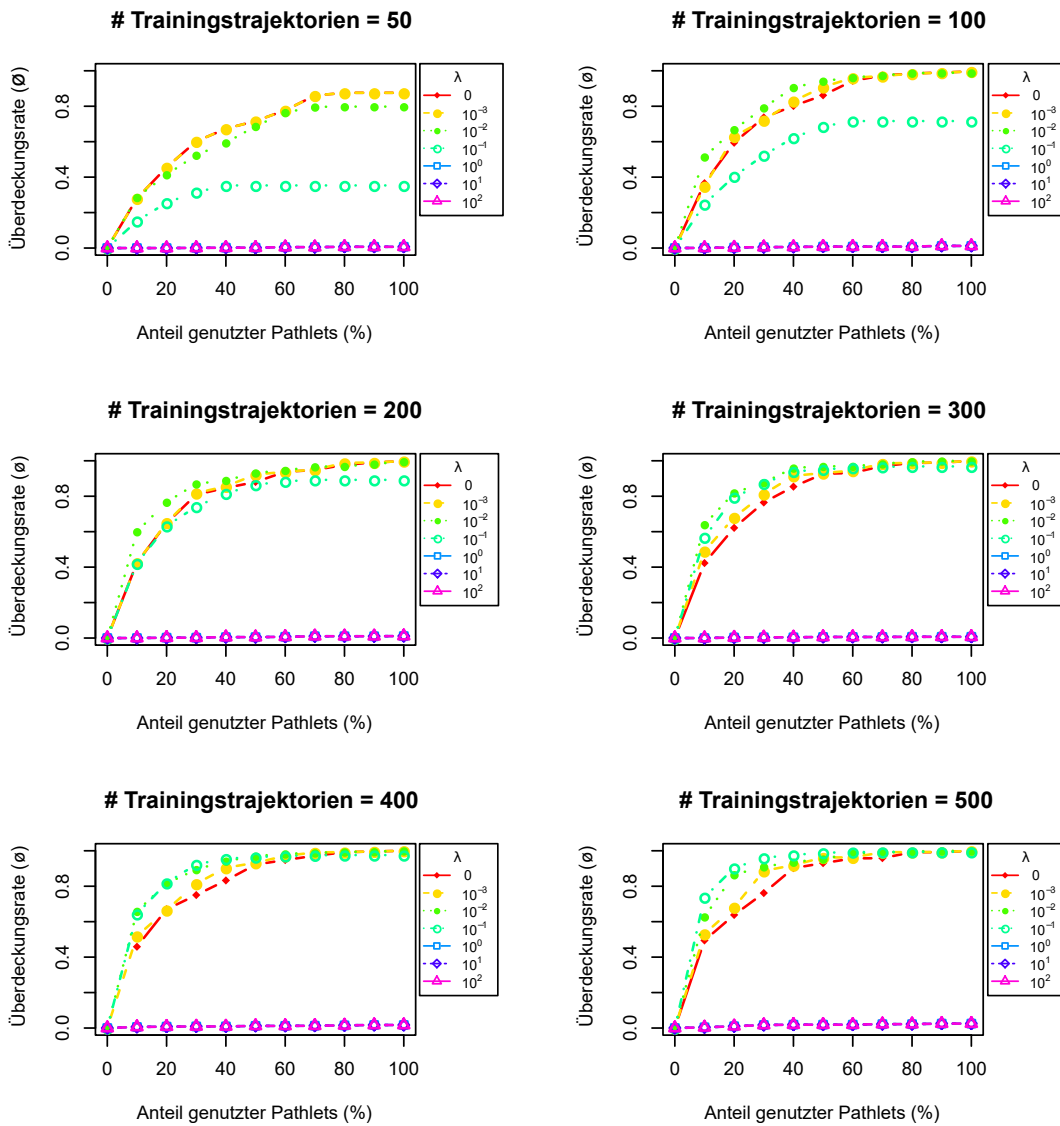


Abbildung 3.5: Durchschnittliche Überdeckungsraten für unterschiedliche Parametereinstellungen.

wird ein Trainingsdatensatz von mindestens 100 Trajektorien benötigt, um ein ausreichend *universelles* Verzeichnis (Überdeckungsrate $\geq 90\%$) zu erhalten. Zusätzlich zeigen diese Ergebnisse, dass die Wahl von λ die *Universalität* der gelernten Pathlets nicht wirklich beeinflusst. Die Kurven für $\lambda = \{0, 10^{-2}, 10^{-1}\}$ sind alle ziemlich ähnlich, wohingegen für $\lambda = \{10^0, 10^1, 10^2, 10^3\}$ die Überdeckungsraten nahezu 0 sind, was nicht verwunderlich ist, da die Verzeichnisse, wie zuvor schon erläutert, in diesen Fällen aus den Trainingstrajektorien selbst bestehen.

Daher kann λ entsprechend der Ergebnisse aus Abbildung 3.4 gewählt werden, wobei ein Kompromiss zwischen der Größe des Pathletverzeichnisses und der durchschnittlichen Anzahl an Pathlets, die pro Trajektorie benötigt werden, gefunden werden muss.

Zusammenfassend lässt sich festhalten, dass mit 100 Trainingstrajektorien und $\lambda = 10^{-2}$ ein geeignetes Pathletverzeichnis für die vorliegende Beispielfabrik erzeugt werden kann. Zusätzlich kann man aus Effizienzgründen die Größe des Pathletverzeichnisses noch weiter reduzieren, indem nur die 80% meist genutzten Pathlets verwendet werden, da der Unterschied bezüglich der durchschnittlichen Überdeckungsraten vernachlässigbar ist. Entsprechend werden alle in diesem Kapitel noch folgenden Experimente mit einem Pathletverzeichnis durchgeführt, das mit dieser Parametereinstellungen erzeugt wurde.

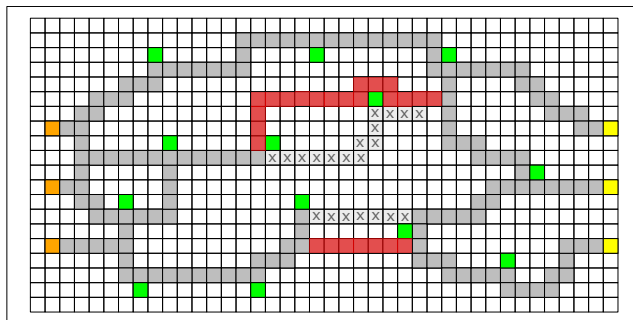
3.6 Erkennung und Analyse von Anomalien

Wie bereits erwähnt, werden zur Erkennung und Analyse von Anomalien im Rahmen der Laufzeitüberwachung Informationen verwendet, die bei der Rekonstruktion der Testdaten aus den Einträgen des Substrukturverzeichnisses anfallen. Im Folgenden werden insgesamt fünf verschiedene Arten von Anomalien vorgestellt, die entsprechend dieses Ansatzes in einem SOIS erkannt und analysiert werden können. Dabei handelt es sich zum einen um Anomalien bezüglich Einzeltrajektorien und globaler anormaler Ereignisse (siehe Abschnitt 3.2). Zusätzlich werden noch Schleifen, Lastenverteilung und Neuartigkeit als SOIS-spezifische Anomalien eingeführt.

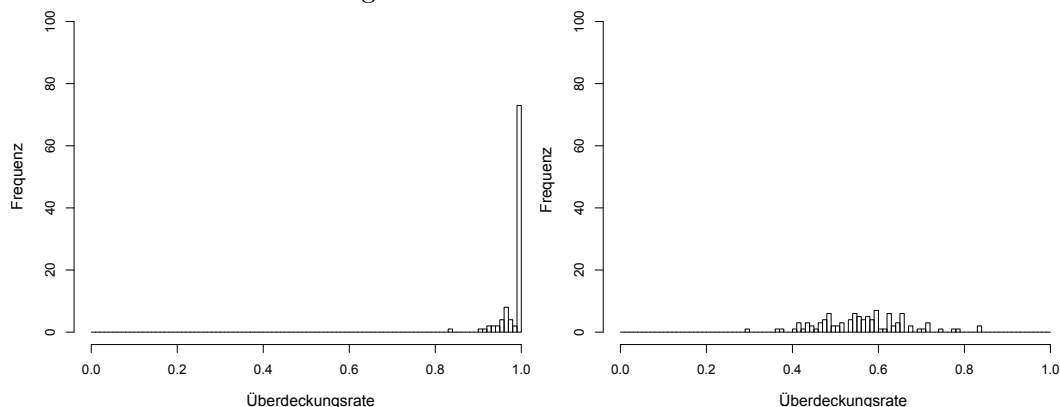
3.6.1 Einzeltrajektorien

Sowohl aus Gründen der Qualitätssicherung, als auch aus Sicherheitsgründen ist es in SOIS wichtig, zu erkennen, ob sich ein Werkteil anormal, das heißt zum Beispiel außerhalb der bisher bekannten Pfade oder in falsche Richtung bewegt. Um Beispiele solcher anormaler Einzeltrajektorien in einem SOIS zu erzeugen, wurde eine Umleitung der Werkstücke auf andere, noch unbekannte Fließbandlinien simuliert. Das entsprechende Simulationsmodell ist in Abbildung 3.6a zu sehen, wobei die neuen Fließbandabschnitte rot markiert und die ursprünglichen ausgeixt sind.

Nachdem es für die neuen Fließbandabschnitte im Pathletverzeichnis keine ent-



(a) Simulationsmodell zur Erzeugung anomaler Einzeltrajektorien, wobei sich der Verlauf der Fließbänder geändert hat.



(b) Überdeckungsrate bei der Rekonstruktion 100 normaler Testtrajektorien. (c) Überdeckungsrate bei der Rekonstruktion 100 anomaler Testtrajektorien.

Abbildung 3.6: Anormale Einzeltrajektorien.

sprechenden Pathlets gibt, lassen sich Trajektorien, die diese neuen Wegstücke beinhalten, nicht mehr vollständig rekonstruieren. Entsprechend kann hier die Überdeckungsrate c als Maß für die (Ab-)Normalität von Einzeltrajektorien verwendet werden. Diese gibt das Verhältnis von der Anzahl an Samplepunkten s_i , die nach dem Rekonstruktionsprozess durch eine Substruktur überdeckt werden, zur Gesamtzahl an Samplepunkten der Trajektorie $t = s_1, \dots, s_n$ an:

$$c(t) = \frac{|\{s_i | s_i \in p_j, \forall p_j \in P(t)\}|}{|s_i \in t|} \tag{3.3}$$

Das Histogramm in Abbildung 3.6c zeigt die Überdeckungsrate von 100 solcher anomalen Trajektorien, die gemäß des neuen Simulationsmodells generiert wurden. Die Überdeckungsrate nehmen Werte zwischen 30% und 85% an, wobei sich ein Cluster um 60% bildet.

Zum Vergleich ist in Abbildung 3.6b das entsprechende Histogramm für 100 Trajektorien zu sehen, die gemäß des ursprünglichen Simulationsmodells (siehe Abbildung 3.2) generiert wurden. Nachdem dieses Modell auch zur Generierung

der Trainingsdaten verwendet wurde, können 73% der Trajektorien vollständig rekonstruiert werden, das heißt, haben eine Überdeckungsrate von 1. Die wenigen, die nicht vollständig rekonstruiert werden können, haben nichtsdestotrotz eine immer noch ziemlich hohe Überdeckungsrate ($\geq 80\%$). Diese falsch-negativ Entscheidungen sind Folge des zusätzlichen Komprimierungsschrittes des Verzeichnisses, bei dem Pathlets mit nur einem geringen Vorkommen in den Trainingsdaten aussortiert werden (siehe Abschnitt 3.5).

Zur Lösung dieses Problems kann ein Schwellwert definiert werden, der die Trajektorien, basierend auf der Überdeckungsrate, in normal und abnormal klassifiziert. Der Einfluss eines solchen Schwellwertes auf die Richtig-positiv-Raten, engl. „True Positives Rates“ (TPR), und Falsch-positiv-Raten, engl. „False Positive Rates“ (FPR), ist in Tabelle 3.1 zu sehen, wobei gilt:

$$\text{TPR} = \frac{\#\text{normale Trajektorien, die als normal klassifiziert wurden}}{\#\text{normale Trajektorien}} \quad (3.4)$$

und

$$\text{FPR} = \frac{\#\text{anormale Trajektorien, die als normal klassifiziert wurden}}{\#\text{anormale Trajektorien}} \quad (3.5)$$

Im Falle eines strikten Schwellwertes von 1 ergibt sich eine FPR von 0 und eine TPR von 0.74. Das heißt, alle anormalen Trajektorien werden korrekt als solche identifiziert, aber 26% der normalen Trajektorien werden fälschlicherweise als abnormal klassifiziert. Durch Abschwächen des Schwellwertes steigt zwar die TPR auf 1, allerdings steigt gleichzeitig die FPR an, was bedeutet, dass immer mehr anormale Trajektorien nicht korrekt klassifiziert werden.

Dementsprechend muss abhängig von den szenariospezifischen Anforderungen der Überdeckungsschwellwert gesetzt werden und damit ein Kompromiss zwischen hoher TRP und kleiner FPR gefunden werden. Dafür kann eine Grenzwertoptimierungskurve, engl. „Receiver-Operating-Characteristic“ (ROC) curve, genutzt werden, die das Verhältnis von TPR und FPR visualisiert.

In Abbildung 3.7 ist die ROC-Kurve für das aktuelle Beispiel zu sehen. Ausgehend von diesem Graf ergibt sich, dass für den vorliegenden Fall die besten Ergebnisse mit einem Überdeckungsschwellwert von 0.9 und 0.8 erreicht werden können, da die entsprechenden ROC-Werte in der linken oberen Ecke im Grafen liegen, was für eine hohe TPR und geringe FPR steht.

Insgesamt konnte in diesem Abschnitt gezeigt werden, dass es möglich ist, anormale Einzeltrajektorien über die sich bei der Rekonstruktion ergebenden Überdeckungsrate zu detektieren. Ganz allgemein wurde damit das Problem der Anomalieerkennung für Trajektorien auf das Problem von Anomalieerkennung in reellen Zahlen, nämlich der Überdeckungsrate, transformiert. Dementsprechend kann bei Bedarf die einfache Schwellwertbestimmung, wie sie hier vorgestellt wurde, auch durch andere Methoden zur Anomalieerkennung in reellen Zahlen ersetzt werden.

Schwellwert	TPR	FPR
1	0.74	0
0.9	0.99	0
0.8	1	0.02
0.7	1	0.09
0.6	1	0.33
0.5	1	0.72
0.4	1	0.97
0.3	1	0.99
0.2	1	1
0.1	1	1
0	1	1

Tabelle 3.1: TPR und FPR bei der Unterscheidung von normalen und anormalen Einzeltrajektorien.

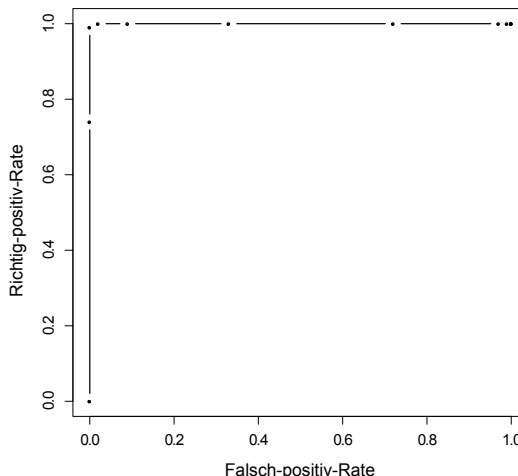


Abbildung 3.7: ROC-Kurve für anormale Einzeltrajektorien.

3.6.2 Globale Ereignisse

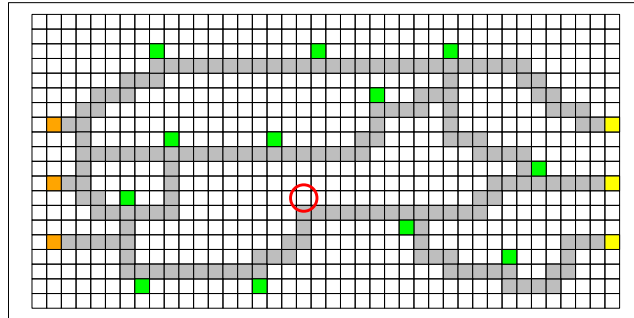
Für die Qualitätssicherung in SOIS ist es wichtig, Produktionsstationen zu identifizieren, die nicht mehr genutzt werden. Das passiert entweder, weil diese defekt sind, oder aber, weil diese aus anderen Gründen beim Routing der Werkteile nicht mehr berücksichtigt werden. Eine solche Nichtnutzung ist ein Beispiel für ein globales anormales Ereignis in einem SOIS.

In Abbildung 3.8a ist ein entsprechende Modell zu sehen, das zur Simulation eines derartigen globalen anormalen Ereignisses verwendet werden kann. Dadurch, dass die Produktionsstätte aus dem Modell entfernt wurde, enthalten die erzeugten Trajektorien den Standort dieser Produktionsstätte nicht mehr. Somit kommen auch Pathlets, die den Standort dieser Produktionsstätte enthalten, nicht mehr bei der Rekonstruktion der Testtrajektorien zum Einsatz. Entsprechend kann die Nutzungshäufigkeit von Pathlets zur Erkennung von globalen anormalen Ereignissen verwendet werden.

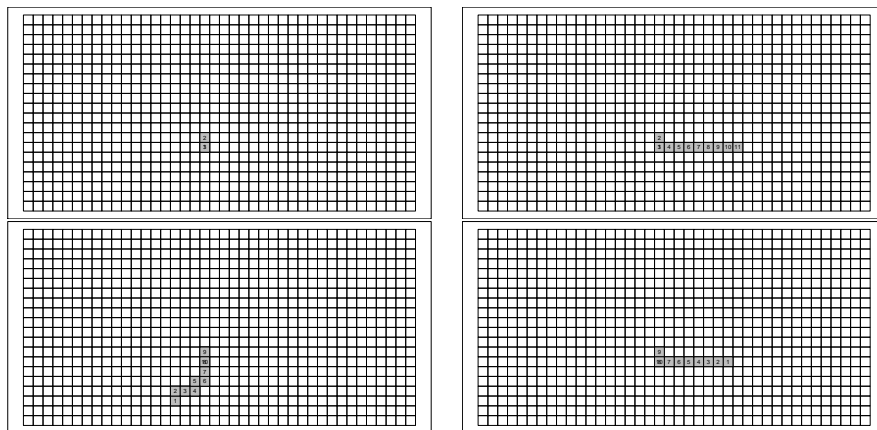
Im Folgenden wird dazu das Maß der *relativen Nutzerhäufigkeit*, engl. „relative user frequency“, eingeführt, das den Anteil an Trajektorien angibt, für die ein Pathlet zur Rekonstruktion benötigt wird. Damit erhält man eine Kennzahl, mit Hilfe derer man die Nutzungshäufigkeit von Pathlets quantifizieren kann. Sei T nun die Menge der Testtrajektorien und $T(p_i) \subseteq T$ die Menge an Trajektorien, die ein Pathlet $p_i \in P$ beinhalten. Dann ist die relative Nutzerhäufigkeit f_r eines Pathlets p_i definiert als:

$$f_r(p_i) = \frac{|T(p_i)|}{|T|} \tag{3.6}$$

In Abbildung 3.9 sind nun die relativen Nutzerhäufigkeiten von Pathlets aus dem aktuellen Beispielszenario in Form von Histogrammen zu sehen. Die blau-

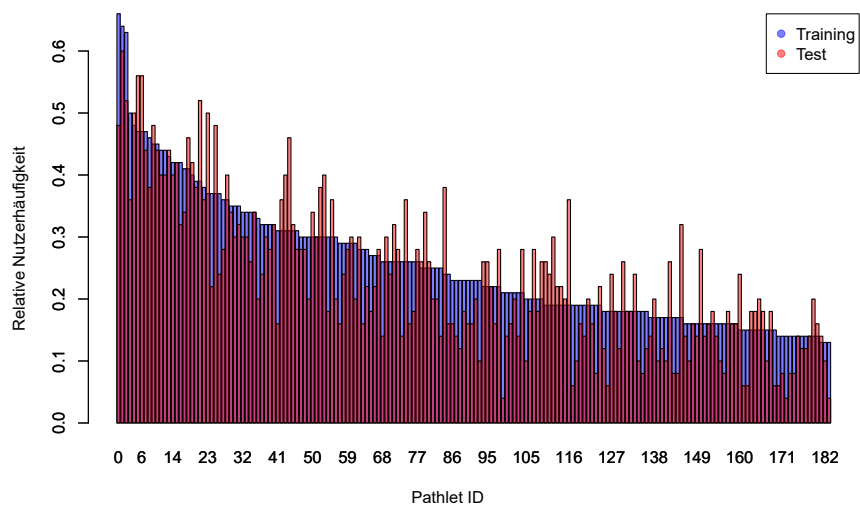


(a) Modell zur Simulation eines globalen anomalen Ereignisses, wobei der rote Kreis eine entfernte Maschine markiert.

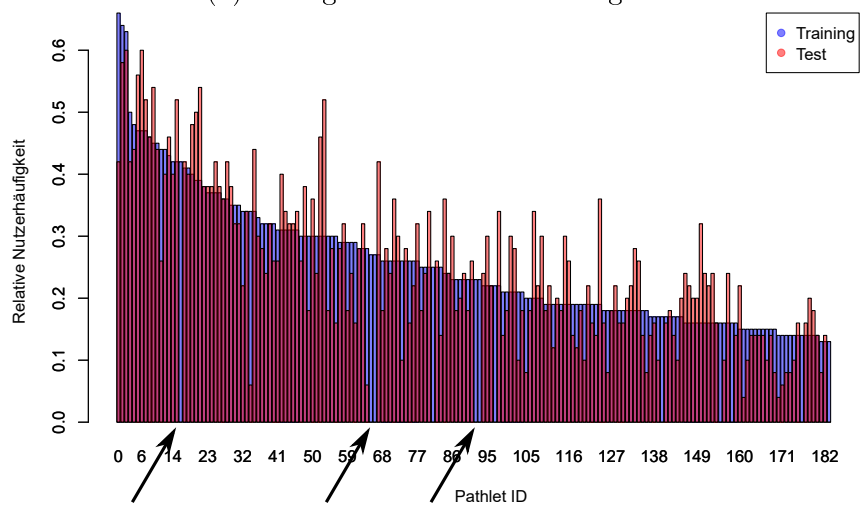


(b) Beispiele von Pathlets, die bei der Rekonstruktion der Testdaten nicht mehr verwendet werden.

Abbildung 3.8: Globales anomales Ereignis.



(a) Kein globales anomales Ereignis.



(b) Globales anomales Ereignis.

Abbildung 3.9: Histogramm der relativen Nutzungshäufigkeit innerhalb eines Zeitintervalls.

en Balken markieren dabei die Nutzerhäufigkeiten über alle Pathlets im Bezug auf ihr Vorkommen im Trainingsdatensatz. Diese dienen als Grundwahrheit, engl. „ground truth“, wohingegen die roten Balken die Nutzerhäufigkeiten der Pathlets bezüglich des Trainingsdatensatzes (50 Trajektorien) angeben.

Im Speziellen zeigt Abbildung 3.9a das Ergebnis für den Fall, dass der Testdatensatz ausschließlich aus normalen Trajektorien besteht, wohingegen Abbildung 3.9b das Histogramm zeigt, das sich bei Testtrajektorien gemäß des modifizierten Simulationsmodells ergibt. Man kann erkennen, dass der Ausfall der Produktionsstätte bei manchen Pathlets einen Rückgang ihrer Nutzerhäufigkeit auf 0 verursacht (siehe schwarze Pfeile).

Neben der grundsätzlichen Erkennung einer solchen Anomalie, können noch – entsprechend des Analyseschritts bei der Laufzeitüberwachung (siehe Abschnitt 2.3) – zusätzliche Informationen über das globale anormale Ereignis gewonnen werden. Dazu werden die Pathlets, die nicht mehr zur Rekonstruktion der Testdaten genutzt werden, genauer analysiert. In Abbildung 3.8b sind beispielhaft vier dieser nicht mehr genutzten Pathlets visualisiert. Offensichtlich enthalten alle den Standort der aus dem Simulationsmodell entfernten Produktionsstätte. Das heißt, eine Analyse der räumlichen Lage der anormal genutzten Pathlets ermöglicht die Lokalisierung des globalen anomalen Ereignisses.

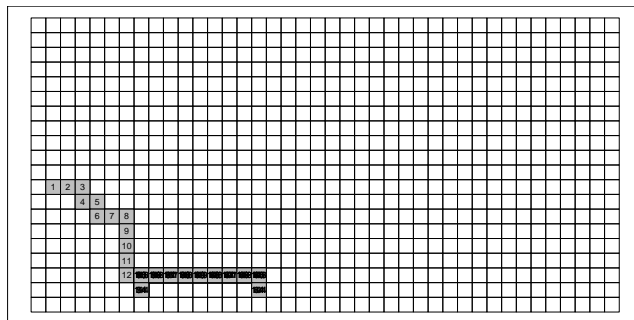
Für die praktische Umsetzung dieses Ansatzes in einem SOIS, bietet sich ein *Sliding Window*-Ansatz, an. Damit kann die relative Nutzerhäufigkeit, die sich aus der Rekonstruktion der Trajektorien innerhalb des aktuellen Sliding Windows ergibt, mit der Grundwahrheit oder Nutzerhäufigkeiten aus vorhergehenden Schiebefenstern verglichen werden, um Änderungen in der Pathletnutzung zu erkennen.

Insgesamt konnte also in diesem Abschnitt gezeigt werden, dass Anomalien wie der Ausfall einer Produktionsstätte über die relative Nutzerhäufigkeit der Pathlets erkennbar sind. Darüber hinaus ist es möglich, globale anormale Ereignisse zu lokalisieren, indem räumliche Überschneidungen bei den nicht mehr genutzten Pathlets gesucht werden.

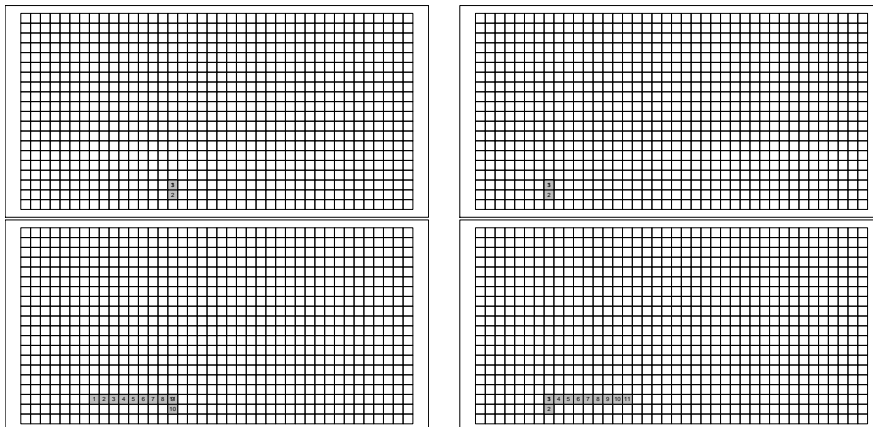
3.6.3 Schleifen

Bei der Laufzeitüberwachung von SOIS ist es zudem wichtig, möglichst schnell zu erkennen, wenn ein Werkteil keinen normalen Prozessfortschritt mehr zeigt, sondern zum Beispiel in eine Schleife gerät und immer dieselben Schritte wiederholt. Zum einen wird dadurch das geforderte Produkt nicht fertiggestellt; zum anderen werden möglicherweise wichtige Ressourcen blockiert, die an anderer Stelle fehlen.

Um derartige Schleifen zu detektieren, wird das Maß des *Pathletvorkommens pro Nutzer*, engl. „pathlet occurrence per user“, eingeführt. Es gibt für jedes Pathlet $p_i \in P(t_j)$ an, wie oft es in der Trajektorie $t_j \in T$ vorkommt, wobei $P(t_j)$ die Menge aller Pathlets ist, die mindestens einmal in t_j vorkommen.



(a) Anormale Trajektorie, die in einer Schleife hängen bleibt.



(b) Beispiele von Pathlets, die auf Grund des Pathletvorkommens bei der Rekonstruktion der Trajektorie als anormal klassifiziert werden.

Abbildung 3.10: Anormale Trajektorie und anormal genutzte Pathlets.

Damit lässt sich wie zuvor bei der Erkennung von globalen anomalen Ereignissen das Problem der Schleifenerkennung in Trajektorien auf das allgemeine Problem der Anomalieerkennung in reellen Zahlen, nämlich das Pathletvorkommen pro Nutzer, transformieren.

Für die Laufzeitüberwachung können daher die Pathletvorkommen aus dem Trainingsdatensatz zum Beispiel in einem Box-Plot dargestellt werden. Damit hat man Referenzwerte, gegen die die eingehenden Trajektorien getestet werden können. Der große Vorteil von Box-Plots ist, dass sich der Grad der tolerierten Abweichung relativ einfach über die Spannweite der Antennen, engl. „whiskers“, kontrollieren lässt. Gleichzeitig hat man eine Visualisierung, die eine schnelle und intuitive Analyse der Daten ermöglicht.

Zur Veranschaulichung wurde eine anormale Trajektorie erzeugt (siehe Abbildung 3.10a). Offensichtlich ist das entsprechende Werkteil in einer Schleife geraten und bewegt sich nur noch zwischen den beiden Produktionsstationen unten links hin und her.

Der Box-Plot, der sich aus den Werten des Pathletvorkommens des Trainingsdatensatzes ergibt, ist in Abbildung 3.11 zu sehen. Zusätzlich ist eingetragen, wenn ein Pathletvorkommenwert einer Testtrajektorie über die Antennen hin-

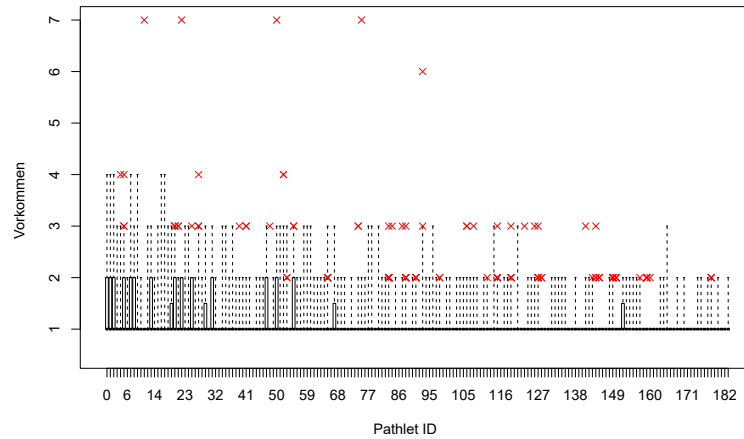


Abbildung 3.11: Boxplot zur Repräsentation normalen Pathletvorkommens, sowie darauf basierend detektierter Ausreißer (rote Kreuze).

ausgeht (siehe rote Kreuzchen). Das heißt, jedes rote Kreuzchen impliziert, dass zur Rekonstruktion einer Trajektorie ein bestimmtes Pathlet anormal oft benötigt wurde.

Zur weiteren Analyse sind in Abbildung 3.10b die entsprechenden Pathlets für das aktuelle Beispiel visualisiert. Offensichtlich handelt es sich genau um die Pathlets, die zur Rekonstruktion des Schleifenabschnitts der anomalen Trajektorie benötigt werden.

Insgesamt konnte also in diesem Abschnitt gezeigt werden, dass über das Pathletvorkommen pro Nutzer Trajektorien detektiert werden können, deren Prozessfortschritt gestört ist. Eine zusätzliche Analyse der betroffenen Pathlets ermöglicht es darüber hinaus die Problemstelle zu lokalisieren und damit unter Umständen beteiligte Produktionsstätten zu identifizieren.

3.6.4 Lastenverteilung

Auch der Aspekt der Ressourcennutzung spielt bei der Qualitätssicherung in SOIS eine wichtige Rolle. Es ist insbesondere wichtig eine Verschiebung hin zu einer ungleichen Lastenverteilung im System zu erkennen, da diese die Qualität der Produktivität des SOIS sinken lässt.

Um diese Lastenverteilung bezüglich der Pfade und Produktionsstationen eines SOIS zu überwachen, kann wie schon zur Erkennung globaler anomaler Ereignisse die Nutzungshäufigkeit der Verzeichniseinträge genutzt werden. In diesem Fall reicht allerdings die Suche nach reellwertigen Ausreißern nicht aus, da die Lastenverteilung nur aus der Gesamtheit des Systems heraus bewertet werden kann. Das heißt, die Nutzungshäufigkeitswerte müssen im Kontext des Gesamtsystems betrachtet werden.

Dazu können Wärmebilder, engl. „heat maps“, genutzt werden, wie sie in Abbildung 3.12b und 3.12c für die Beispielfabrik zu sehen sind. Der Vorteil dieser Wärmebilder ist, dass die Nutzungshäufigkeitswerte einzelner Pathlets direkt

räumlich zugeordnet werden. Dadurch können Änderungen bei einzelnen Pathlets unmittelbar im Zusammenhang mit möglichen globalen Verschiebungen betrachtet werden.

Sei $P(l_i)$ die Menge an Pathlet, die eine bestimmte Rasterzelle $l_i \in L$ enthalten und L die Menge aller Rasterzellen im Simulationsmodell. Der Wert $c(l_i)$ zur Einfärbung des Wärmebildes ist dann gegeben durch:

$$c(l_i) = \max_{p_j \in P(l_i)} f_r(p_j), \quad (3.7)$$

wobei $f(p_j)$ die relative Nutzerhäufigkeit des Pathlets p_j bezeichnet (siehe Gleichung 3.6).

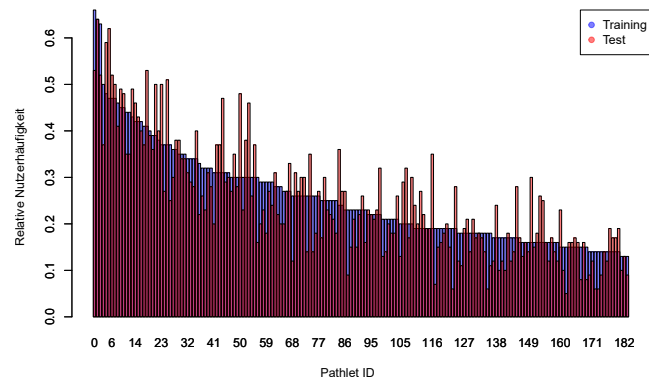
Damit kann nun statt die Nutzungshäufigkeit einer jeden Rasterzelle separat zu bestimmen, die Färbung des Wärmebildes aus den Nutzungshäufigkeitswerten der Pathlets abgeleitet werden, die sich ohnehin bei der Rekonstruktion ergeben. Im vorliegenden Beispiel wurde sich für eine 10-teilige Farbabstufung von dunkel grün für eine sehr geringe Nutzungshäufigkeit ($0 \leq f_r \leq 0.1$) bis dunkel rot für eine sehr hohe Nutzungshäufigkeit ($0.9 < f_r \leq 1$) entschieden, wobei jedes Farbintervall einer Änderung der relativen Nutzerhäufigkeit von 0.1 entspricht.

In Abbildung 3.12a, 3.12b und 3.12c sind das entsprechende Histogramm bzw. die Wärmebilder zu sehen, die sich bei der Rekonstruktion von 100 Trainings- und Testtrajektorien ergeben. Man sieht, dass das Histogramm, das lediglich einen Balken-weisen Vergleich der relativen Nutzerhäufigkeitswerte erlaubt, selbst bei den normalen Testdaten durchaus erhebliche Unterschiede bei der gelernten (blaue Balken) und getesteten (rote Balken) Nutzerhäufigkeiten zeigt. Die Färbung der beiden Wärmebilder dagegen ist sehr ähnlich, was vielmehr dem entspricht, dass es aus globaler Sicht keine Verschiebung der Lastenverteilung gegeben hat.

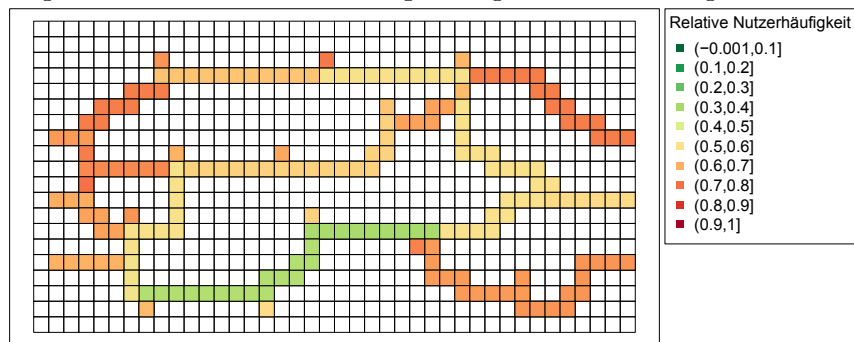
Zur Simulation von Szenarien, in denen sich die Lastenverteilung der Pfade ändert, wurde die oberste Fließbandlinie zusammen mit den angehängten Produktionsstationen aus dem ursprüngliche Simulationsmodell entfernt (siehe Abbildung 3.13). Da Trajektorien, die gemäß dieses modifizierten Modells generiert werden, diese obere Fließbandroute nicht mehr nutzen können, wird stattdessen die gesamte Last auf die anderen Montagebänder und Produktionsstätten verteilt.

Insgesamt wurden sechs Szenarien definiert, die verschiedene Abstufungen der Verschiebung der Lastenverteilung zeigen. Die entsprechenden Datensätze sind in Tabelle 3.2 aufgeschlüsselt, wobei für jeden Datensatz angegeben ist, wie viele Trajektorien die aus dem Simulationsmodell entfernten Fließbandabschnitte enthalten (*standard*) und wie viele nicht (*modifiziert*).

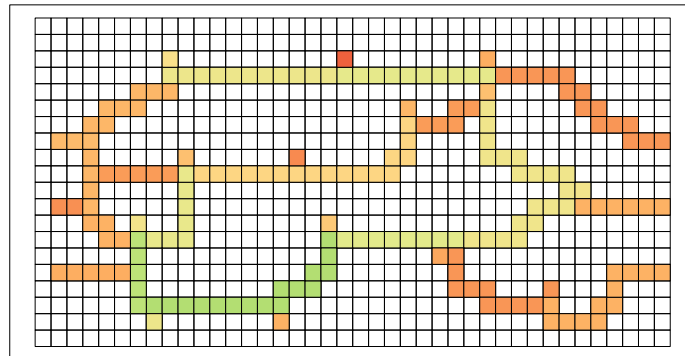
In Datensätzen, die mit dem ursprünglichen Simulationsmodell generiert wurden, das heißt, im Trainingsdatensatz, sowie allen „normalen“ Testdatensätzen, wird die obere Fließbandlinie ungefähr von der Hälfte der Werkteile genutzt. Für die „verlagerten“ Datensätze wurde das Verhältnis von Trajektorien, die



(a) Histogramm der relativen Nutzungshäufigkeit der Trainings- und Testdaten.



(b) Wärmebild der relativen Nutzungshäufigkeit der Trainingsdaten.



(c) Wärmebild der relativen Nutzungshäufigkeit der Testdaten.

Abbildung 3.12: Vergleich Histogramm- und Wärmebilddarstellung der relativen Nutzungshäufigkeit von Pathlets.

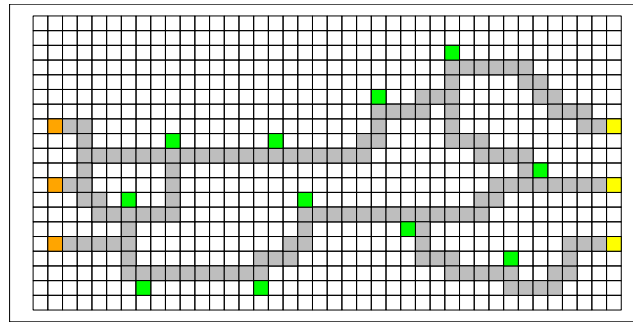


Abbildung 3.13: Modell zur Simulation der Lastenverteilung.

Tabelle 3.2: Evaluationsszenarien für Lastenverschiebung.

	# Trajektorien <i>standard</i>	# Trajektorien <i>modifiziert</i>
Trainingsdaten	45	55
normal 1	52	48
normal 2	97	103
normal 3	168	132
verlagert 1	0	100
verlagert 2	10	90
verlagert 3	30	70
verlagert 4	70	30
verlagert 5	90	10
verlagert 6	100	0

das obere Montageband nutzen und solchen, die es nicht nutzen, variiert. Dabei sind *verlagert 1* und *verlagert 6* die beiden Extrema, sodass die oberen Fließbandabschnitte entweder von allen Werkteilen genutzt werden, oder von keinem.

In Abbildung 3.14a, 3.14b und 3.14c sind die Wärmebilder für die „normalen“ Testdatensätze zu sehen. Es ist gut zu sehen, dass die Einfärbungen ähnlich zu denen aus Abbildung 3.12b sind, die sich aus dem Trainingsdatensatz ergeben haben, sodass damit korrekterweise keine Verschiebung impliziert wird.

In Abbildung 3.15a, 3.15b und 3.15c sind die Wärmebilder für die Szenarien mit steigender Last auf der oberen Fließbandlinie zu sehen. Man sieht, dass je mehr das obere Montageband belastet ist, desto mehr verschiebt sich die Farbe der entsprechenden Rasterzellen im Wärmebild hin zu rot. Hier zeigt sich in aller Deutlichkeit der Vorteil der Wärmebilder gegenüber der einzelwertbasierten Analyse in Form von Histogrammen. Dadurch, dass die Nutzungshäufigkeiten aller Pathlets nicht unabhängig von einander analysiert werden, sondern die Werte stattdessen mit ihrer räumlichen Lage verknüpft werden, können globale Trends relativ einfach und intuitiv detektiert und mögliche Anomalien analysiert werden.

Ähnliches gilt für die Erkennung von sinkender Last bei einem Fließbandabschnitt in einem SOIS. In Abbildung 3.16a, 3.16b und 3.16c sind die entsprechenden Wärmebilder für die Szenarien mit abnehmender Last auf dem oberen

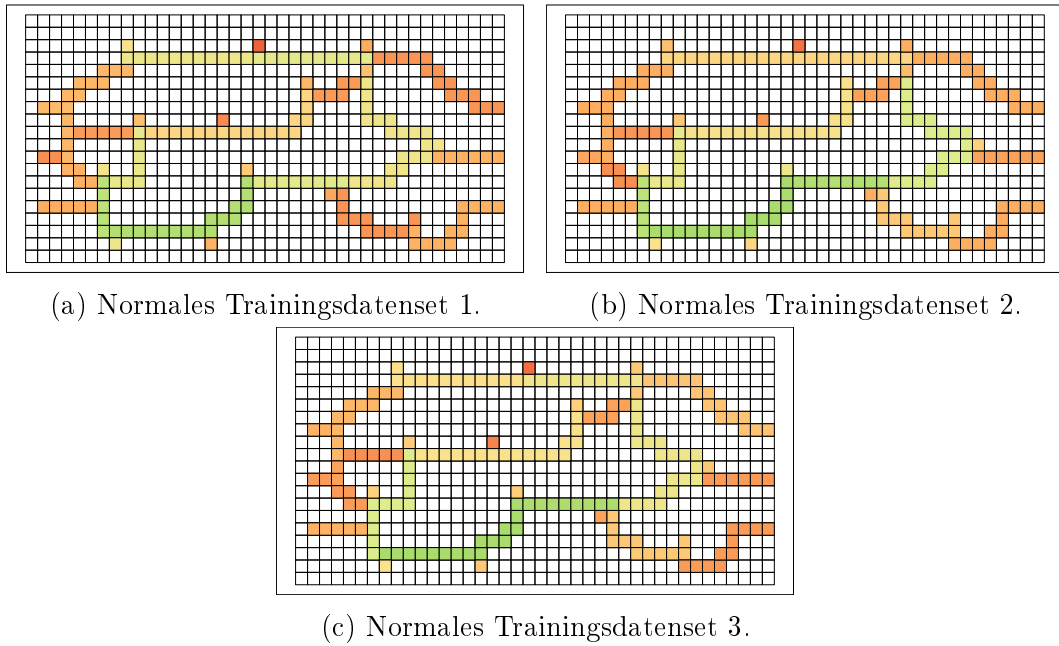


Abbildung 3.14: Wärmebilder der relativen Nutzungshäufigkeit von Pathlets für verschiedene Evaluationsszenarien.

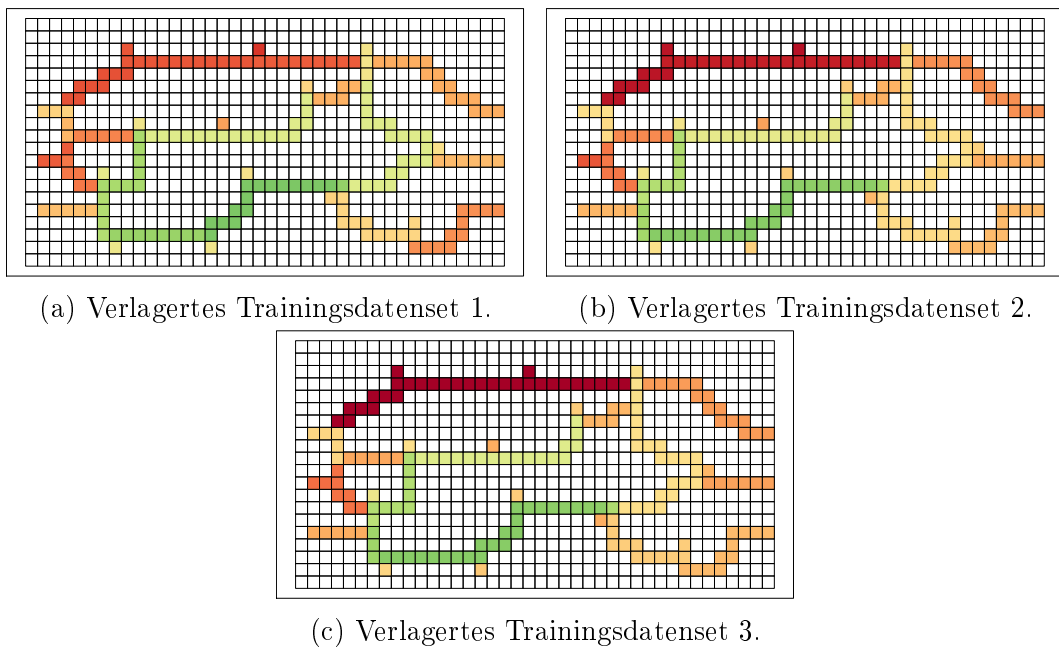


Abbildung 3.15: Wärmebilder der relativen Nutzungshäufigkeit von Pathlets für verschiedene Evaluationsszenarien.

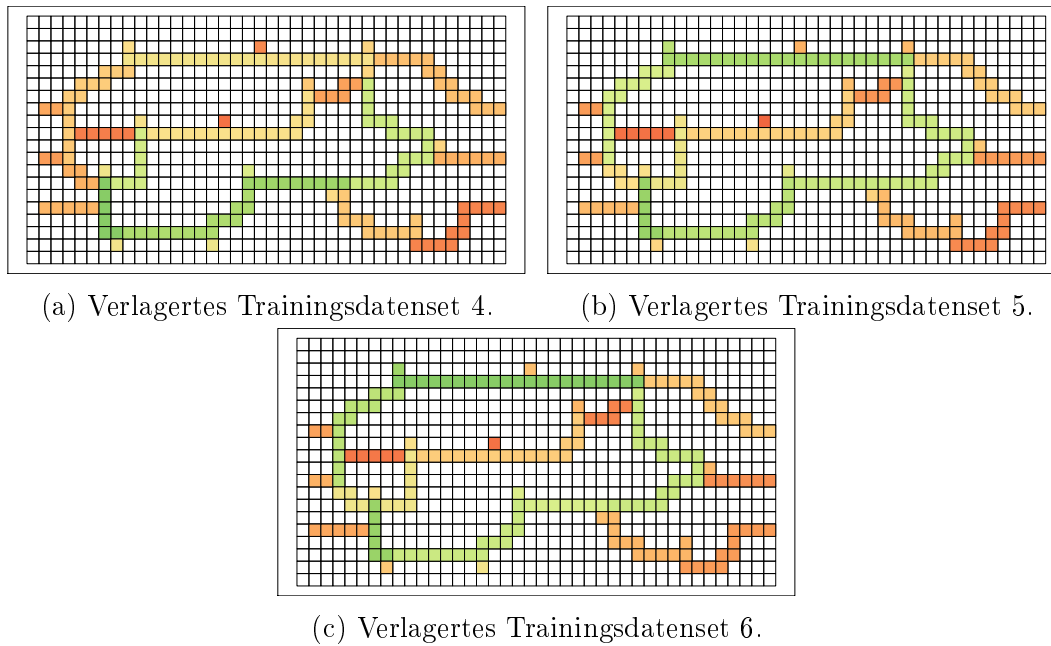


Abbildung 3.16: Wärmebilder der relativen Nutzungshäufigkeit von Pathlets für verschiedene Evaluationsszenarien.

Montagebandabschnitt zu sehen. Während im Referenzwärmebild aus Abbildung 3.12b die Rasterzellen des oberen Fließbandes mehrheitlich hellrot oder orange sind, sind diese in Abbildung 3.16a orange und gelb und schließlich in Abbildung 3.16b und Abbildung 3.16c grün, was einen deutliche Abnahme der Nutzungshäufigkeit der Pathlets dieses Teilstückes, bedeutet.

Insgesamt konnte also in diesem Abschnitt gezeigt werden, dass durch die Verknüpfung der Nutzungshäufigkeiten der Pathlets mit deren räumlichen Lage über Wärmebilder eine Bewertung der Lastverteilung möglich ist. Dadurch, dass die Nutzungshäufigkeitswerte im Kontext des Gesamtsystems betrachtet werden, können damit Verschiebungen beziehungsweise Anomalien in der Lastverteilung auf Pfade und Produktionsstationen erkannt und in ihrer globalen Auswirkung analysiert werden.

3.6.5 Neuartigkeit

Die Flexibilität eines SOIS, zum Beispiel neue Produktionsstationen während des laufenden Betriebs automatisch in den Produktionsprozess zu integrieren, ist eine enorme Herausforderung bei der Laufzeitüberwachung. Es entstehen beispielsweise neue Pfade beziehungsweise werden Fließbandabschnitte genutzt, wie es so vorher noch nie beobachtet wurde. Um bei derartige Abweichungen zu unterscheiden, ob es sich um eine Anomalie wie in Abschnitt 3.6.1 handelt, oder um eine *Neuartigkeit* im Systemverhalten, welches eine Aktualisierung des bisherigen Normalitätsmodells erfordert, kann auch hier die Nutzungshäufigkeit als Bewertungsmaß genutzt werden.

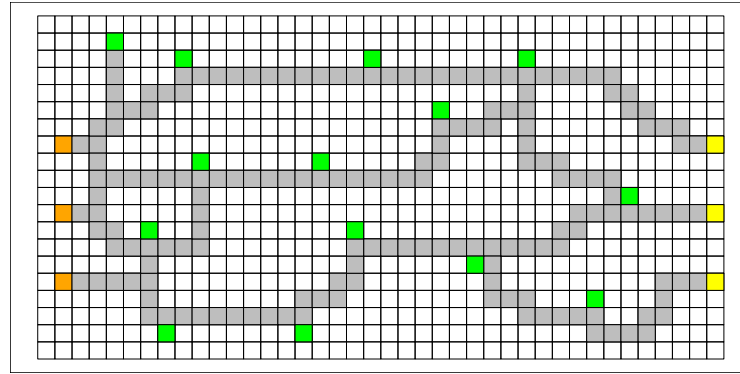


Abbildung 3.17: Simulationsmodell mit zusätzlicher neuer Maschine und Fließbandanbindung.

Allerdings wird in diesem Fall statt der relativen Nutzerhäufigkeit der zur Rekonstruktion verwendeten Pathlets im Falle einer anormalen Einzeltrajektorie für jede Rasterzelle l_i , die nicht rekonstruiert werden kann, die relative Nutzerhäufigkeit $f_r(l_i)$ berechnet:

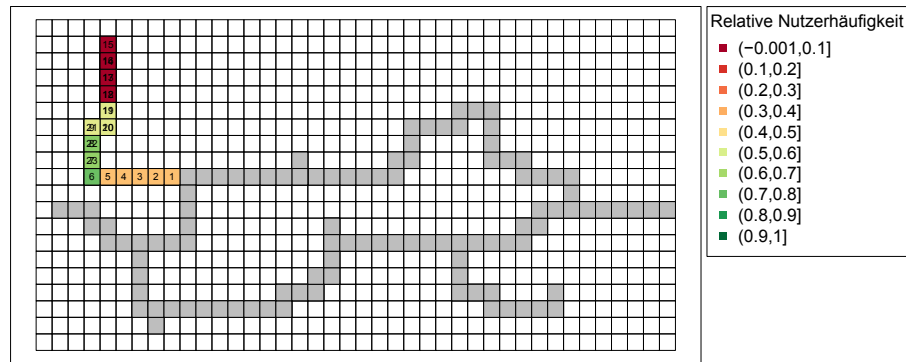
$$f_r(l_i) = \frac{|T(l_i)|}{|T|} \quad (3.8)$$

Das Ergebnis kann dann in einem *partiellen Wärmebild*, engl. „partial heat map“, visualisiert werden, in dem nur die betroffenen Rasterzellen eingefärbt sind.

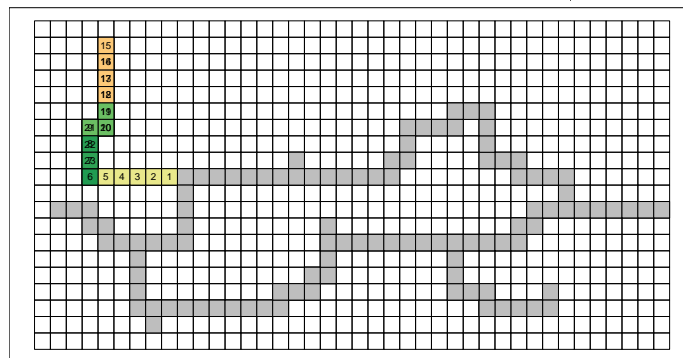
Zur Simulation einer derartigen Situation in einem SOIS wurde das ursprüngliche Simulationsmodell um eine neue Produktionsstation und entsprechende Fließbandmodule ergänzt (siehe Abbildung 3.17). Damit können zum einen anormale Einzeltrajektorien generiert werden, die den neuen Fließbandabschnitt enthalten; zum anderen kann ein Satz an Testtrajektorien erzeugt werden, bei dem die neue Produktionsstation vollständig in den Produktionsprozess integriert ist und damit in derselben Form von Werkteilen genutzt wird, wie auch alle anderen Produktionsstationen.

In Abbildung 3.18a ist ein solches partielles Wärmebild zu sehen, das sich ergibt, wenn nur eins von 100 Werkteilen die neue Produktionsstation und die neuen Fließbandabschnitte nutzt. Auch wenn mehr als die im Simulationsmodell ergänzten Komponenten nicht rekonstruierbar und damit eingefärbt sind, lässt sich über den Farbton eine Bewertung der Abnormalität vornehmen. In diesem Fall steht das Dunkelrot für eine sehr geringe relative Nutzerhäufigkeit ($0 \leq f_r \leq 0.1$) der Rasterzelle, was auf eine tatsächliche Anomalie schließen lässt. Dass die anderen Stücke trotz einer höheren relativen Nutzerhäufigkeit der Rasterzellen nicht rekonstruiert werden können, liegt daran, dass die Rasterzellen in einer Reihenfolge (siehe Zahlen in den Rasterzahlen) in der Trajektorie vorkommen, für das es kein Pathlets im Verzeichnis gibt.

Das heißt, durch die Analyse des Wärmebildes kann nicht nur abgelesen wer-



(a) Nur eine einzige Trajektorie nutzt den neuen Pfad/die neue Maschine.



(b) Normales Trainingsdatenset (40 von 100 Trajektorien nutzen den neuen Pfad/die neue Maschine).

Abbildung 3.18: Partielle Wärmebilder der relativen Nutzungshäufigkeit von Pathlets zur Analyse von nicht rekonstruierbaren Teilstücken von anormalen Trajektorien.

den, welche Teilstücke der Trajektorie nicht rekonstruierbar sind, sondern auch, welcher Grad an Abnormalität vorliegt.

In Abbildung 3.18b ist dagegen das partielle Wärmebild derselben Trajektorie zu sehen, wobei sie diesmal Teil des Testdatensatzes ist, der die neue Produktionsstation bereits vollständig in den Produktionsprozess integriert hat, was heißt, dass die Produktionstation genauso genutzt wird, wie alle anderen auch. Erwartungsgemäß ist die Trajektorie wieder nicht vollständig rekonstruierbar, allerdings hat sich die Färbung im Wärmebild deutlich geändert. Die relative Nutzerhäufigkeit der betroffenen Rasterzellen ist in allen Fällen gestiegen. Im Speziellen ist der neue Fließbandabschnitt mit Produktionsstation nicht mehr dunkelrot, sondern hell-orange ($0.4 < f_r \leq 0.5$) eingefärbt, was auf eine Neuartigkeit im Systemverhaltens, wie eben eine neue Produktionsstation hindeutet. Zusammenfassend konnte also in diesem Abschnitt gezeigt werden, dass durch die Verwendung von partiellen Wärmebildern, die die relativen Nutzerhäufigkeiten von nicht rekonstruierbaren Teilstücken einer Testtrajektorie visualisieren, der Grad der Abnormalität bewertet werden kann. Damit ist es im Falle einer anormalen Einzeltrajektorie möglich, zu unterscheiden, ob es sich um

das Fehlverhalten eines einzelnen Werkteils oder aber um eine Neuartigkeit im gesamten Systemverhalten handelt.

3.7 Zusammenfassung

In diesem Kapitel wurde ein Konzept vorgestellt, das trotz des hohen Freiheitsgrads eines SOIS eine Erkennung und Analyse von Anomalien in darin anfallenden raumbezogenen Daten ermöglicht (siehe Abschnitt 3.1). Dazu wird aus einer Menge von Trainingsdaten ein Modell erzeugt, das das Normalverhalten des Systems repräsentiert. Um den hohen Freiheitsgrad des Systems nachbilden zu können, ist das Modell in diesem Fall ein Verzeichnis an Substrukturen, sodass diese Teilstücke genauso flexibel zusammengesetzt werden können, wie es der Prozessablauf erfordert. Ausgehend von einem derartigen Modell können nun im Rahmen der Laufzeitüberwachung Testdaten rekonstruiert werden, wobei die dabei anfallende Informationen die Erkennung und Analyse von Anomalien ermöglichen.

Dieses Konzept wurde am Beispiel von Bewegungsdaten mit konstanter Geschwindigkeit eingeführt und evaluiert. Entsprechend wurde in Abschnitt 3.3 ein Simulationsmodell in Form einer kleinen Beispielfabrik vorgestellt, das es erlaubt, Trainings- und Testdaten zu generieren, die Routen mobiler Werkteile in einem SOIS entsprechen.

Zur Extraktion geeigneter Substrukturen wurde das Prinzip des Pathlet Learning verwendet, das durch die Formulierung eines bedingten Optimierungsproblems sowohl die Kompaktheit des sich ergebenden Verzeichnisses, als auch die Effizienz des Rekonstruktionprozesses optimiert (siehe Abschnitt 3.4).

Es konnte gezeigt werden, dass mit Hilfe dieses Verfahrens aus den Trainingsdaten des simulierten SOIS geeignete Teilpfade extrahiert werden können, die neben der *Kompaktheit* und *Rekonstruktionseffizienz* eine ausreichende *Universalität* des Verzeichnisses garantieren. Damit kann sichergestellt werden, dass auch Trajektorien, die nicht explizit im Trainingsdatensatz enthalten sind, aber trotzdem mögliche Produktionsabläufe zeigen, rekonstruiert werden können (siehe Abschnitt 3.5).

Die anschließende Erkennung und Analyse von Anomalien im Rahmen der Laufzeitüberwachung wurde in Abschnitt 3.6 anhand von fünf verschiedenen für die Qualitätssicherung relevanten Arten von Anomalien ausgeführt und evaluiert. So können anormale Einzeltrajektorien, die sich ergeben, wenn sich zum Beispiel mobile Werkteile außerhalb der bisher bekannten Pfade oder in falscher Richtung bewegen, über die sich bei der Rekonstruktion ergebenden Überdeckungsrate detektiert werden (siehe Abschnitt 3.6.1).

In gleicher Weise können globale anormale Ereignisse wie zum Beispiel der Ausfall einer Maschine durch das Beobachten der relativen Nutzerhäufigkeiten der einzelnen Verzeichniseinträge erkannt werden. Zusätzlich können derartige Nichtnutzungen einer Maschine durch eine entsprechende Visualisierung der räumlichen Überschneidungen der nicht genutzten Pathlets lokalisiert werden

(siehe Abschnitt 3.6.2).

Eine weitere zentrale Herausforderung in der Qualitätssicherung ist das Erkennen von anormalem Prozessfortschritt, in dem das Werkteil in eine Schleife gerät und immer dieselben Schritte wiederholt. Dies kann, wie in Abschnitt 3.6.3 gezeigt, ebenfalls unter Verwendung des Substrukturverzeichnisses als Normalitätsmodell detektiert werden, indem das Pathletvorkommen pro Nutzer bei der Rekonstruktion mit ausgewertet wird. Eine zusätzliche visuelle Analyse der betroffenen Pathlets erlaubt auch hier die Lokalisierung der Problemstelle. Ein weiterer bei der Qualitätssicherung in SOIS zu beobachtender Aspekt ist die Ressourcennutzung. Das heißt, auch Änderungen bezüglich der Lastenverteilung auf die Fließbänder und Produktionsstätten müssen detektiert werden. Dazu können die bei der Rekonstruktion beobachteten Nutzungshäufigkeiten der Pathlets mit deren räumlichen Lage verknüpft werden, sodass durch die Auswertung entsprechender Wärmebilder Verschiebungen bezüglich der Ressourcennutzung detektiert und deren globale Auswirkungen analysiert werden können (siehe Abschnitt 3.6.4).

Als letzte der fünf Anomalien wurde in Abschnitt 3.6.5 die Problematik der Neuartigkeit ausgeführt, das heißt, die Fragestellung betrachtet, ob es sich bei einer detektierten Anomalie um das Fehlverhalten eines einzelnen Werkteils oder aber um eine Neuigkeit im gesamten Systemverhalten handelt. Dazu werden diesmal die relativen Nutzerhäufigkeiten der nicht rekonstruierbaren Teile ausgewertet und anschließend wie bereits bei der Überwachung der Lastenverteilung mit deren räumlichen Lage verknüpft. An den sich ergebenden partiellen Wärmebilder lässt sich schließlich der Grad der Anomalität ablesen. Damit konnte in diesem Kapitel also sowohl das grundsätzliche Konzept zur Erkennung und Analyse von Anomalien in raumbezogenen Daten mittels Substrukturverzeichnissen vorgestellt, als auch dessen Umsetzung und Anwendung im Rahmen der Laufzeitüberwachung für ein SOIS mit Fließbandarchitektur gezeigt und evaluiert werden.

4 Effiziente Umsetzung der Laufzeitüberwachung

In Kapitel 3 wurde das Konzept zur Erkennung und Analyse von Anomalien in SOIS mit Hilfe von Substrukturverzeichnissen eingeführt. Für eine möglichst effiziente Umsetzung dieses Ansatzes werden im Folgenden die Themen Skalierbarkeit der Substrukturextraktion und Onlinefähigkeit des Rekonstruktionsprozesses diskutiert.

Die Skalierbarkeit der Substrukturextraktion ist gerade im Fall von Trajektorien, die aus vielen Punktkoordinaten bestehen, von besonderer Bedeutung, um weiterhin mit realistischem Aufwand ein Substrukturverzeichnis erstellen zu können. Dazu wird in Abschnitt 4.1 eine Alternative zu dem in Abschnitt 3.4 eingeführten Pathlet Learning vorgestellt, das im Falle von punktreichen Trajektorien eine effiziente Substrukturextraktion bei gleichzeitig vernachlässigbaren Einbußen bezüglich der Ergebnisse der Anomalieerkennung ermöglicht.

Die Onlinefähigkeit des Rekonstruktionsprozesses ist ebenso ein zentraler Punkt für die erfolgreiche Umsetzung des Verfahrens, da gerade bei der Laufzeitüberwachung das schnelle Vorhandensein und eine mindestens genauso unmittelbare Analyse der Ergebnisse essentiell sind, um die Qualität im System konstant sicherzustellen. Dementsprechend wird in Abschnitt 4.2 eine Datenstruktur vorgestellt, die eine effiziente Rekonstruktion von Trajektorien aus Substrukturen bereits zur Laufzeit ermöglicht, um frühzeitig an die für Anomalieerkennung und -analyse notwendigen Informationen zu gelangen.

Die Ergebnisse zum Thema Skalierbarkeit der Substrukturextraktion wurden bereits in [Kier 17d] veröffentlicht. Die Ergebnisse zur Onlinefähigkeit des Rekonstruktionsprozesses wurden in [Kier 17a] vorveröffentlicht.

Wie in Abschnitt 1.3 beschrieben, stammen Idee, Implementierung und Evaluation in beiden Fällen vom Autor der vorliegenden Arbeit.

4.1 Skalierbarkeit der Substrukturextraktion

Wie in Kapitel 3 beschrieben, findet die initiale Erstellung des Substrukturverzeichnisses nicht zur Laufzeit, sondern in einer vorgelagerten Trainingsphase statt (siehe Abbildung 3.1). Damit ist der Lernprozess im Grunde nicht zeitkritisch beziehungsweise unterliegt keiner Echtzeitanforderung. Nichtsdestotrotz ist eine effiziente Substrukturextraktion vor allem bei punktreichen Trajektorien wünschenswert. Insbesondere wenn es darum geht auf Änderungen im SOIS

zu reagieren, die eine Anpassung des Normalitätsmodells erfordern und wofür ein neuer Lernprozess angestoßen werden muss.

Entsprechend wird in diesem Abschnitt eine Alternative zum Pathlet Learning vorgestellt, mit der Substrukturen für die Anomalieerkennung auch bei punktreichen Trainingstrajektorien effizient extrahiert werden können. Konkret kommt dabei der Grammatikinduktions-Algorithmus SEQUITUR [Nevi 97] zum Einsatz, der in linearer Zeit wiederkehrende Strukturen in einer Sequenz identifiziert.

Das heißt, statt des speziell für Trajektorien entwickelten Kompressionsverfahren mit quadratischer Komplexität, wird ein allgemeineres Kompressionsverfahren eingesetzt, wodurch zwar die Qualität der extrahierten Substrukturen etwas abnimmt, was aber im Falle von punktreichen Trajektorien mit dem deutlichen Effizienzgewinn gerechtfertigt werden kann.

Damit soll gezeigt werden, dass auch alternative Datenkompressionsverfahren, die wiederkehrende Strukturen in Datensätzen identifizieren, durchaus zur Erstellung eines Substrukturverzeichnisses zur Anomalieerkennung verwendet werden können. SEQUITUR als Grammatikinduktions-Algorithmus wurde hier als Beispiel gewählt, da sich Trajektorien gut in entsprechende Datensequenzen umwandeln lassen, das Verfahren damit lineare Laufzeit hat und einfach umzusetzen ist.

4.1.1 Substrukturextraktion mittels SEQUITUR

SEQUITUR ist ein Grammatikinduktionsverfahren, das zur Datenkompression eingesetzt werden kann [Nevi 97]. Dazu werden wiederkehrende Phrasen in einer Datensequenz identifiziert und durch Grammatikregeln ersetzt, wobei folgende Restriktionen zu beachten sind:

1. Digram-Eindeutigkeit: Kein Symbolpaar darf mehr als einmal in der Grammatik vorkommen.
2. Regelnutzung: Jede Grammatikregel muss mindestens einmal verwendet werden.

Im Fall von Trajektorien ist die Eingabe für SEQUITUR die Konkatenation aller Trainingstrajektorien zu einer langen Sequenz. Dazu werden die Trajektorien nicht durch ihre raumbezogenen Koordinaten x und y , sowie einen Zeitstempel t repräsentiert (siehe Abschnitt 2.1.3.1), sondern durch die eindeutigen Identifikationsnummern der Rasterzellen des gerasterten Raumplans des SOIS (siehe Abschnitt 3.3).

Zur Veranschaulichung ist in Abbildung 4.1 ein Beispiel skizziert, wie mittels SEQUITUR Substrukturen aus zwei Trainingstrajektorien extrahiert werden. Wie bereits erwähnt, sind die Trajektorien dafür über eine Sequenz von eindeutigen Identifikationsnummern der Rasterzellen des Raumplans repräsentiert (siehe Abbildung 4.1a und 4.1b). Aus der Konkatenation der Einzelsequenzen ergibt sich in diesem Fall die Eingabe

0	1	2	3	4
5	6	7	8	9
10	11	12	13	14
15	16	17	18	19
20	21	22	23	24

0	1	2	3	4
5	6	7	8	9
10	11	12	13	14
15	16	17	18	19
20	21	22	23	24

0	1	2	3	4
5	6	7	8	9
10	11	12	13	14
15	16	17	18	19
20	21	22	23	24

(a) Trajektorie 1:
(10,11,12,7,12,13,14)

(b) Trajektorie 2:
(10,11,12,7,12,17,22)

(c) Resultierende Regel:
(10,11,12,7,12)

Abbildung 4.1: Beispieltrajektorien und resultierende Regel nach Anwendung von SEQUITUR.

Tabelle 4.1: SEQUITUR für Eingabe 10,11,12,7,12,13,14,10,11,12,7,12,17,22

bisheriger Eingabe	resultierende Grammatik	Anmerkungen
10,11,12,7,12,13,14,10,11	$S \rightarrow 10,11,12,7,12,13,14,10,11$	2x 10,11
10,11,12,7,12,13,14,10,11	$S \rightarrow A,12,7,12,13,14,A$ $A \rightarrow 10,11$	Digram-Eindeutigkeit
10,11,12,7,12,13,14,10,11,12	$S \rightarrow A,12,7,12,13,14,A,12$ $A \rightarrow 10,11$	2x A,12
	$S \rightarrow B,7,12,13,14,B$ $A \rightarrow 10,11$ $B \rightarrow A,12$	Digram-Eindeutigkeit Regelnutzung (A)
	$S \rightarrow B,7,12,13,14,B$ $B \rightarrow 10,11,12$	
10,11,12,7,12,13,14,10,11,12,7	$S \rightarrow B,7,12,13,14,B,7$ $B \rightarrow 10,11,12$	2x B,7
	$S \rightarrow C,12,13,14,C$ $B \rightarrow 10,11,12$ $C \rightarrow B,7$	Digram-Eindeutigkeit Regelnutzung (B)
	$S \rightarrow C,12,13,14,C$ $C \rightarrow 10,11,12,7$	
10,11,12,7,12,13,14,10,11,12,7,12	$S \rightarrow C,12,13,14,C,12$ $C \rightarrow 10,11,12,7$	2x C,12
	$S \rightarrow D,13,14,D$ $C \rightarrow 10,11,12,7$ $D \rightarrow C,12$	Digram-Eindeutigkeit Regelnutzung (C)
	$S \rightarrow D,13,14,D$ $D \rightarrow 10,11,12,7,12$	
10,11,12,7,12,13,14,10,11,12,7,12,17,22	$S \rightarrow D,13,14,D,17,22$ $D \rightarrow 10,11,12,7,12$	

10, 11, 12, 7, 12, 13, 14, 10, 11, 12, 7, 12, 17, 22.

In Tabelle 4.1 sind die wichtigsten Schritte von SEQUITUR detaillierter aufgeschlüsselt, wobei die Eingabe sequentiell von vorne nach hinten durchlaufen wird, bzw. immer um ein weiteres Symbol der Sequenz ergänzt wird. Die erste Regel $A \rightarrow 10, 11$ ergibt sich auf Grund der Digram-Eindeutigkeitsrestriktion, sobald das Paar 10, 11 ein zweites Mal vorkommt. Im nächsten Schritt kommt erneut die Digram-Eindeutigkeitsrestriktion zum Einsatz, da $A, 12$ zweimal vorkommt. Allerdings macht diese neue Regel $B \rightarrow A, 12$ die erste Regel überflüssig, sodass entsprechend der Regelnutzungsrestriktion A wieder aus der Grammatik entfernt wird.

Die Umsetzung der beiden Restriktionen bei den weiteren Schritten resultiert schließlich neben der Gesamtsequenz S in der Regel $D \rightarrow 10, 11, 12, 7, 12$, die in Abbildung 4.1c visualisiert ist. Man sieht, dass es sich dabei genau um das sich in beiden Trainingstrajektorien wiederholende Teilstück handelt. Dass diese rechten Seiten der sich bei SEQUITUR ergebenden Grammatikregeln zur alternativen Erzeugung eines Substrukturverzeichnis für die Anomalieerkennung in SOIS verwendet werden können, wird in Abschnitt 4.1.4 gezeigt.

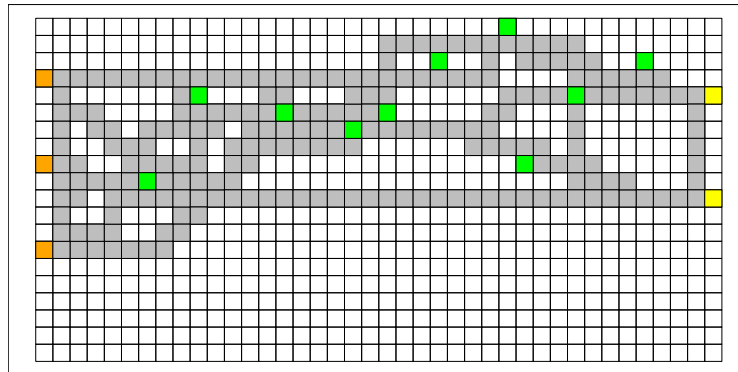
4.1.2 Simulationsmodelle

Zur Erzeugung von Trajektorien datensätze zur Evaluation des hier vorgestellten alternativen Ansatzes zur Substrukturextraktion wurden Simulationsmodelle ähnlich denen aus Kapitel 3 (siehe Abschnitt 3.3) erzeugt. Allerdings wurde in diesem Fall die Größe der SOIS und damit die Anzahlen an Samplepunkten pro Trajektorien variiert, um zu zeigen dass der SEQUITUR-basierte Ansatz insbesondere auch bei punktreichen Trajektorien funktioniert, wo der Effizienzgewinn schwerer wiegt.

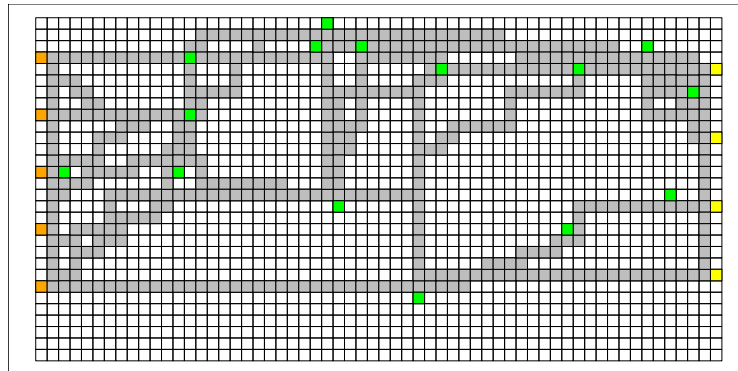
Insgesamt wurden drei verschiedene Simulationsmodelle generiert, wobei neben der Größe des SOIS die Anzahl an Produktionsstationen sowie Start- und Endpunkten vorgegeben werden konnten. Die Start- und Endpunkte werden dabei jeweils gleichmäßig auf die erste oder letzte Spalte des gerasterten Raumplanes verteilt, wohingegen die Produktionsstationen zufällig im Raster platziert und über Fließbänder verbunden werden, deren Struktur mit Hilfe von Delaunay Triangulation erzeugt werden. Die sich dabei für unterschiedliche Parameter ergebenden Simulationsmodelle sind in Abbildung 4.2 zu sehen.

Das Simulationsmodell aus Abbildung 4.2a hat dabei dieselbe Größe (40x20 Rasterzellen) wie die Simulationsmodelle aus Kapitel 3, wohingegen die beiden anderen deutlich größer sind (60x30 Rasterzellen in Abbildung 4.2b, bzw. 80x40 Rasterzellen in Abbildung 4.2c).

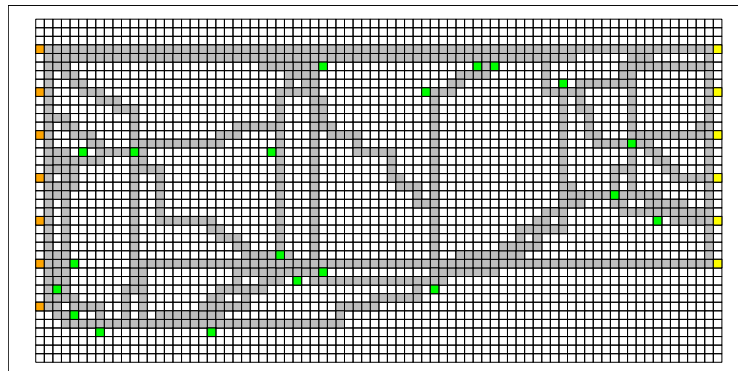
Wie bereits angedeutet, steigen mit zunehmender Größe des SOIS auch die Anzahl an Samplepunkten pro Trajektorie, da die Routen der Werkteile innerhalb der Fabrik auch entsprechend länger werden (siehe Tabelle 4.2).



(a) Simulationsmodell mit 40x20 Rasterzellen, drei Start-, zwei Endpunkten und zehn Produktionsstationen.



(b) Simulationsmodell mit 60x30 Rasterzellen, fünf Start-, vier Endpunkten und 15 Produktionsstationen.



(c) Simulationsmodell mit 80x40 Rasterzellen, sieben Start-, sechs Endpunkten und 20 Produktionsstationen.

Abbildung 4.2: Simulationsmodelle für SOIS unterschiedlicher Größen mit Startpunkten (orange), Endpunkten (gelb) und Produktionsstationen (grün), die durch Fließbänder (grau) verbunden sind.

Tabelle 4.2: Durchschnittliche Anzahl an Samplepunkten pro Trajektorie

Szenario	# Punkte pro Trajektorie (\emptyset)
40x20	~ 150
60x30	~ 350
80x40	~ 580

4.1.3 Evaluation

Zur Evaluation des SEQUITUR-basierten Ansatzes zur Substrukturextraktion wird dieser im Folgenden bezüglich Rechenkomplexität des Lernprozesses, *Universalität* der Substrukturen, *Kompaktheit* des Verzeichnisses und *Effizienz* des Rekonstruktionsprozesses mit dem Pathlet Learning-Ansatz aus Kapitel 3 verglichen.

4.1.3.1 Rechenkomplexität

Ein großer Vorteil des Pathlet Learning ist, dass die „optimale“ Zerlegung für jede Trajektorie unabhängig und damit, rein theoretisch, zeitgleich berechnet werden kann (siehe Abschnitt 3.4), sodass der Gesamtrechenaufwand linear mit der Anzahl der Trainingstrajektorien steigt. Die Rechenkomplexität jeder einzelner Zerlegung hängt allerdings von der Anzahl der Samplepunkte pro Trajektorie ab. Der für die Zerlegung verwendete Kürzeste-Pfad-Suchalgorithmus von Dijkstra hat eine Komplexität von $O(|E| \log(|V|))$, wobei $|E|$ die Anzahl an Kanten und $|V|$ die Anzahl an Knoten im Grafen bezeichnet. Sei nun n die Anzahl an Samplepunkten einer Trajektorie, dann gilt für das Pathlet Learning $|V| = n$ und $|E| = \binom{n}{2}$ (alle möglichen Teiltrajektorien). Dies resultiert in einer Rechenkomplexität von $O(n^2 \log(n))$ für die Zerlegung einer Trajektorie. Das heißt, insgesamt ergibt sich für eine Menge von m Trainingstrajektorien, die im Durchschnitt aus n Samplepunkten bestehen, ein Rechenaufwand von $O(mn^2 \log(n))$.

Im Gegensatz dazu steigt der Rechenaufwand bei dem SEQUITUR-basierten Ansatz linear mit der Anzahl an Samplepunkten aller Trainingstrajektorien. Dies entspricht in diesem Fall einer Gesamtrechenkomplexität von $O(mn)$.

Das heißt, statt einer Komplexität, die quadratisch mit der Anzahl der Samplepunkte pro Trajektorie steigt, ermöglicht der SEQUITUR-basierte Ansatz mit seiner linearen Komplexität eine deutliche Effizienzsteigerung bei der Substrukturextraktion.

4.1.3.2 Universalität der Substrukturen

Wie bereits in Abschnitt 3.5 erläutert, ist die *Universalität* der Substrukturen wichtig, um sicherzustellen, dass auch Trajektorien rekonstruiert werden können, die nicht im Trainingsdatensatz enthalten sind, aber durch denselben Prozess erzeugt wurden. Zur Evaluation dieser *Universalität* kann die *durch-*

schnittliche Überdeckungsrate verwendet werden, die sich bei der Rekonstruktion von in diesem Fall 1000 Testtrajektorien ergibt. Entsprechend wird im Folgenden nun untersucht, ob sich für die gegebenen Szenarien sowohl mittels Pathlet Learning als auch mittels SEQUITUR Substrukturen extrahieren lassen, die *universell* genug sind, um schließlich zur Anomalieerkennung eingesetzt werden zu können.

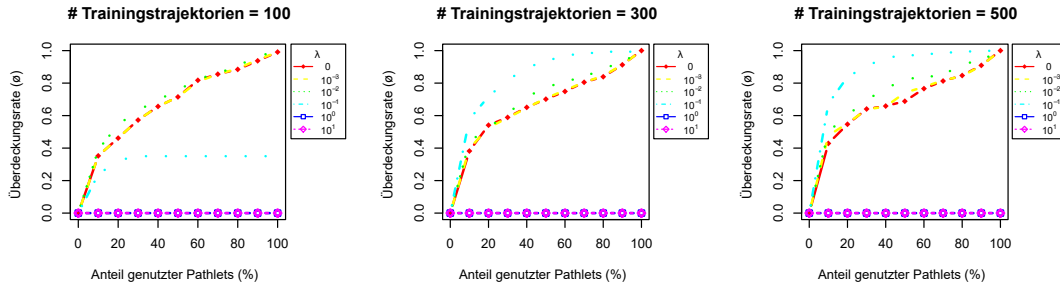
Dazu wird zunächst für das Pathlet Learning für jedes Szenario der Parameter λ und die Anzahl an Trainingsdaten bestimmt, die benötigt werden, um mit dem resultierenden Verzeichnis eine hohe Überdeckungsrate bei den Testdaten zu erzielen. Die Ergebnisse für $\lambda = \{0, 10^{-3}, 10^{-2}, 10^{-1}, 10^0, 10^1, 10^2\}$ und Trainingsdatensatzgrößen $\# \text{ Trainingstrajektorien} = \{100, 300, 500\}$ sind in Abbildung 4.3 zu sehen, wobei zusätzlich jeweils bereits die Größe des Pathletverzeichnisses wie in Abschnitt 3.5 beschrieben, abhängig vom Anteil genutzter Pathlets variiert wird.

Offensichtlich kann unabhängig vom Szenario für $\lambda \geq 1$ ($= 10^0$) kein brauchbares Pathletverzeichnis erzeugt werden, da in diesem Fall die Testtrajektorien nicht mehr rekonstruiert werden können (Überdeckungsrate = 0). Außerdem sieht man, dass lediglich für das kleinste Szenario 100 Trainingstrajektorien ausreichen; für das zweite und dritte Szenario werden mindestens 300 Trainingstrajektorien benötigt, um eine Überdeckungsrate von $> 80\%$ zu erhalten. Grundsätzlich lässt sich festhalten, dass für alle drei Szenarien Substrukturverzeichnisse mittels Pathlet Learning erzeugt werden können, die ausreichend *universell* sind, das heißt, eine Überdeckungsrate von nahezu 100% bei der Rekonstruktion der 1000 Testtrajektorien vorweisen können.

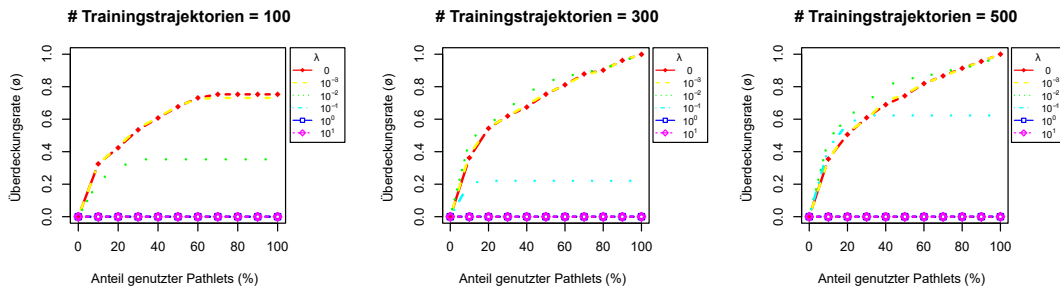
Für die Substrukturextraktion mittels SEQUITUR ist die Anzahl der Trainingstrajektorien der einzige Parameter, der bei der Erzeugung von Regelverzeichnissen abgestimmt werden muss. Dementsprechend sind in Abbildung 4.4 die durchschnittlichen Überdeckungsraten in Abhängigkeit von der Trainingsdatensatzgröße ($\# \text{ Trainingstrajektorien} = \{100, 300, 500\}$) für die drei Szenarien zu sehen, wobei auch hier die Verzeichnisgröße auf der x-Achse variiert wird.

Man sieht, dass in diesem Fall 100 Trainingstrajektorien in allen drei Szenarien ausreichen, um ein adäquates Substrukturverzeichnis mit Überdeckungsraten von nahezu 100% zu erzeugen. Weiterhin ist interessant, dass diese hohen Werte auch mit nur 30% der Regeln erreicht werden können. Diese Beobachtung ist gerade für die Bewertung des SEQUITUR-Ansatzes bezüglich der noch folgenden Evaluationskriterien *Kompaktheit* des Substrukturverzeichnisses und *Effizienz* des Rekonstruktionsprozesses von Bedeutung.

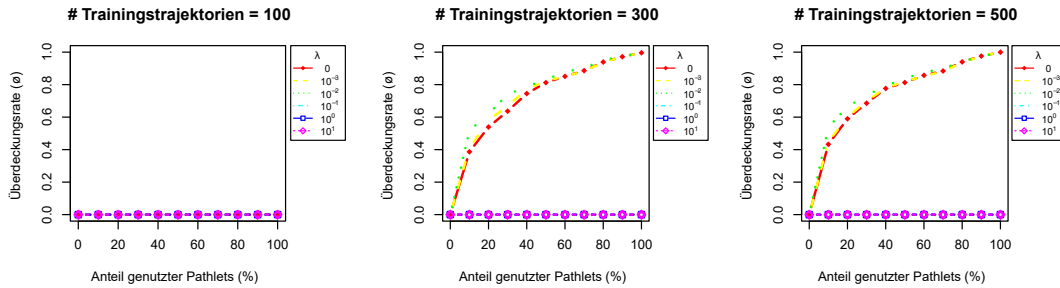
Insgesamt konnte also gezeigt werden, dass sich durch die Wahl geeigneter Parametergrößen sowohl mittels Pathlet Learning, als auch mittels SEQUITUR für alle drei Szenarien *universelle* Substrukturen aus Trajektorien extrahieren lassen. Die sich daraus ergebenden Substrukturverzeichnisse haben durchschnittliche Überdeckungsraten von nahezu 100% und können damit zur Anomalieerkennung eingesetzt werden.



(a) Durchschnittliche Überdeckungsrate für das 40x20 Szenario.



(b) Durchschnittliche Überdeckungsrate für das 60x30 Szenario.



(c) Durchschnittliche Überdeckungsrate für das 80x40 Szenario.

Abbildung 4.3: Durchschnittliche Überdeckungsrate für Substrukturverzeichnisse, die mittels Pathlet Learning erstellt wurden.

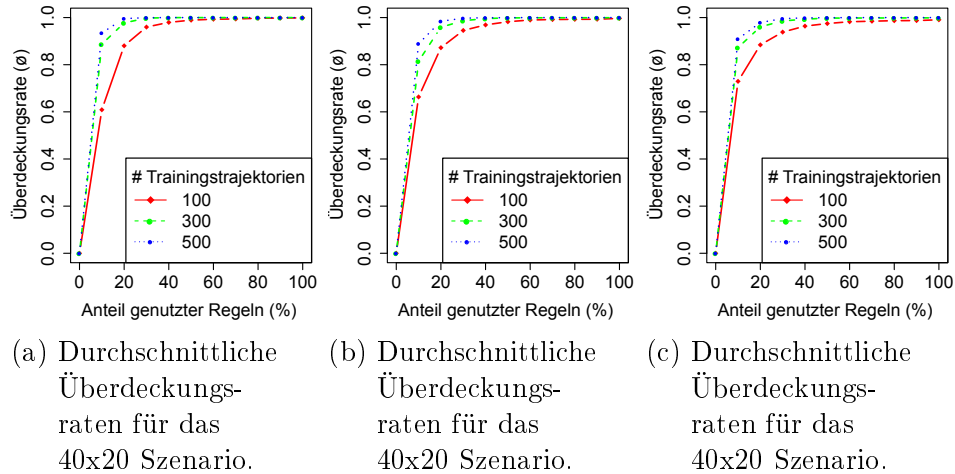


Abbildung 4.4: Durchschnittliche Überdeckungsrate für Substrukturverzeichnisse, die mittels SEQUITUR erstellt wurden.

4.1.3.3 Kompaktheit des Substrukturverzeichnisses

Die Kompaktheit des Substrukturverzeichnisses ist ein weiteres wichtiges Evaluationskriterium, da man sowohl aus Speicher technischen, aber auch kombinatorischen Gründen mit möglichst wenigen Substrukturen auskommen möchte. Zur Evaluation der Kompaktheit der Substrukturverzeichnisse werden daher im Folgenden für die drei Szenarien die Verzeichnisgrößen verglichen, die sich durch die Verwendung von Pathlet Learning bzw. SEQUITUR ergeben. Im Fall von Pathlet Learning werden dafür ausschließlich Verzeichnisse betrachtet, die eine Überdeckungsrate von $\geq 95\%$ haben (siehe Abschnitt 4.1.3.2). Das heißt, für das zweite und dritte Szenario entfallen die Ergebnisse für eine Trainingsdatensatzgröße von 100 und für die unterschiedlichen λ -Werte ($\lambda = \{0, 10^{-3}, 10^{-2}, 10^{-1}\}$) ergibt sich eine Spanne an minimalen und maximalen Anzahlen an Pathlets, da die Anzahl an Pathlets gleichzeitig mit λ steigt (siehe Gleichung 3.1).

In Tabelle 4.3 sind für alle drei Szenarien die jeweiligen Anzahlen von sich ergebenden Pathlets bzw. Regeln aufgeführt. Offensichtlich ist die Anzahl von Regeln unabhängig vom Szenario und der Anzahl an Trainingstrajektorien deutlich höher als die Anzahl an Pathlets. Allerdings werden, wie bereits in Abschnitt 4.1.3.2 erwähnt, nicht alle Regeln benötigt, um eine ausreichend hohe durchschnittlich Überdeckungsrate zu erzielen. Im Speziellen haben die Experimente gezeigt, dass bereits 30% der Regeln ausreichen (siehe Abbildung 4.3). Dementsprechend sind in der letzten Spalte von Tabelle 4.3 zusätzlich die Werte für das entsprechend reduzierte Verzeichnis zu sehen. Durch diese Optimierung der Verzeichnisgröße sinkt in allen Fällen die Anzahl an benötigten Regeln sogar unter die minimale Anzahl an Pathlets, die sich bei einem λ -Wert von 0 ergibt.

Damit konnte insgesamt also gezeigt werden, dass der SEQUITUR-basierte

Tabelle 4.3: Anzahl an Verzeichniseinträgen.

Szenario	# Trainingstrajektorien	# Pathlets	# Regeln	# Regeln'
40x20	100	280 – 287	552	166
	300	306 – 585	1004	302
	500	334 – 945	1298	390
60x30	100	-	1023	307
	300	612 – 735	1810	543
	500	688 – 935	2244	674
80x40	100	-	1468	441
	300	984 – 1096	2684	806
	500	1050 – 1397	3384	1016

Ansatz bezüglich der *Kompaktheit* der sich ergebenden Substrukturverzeichnisse unabhängig vom Szenario durchaus mit dem Pathlet Learning-Ansatz vergleichbar ist, bzw. sich nach dem Optimierungsschritt damit sogar kompaktere Substrukturverzeichnisse ergeben.

4.1.3.4 Effizienz des Rekonstruktionsprozesses

Bei der *Effizienz* des Rekonstruktionsprozesses geht es darum, die Anzahl an Verzeichniseinträgen, die im Durchschnitt für die Rekonstruktion einer Trajektorie benötigt werden, möglichst gering zu halten, um möglichst wenig Abgleichschritte durchführen zu müssen.

Für die Evaluation des Pathlet Learning-, beziehungsweise SEQUITUR-basierten Ansatzes bezüglich dieser *Effizienz* werden dieselben Verzeichnisse herangezogen wie bereits in Abschnitt 4.1.3.3. Das heißt, auch hier gibt es für das Pathlet Learning beim zweiten und dritten Szenario für eine Trainingsdatensatzgröße von 100 keine Ergebnisse und in allen anderen Fällen wird die Spanne an Ergebnissen angegeben, die sich für die unterschiedlichen λ -Werte und damit verschieden große Verzeichnisse ergeben. Für den SEQUITUR-basierten Ansatz wird hier bereits von dem um 70% reduzierten Regelverzeichnis ausgegangen.

Die entsprechenden Ergebnisse sind in Tabelle 4.4 zu sehen. Dazu wurden 1000 Testtrajektorien mit Hilfe des jeweiligen Substrukturverzeichnisses rekonstruiert, die Anzahl an Pathlets bzw. Regeln, die zur Rekonstruktion der Trajektorien verwendet wurden, aufsummiert und anschließend über die Anzahl an Testtrajektorien gemittelt.

Man sieht, dass unabhängig vom Szenario und der Anzahl an Trainingsdaten die durchschnittliche Anzahl an Regeln pro Trajektorie deutlich größer ist, als der entsprechende Maximalwert beim Pathlet Learning. Der Grund dafür zeigt sich, wenn man die Länge, das heißt, die Anzahl an Samplepunkten der Regeln bzw. Pathlets in den jeweiligen Verzeichnissen betrachtet (siehe Tabelle 4.5). Offensichtlich sind unabhängig vom Szenario die Regeln im Durch-

Tabelle 4.4: Durchschnittliche Anzahl an Verzeichniseinträgen, die zur Rekonstruktion verwendet werden.

Szenario	# Trainings- trajektorien	# Pathlets pro Trajektorie (\emptyset)	# Regeln pro Trajektorie' (\emptyset)
40x20	100	57.4 – 66.6	79.6
	300	32.4 – 82.6	99.5
	500	37.0 – 91.1	88.6
60x30	100	-	159.3
	300	127.5 – 153.6	225.2
	500	147.6 – 179.9	219.9
80x40	100	-	232.5
	300	160.6 – 213.2	303.2
	500	185.2 – 208.3	307.5

Tabelle 4.5: Durchschnittliche Länge von Verzeichniseinträgen.

Szenario	# Trainings- trajektorien	Pathletlänge (\emptyset)	Regellänge' (\emptyset)
40x20	100	20.9 – 26.3	6.1
	300	13.3 – 52.9	8.8
	500	13.5 – 33.4	11.5
60x30	100	-	8.2
	300	20.8 – 26.9	11.7
	500	20.4 – 27.7	14.5
80x40	100	-	9.3
	300	33.9 – 76.8	13.0
	500	26.5 – 33.6	16.2

schnitt deutlich kürzer als die Pathlets, was wiederum erklärt, warum beim SEQUITUR-basierten Ansatz im Durchschnitt mehr Einträge zur Rekonstruktion einer Trajektorie benötigt werden.

Insgesamt hat sich also gezeigt, dass im direkten Vergleich der Pathlet Learning-basierte Ansatz bezüglich der *Effizienz* des Rekonstruktionsprozess in allen drei Szenarien besser abschneidet als der SEQUITUR-basierte Ansatz. Die Tatsache, dass beim SEQUITUR-basierten Ansatz durchschnittlich mehr Verzeichniseinträge zur Rekonstruktion einer Trajektorie benötigt werden, ergibt sich daraus, dass die extrahierten Regeln im Durchschnitt deutlich kürzer sind als die Pathlets.

4.1.4 Erkennung und Analyse von Anomalien

Wie gerade gezeigt, können mittels SEQUITUR ähnlich zum Pathlet Learning wiederkehrende Teiltrajektorien aus einer Menge von Trainingstrajektorien ex-

trahiert werden. Deren Einsatzmöglichkeit in Form von Substrukturverzeichnissen zur Erkennung und Analyse von Anomalien in SOIS wird im Folgenden für die beiden zentralen Anomaliearten – anormale Einzeltrajektorien und globale anormale Ereignisse – gezeigt.

Dazu wird sich auf das dritte Szenario mit durchschnittlich 580 Samplepunkten pro Trajektorie (siehe Abbildung 4.2c und Tabelle 4.2) konzentriert, da die lineare Skalierbarkeit von SEQUITUR insbesondere bei punktreichen Trajektorien von besonderem Interesse ist.

4.1.4.1 Einzeltrajektorien

Zur Erkennung anormaler Einzeltrajektorien kann wie in Abschnitt 3.6.1 beschrieben die Überdeckungsrate als Evaluationsmaß genutzt werden. Entsprechend sind in Abbildung 4.5 Histogramme zu sehen, die die Überdeckungsraten für die Rekonstruktion von 100 normalen Testtrajektorien zeigen. In Abbildung 4.5a wurde dazu ein reduziertes Regelverzeichnis angewendet, das mit Hilfe von 100 Trainingstrajektorien erzeugt wurde. Dem gegenüber steht das Histogramm aus Abbildung 4.5b, das die Ergebnisse für ein Regelverzeichnis zeigt, das für das Training 300 Trajektorien zur Verfügung hatte.

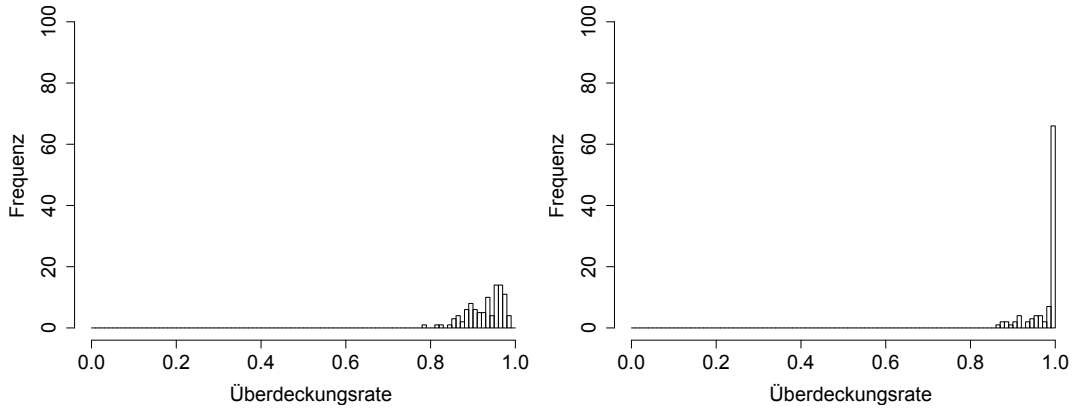
In Übereinstimmung mit den durchschnittlichen Überdeckungsraten aus Abbildung 4.4 fallen die Überdeckungsraten für das Regelverzeichnis mit 100 Trainingstrajektorien insgesamt etwas niedriger aus als bei der Verwendung von 300 Trainingstrajektorien. Im Speziellen sind die Balken in Abbildung 4.5a vor allem um den Wert 0.95 geclustert, wobei die Werte bis zu 0.78 nach links hin auslaufen.

Weiterhin ist für dieses Szenario interessant, dass scheinbar keine Testtrajektorie vollständig rekonstruiert werden kann (Überdeckungsrate = 1). Im Gegensatz dazu können im Fall von 300 Trainingstrajektorien 81% der Testtrajektorien vollständig rekonstruiert werden und selbst die wenigen Ausreißer nach links haben immer noch eine Überdeckungsrate von ≥ 0.87 (siehe Abbildung 4.5b).

An dieser Stelle stellt sich wieder die Frage nach dem geeigneten Schwellwert zur Definition der (Ab-)Normalität von Trajektorien. Für das Regelverzeichnis, das mit 100 Trainingstrajektorien erzeugt wurde, könnte zum Beispiel ein Schwellwert von 0.8 gewählt werden. Damit würden Trajektorien, die zu mehr als 80% rekonstruiert werden können, als normal klassifiziert werden. Die Richtig-Postiv-Rate wäre dadurch sehr hoch, aber Trajektorien würden erst ab einer Abweichung von mehr als 20% von den bekannten Pfaden als anormal gelten.

Im Gegensatz dazu könnte im zweiten Fall mit 300 Trainingstrajektorien der Schwellwert höher angesetzt werden, da hier die Überdeckungsraten für die normalen Testtrajektorien insgesamt deutlich höher sind. Damit könnten auch Trajektorien, die um weniger als 20% von den bekannten Pfaden abweichen als anormal erkannt werden.

Das heißt, ist ein strikter Schwellwert aus sicherheitskritischen Gründen erfor-



(a) Ergebnis unter Verwendung von 100 Trainingstrajektorien. (b) Ergebnis unter Verwendung von 300 Trainingstrajektorien.

Abbildung 4.5: Überdeckungsrate bei der Rekonstruktion von 100 Testtrajektorien für unterschiedlich große Trainingsdatensets.

derlich, sollten zur Erstellung des Regelverzeichnis deutlich mehr Trainings-trajektorien verwendet werden; wenn auch ein etwas schwächerer Schwellwert ausreicht, kann aber auch mit weniger Trainingstrajektorien ein geeignetes Substrukturverzeichnis erstellt werden.

Insgesamt konnte also gezeigt werden, dass auch mittels SEQUITUR erstellte Substrukturverzeichnisse zur Erkennung von anormalen Einzeltrajektorien in einem SOIS eingesetzt werden können. Dazu wird über die Ergebnisse der Rekonstruktion von Testtrajektorien ein Schwellwert definiert, der anzeigt zu welchem Grad diese normalen Trajektorien rekonstruiert werden können und ab welchem Abweichungsgrad damit eine Route als abnormal erkannt werden kann.

4.1.4.2 Globale Ereignisse

Zur Erkennung globaler anormaler Ereignisse in einem SOIS kann wie in Abschnitt 3.6.2 beschrieben die Nutzungshäufigkeit als Evaluationsmaß verwendet werden. Zusätzlich kann ein solches Ereignis lokalisiert werden, indem räumliche Überschneidungen der betroffenen Verzeichniseinträge gesucht werden.

Zur Simulation eines solchen globalen anormalen Ereignisses wurde zufällig eine der Produktionsstationen aus dem dritten Simulationsmodell entfernt (siehe Abbildung 4.6a) und damit entsprechende Testtrajektorien erzeugt.

Für die Anomalieerkennung wurde ein reduziertes Regelverzeichnis genutzt, das unter Verwendung von 100 Trainingstrajektorien erzeugt wurde. Wie in Abbildung 4.6c zu sehen ist, sinken wie erwartet die relativen Nutzerhäufigkeiten für einige der Verzeichniseinträge auf 0 (schwarze Pfeile).

Zur Lokalisierung des globalen anormalen Ereignisses können die entsprechenden Verzeichniseinträge zusätzlich in Form eines Wärmebildes visualisiert werden (siehe Abbildung 4.6b). Dabei entspricht der Farbton der Überlagerungs-

häufigkeit der nicht mehr genutzten Verzeichniseinträge (dunkelrot für eine Überlagerung von sechs Verzeichniseinträgen, dunkel grün für eine einfache Nichtnutzung). Man sieht deutlich, dass der Standort der zuvor entfernten Maschine tatsächlich als einzige Rasterzelle dunkelrot eingefärbt ist, weil sich hier alle nicht mehr genutzten Verzeichniseinträge räumlich überschneiden.

Insgesamt konnte also gezeigt werden, dass Substrukturverzeichnisse, die mittels SEQUITUR erstellt wurden, auch für die Erkennung von globalen anormalen Ereignissen, wie der Nichtnutzung einer Produktionsstation in einem SOIS, genutzt werden kann. Zur weiteren Analyse beziehungsweise Lokalisierung der detektierten Anomalie kann zusätzlich ein Wärmebild erstellt werden, das die Lage der Verzeichniseinträge, deren relative Nutzerhäufigkeiten auf 0 gesunken sind, räumlich zuordnet.

Diese Analyse ist insbesondere bei großen SOIS mit vielen Produktionsstationen hilfreich, da ein einfacher visueller Vergleich der nicht mehr genutzten Verzeichniseinträge in solchen Fällen meist nicht mehr zielführend ist, sondern stattdessen eine automatisiert Lokalisierung wie zum Beispiel durch ein Wärmebild benötigt wird.

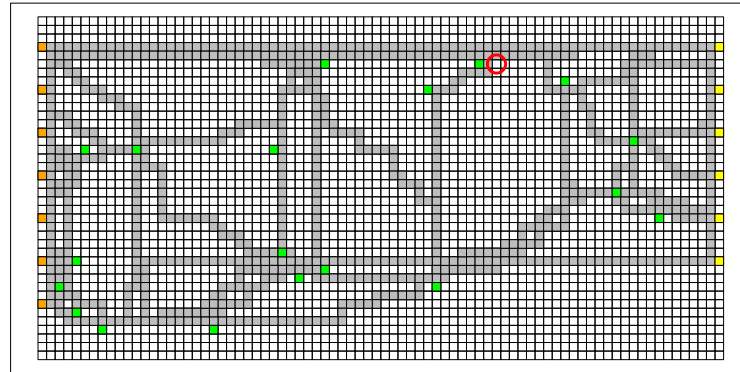
4.1.5 Zusammenfassung

In diesem Abschnitt wurde der Aspekt der Skalierbarkeit des Substrukturextraktionprozesses für das in Kapitel 3 vorgestellte Konzept zur Erkennung und Analyse von Anomalien in raumbezogenen Daten von SOIS thematisiert. Dazu wurde SEQUITUR als Effizienz optimierende Alternative zu dem in Abschnitt 3.4 vorgestellten Qualität optimierenden Pathlet Learning vorgestellt und evaluiert.

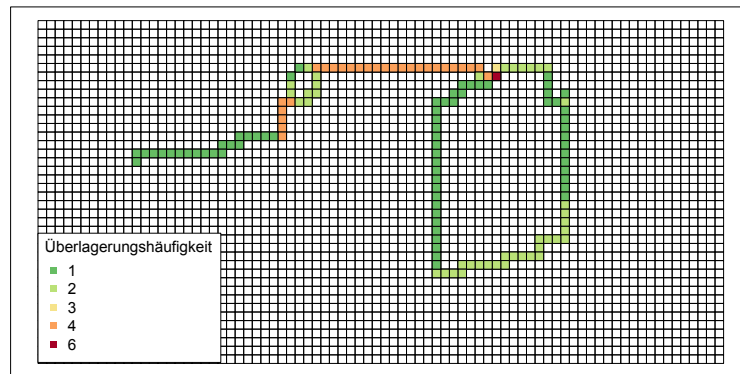
Zur Erzeugung des Substrukturverzeichnisses mittels SEQUITUR werden die Trainingstrajektorien zu einer einzigen Datenreihe konkateniert, sodass nach Anwendung des Grammatikinduktionsverfahrens auf diese Sequenz am Ende die rechten Seiten der sich dabei ergebenden Regeln die Einträge des Verzeichnisses ergeben (siehe Abschnitt 4.1.1).

Da die Skalierbarkeit insbesondere bei Trajektorien mit vielen Punktkoordinaten relevant ist, wurden in diesem Abschnitt noch drei weitere Simulationsmodelle von SOIS eingeführt, mit denen sich unterschiedlich punktreiche Trainings- und Testdaten erzeugen lassen (siehe Abschnitt 4.1.2). Anhand dieser drei Evaluationsszenarien konnten dann in Abschnitt 4.1.3 die beiden Verfahren – Pathlet Learning und SEQUITUR – bezüglich der Rechenkomplexität des Lernprozesses, der *Universalität* der sich ergebenden Substrukturen, der *Kompaktheit* des Verzeichnisses und der *Effizienz* des Rekonstruktionsprozesses verglichen werden.

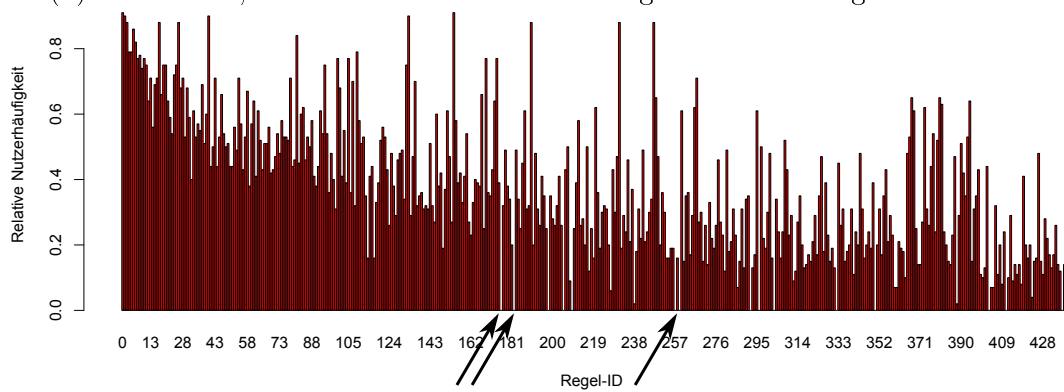
Für die Rechenkomplexität ergab sich, dass die Komplexität beim SEQUITUR-basierenden Ansatz lediglich linear mit der Anzahl der Samplepunkte pro Trajektorie steigt und nicht quadratisch wie beim Pathlet Learning, was eine enorme Effizienzsteigerung bei der Substrukturextraktion ermöglicht (siehe



(a) Simulationsmodell des hochauflösenden Beispielszenarios, wobei der rote Kreis eine entfernte Maschine markiert.



(b) Wärmebild, das die Rasterzellen der nicht genutzten Einträge visualisiert.



(c) Relative Nutzerhäufigkeit der Testtrajektorien.

Abbildung 4.6: Globales anomales Ereignis.

Abschnitt 4.1.3.1).

Hinsichtlich der Universalität konnten unabhängig vom Szenario mit beiden Verfahren durch die geeignete Wahl von Parametern Substrukturverzeichnisse erzeugt werden, die ausreichend hohe durchschnittliche Überdeckungsraten bei der Rekonstruktion der Testdaten erzielten, wobei hervorzuheben ist, dass im Falle sehr punktreicher Trajektorien der SEQUITUR-basierte Ansatz mit weniger Trainingstrajektorien auskommt als das Pathlet Learning (siehe Abschnitt 4.1.3.2).

Bei der Kompaktheit der Substrukturverzeichnisse ergeben sich bei einfacher Anwendung der beiden Verfahren zunächst große Unterschiede auf Grund einer hohen Redundanz in den durch SEQUITUR erzeugten Regeln. Diese Differenz löst sich allerdings bei dem in beiden Fällen durchgeführten zusätzlichen Optimierungsschritt, bei dem der tatsächlich zur Rekonstruktion benötigte Anteil der Verzeichniseinträge bestimmt wird, auf beziehungsweise kehrt sich sogar um (siehe Abschnitt 4.1.3.3).

Im Bezug auf die Effizienz des Rekonstruktionsprozesses zeigte sich allerdings, dass die mittels Pathlet Learning erzeugten Substrukturen in allen drei Szenarien deutlich länger waren und damit auch im Durchschnitt offensichtlicher weniger Pathlets pro Rekonstruktion benötigt wurden (siehe Abschnitt 4.1.3.4). Damit konnte im direkten Vergleich gezeigt werden, dass SEQUITUR eine adäquate Alternative zur Substrukturextraktion mittels Pathlet Learning ist, bei der man zwar geringfügige Einbußen bei der effizienten Rekonstruktion hinnehmen muss, allerdings die Rechenkomplexität linear und nicht quadratisch mit der Anzahl an Koordinatenpunkten pro Trajektorie skaliert.

Die Anwendbarkeit der mittels SEQUITUR erzeugten Substrukturverzeichnisse zur Erkennung und Analyse von Anomalien in SOIS wurde beispielhaft für das Evaluationsszenario mit den punktreichsten Trajektorien in Abschnitt 4.1.3 evaluiert. Analog zu dem in Kapitel 3 vorgestellten Vorgehen wurden dafür die entsprechend bei der Rekonstruktion anfallenden Informationen ausgewertet, um in diesem Fall anormale Einzeltrajektorien und globale anormale Ereignisse in dem vorliegenden SOIS erfolgreich zu detektieren und analysieren (siehe Abschnitt 4.1.4.1 bzw. Abschnitt 4.1.4.2).

4.2 Onlinefähigkeit des Rekonstruktionsprozesses

Im Gegensatz zur Erstellung des Substrukturverzeichnisses findet der Rekonstruktionsprozess zur Laufzeit statt (siehe Abbildung 3.1) und ist damit zeitkritisch. Das heißt, die Ergebnisse müssen möglichst in Echtzeit vorliegen, um diese unmittelbar zu analysieren und relevante Informationen für die Anomalieerkennung abzuleiten. Insbesondere in Fällen, in denen das Andauern der Anomalie weitere, unter Umständen sogar sicherheitskritische Auswirkungen auf das restliche System hat, ist die Onlinefähigkeit des Rekonstruktionspro-

zesses unabdingbar.

Besonders deutlich wird das am Beispiel eines Werkteiles, das sich anormal durch die Fabrik bewegt und dabei andere Werkteile beschädigen oder Unfälle verursachen kann. Hier ist eine schnelle Detektion der Anomalie notwendig, um weiteren Schaden zu verhindern.

Dementsprechend wird in diesem Abschnitt ein Verfahren für einen inversen Substrukturabgleich, engl. „reverse subpattern matching“, vorgestellt, der einen effizienten Rekonstruktionsprozess zur Laufzeit ermöglicht. Dazu wird eine Datenstruktur zur Indexierung von Substrukturen eingeführt, die das Erstellen von *Kandidatenlisten*, engl. „candidate lists“, und *Listen übereinstimmender Substrukturen*, engl. „matched pattern lists“, ermöglicht. Basierend auf den Zwischenergebnissen der *Listen übereinstimmender Substrukturen* können bereits zur Laufzeit beziehungsweise unmittelbar danach die sich bei der Rekonstruktion ergebenden Informationen wie die *Überdeckungsrate* abgegriffen und zur Anomalieerkennung und -analyse verwendet werden.

4.2.1 Musterabgleich in Trajektorien

Beim Musterabgleich, engl. „pattern matching“, in Trajektorien geht es darum eine *Anfragetrajektorie*, engl. „query trajectory“, mit *Mustertrajektorien*, engl. „trajectory patterns“, zu vergleichen, um den Grad ihre Überdeckung bestimmen zu können. Dabei unterscheidet man drei Arten des Musterabgleichs: Vollständiger Musterabgleich, Substrukturabgleich und Inverser Substrukturabgleich [Roh 11b, Roh 11a]. In Abbildung 4.7 sind jeweils entsprechende Beispiele skizziert, wobei die durchgezogene Linie die Anfragetrajektorie und die gestrichelte Linie die Mustertrajektorie repräsentiert.

Beim **vollständigen Musterabgleich** haben beide Trajektorien eine ähnliche Länge (siehe Abbildung 4.7a). Eine typische Anwendung für diese Art des Musterabgleichs ist das Finden von Personengruppen, die eine ähnliche Route genommen haben. Beim **Substrukturabgleich**, engl. „subpattern matching“, ist die Anfragetrajektorie deutlich kürzer als die Mustertrajektorie, mit der sie verglichen wird (siehe Abbildung 4.7b). Dementsprechend wird ein derartiger Abgleich zum Finden von Routen eingesetzt, die alle einen bestimmten Teilpfad gemeinsam haben. Im letzten Fall, dem **inversen Substrukturabgleich**, ist es genau andersherum. Das heißt, die Anfragetrajektorie ist länger als die Mustertrajektorien (siehe Abbildung 4.7c). Damit lassen sich aus einer Menge von Routenabschnitten die herausfiltern, die im angefragten Pfad vorkommen.

Während in der Praxis hauptsächlich die beiden ersten Formen des Musterabgleichs zum Einsatz kommen, gibt es vergleichsweise wenige Anwendungen, die einen inversen Substrukturabgleich erfordern. Der Rekonstruktionsprozess als Teil des in der vorliegenden Arbeit vorgestellten Anomalieerkennungsverfahrens (siehe Abschnitt 3.1) ist allerdings ein solcher Anwendungsfall.

Die extrahierten Substrukturen entsprechen den Mustertrajektorien und die zu evaluierende Route eines Werkteiles ist die Anfragetrajektorie. Zur Routen-

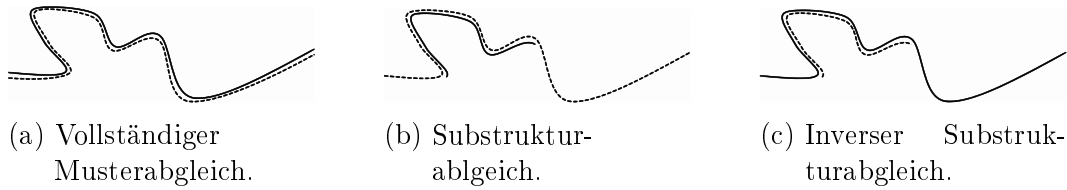


Abbildung 4.7: Drei Arten von Musterabgleich.

rekonstruktion müssen zunächst alle Substrukturen im Verzeichnis gefunden werden, die sich mit der Route des Werkteils überdecken. Anschließend kann der Grad der Rekonstruktion und damit die (Ab-)Normalität der Route ermittelt werden, indem der Anteil an durch Substrukturen überdeckten Teile der Trajektorie berechnet wird.

Wie eingangs erwähnt, findet dieser Rekonstruktionsprozess zur Laufzeit statt und unterliegt damit dem Anspruch, möglichst schnell Zwischenergebnisse bezüglich der Überdeckungsrate zu liefern. Das heißt, der inverse Substrukturabgleich muss auch mit Anfragetrajektorien umgehen können, die noch nicht vollständig vorliegen, da zum Beispiel das Werkteil im SOIS noch in Bewegung ist.

Dementsprechend wird im Folgenden ein Ansatz zum inversen Substrukturabgleich vorgestellt, der im Speziellen auch diesen Aspekt der Onlinefähigkeit erfüllt und damit zur online Rekonstruktion von Trajektorien eingesetzt werden kann.

Für diesen Ansatz wird eine netzwerkbasierte Repräsentation der Trajektorien angenommen, engl. „network-constrained trajectories“. Diese wird verwendet, wenn die zu beobachtenden Bewegungen nicht beliebig sind, sondern einer zugrundeliegenden Struktur folgen, wie zum Beispiel Autos dem Straßennetz. Im Gegensatz zur bisher verwendeten Rasterstruktur ist diese Form der Darstellung gerade bei großen Flächen effizienter, da nicht die gesamte Grundfläche gerastert werden muss und damit auch Rasterzellen inklusive eindeutiger Identifikationsnummer vorgehalten werden, die eigentlich überhaupt nicht benötigt werden, sondern nur die tatsächlich genutzten Kanten. Die Trajektorien werden dann als Sequenz von Netzwerkkanten, die mit einer eindeutigen Identifikationsnummer versehen sind, dargestellt.

Dementsprechend wird im Folgenden auf eine netzwerkbasierte Repräsentation der Trajektorien umgestellt, wobei die Konformität zur bisherigen Rasterstruktur des SOIS erhalten wird, indem die dabei zugrunde liegende Netzwerkstruktur ein Raster an Kanten ist, das über die Grundfläche gelegt wird. Damit ist der im Folgenden vorgestellte Ansatz zum inversen Substrukturabgleich unabhängig von einer Rasterung der Grundfläche anwendbar, wenn stattdessen eine netzwerkbasierter Trajektorienrepräsentation vorliegt.

Darüber hinaus ist für eine effiziente Bearbeitung einer Anfrage wichtig, dass es eine geeignete Datenstruktur zur Indexierung der Daten gibt. Damit wird jeder Dateneintrag einmalig entsprechend der vorgegebenen Struktur indexiert, so-

dass dann möglichst viele für die Anfrage irrelevante Daten ohne großen weiteren Rechenaufwand basierend auf diesem Index bereits ausgeschlossen werden können. Entsprechende Datenstrukturen, die einen effizienten inversen online Substrukturabgleich ermöglichen, werden ebenfalls im Folgenden präsentiert.

4.2.2 Verwandte Arbeiten

Wie bereits erwähnt, sind geeignete Datenstrukturen essentiell für eine effiziente Datenanalyse. Dementsprechend findet man in der Literatur eine Vielzahl solcher Indexierungsverfahren. Eine allgemeine Einführung, sowie einen Überblick über das grundsätzliche Vorgehen zur Indexierung von Trajektorien und entsprechende Methoden findet man in [Deng 11].

Zusätzlich gibt es verschiedene Ansätze, die sich auf die Indexierung der netzwerk-basierten Repräsentationen von Trajektorien spezialisiert haben, zum Beispiel [De A 05, Fren 03, Sand 11, Pfos 05].

Während diese Verfahren ausschließlich für allgemeine *Bereichsanfragen*, engl. „range queries“, und *k-nächste-Nachbarn Anfragen*, engl. „k nearest neighbor queries“, geeignet sind, präsentiert Krogh et al. in [Krog 14] einen Index für *exakte Pfadanfragen*, engl. „strict path queries“. Das heißt, dieser Index ist darauf ausgelegt, möglichst effizient Trajektorien zu finden, die einen spezifischen Pfad beinhalten, wodurch er effiziente Substrukturabgleiche ermöglicht. Dabei kommt ein Kodierungsschema für Trajektorien zum Einsatz, sodass für die Beantwortung der exakten Pfadanfrage nur noch die Start- und Endkanten der Trajektorie betrachtet werden müssen.

In [Koid 15] wird ein weiterer Index für exakte Pfadanfragen vorgestellt. Dabei wird der FM-Index, der seinen Ursprung in der Informationsgewinnung, engl. „information retrieval“, und dem Abgleich von Teilzeichenabfolgen, engl. „substring matching“, hat, derart erweitert, dass auch die zeitliche Komponente berücksichtigt wird.

Alle gerade erwähnten Verfahren zur Indexierung von Trajektorien-daten wurden bisher lediglich für Anwendungen des offline Muster- und Substrukturabgleichs evaluiert. Das heißt, es wird davon ausgegangen, dass die Anfrage-trajektorien stets bereits vollständig vorliegen. Im Gegensatz dazu stehen nun die Anwendungen, wie die Anomalieerkennung in SOIS basierend auf Substrukturverzeichnissen, wo dies nicht der Fall ist, die Anfragetrajektorien nur sukzessive verfügbar sind und trotzdem möglichst schnell erste Zwischenergebnisse des Rekonstruktionsprozesses vorliegen müssen.

Dementsprechend ist die Idee des im Folgenden vorgestellten Ansatzes, Datenstrukturen zu verwenden, die es nun auch erlauben, Informationen bereits während des Abgleichprozesses abzugreifen, ohne dass dafür die Anfragetrajektorie vollständig vorliegen muss. Mit Hilfe eines solchen Index kann dann ein inverser online Substrukturabgleich effizient umgesetzt werden.

4.2.3 Inverser online Substrukturabgleich

Für den inversen Substrukturabgleich ist initial ein Verzeichnis an Substrukturen (Mustertrajektorien) gegeben. Die Herausforderung besteht darin, eine geeignete Datenstruktur zu finden, die einen effizienten Zugriff auf diese Substrukturen erlaubt. Dafür werden im Folgenden *Kantenlisten* eingeführt, die eine einfache Identifizierung von Substrukturen erlaubt, die in der Anfragetrajektorie enthalten sind.

4.2.3.1 Kantenlisten

Ein großer Vorteil der netzwerkbasierten Repräsentation von Trajektorien ist, dass jede Kante $e_i \in E$ im Netzwerk mit einer eindeutigen Identifikationsnummer $ID(e_i) = i, i = 0, 1, \dots, n$ beschrieben werden kann (siehe Abbildung 4.8). Dieser Kante e_i kann dann eine Liste $list(ID(e_i))$ zugeordnet werden. In diese Liste werden alle Substrukturen p_j aus dem Verzeichnis P eingetragen, die die Kante e_i enthalten. Dabei wird neben der eindeutigen Identifikationsnummer der Substruktur $ID(p_j) = j, j = 0, 1, \dots, m$, auch der Start bzw. das Ende der Kantennutzung innerhalb der Zeitstruktur der Substruktur in Form von t_{in} bzw. t_{out} angegeben. Die Differenz von t_{in} und t_{out} steht also für die Dauer, die benötigt wird, um die Wegstrecke, die durch Kante e_i repräsentiert wird, zu befahren. Die Einträge der Liste $list(ID(e_i))$ haben damit die Form $(ID(p_j), t_{in}, t_{out})$.

Am Beispiel der gerasterten Netzstruktur aus Abbildung 4.8 soll nun gezeigt werden, wie sich die Kantenliste von Kante e_{46} ergibt. Dafür werden zunächst alle Substrukturen p_j identifiziert, die e_{46} enthalten. In diesem Fall sind es p_4 , p_7 und p_{12} , die entsprechend in Abbildung 4.9 sowohl grafisch in der Netzstruktur, als auch formal in tabellarischer Form dargestellt sind. Da Substruktur p_4 die Kante e_{46} vom Zeitpunkt $t_{in} = 4$ an für zwei Zeiteinheiten ($t_{out} - t_{in} = 6 - 4 = 2$) nutzt (siehe Tabelle 4.9d), ist der erste Eintrag der Kantenliste $(ID(p_j), t_{in}, t_{out}) = (4, 4, 6)$ (siehe Tabelle 4.6). Desweiteren nutzen p_7 die Kante zum Zeitpunkt $t_{in} = 11$ bis $t_{out} = 13$ und p_{12} von $t_{in} = 7$ bis $t_{out} = 9$, was in den weiteren Listeneinträgen $(7, 11, 13)$ und $(12, 7, 9)$ resultiert. Die finale Kantenliste für Kante e_{46} ist in Tabelle 4.6 zu sehen.

Nachdem für alle Kanten der Netzstruktur eine derartige Datenstruktur erstellt wurde, indem über alle Substrukturen im Verzeichnis iteriert wurde, können darauf basierend mögliche Substrukturkandidaten für den inversen Substrukturabgleich identifiziert werden. Das genaue Vorgehen zum Finden dieser Kandidaten wird im Folgenden näher erläutert.

4.2.3.2 Identifikation von Kandidaten

Nachdem der hier vorgestellte Ansatz ein Online-Verfahren ist, das bereits Zwischenergebnisse liefert, ohne dass die Anfragetrajektorie vollständig vorliegt, müssen die Anfragetrajektorien ebenso sukzessive verarbeitet werden, wie neue

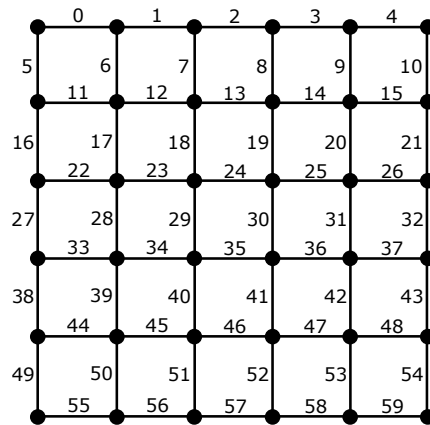
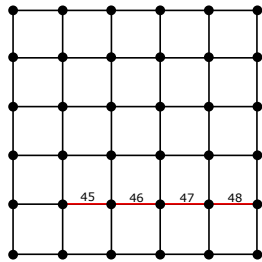


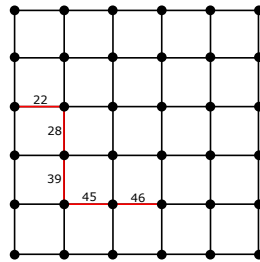
Abbildung 4.8: Jeder Kante ist eine eindeutige Identifikationsnummer $ID(e_i) = i, i = 0, 1, \dots, 59$ zugewiesen.



(a) Substruktur p_4 .

$ID(e)$	t_{in}	t_{out}
45	0	4
46	4	6
47	6	10
48	10	11

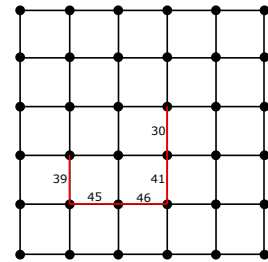
(d) Formale Repräsentation von Substruktur p_4 .



(b) Substruktur p_7 .

$ID(e)$	t_{in}	t_{out}
22	0	2
28	2	5
39	5	7
45	7	11
46	11	13

(e) Formale Repräsentation von Substruktur p_7 .



(c) Substruktur p_{12} .

$ID(e)$	t_{in}	t_{out}
39	0	3
45	3	7
46	7	9
41	9	13
30	13	14

(f) Formale Repräsentation von Substruktur p_{12} .

Abbildung 4.9: Substrukturen, die Kante e_{46} enthalten.

Tabelle 4.6: Kantenliste für Kante e_{46} .

$ID(p)$	t_{in}	t_{out}
4	4	6
7	11	13
12	7	9

Daten verfügbar sind. Dementsprechend wird der inverse Submusterabgleich kantenweise durchgeführt.

Das Schlüsselement bei diesem Vorgehen ist dabei die *Kandidatenliste*, die für jede Anfragetrajektorie vorliegt. Diese enthält Substrukturen, bei denen eine Überschneidung mit dem aktuell vorliegenden Teil der Anfragetrajektorie besteht, und somit noch nicht ausgeschlossen werden kann, dass diese möglicherweise auch in der finalen Anfragetrajektorie enthalten sein werden.

Die Struktur einer solchen Kandidatenliste ist in Abbildung 4.7a zu sehen: Die erste Spalte enthält die eindeutige Identifikationsnummer des Substrukturkandidaten; die zweite Spalte wird als Zähler, engl. „counter“, für die Anzahl der Kanten verwendet, in denen die Substruktur und die Anfragetrajektorie übereinstimmen. Diese Kandidatenliste wird jedes mal, wenn neue Daten der Anfragetrajektorie vorliegen, entsprechend aktualisiert. Das genaue Vorgehen für diese Kanten-weise Aktualisierung ist in Algorithmus 1 im Detail aufgeschlüsselt.

Grundsätzlich unterteilt sich der Aktualisierungsprozess in zwei Schritte: zunächst wird geprüft, ob Substrukturen aus der Kandidatenliste entfernt werden müssen (Zeilen 5-16). Dies sind entweder Kandidaten, die durch die neu hinzugefügte Kante nun vollständig in der aktuellen Anfragetrajektorie enthalten sind und damit stattdessen in die *Liste übereinstimmender Substrukturen* eingetragen werden können. Alternativ, können Substrukturen auch aus der Kandidatenliste entfernt werden, wenn sie einfach auf Grund der neuen Kante nicht mehr mit der aktuellen Anfragetrajektorie übereinstimmen.

Im zweiten Schritt werden neue Kandidaten identifiziert, indem die Kantenliste der neu hinzugefügten Kante nach Substrukturen durchsucht wird, die diese Kante als Startkante haben (Zeilen 17-21). Für den Vergleich in Zeile 7 werden sowohl die Abfolge der Kanten, als auch die durch t_{in} und t_{out} gegebenen zeitlichen Abläufe abgeglichen.

Zur Veranschaulichung sind in Tabelle 4.7 die Kandidatenliste und Liste übereinstimmender Substrukturen zu sehen, die sich für die Beispieltrajektorie aus Abbildung 4.10a ergeben. Der Zeitpunkt der Betrachtung ist $t = 11$. Das heißt, zu diesem Zeitpunkt sind nur die ersten vier Kanten der Anfragetrajektorie bekannt ($e_{33}, e_{39}, e_{45}, e_{46}$). Nachdem Substruktur p_1 (siehe Abbildung 4.10b und Tabelle 4.10d) vollständig in der aktuellen Anfragetrajektorie enthalten ist, wird deren eindeutige Identifikationsnummer der Liste übereinstimmender Substrukturen hinzugefügt (siehe Tabelle 4.7b). Die Substrukturen p_4 und p_{12} sind derweil als Kandidaten erfasst (siehe Tabelle 4.7a), da die ersten beiden Kanten von p_4 bzw. die ersten drei Kanten von p_{12} mit Teilen der aktuellen Anfragetrajektorie übereinstimmen. Dementsprechend besteht die aktuelle Kandidatenliste aus den beiden Einträgen (4,2) und (12,3). Andere Substrukturen, wie zum Beispiel p_7 (siehe Abbildung 4.9b und Tabelle 4.9e) werden nicht berücksichtigt, da sie zum aktuellen Zeitpunkt nicht relevant sind für die Reproduktion der aktuellen Anfragetrajektorie.

Das heißt, durch das Vorhalten derartiger Kandidatenlisten wird der Suchraum

Algorithmus 1: Inverser online Substrukturabgleich

```

input : Substrukturverzeichnis, Kantenlisten
output: Übereinstimmende Substrukturen
1  $C \leftarrow \emptyset$  // Kandidatenliste
2  $M \leftarrow \emptyset$  // Liste übereinstimmender Substrukturen
3 foreach Eingehende Kante e do
    // hole entsprechende Kantenliste
4    $l \leftarrow list(ID(e))$ 
    // aktualisiere Kandidatenliste:
    // 1. entferne Kandidaten
5   foreach  $c \in C$  do
        // hole nächste Kante der Substruktur c
6        $e_{next} \leftarrow next\_edge(c.ID, c.matchedEdges)$ 
7       if  $e == e_{next}$  then
            // überprüfe, ob die Substruktur nun vollständig
            // enthalten ist
8           if  $e_{next}.ID == lastElement(c.ID)$  then
                // entferne c von der Kandidatenliste und füge
                // sie zur Liste übereinstimmender Substrukturen
                // hinzu
9                $C \leftarrow C.remove(c)$ 
10               $M \leftarrow M.add(c)$ 
11           else
                // erhöhe den Zähler für übereinstimmende Kanten
12               $c.matchedEdges \leftarrow c.matchedEdges + 1$ 
13           end
14       else
            // entferne c von der Kandidatenliste
15               $C \leftarrow C.remove(c)$ 
16       end
        // 2. überprüfe Kantenliste bezüglich neuer
        // Kandidaten
        // add all patterns which starts at edge e
17       foreach  $p \in l$  do
18           if  $p.t_{in} == 0$  then
19                $C \leftarrow C.add(p)$ 
20           end
21       end
22   end
23 end

```

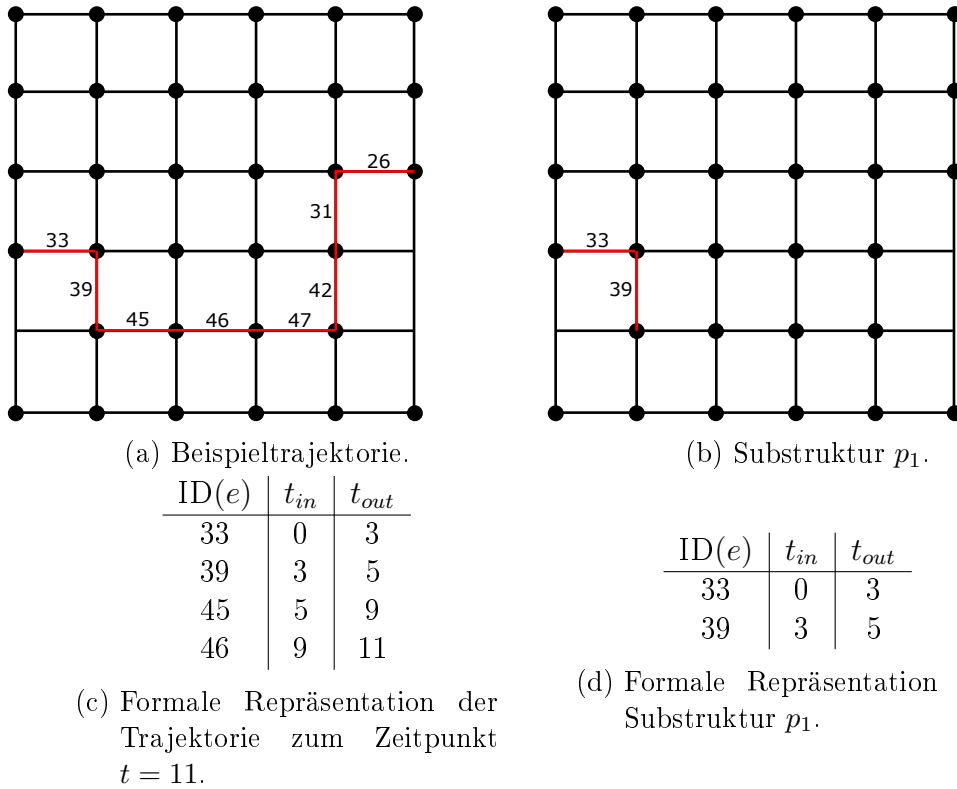


Abbildung 4.10: Beispieltrajektorie und Substruktur ID= 1.

ID(p)	# übereinstimmender Kanten	ID(p)
4	2	1
12	3	

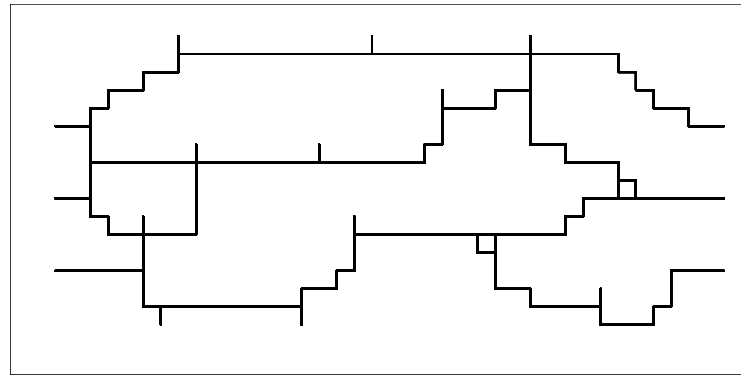
(a) Kandidatenliste. (b) Liste übereinstimmender Substrukturen.

Tabelle 4.7: Listen für Beispieltrajektorie zum Zeitpunkt $t = 11$.

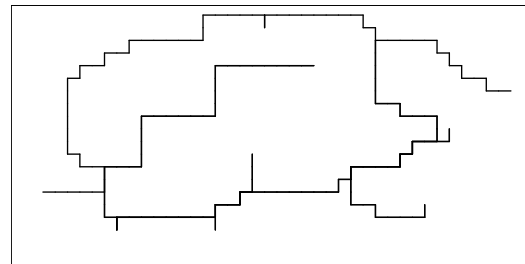
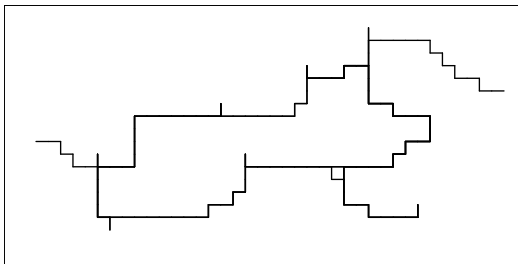
für die Rekonstruktion auf Substrukturen reduziert, bei denen zumindest eine anfängliche Teilüberdeckung vorliegt. Damit sinkt die Anzahl an vollständig durchzuführenden inversen Substrukturabgleichen, da bereits nach dem Vergleich der ersten Kanten entschieden werden kann, ob eine Übereinstimmung grundsätzlich möglich ist. Insgesamt ergibt sich damit die Möglichkeit, effizient aus einer Menge von Substrukturen die Elemente herauszufiltern, die in den Anfragetrajektorien enthalten sind.

4.2.4 Evaluation

Die Funktionsweise des hier vorgestellten Algorithmus zum inversen online Substrukturabgleich wird im Folgenden am Beispiel der Erkennung und Analyse von Anomalien in SOIS mittels Substrukturverzeichnissen (siehe Abschnitt 3.1) evaluiert. Dazu wird die Rasterstruktur aus dem Simulationsmodell (siehe



(a) Kanten der Substrukturen des Simulationsmodells.



(b) Beispiel einer normalen Trajektorie. (c) Beispiel einer anormalen Trajektorie.

Abbildung 4.11: Evaluationsszenario.

Abbildung 3.2) wie bereits in Abschnitt 4.2.1 erläutert in eine Netzwerkstruktur überführt. Zur Veranschaulichung sind in Abbildung 4.11a alle Kanten, die in mindestens einer Substruktur vorkommen, in den Raumplan des SOIS eingezeichnet.

Wie bei der rasterbasierten Darstellung enthält eine solche Darstellung lediglich die räumlichen Aufenthaltsmöglichkeiten der Werkteile, aber darüber hinaus keine Informationen über die anwendungsspezifischen Nutzungseigenarten, wie Richtungsinformationen bzw. konkrete Verbindungsabfolgen. Diese Informationen sind in den Substrukturen enthalten und ermöglichen damit eine Anomalieerkennung in SOIS (siehe Abschnitt 3.1).

Weiterhin starten die mobilen Werkteile auf der linken Seite in der Fabrik (drei mögliche Startpunkte), suchen sich dann über die Fließbänder ihren Weg durch die Fabrik, wobei sie immer wieder verschiedene Produktionsstationen anfahren, bis das fertige Produkt die Fabrik auf der rechten Seite verlässt (ebenfalls drei mögliche Endpunkte) (siehe Abbildung 4.11a).

In Abbildung 4.11b ist entsprechend beispielhaft eine normale Testtrajektorie zu sehen, die aus den bekannten Substrukturen rekonstruiert werden kann. Im Gegensatz dazu ist in Abbildung 4.11c die Route eines Werkteiles visualisiert, die in diesem Fall sogar sehr offensichtlich auch Teilstücke enthält, die nicht rekonstruiert werden können (oberer und mittlerer Pfad).

Im Folgenden wird anhand der normalen Testtrajektorie zunächst demonstriert, wie der hier vorgestellte Ansatz zum inversen online Substrukturabgleich

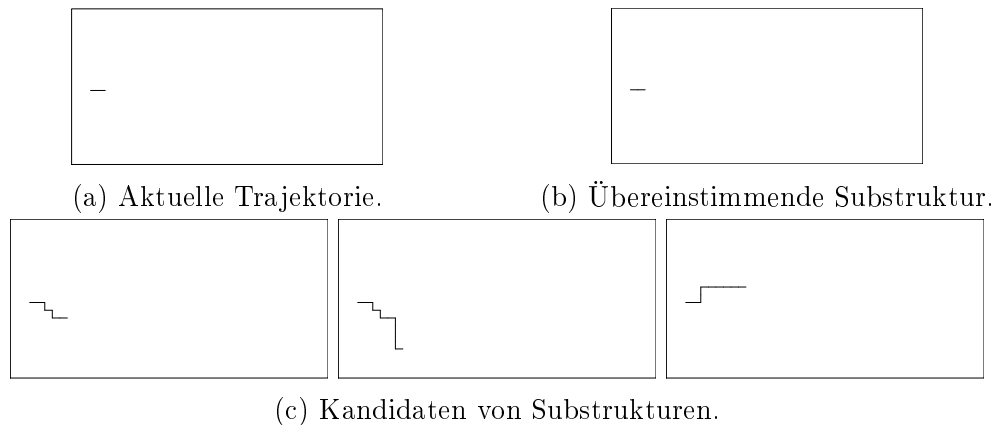


Abbildung 4.12: Schritt 1.

zur Online Rekonstruktion von Trajektorien einer SOIS verwendet werden kann. Anschließend wird noch darauf eingegangen, wie Zwischenergebnisse des inversen Substrukturabgleichs bereits zur Laufzeit abgegriffen werden können, was insbesondere für die Laufzeitüberwachung in SOIS von enormer Bedeutung ist.

4.2.4.1 Online Rekonstruktion

Für die Rekonstruktion einer Testtrajektorie werden über inverse Substrukturabgleiche diejenigen Substrukturen aus dem Verzeichnis identifiziert, die in ebenjener Anfragetrajektorie enthalten sind. Da dies für die Anomalieerkennung in SOIS zur Laufzeit erfolgen muss, ohne dass die Anfragetrajektorie vollständig vorliegt, wird der inverse Substrukturabgleich sukzessive für jede neu eintreffende Kante durchgeführt (siehe Abschnitt 4.2.3).

Dementsprechend sind in den Abbildungen 4.12 und 4.13 die Kandidatensubstrukturen und die übereinstimmenden Substrukturen nach der Auswertung der ersten bzw. zweiten Kante der normalen Testtrajektorie aus Abbildung 4.11b visualisiert.

Wie in Abbildung 4.12 zu sehen ist, gibt es bereits die erste Substruktur, die vollständig in der aktuellen Anfragetrajektorie enthalten ist (siehe Abbildung 4.12b) und daher entsprechend in der Liste übereinstimmender Substrukturen eingetragen wird. Daneben gibt es insgesamt drei Substrukturen, die in der ersten Kante mit der Anfragetrajektorie übereinstimmen (siehe Abbildung 4.12c) und damit die aktuellen Einträge der Kandidatenliste ergeben.

Abbildung 4.13a zeigt die aktuelle Anfragetrajektorie, nachdem die nächste Kante für den inversen online Abgleich zur Verfügung steht. Nachdem die ersten beiden Substrukturen der Kandidatenliste auch in der zweiten Kante mit der neu hinzugefügten Kante der Anfragetrajektorie übereinstimmen (siehe Abbildung 4.13c), bleiben die entsprechenden Einträge in der Kandidatenliste bestehen. Im Gegensatz dazu wird der dritte Eintrag der Kandidatenliste gelöscht, da die entsprechende Substruktur in der zweiten Kante nicht mehr mit

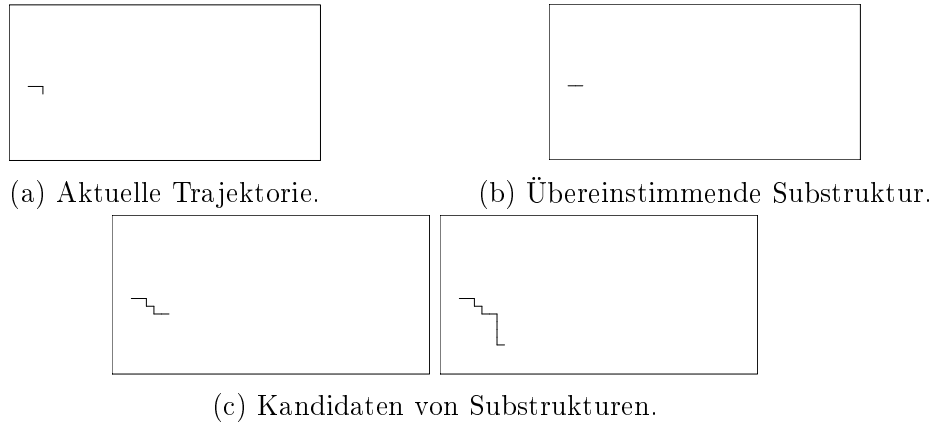


Abbildung 4.13: Schritt 2.

der Anfragetrajektorie übereinstimmt. Damit ergeben sich keine Änderungen für die Liste übereinstimmender Substrukturen (siehe Abbildung 4.13b).

Dieser Prozess wird kontinuierlich für jede neu eintreffende Kante fortgeführt, wodurch die Testtrajektorie Schritt für Schritt zur Laufzeit aus den gegebenen Substrukturen rekonstruiert wird.

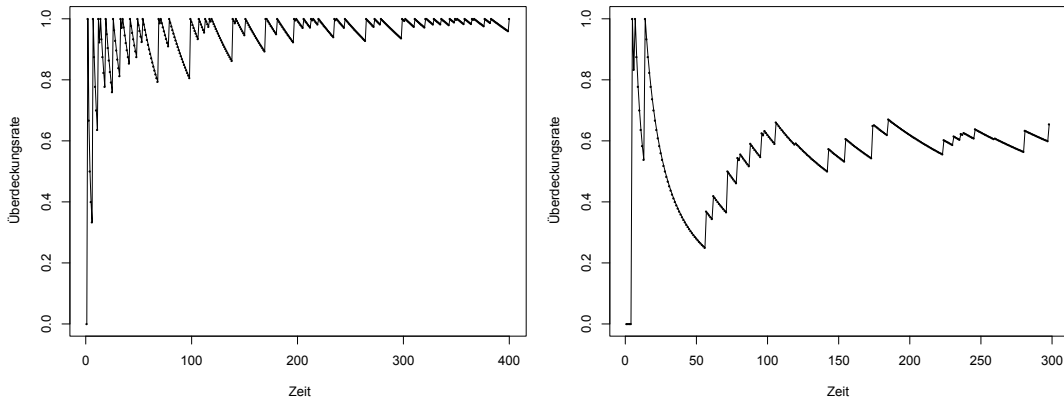
An diesem Beispiel wird deutlich, dass der hier vorgestellte Ansatz für inversen online Substrukturabgleich zur online Rekonstruktion von Trajektorien in einem SOIS eingesetzt werden kann. Damit ist er ein wichtiger Baustein bei der technischen Umsetzung des Anomalieerkennungsansatzes aus Abschnitt 3.1.

4.2.4.2 Überdeckungsrate

Wie eingangs erwähnt, ist es im Fall des inversen online Substrukturabgleichs notwendig, dass bereits zur Laufzeit, wenn die Anfragetrajektorie noch nicht vollständig vorliegt, Informationen über die Rekonstruierbarkeit und damit die (Ab-)Normalität der Testtrajektorie verfügbar sind. Dafür wird für jede Anfragetrajektorie deren Überdeckungsrate über die Zeit beobachtet, sodass sich unmittelbar Tendenzen erkennen lassen, wie gut die Trajektorie rekonstruierbar ist.

Die Überdeckungsrate ist in diesem Fall das Verhältnis von Kanten, die entsprechend der Rekonstruktion durch Substrukturen abgedeckt sind, und solchen, die dabei von keiner Substruktur überdeckt werden. Dementsprechend wird während des Rekonstruktionsprozesses für jeden Zeitpunkt t die Überdeckungsrate $c(t)$ bestimmt, indem die Anzahl an Kanten, die durch Substrukturen aus der Liste übereinstimmender Substrukturen $M(t)$ überdeckt werden, durch die Anzahl der aktuell verfügbaren Kanten der Anfragetrajektorie $T(t)$ geteilt wird:

$$c(t) = \frac{|\{e | e \in p_i, \forall p_i \in M(t)\}|}{|\{e | e \in T(t)\}|}, \quad (4.1)$$



(a) Zeitlicher Verlauf der Überdeckungsrate einer normalen Trajektorie. (b) Zeitlicher Verlauf der Überdeckungsrate einer anormalen Trajektorie.

Abbildung 4.14: Überdeckungsrate.

Zur Illustration sind in den Abbildungen 4.14a und 4.14b die zeitlichen Verläufe von Überdeckungsrate von zwei Beispieltrajektorien zu sehen. In beiden Fällen, ergibt sich für den Graph ein Sägezahnmuster. Der Grund dafür liegt darin, dass lediglich Substrukturen aus der Liste übereinstimmender Substrukturen für die Berechnung der Überdeckungsrate berücksichtigt werden. Das heißt, so lange die zuletzt hinzugefügten Kanten der Anfragetrajektorie lediglich durch Kanten von Kandidatensubstrukturen abgedeckt werden, sinkt die Überdeckungsrate ab, bis einer der Kandidaten vollständig übereinstimmt. Dann schnell die Überdeckungsrate entsprechend der Anzahl an nun auch überdeckten Kanten der Anfragetrajektorie nach oben.

Betrachtet man die Überdeckungsrate der normalen Testtrajektorie in Abbildung 4.14a findet dieses Auf und Ab auf einem Level von 0.8 und 1 statt. Im Gegensatz dazu brechen die Überdeckungsrate bei der anormalen Testtrajektorie zum Zeitpunkt $t = 25$ ein und erreichen danach auch nur noch einen maximalen Wert von 0.65 (siehe Abbildung 4.14b). Das heißt, in diesem Fall müsste man nicht die finale Überdeckungsrate abwarten, um zu sehen, dass hier Teile der Anfragetrajektorie nicht rekonstruiert werden können. Stattdessen kann man bereits bei dem starken Abfall nach $t = 25$ einschreiten und die Anormalität der Trajektorie analysieren.

Insgesamt konnte also in diesem Abschnitt gezeigt werden, dass durch das Beobachten der Überdeckungsrate über die Zeit bereits zur Laufzeit Rückschlüsse über die Rekonstruierbarkeit und damit (Ab-)Normalität einer Anfragetrajektorie möglich sind. Dies ist insbesondere im Falle von Anomalieerkennung in SOIS von zentraler Bedeutung, um frühzeitig Fehlverhalten des Systems identifizieren zu können und so weiteren Schaden zu vermeiden.

4.2.5 Zusammenfassung

In Abschnitt 4.2.1 wurde die Onlinefähigkeit des Rekonstruktionsprozesses bei der Erkennung und Analyse von Anomalien mittels Substrukturverzeichnis in raumbezogenen Daten von SOIS thematisiert. Dazu wurden zunächst die bestehenden Anwendungsfälle und Lösungsansätze zum Musterabgleich in Trajektorien Daten diskutiert.

Gleichzeitig wurden die speziellen Anforderungen beim Rekonstruktionsprozess im Rahmen der Anomalieerkennung in SOIS, wie Echtzeitfähigkeit und Verfügbarkeit von Teilergebnisse, herausgearbeitet, die allerdings in dieser Form nicht von bestehenden Verfahren berücksichtigt werden (siehe Abschnitt 4.2.2).

Entsprechend wurde in Abschnitt 4.2.3 ein Verfahren zum inversen online Substrukturabgleich vorgestellt, das einen ebensolchen effizienten Rekonstruktionsprozess zur Laufzeit ermöglicht. Dazu werden zunächst Kantenlisten als Indexstrukturen zur Identifizierung von Substrukturkandidaten eingeführt, die für die Rekonstruktion in Frage kommen (siehe Abschnitt 4.2.3.1). Die Anfragetrajektorien können dann zur Laufzeit verarbeitet werden, ohne dass diese vollständig vorliegen. Dafür wird eine Kandidatenliste angelegt und sukzessive aktualisiert, indem mit Hilfe der Kantenlisten mögliche Substrukturen für die Rekonstruktion identifiziert werden (siehe Abschnitt 4.2.3.2). Zeitgleich werden vollständig passende Kandidaten von der Kandidatenliste in eine Liste übereinstimmender Substrukturen übertragen, wodurch bereits zur Laufzeit Zwischenergebnisse bezüglich der Überdeckung von Anfragetrajektorie durch verfügbare Substrukturen verfügbar sind.

Zur Evaluation dieses Vorgehens wurde in Abschnitt 4.2.4.1 die Rekonstruktion von Testtrajektorien im Rahmen der Anomalieerkennung in SOIS entsprechend des vorgestellten Verfahrens ausgeführt. In diesem Zusammenhang konnte auch gezeigt werden, dass durch die Auswertung der zur Verfügung stehenden Zwischenergebnisse bereits zur Laufzeit Rückschlüsse über die (Ab-)Normalität einer Trajektorie möglich sind (siehe Abschnitt 4.2.4.2), was für die Qualitätssicherung in einem SOIS von zentraler Bedeutung ist, um frühzeitig Änderungen zu detektieren und entsprechende (Gegen-)Maßnahmen einleiten zu können. Damit konnte in diesem Abschnitt gezeigt werden, dass durch den vorgestellten inversen online Substrukturabgleich ein effizienter Rekonstruktionsprozess im Rahmen der Anomalieerkennung in SOIS zur Laufzeit möglich ist und gleichzeitig wertvolle Teilergebnisse liefert.

4.3 Zusammenfassung

In diesem Kapitel wurden die Themen Skalierbarkeit der Substrukturextraktion und Onlinefähigkeit des Rekonstruktionsprozesses bezüglich der Umsetzung des in Kapitel 3 vorgestellten Konzepts zur Erkennung und Analyse von Anomalien in raumbezogenen Daten von SOIS diskutiert.

In Abschnitt 4.1 wurde dazu zunächst ein alternatives Verfahren zur Substrukturextraktion mittels Pfächen-Lernens vorgestellt, das insbesondere im Falle von Trainingstrajektorien mit vielen Koordinatenpunkten deutlich besser skaliert. Es konnte gezeigt werden, dass statt eines quadratischen Anstiegs der Rechenkomplexität mit der Anzahl an Samplepunkten pro Trajektorie, unter Verwendung des Grammatikinduktionsverfahrens SEQUITUR die Komplexität linear steigt. Gleichzeitig muss lediglich bei der *Effizienz* der Rekonstruktion mit leichten Einbußen gegenüber den Ergebnissen des Pathlet Learning-basierten Ansatzes gerechnet werden, da dieser im Durchschnitt längere Substrukturen extrahiert. Bezüglich aller anderen Qualitätskriterien, wie *Universalität* der extrahierten Substrukturen und *Kompaktheit* des Verzeichnisses, liefern beide Verfahren vergleichbare Ergebnisse.

Damit ist das in Abschnitt 4.1 vorgestellte Verfahren durchaus eine gute Alternative, wenn es um eine skalierbare Umsetzung des Substrukturextraktionsprozesses gemäß des Konzeptes aus Kapitel 3 geht. Dass die mittels SEQUITUR erzeugten Substrukturverzeichnisse ebenfalls erfolgreich zur Erkennung und Analyse von Änderungen in SOIS eingesetzt werden können, wurde in Abschnitt 4.1.3 an den Beispielen anormaler Einzeltrajektorien und globaler anormaler Ereignisse gezeigt.

Für eine effiziente Umsetzung des Rekonstruktionsprozesses im Rahmen der Anomalieerkennung gemäß des Konzeptes aus Kapitel 3 zur Laufzeit wurde zudem in Abschnitt 4.2 ein Verfahren zum inversen online Substrukturabgleich vorgestellt. Dieses ermöglicht durch Anwendung einer geeigneten Indexstruktur auf die Verzeichniseinträge eine effiziente Identifizierung von Substrukturkandidaten, die für die Rekonstruktion einer Testtrajektorie in Frage kommen. Zusätzlich können mit diesem Verfahren bereits zur Laufzeit wertvolle Informationen über den Grad der Rekonstruierbarkeit in Form von Überdeckungsratewerten abgefragt werden. Die Umsetzung dieses Ansatzes für die Erkennung und Analyse von Anomalien in raumbezogenen Daten von SOIS zeigt eindeutig, dass der inverse online Substrukturabgleich die erforderliche Onlinefähigkeit des Rekonstruktionsprozesses erfolgreich umsetzt.

Damit wurden in diesem Kapitel zwei wichtige Modullösungen erarbeitet, die eine effiziente Umsetzung des in Kapitel 3 vorgestellten Konzeptes zur Erkennung und Analyse von Anomalien in raumbezogenen Daten von SOIS ermöglichen.

5 Erweiterungen zur Laufzeitüberwachung komplexerer Datenstrukturen

In diesem Kapitel wird das Konzept zur Erkennung und Analyse von Anomalien in SOIS mit Hilfe von Substrukturverzeichnissen aus Kapitel 3 auch für komplexere raumbezogene Bewegungsdaten umgesetzt. Im Speziellen werden zwei Verfahren vorgestellt: das eine für Bewegungsdaten von Objekten mit variabler Geschwindigkeit und das zweite für Zustandsbeschreibungsdaten des Gesamtsystems. In beiden Fällen handelt es sich um raumbezogene Daten, die zur Qualitätssicherung im SOIS zur Laufzeit überwacht werden müssen.

Bewegungsdaten von Objekten mit variabler Geschwindigkeit entstehen, wenn im SOIS mobile Roboter eingesetzt werden, um Werkteile von Maschine zu Maschine zu transportieren. Im Gegensatz zu den mobilen Werkteilen aus Kapitel 3, die sich auf Fließbändern bewegen, können sich die mobilen Roboter frei im Raum bewegen und auch ihre Geschwindigkeit selbst steuern. Damit ergibt sich ein deutlich größerer Zustandsraum. In Abschnitt 5.1 wird daher eine entsprechende Umsetzung des Konzeptes aus Abschnitt 3.1 vorgestellt, die trotz des größeren Zustandsraumes eine zuverlässige Erkennung und Analyse von Anomalien, basierend auf den Bewegungsdaten solcher mobiler Roboter, im Rahmen der Laufzeitüberwachung ermöglicht.

Die Zustandsbeschreibungsdaten, als ein weiterer komplexer raumbezogener Datentyp, ergeben sich, wenn zur Erstellung des kognitiven Systems *bestärkendes Lernen* verwendet wird. Dadurch, dass dafür die räumlichen Lagen verschiedener Merkmale zusammengefasst werden, ergibt sich erneut ein enorm großer Zustandsraum. Dieses Problem kann ebenfalls durch eine geeignete Umsetzung des Konzepts aus Abschnitt 3.1 gelöst werden, sodass anormale Zustandsbeschreibungen zur Laufzeit detektiert werden können (siehe Abschnitt 5.2).

Die Ergebnisse für die Anomalieerkennung in Bewegungsdaten von Objekten mit variabler Geschwindigkeit wurden bereits in [Kier 18c] vorgestellt. Die Ergebnisse für die Anomalieerkennung in Zustandsbeschreibungsdaten wurden in [Kier 18b] veröffentlicht.

Wie in Abschnitt 1.3 beschrieben, sind in beiden Fällen Idee, Umsetzung und Evaluation der vorgestellten Ansätze vom Autor der vorliegenden Arbeit, wobei für die Evaluation in [Kier 18c] Daten aus einer entsprechenden Simulation von Thomy Phan verwendet wurden.

5.1 Laufzeitüberwachung von Objekten mit variabler Geschwindigkeit

Die Herausforderung bei der Umsetzung des Konzepts aus Abschnitt 3.1 für Bewegungsdaten von Objekten mit variabler Geschwindigkeit ist, dass neben der räumlichen Bewegung nun auch der zeitliche Ablauf überwacht werden muss. Im Falle von mobilen Robotern in einem SOIS bedeutet das, dass nicht nur deren Routen, sondern auch ihre Geschwindigkeiten auf Konformität mit dem Normalitätsmodell getestet werden müssen.

Entsprechend wird das bisherige Normalitätsmodell (Substrukturverzeichnis) um sogenannte *Zeitdauermodelle* erweitert (siehe Abschnitt 5.1.2). Diese geben für jede Substruktur die Wahrscheinlichkeitsverteilung der Zeitspannen an, die in der Trainingsphase von einem mobilen Roboter für die Nutzung, beziehungsweise das Abfahren der Substruktur benötigt wurden.

Damit kann die (Ab-)Normalität bezüglich der räumlichen Bewegung wie gewohnt über die Rekonstruierbarkeit der Trajektorie durch gegebene Substrukturen bestimmt werden. Gleichzeitig kann die (Ab-)Normalität im Bezug auf den zeitlichen Ablauf bewertet werden, indem für jede der zur Rekonstruktion verwendeten Substrukturen die dafür von der Testtrajektorie benötigte Zeitspanne mit dem entsprechenden Zeitdauermodell verglichen wird.

Wie in Abschnitt 3.6 können über die dabei anfallenden Informationen zudem verschiedene Arten von Anomalien erkannt und analysiert werden. Insgesamt werden hier erneut lokale und globale Änderungen unterschieden (vgl. Einzeltrajektorien und globale Ereignisse in Abschnitt 3.6.1 bzw. 3.6.2). Allerdings können diese Anomalien nun sowohl in Bezug auf die räumliche Bewegung (siehe Abschnitt 5.1.3.2), als auch bezüglich des zeitlichen Ablaufs (siehe Abschnitt 5.1.3.3) auftreten.

5.1.1 Simulationsmodell

Für die Simulation eines SOIS mit mobilen Robotern, die die Werkteile von einer Maschine zur nächsten transportieren und dabei sowohl ihre Route, als auch ihre Geschwindigkeit selbst bestimmen, wurde eine entsprechende Simulation von Thomy Phan verwendet. Diese kombiniert bestärkendes Lernen und simulationsbasiertes Online Planen und Entscheiden (siehe Abschnitt 2.1.2), um emergentes Verhalten nachbilden zu können [Phan 18].

Durch das bestärkende Lernen wird eine Nutzenfunktion, engl. „value function“, gelernt, die angibt, welchen Nutzen ein gegebener Systemzustand auf lange Sicht hat. Beim Online Planen und Entscheiden geht es darum, zu jedem Zeitpunkt die aktuell beste (gemeinsame) Aktion für die mobilen Roboter zu finden, indem mögliche Aktionen gesampelt, simuliert und dann gemäß der gelernten Nutzenfunktion bewertet werden. Die derart bestbewertete Aktion wird schließlich ausgewählt und ausgeführt.

Der Raumplan des simulierten SOIS ist in Abbildung 5.1a zu sehen. Wie in

Kapitel 3 ist die Grundfläche in gleichgroße Rasterzellen unterteilt, um Objektpositionen in Form von x - und y -Koordinaten oder eindeutigen Identifikationsnummern der Rasterzellen darstellen zu können.

In diesem Fall starten die Roboter mit dem Werkteilrohling an den gelb markierten Zellen und verlassen die Fabrik mit dem fertigen Produkt an den orange markierten Zellen. Dazwischen sind verschiedene Maschinen platziert, die von den mobilen Robotern angefahren werden können. In Abbildung 5.1a sind entsprechende Zellen grün markiert, und der Buchstabe steht für den jeweiligen Maschinentyp.

Jedes Werkteil hat, wenn es in das SOIS kommt, eine Aufgabenliste, die aus Maschinentypen besteht und entsprechend abgearbeitet werden muss, indem passende Maschinen angefahren werden. Die Bearbeitungszeit an jeder Maschine beträgt dabei zehn Zeitschritte. Das heißt, ein mobiler Roboter verweilt zehn Zeitschritte an der Maschine, bevor er mit der Abarbeitung der verbleibenden Aufgabenliste fortfahren kann beziehungsweise bis ein anderes Werkteil an der Maschine bearbeitet werden kann.

Dabei ist zu beachten, dass im Gegensatz zur Fließbandarchitektur aus den vorherigen Kapiteln in diesem Simulationsmodell die Maschinen räumlich nicht von der durch die mobilen Roboter befahrbaren Fläche separiert sind. Das heißt, ein mobiler Roboter kann räumlich derselben Rasterzelle zugeordnet sein wie eine Maschine entweder, weil gerade sein Werkteil entsprechend von der Maschine bearbeitet wird, oder aber, weil er auf dem Weg zu einer anderen Maschine an dieser Maschine vorbeifährt.

5.1.2 Erweiterung des Substrukturverzeichnis

Wie eingangs beschrieben, muss zur Erkennung und Analyse von Anomalien in Bewegungsdaten von Objekten mit variabler Geschwindigkeit neben der räumlichen Bewegung auch der zeitliche Ablauf der Objekte überwacht werden. Dazu wird das Substrukturverzeichnis um eine zeitliche Komponente, sogenannte *Zeitdauermodelle*, erweitert.

Dabei handelt es sich um Wahrscheinlichkeitsverteilungen über die Zeitspannen, die in der Trainingsphase für das Abfahren der durch die jeweilige Substruktur definierten Pfade beobachtet wurden. Diese Wahrscheinlichkeitsverteilungen werden für jede Substruktur $p = (x_1, y_1, t_1), \dots, (x_n, y_n, t_n)$ berechnet, wobei für jede Trainingstrajektorie $t_i \in T(p)$, die p enthält, eine Zeitspannenbeobachtung $d_i = t_{n+1} - t_1$ in die Dichteschätzung eingeht.

In Abbildung 5.1c ist das sich so ergebende Zeitdauermodell für die Beispielsubstruktur aus Abbildung 5.1b zu sehen. Offensichtlich gibt es bei einer Zeitdauer rund um den Wert 3 eine signifikante Spitze und einen weiteren, wenn auch geringeren Ausschlag nach oben etwa um den Zeitdauerwert 20. Größere Zeitspannenwerte sind zwar dann auch noch möglich, aber haben deutlich geringere Wahrscheinlichkeitswerte.

Dieses Modell spiegelt damit wider, dass die untersuchte Substruktur einen

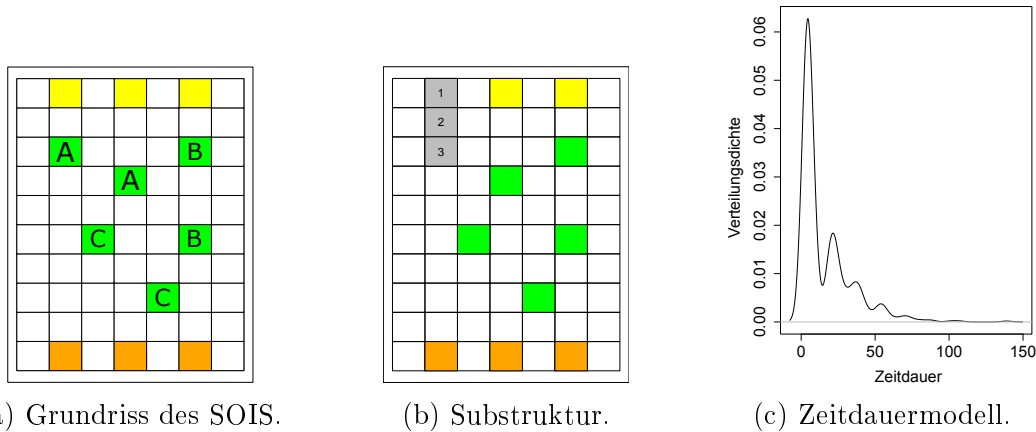


Abbildung 5.1: Raumplan des SOIS, sowie eine Beispielsubstruktur mit dazugehörigem Zeitdauermodell.

Maschinenstandort beinhaltet (siehe Abbildung 5.1b). Entsprechend variieren die Zeitspannen in den Trainingstrajektorien abhängig davon, ob der mobile Roboter an der Maschine vorbeifährt (kurze Zeitspannen, hier: die Spitze rund um den Wert 3), oder ob das jeweilige Werkstück von der Maschine bearbeitet werden soll (größere Zeitspannen wegen der Bearbeitungszeit von 10 Zeitschritten) und der mobile Roboter sich dafür möglicherweise sogar zuvor noch in eine Warteschlange einreihen muss (noch größere Zeitspannen, hier: zum Beispiel der Ausschlag beim Zeitdauerwert von 20).

Das heißt, mit einer derartigen Erweiterung des Substrukturverzeichnisses erhält man für jede Substruktur ein Normalitätsmodell, das die Bewertung der Geschwindigkeit eines Objekts im Bezug auf diese Substruktur und den dadurch definierten Pfad, ermöglicht. Durch den Abgleich der eingehenden Beobachtungen aus einem SOIS mit solchen Zeitdauermodellen können nun auch Anomalien im zeitlichen Ablauf erkannt und analysiert werden. Das genaue Vorgehen dabei wird in den folgenden Abschnitten näher erläutert.

5.1.3 Erkennung und Analyse von Anomalien

Im Folgenden wird anhand der SOIS-Simulation mit mobilen Robotern gezeigt, dass mit der in Abschnitt 5.1.2 erläuterten Erweiterung das Konzept aus Abschnitt 3.1 auch für Bewegungsdaten von Objekten mit variabler Geschwindigkeit umgesetzt werden kann. Dazu wird zunächst die geeignete Parametrisierung zur Erzeugung eines Substrukturverzeichnisses experimentell ermittelt (siehe Abschnitt 5.1.3.1), um anschließend anhand verschiedener Szenarien das gesamte Verfahren zu evaluieren. Diese Szenarien umfassen sowohl Situationen im SOIS, in denen Anomalien bezüglich der räumlichen Bewegung (siehe Abschnitt 5.1.3.2) als auch des zeitlichen Ablaufs (siehe Abschnitt 5.1.3.3) auftreten und entsprechend erkannt und analysiert werden müssen.

5.1.3.1 Substrukturverzeichnis

Bei der Erstellung des Substrukturverzeichnisses geht es zunächst wieder darum, die beiden Parameter – Anzahl an Trainingsdaten und λ – dem Szenario entsprechend zu bestimmen (siehe Abschnitt 3.5). Dafür werden, wie bereits in Abschnitt 3.5, die durchschnittlichen Überdeckungsraten als Bewertungskriterium herangezogen, die sich in diesem Fall bei der Rekonstruktion von 500 Testtrajektorien ergeben. Die Überdeckungsraten müssen ausreichend groß sein (>0.99), um sicherzustellen, dass normale Testtrajektorien nicht fälschlicherweise als anormal klassifiziert werden.

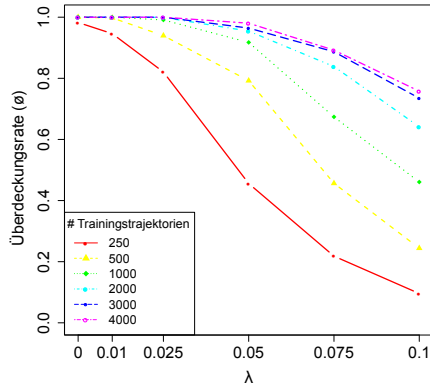
Aus den Ergebnissen aus Abbildung 5.2a ergibt sich, dass für das vorliegende Szenario $0 \leq \lambda \leq 0.025$ gesetzt werden muss und mindestens 1000 Trainings-trajektorien benötigt werden, um ein ausreichend *universelles* Substrukturverzeichnis zu erzeugen. Außerdem lässt sich erneut beobachten, dass gemäß Gleichung 3.1 die Größe des Substrukturverzeichnisses mit steigendem λ zunimmt (siehe Abbildung 5.2b), wohingegen die durchschnittliche Anzahl an Substrukturen, die für die Rekonstruktion einer Testtrajektorie benötigt werden, sinkt (siehe Abbildung 5.2c).

Die auf den ersten Blick dieser Aussage widersprechenden kleine Werte für $\lambda = 0$ in Abbildung 5.2c lassen sich durch die durchschnittlichen Längen der zur Rekonstruktion verwendeten Substrukturen erklären (siehe Abbildung 5.2d). Mit steigendem λ werden längere Substrukturen zur Rekonstruktion der Testtrajektorien verwendet. Das heißt, auch wenn sich im Bezug auf die Rekonstruktion mit $\lambda = 0$ und $\lambda = 0.025$ ähnlich effiziente Substrukturverzeichnisse erzeugen lassen (siehe Abbildung 5.2c), gilt, dass längere Substrukturen mehr Informationen für eine weitere Analyse bereitstellen. Insbesondere im Hinblick auf rückwärtige Fehlersuche, engl. „back tracing“, und dem Erkennen von systemspezifischen Prozessabfolgen sind daher längere Substrukturen bei ähnlicher Rekonstruktionseffizienz zu bevorzugen.

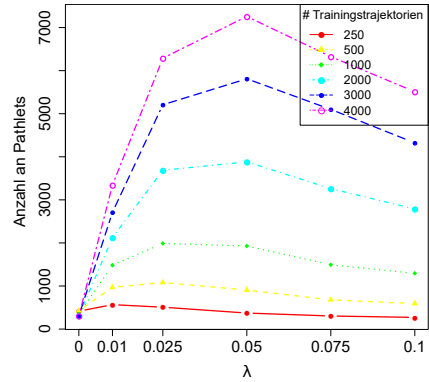
Dementsprechend wird für die weitere Evaluation ein Substrukturverzeichnis verwendet, das mit $\lambda = 0.025$ und 2000 Trainingsdaten erzeugt wurde. Diese Parameterwahl ist ein sinnvoller Kompromiss aus Rekonstruktionseffizienz (siehe Abbildung 5.2c), Trainingsaufwand (die Rechenkomplexität beim Pathlet Learning steigt linear mit der Anzahl an Trainingstrajektorien, siehe Abschnitt 4.1.3.1), sowie langen und damit für die Analyse hilfreichen Substrukturen.

Darüber hinaus kann die Größe des Substrukturverzeichnisses wie in Abschnitt 3.5 zusätzlich optimiert werden, da nicht alle extrahierten Substrukturen tatsächlich benötigt werden. Im Falle des für die Evaluation ausgewählten Substrukturverzeichnisses genügen bereits die 40% der in den meisten Trainings-trajektorien vorkommenden Substrukturen, um weiterhin eine ausreichend hohe durchschnittliche Überdeckungsrate (>0.99) zu erzielen (siehe Abbildung 5.2e).

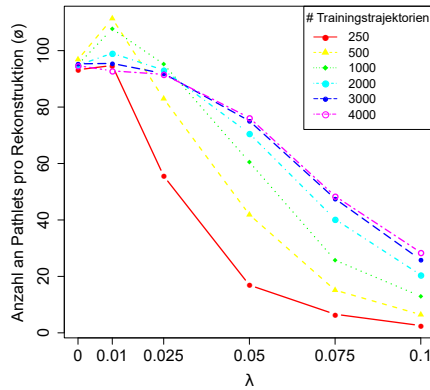
Insgesamt konnte also in diesem Abschnitt gezeigt werden, dass sich auch aus Bewegungsdaten von mobilen Robotern, die sich frei in einem SOIS bewegen, mittels Pathlet Learning ein Substrukturverzeichnis erzeugen lässt, das damit



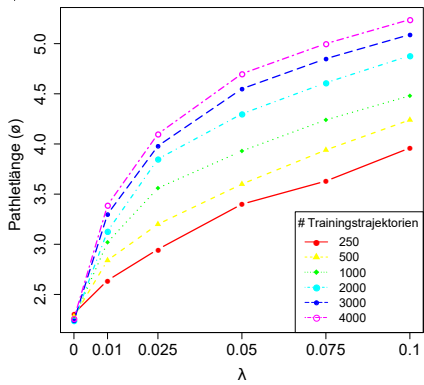
(a) Durchschnittliche Überdeckungsraten.



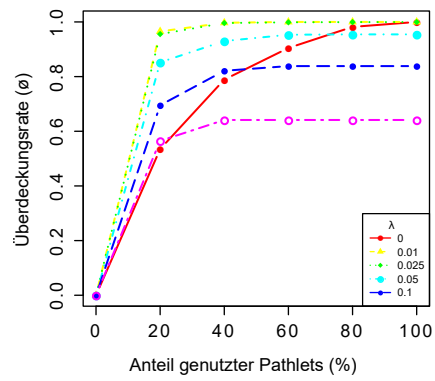
(b) Verzeichnisgrößen.



(c) Durchschnittliche Anzahl an zur Rekonstruktion benötigter Substrukturen.



(d) Durchschnittliche Substruktur­längen.



(e) Durchschnittliche Überdeckungs­raten.

Abbildung 5.2: Evaluationsergebnisse für die Trainingsphase.

die Anforderungen der *Rekonstruktionseffizienz*, *Kompaktheit* und *Universalität* erfüllt.

5.1.3.2 Bewegung

Die Überwachung der räumlichen Bewegung von mobilen Robotern in einem SOIS erfolgt analog zu dem Ansatz für SOIS mit Fließbandarchitektur aus Abschnitt 3.6. Das heißt, zur Erkennung und Analyse der Anomalien werden Informationen verwendet, die bei der Rekonstruktion der Testtrajektorien aus den Substrukturen des Verzeichnisses anfallen.

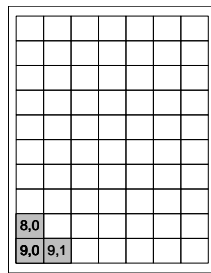
Dabei kann ebenso unterschieden werden zwischen Anomalien innerhalb einer einzelnen Route (siehe Abschnitt 5.1.3.2.1) und Anomalien, die sich bei Betrachtung des globalen Systemverhaltens ergeben (siehe Abschnitt 5.1.3.2.2). Allerdings ist zu beachten, dass insbesondere in letzterem Fall die Aussage einer detektierten Änderung des globalen Systemverhaltens eine andere ist als in der zuvor diskutierten Fließbandarchitektur.

Entsprechend wird im Folgenden anhand geeigneter Evaluationsszenarien gezeigt, an welchen Stellen das Vorgehen aus Abschnitt 3.6 übernommen werden kann und an welchen Stellen Modifikationen beziehungsweise Erweiterungen nötig sind, um auch in SOIS mit mobilen Robotern zur Laufzeit Anomalien bezüglich der räumlichen Bewegung detektieren und analysieren zu können.

5.1.3.2.1 Einzeltrajektorien Für die Erkennung von anormalen räumlichen Bewegungen wird in Abschnitt 3.6.1 die *Überdeckungsrate* als Maß für die (Ab-)Normalität von Einzeltrajektorien eingeführt. Abhängig von den szenariospezifischen Anforderungen kann ein Schwellwert definiert werden, sodass Routen, deren Überdeckungsrate diesen Schwellwert unterschreiten, als abnormal klassifiziert werden. Dass dieses Vorgehen auch für SOIS mit mobilen Robotern sinnvoll ist, wird deutlich, wenn man in der vorliegenden Simulation einen entsprechenden Schwellwert definiert und sich anschaut, welche Routen dadurch als anormal klassifiziert werden.

Im Folgenden wurde dafür ein Schwellwert von 0.95 angenommen. Das heißt, eine Trajektorie muss zu mindestens 95% aus den Substrukturen des Verzeichnisses rekonstruierbar sein, um nicht als anormal klassifiziert zu werden. Bei der Untersuchung von 500 Testtrajektorien, die durch den Ablauf der Simulation generiert wurden, wurden so sieben Trajektorien als abnormal eingestuft, da ihre Überdeckungsrate < 0.95 war.

Wesentlich interessanter an dieser Stelle ist allerdings die Analyse der detektierten Abnormalitäten. Entsprechend sind in Abbildung 5.3 für eine der anormalen Testtrajektorien die nicht rekonstruierbaren Teilstücke dargestellt. Zum einen sind in Abbildung 5.3a die Rasterzellen und damit die räumliche Lage der nicht rekonstruierbaren Trajektorienteile visualisiert und mit den entsprechenden x - und y -Koordinaten versehen. Zusätzlich sind in Tabelle 5.3b die formalen Datenrepräsentationen der als anormal klassifizierten Verläufe zu sehen.



(a) Räumliche Lage.

x	y	t
9	0	1396
9	1	1397
9	0	1398
9	1	1405

x	y	t
9	0	1414
8	0	1416
9	0	1419
8	0	1420
9	0	1423
8	0	1424

(b) Anormale Daten.

Abbildung 5.3: Nicht rekonstruierbare Teile einer anormalen Trajektorie.

Offensichtlich gibt es keine Substruktur, die räumliche Bewegungen von der Rasterzelle (9,0) zur Rasterzelle (9,1) abdeckt beziehungsweise auch in die Gegenrichtung. Das gleiche gilt für die Bewegung von (9,0) nach (8,0), oder andersherum. Betrachtet man den der Simulation zugrunde liegenden Raumplan (siehe Abschnitt 5.1a), macht das Sinn, da die Roboter keinen plausiblen Grund, wie zum Beispiel eine Maschine, haben, sich in dieser Ecke zu bewegen. Das heißt, auch in diesem Fall kann die Überdeckungsrate als Maß der (Ab-)Normalität von Einzeltrajektorien genutzt werden. Darüber detektierte anormale Bewegungen von Robotern können zusätzlich durch entsprechende Visualisierung analysiert werden.

5.1.3.2.2 Globale Ereignisse In Abschnitt 3.6.2 und Abschnitt 3.6.4 wurde gezeigt, dass durch Beobachtung der *relativen Nutzerhäufigkeiten* der Substrukturen globale anormale Ereignissen erkannt werden können. Dabei ging es darum, zu sehen, wenn bestimmte räumliche Bereiche nicht mehr oder signifikant öfter beziehungsweise seltener genutzt werden.

Derartige Analysen haben in SOIS mit mobilen Robotern allerdings wenig bis gar keine Aussagekraft mehr. Zum einen sind Maschinenstandorte räumlich nicht von der frei befahrbaren Fläche zu unterscheiden. Damit schlägt sich die Nichtnutzung einer Maschine auch nicht in der Nichtnutzung der Substrukturen, die den Maschinenstandort beinhalten, nieder.

Zum anderen sind die Bewegungen der Roboter zwischen den Maschinen nicht auf wenige, klar separierte vordefinierten Verbindungen beschränkt. Stattdessen ergeben sich eine Vielzahl an Verbindungspfaden, deren Überwachung bezüglich Lastenverteilung damit nicht nur auf Grund ihrer Anzahl, sondern auch durch Routenüberlagerungen nicht mehr umsetzbar ist, beziehungsweise selbst dann wenig Aussagekraft hätte.

Um trotzdem Aussagen über Änderungen im globalen Bewegungsverhalten der mobilen Roboter treffen zu können, wird weiterhin die *relative Nutzerhäufigkeit* der Substrukturen ausgewertet. Darauf basierend wird die Menge an anormal oft beziehungsweise selten genutzter Substrukturen dahingehend untersucht, ob die betroffenen Substrukturen lediglich eine zufällige Untermenge des Verzeichnisses sind, oder ob sich innerhalb dieser Menge ein Muster erkennen lässt,

das auf eine Änderung des globalen Bewegungsverhaltens schließen lässt. Die Erkennung und Analyse derartiger globaler Änderungen in den Bewegungen der Roboter ist insbesondere für die Qualitätssicherung in SOIS, in denen das kognitive System im laufenden Betrieb noch weiterlernt, von enormer Bedeutung, um ebendiesen Lernprozess überwachen zu können.

Für die Analyse der Menge an anormal genutzten Substrukturen werden im Folgenden drei sogenannte *Bewegungsparameter* eingeführt, mit deren Hilfe nun auch globale Ereignisse in einem SOIS mit mobilen Robotern erkannt werden können:

1. **Räumliche Lage:** Zwar ist im Gegensatz zur Fließbandarchitektur in SOIS mit mobilen Robotern, wie eingangs erläutert, keine so aussagekräftige Analyse der Lastenverteilung auf einzelne Verbindungspfade möglich, eine Änderung der Lastenverteilung auf die genutzten Raumflächen in der Fabrik kann allerdings erkannt werden.

Dazu wird in der Menge an anormal genutzten Substrukturen nach Mustern bezüglich der räumlichen Lage der betroffenen Substrukturen gesucht. Wie in Abschnitt 3.6.4 werden dafür Wärmebilder genutzt.

Im Falle von Substrukturen, die signifikant öfter zur Rekonstruktion genutzt werden als in der Trainingsphase, wird dazu für jede Rasterzelle im Raumplan der Maximalwert der Differenz der relativen Nutzerhäufigkeiten berechnet; für Substrukturen, die signifikant weniger genutzt werden, wird analog in einem separaten Wärmebild für jede Rasterzelle, der entsprechend minimale Wert der Differenz aus den relativen Nutzerhäufigkeiten berechnet.

Damit lässt sich die Berechnung des Wertes $col(a)$ zur Einfärbung einer Rasterzelle a folgendermaßen formalisieren:

$$col(a) = \begin{cases} \min_{a \subset p \in P} \Delta f_r(p), & \Delta f_r(p) < t_f \\ \max_{a \subset p \in P} \Delta f_r(p), & \Delta f_r(p) > t_f, \end{cases} \quad (5.1)$$

wobei t_f der zuvor definierte Schwellwert für die relative Nutzerhäufigkeit ist, ab dem die Nutzung einer Substruktur als anormal gilt.

Für das Farbschema bietet sich eine Intensivierung der Farbstärke mit steigender Differenz der Nutzerhäufigkeit an. Das hieße dann, je dunkler die Farbe, um so größer ist der Unterschied bezüglich der Nutzerhäufigkeit.

2. **Richtung:** Eine weitere Möglichkeit, nach Mustern in den anormal genutzten Substrukturen zu suchen, die auf eine globale Änderung des Bewegungsverhalten der mobilen Roboter schließen lassen, ist die Analyse der Richtung der betroffenen Substrukturen. Diese würde zum Beispiel aufdecken, wenn es eine Umstellung von geradlinigen auf eher kurvenreiche Routen gibt, oder, wenn die Maschinen nicht mehr in der für den

Gesamtprozess günstigen Reihenfolge von den Start- zu den Endpunkten angefahren werden, sondern vermehrt auch ungünstigerweise „zurückgefahren“ wird.

Dafür wird für jede der anormal genutzten Substrukturen die Hauptausrichtung bestimmt, indem der Rotationswinkel ρ (in Grad) zwischen dem ersten Samplepunkt der Substruktur (x_1, y_1) und dem letzten Punkt (x_n, y_n) mit Hilfe der Arcus-Tangens-Funktion berechnet wird:

$$\rho = \frac{-\operatorname{atan}\left(\frac{\Delta y}{\Delta x}\right) * 180}{\pi} = \frac{-\operatorname{atan}\left(\frac{|y_n - y_1|}{|x_n - x_1|}\right) * 180}{\pi} \quad (5.2)$$

3. **Länge:** Als dritter Bewegungsparameter kann noch die Länge der anormal genutzten Substrukturen untersucht werden. Grundsätzlich gilt, dass längere Substrukturen prozessspezifischer sind, während kürzere Substrukturen eher wenige Informationen über typische Prozessabläufe beinhalten. Sollten daher hauptsächlich längere Substrukturen signifikant weniger genutzt werden, deutet dies auf eine Abweichung von bisher typischen Prozessabläufen hin.

Zur Erkennung eines derartigen globalen Ereignisses, wird die Länge $l(p)$ einer Substruktur $p = s_1, s_2, \dots, s_n$ entsprechend definiert als:

$$l(p) = |s_i|, s_i \in p \quad (5.3)$$

Im Folgenden wird anhand von 500 Testtrajektorien, die sich maßgeblich vom Bewegungsverhalten der gemäß des Simulationsmodelles aus Abschnitt 5.1.1 generierten Trainingstrajektorien unterscheiden, gezeigt, wie eine derartige Analyse von anormal genutzten Substrukturen ablaufen kann. Die Testtrajektorien starten beziehungsweise enden ganz normal entsprechend des Simulationsmodells aus Abschnitt 5.1.1 an einem Start- bzw. Endpunkt in der Fabrik. Der Pfad dazwischen besteht allerdings aus der Konkatenation von kürzesten Pfaden zwischen Zufallspunkten. Damit ergibt sich ein grundlegend anderes Bewegungsverhalten.

In Abbildung 5.4a sind zunächst die sich für jede Substruktur ergebenden Differenzen bezüglich der relativen Nutzerhäufigkeiten zu sehen. Offensichtlich gibt es Substrukturen, die bei einem Schwellwert von $t_f = \pm 0.3$ anormal oft zur Rekonstruktion der Testtrajektorien genutzt werden.

Für die weitere Analyse dieser Menge anormal genutzter Substrukturen sind in den Abbildungen 5.4b - 5.4e die Auswertungen der entsprechenden Bewegungsparameter visualisiert. In Abbildung 5.4b zum Beispiel ist das Wärmebild zu sehen, das die *räumliche Lage* der betroffenen Substrukturen zusammen mit der entsprechenden Intensität der Differenz der relativen Nutzerhäufigkeiten zeigt. Nachdem es in diesem Fall keine Substrukturen gibt, die bezüglich der Differenz der Nutzerhäufigkeiten nicht unter den negativen Schwellwert fallen, muss kein weiteres Wärmebild ausgewertet werden. Offensichtlich werden

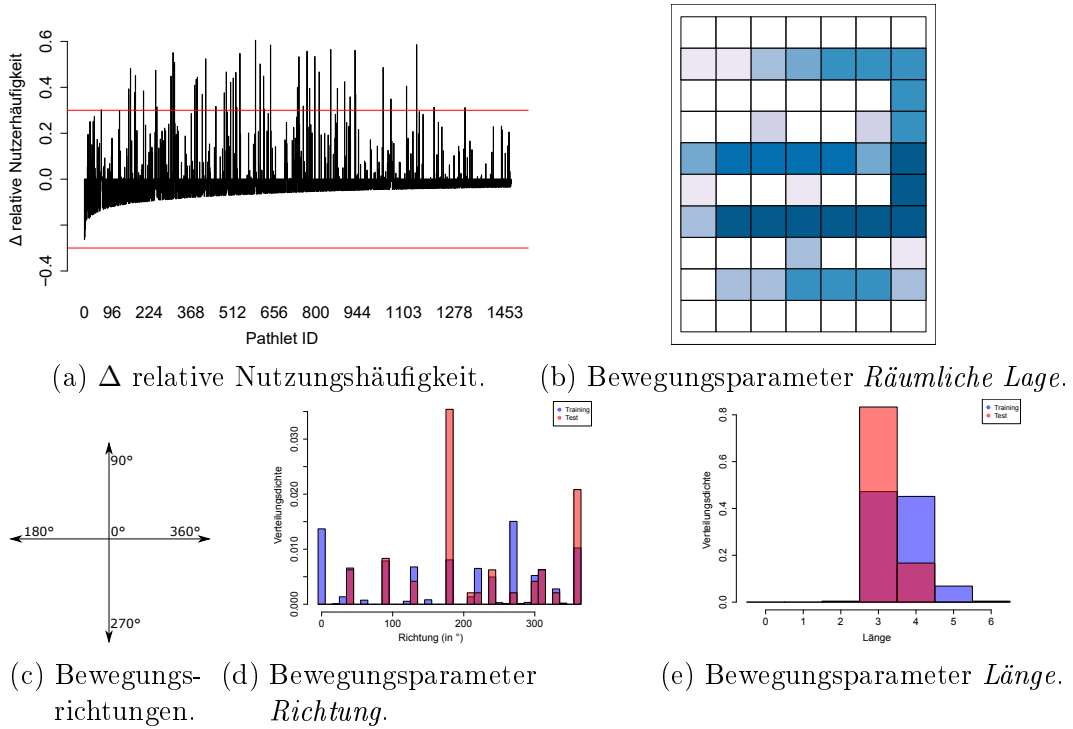


Abbildung 5.4: Erkennung und Analyse einer anormalen Bewegungsstrategie.

hauptsächlich Substrukturen signifikant öfter genutzt, die eine Rekonstruktion relativ langer und geradliniger Pfade im Zentrum der Fabrik ermöglichen. In Abbildung 5.4d ist der Bewegungsparameter *Richtung* für die anormal genutzten Substrukturen ausgewertet. Dazu wird die Dichteverteilungsfunktion für die möglichen Richtungen ($\rho = [0^\circ, 360^\circ]$, siehe Abbildung 5.4c) aller Substrukturen (blaue Balken) mit denen der anormalen Menge (rote Balken) verglichen. Offensichtlich ist die Mehrheit der anormal oft genutzten Substrukturen horizontal ($\rho = 180^\circ/360^\circ$) ausgerichtet. Diese Beobachtung deutet darauf hin, dass die Untermenge der anormal genutzten Substrukturen eben keine zufällige Teilmenge des Verzeichnisses ist, sondern, dass es hauptsächlich horizontal ausgerichtete Substrukturen sind, die öfter zur Rekonstruktion benötigt werden.

Zuletzt wird in Abbildung 5.4e der Bewegungsparameter *Länge* analysiert. Man erkennt, dass die meisten der anormal oft genutzten Substrukturen eine Länge von 3 (rote Balken) haben. Auch hier legt der Vergleich mit der Grundverteilung (blaue Balken) die Vermutung nahe, dass es sich um ein Muster eines möglichen, neuen Bewegungsverhaltens und nicht um eine Zufallsmenge an Substrukturen handelt.

Dementsprechend ergänzen die beiden Analysen bezüglich *Richtung* und *Länge* nun die Erkenntnisse aus dem Wärmebild, das die *Räumliche Lage* der anormal genutzten Substrukturen zeigt. Offensichtlich sind in dem anormalen Verhalten, das sich in einer signifikanten Änderung bezüglich der relativen

Nutzerhäufigkeit zeigt, Muster zu erkennen. Nimmt man nun die Auswertung aller drei Bewegungsparameter zusammen, unterscheidet sich das beobachtete Bewegungsverhalten vom ursprünglichen insofern, als dass im Speziellen horizontal ausgerichtete Substrukturen der Länge 3 für die Rekonstruktion von länglichen und geradlinigen Pfaden im Zentrum (nicht am Rand) der Fabrik zum Einsatz kommen.

Diese Beobachtung stimmt mit dem den Testtrajektorien zugrunde liegenden Bewegungsverhalten der Kürzester-Pfad-Routen überein. Die Routen setzen sich aus langen und geradlinigen Pfadabschnitten zusammen, da für den kürzesten Pfad zwischen zwei Punkten, der Roboter erst direkt in eine Richtung geht, bis er diesbezüglich die finale Koordinate erreicht hat und dann direkt in die zweite Richtung, um den Zielpunkt zu erreichen.

Insgesamt konnte in diesem Abschnitt also gezeigt werden, dass durch die Auswertung der Bewegungsparameter *Räumliche Lage*, *Richtung* und *Länge* Änderungen im globalen Bewegungsverhalten der Roboter in einem SOIS erkannt und analysiert werden können. Im Speziellen werden diese drei Bewegungsparameter für die Menge an anormal oft oder wenig genutzten Substrukturen ausgewertet, um darin nach Mustern zu suchen, die auf eine derartige globale Änderung hindeuten.

5.1.3.3 Geschwindigkeit

Zur Überwachung der Geschwindigkeiten der mobilen Roboter in einem SOIS findet, wie eingangs erwähnt, ein Abgleich mit dem zeitlichen Normalitätsmodell – den Zeitdauermodellen (siehe Abschnitt 5.1.2) – statt. Das heißt, sobald eine Substruktur zur Rekonstruktion einer Route benötigt wird, werden die entsprechenden Zeitdaten t_i der beobachteten Trajektorie mit dem Zeitdauermodell der Substruktur verglichen.

Die dabei anfallenden Informationen wie der Grad der Abweichung, können nun zur Erkennung und Analyse von Anomalien bezüglich des zeitlichen Ablaufs verwendet werden. Dabei kann erneut unterschieden werden zwischen Anomalien, die sich bei Einzeltrajektorien ergeben (siehe Abschnitt 5.1.3.3.1) und globalen anormalen Ereignissen (siehe Abschnitt 5.1.3.3.2).

5.1.3.3.1 Einzeltrajektorien Zur Erkennung einer Anomalie bezüglich des zeitlichen Ablaufs einer Einzeltrajektorie wird für jede Trajektorie überwacht, in wie fern die beobachteten Zeitdauerwerte mit den Zeitdauermodellen der Substrukturen, die zur Rekonstruktion verwendet wurden, konform sind. Damit kann zum einen erkannt werden, wenn ein Roboter seine Geschwindigkeit drastisch erhöht und damit zu einem Risiko für die anderen Roboter wird, die ein solches Verhalten nicht erwarten. Umgekehrt kann aber auch erkannt werden, wenn ein Roboter auf einmal signifikant langsamer wird oder an Stellen in der Fabrik stehen bleibt, wo es nicht nötig ist.

Dazu muss auch hier Szenario spezifisch ein Schwellwert definiert werden, der

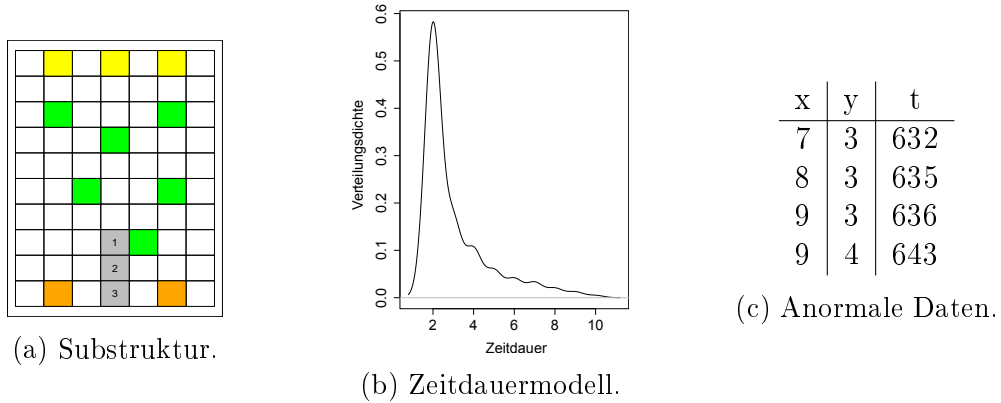


Abbildung 5.5: Beispiel einer Substruktur, das entsprechende Zeitdauermodell und anormale Daten, die eine Anomalie bezüglich zeitlichen Ablaufs induzieren.

angibt, welche Wahrscheinlichkeitsdichte die Zeitdauerbeobachtung im entsprechenden Zeitdauermodell mindestens haben muss, um noch als normal klassifiziert zu werden. Für das vorliegende Simulationsszenario wurde ein Schwellwert von 0.001 ermittelt, um eine sinnvolle Unterscheidung von normalen und anomalen Verhalten zu erhalten.

In Abbildung 5.5a ist nun beispielhaft eine Substruktur aus dem verwendeten Verzeichnis und ihr entsprechendes Zeitdauermodell (siehe Abbildung 5.5b) zu sehen. Das Zeitdauermodell hat eine Wahrscheinlichkeitsspitze um den Wert 2 herum. Dies passt zur intuitiven Annahme, dass Roboter diesen Pfadabschnitt eigentlich nur dann nutzen, wenn die Aufgabenliste ihrer Werkteile bereits vollständig abgearbeitet wurde und sie daher nun auf dem Weg zu einem Endpunkt in der Fabrik – in diesem Fall, dem mittleren – sind.

In Tabelle 5.5c ist ein Ausschnitt aus der formalen Datenrepräsentation einer Beispieltrajektorie zu sehen, der auf Grund des zuvor definierten Schwellwertes eine Anomaliemeldung auslöst, da der entsprechende Roboter $d = t_{n+1} - t_1 = 643 - 632 = 11$ Zeitschritte für den durch die Substruktur beschriebenen Pfad benötigt, was aber laut Zeitdauermodell eine Wahrscheinlichkeitsdichte < 0.001 hat. Der zeitliche Verlauf der Route stimmt also nicht mit dem für diese Substruktur gegebenen Zeitdauermodell überein.

Die bereits erwähnte Lokalitätsinformation („Pfad zum mittleren Endpunkt“) ermöglicht dahingehend eine weitergehend Analyse der Anomalie, dass vermutlich die Aufgabenliste des vom mobilen Roboter transportieren Werkteils noch gar nicht vollständig abgearbeitet wurde, sonst würde der Roboter ohne Verzögerung zum Endpunkt durchfahren. Stattdessen scheint er Probleme in der Planungsphase zu haben, um zu seiner nächsten Maschine zu gelangen und hält sich dabei anormalerweise an diesem Endpunkt auf.

Entsprechend ist es für die Qualitätssicherung in SOIS von enormer Bedeutung, dass derartige Einzeltrajektorien als anormal klassifiziert werden, um zu erkennen, dass es möglicherweise Probleme in der Planungsphase gibt.

Insgesamt konnte in diesem Abschnitt gezeigt werden, dass eine Erkennung und Analyse von Anomalien im zeitlichen Ablauf von Einzeltrajektorien möglich ist, indem die beobachteten Zeitspannen der Trajektorien mit den Laufzeitmodellen der zur Rekonstruktion verwendeten Substrukturen abgeglichen werden. Die Analyse der detektierten Anomalien erfolgt über Lokalitätsinformationen, die sich durch eine entsprechende Visualisierung der betroffenen Substruktur ergibt.

5.1.3.3.2 Globale Ereignisse Wie in Abschnitt 5.1.3.2.2 bereits erläutert, schlägt sich in einem SOIS mit mobilen Robotern der Ausfall einer Maschine nicht unbedingt in anormalen *relativen Nutzerhäufigkeitswerten* der Substrukturen nieder, da der Maschinenstandort weiterhin als Wegstrecke genutzt werden kann. Stattdessen können Informationen über die Nutzung einer Maschine über das zeitliche Normalitätsmodell – die Zeitdauermodelle – gewonnen werden (siehe Abschnitt 5.1.2). Das heißt, zur Erkennung und Analyse derartiger globaler Ereignisse muss ein Abgleich der beobachteten Zeitdauerwahrscheinlichkeitsverteilung mit den Zeitdauermodellen des Substrukturverzeichnis erfolgen.

Damit lässt sich zum Beispiel erkennen, wenn eine Wahrscheinlichkeitsspitze im Zeitdauermodell, die durch einen Maschinenstandort innerhalb der Substruktur verursacht wird (siehe zum Beispiel Abbildung 5.1c), auf einmal abflacht, während gleichzeitig die Spitze, die die Zeitdauer für die einfache Wegstrecke markiert, ansteigt. Dies würde daraufhin deuten, dass die betroffene Maschine weniger oder im Extremfall gar nicht mehr genutzt wird.

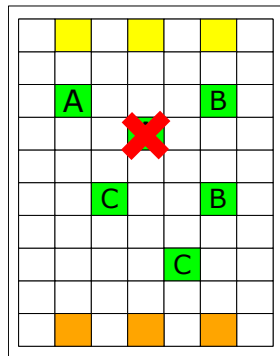
Das gleiche gilt für Situation, in denen neue Maschinen in der Fabrik installiert werden oder anderweitige signifikante Änderungen in der Maschinennutzung auftreten.

In jedem Fall unterscheidet sich die Wahrscheinlichkeitsverteilung der Zeitdauerwerte der Testdaten signifikant von den zur Trainingszeit erstellten Laufzeitmodellen.

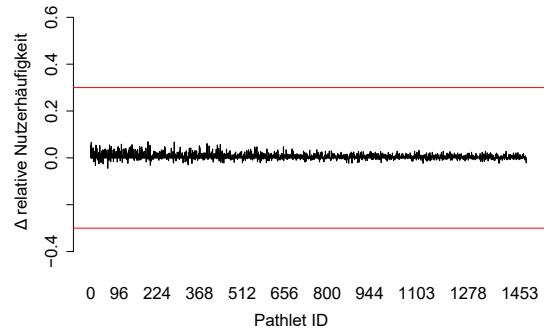
Der Abgleich der beiden Zeitdauerwahrscheinlichkeitsverteilungen kann über einen Kolmogorov-Smirnov (KS)-Test erfolgen [Lope 11]. Dieser vergleicht die empirisch kumulativen Verteilungsfunktionen zweier Samplepunktmengen. Dazu wird die Hypothese überprüft, ob die sich aus den Zeitdauerwerten der Testtrajektorien ergebende Wahrscheinlichkeitsverteilung mit der Wahrscheinlichkeitsverteilung aus der Trainingsphase übereinstimmt. Entsprechend ergibt sich für jede Substruktur, die für die Rekonstruktion der Testtrajektorien benötigt wird, ein p -Wert, der dann mit einem szenariospezifischen Signifikanzniveau α verglichen werden kann.

Zur Evaluation dieses Ansatzes wurde nun in der SOIS-Simulation ein derartiges anormales globales Ereignis ausgelöst, indem eine Maschine beispielhaft entfernt wurde (siehe Abbildung 5.6a).

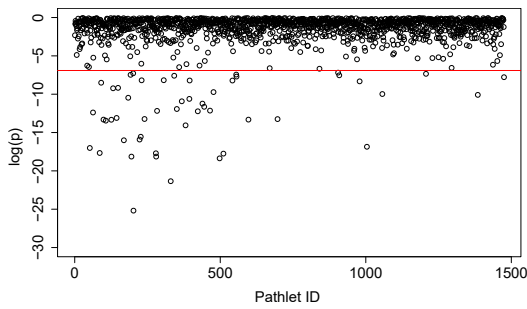
In Abbildung 5.6b sieht man die entsprechenden Differenzen der *relativen Nutzerhäufigkeiten* für jede Substruktur. Wie zu erwarten ergeben sich keine signi-



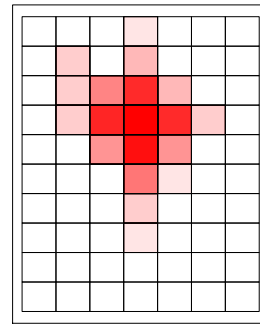
(a) Evaluationszenario zur Simulation eines Maschinenausfalls.



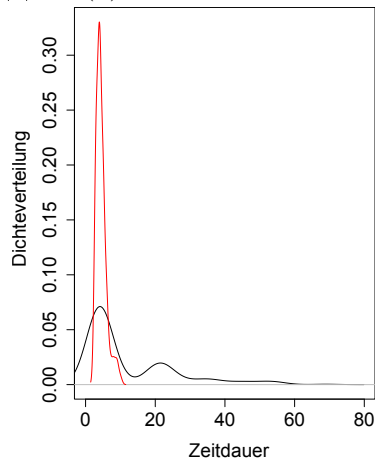
(b) Δ relative Nutzungshäufigkeit.



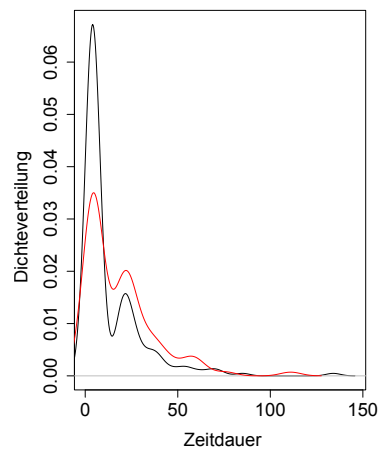
(c) $\log(p)$ -Werte des KS-Tests.



(d) Überlappungskarte.



(e) Zeitdauermodell 1.



(f) Zeitdauermodell 2.

Abbildung 5.6: Erkennung und Analyse eines anomalen globalen Ereignisses.

fikanten Änderungen, wobei in diesem Fall ein szenariospezifischer Schwellwert von $t_f = \pm 0.3$ angesetzt wurde (rote Linie).

Im Gegensatz zeigt sich die Nichtnutzung der aus der Fabrik entfernten Maschine in den p -Werten der KS-Tests (siehe Abbildung 5.6c). In diesem Fall wurde ein Signifikanzniveau von $\alpha = 0.001$ festgelegt, das bei einigen Substrukturen deutlich unterschritten wird.

Für die Analyse der auf diese Art detektierten Anomalien können die räumlichen Informationen der betroffenen Substrukturen erneut ausgewertet werden. Dafür werden alle zeitlich anormal genutzten Substrukturen mit einem transparenten Farbton im Raumplan der Fabrik visualisiert. Dadurch ergibt sich eine Überlappungskarte wie sie in Abbildung 5.6d zu sehen ist, bei der die Intensität der Farbe eine Lokalisierung der anormal genutzten Maschine ermöglicht (dunkelster Farbton = größte Überscheidung).

Offensichtlich ist der KS-Test im vorliegenden Fall ausschließlich bei Substrukturen negativ ausgefallen, die den Standort der entfernten Maschine oder Bereiche rund um den ehemaligen Maschinenstandort enthalten.

Zusätzlich sind in Abbildung 5.6e und Abbildung 5.6f beispielhaft zwei Zeitdauermodelle von anormal genutzten Substrukturen zu sehen. Dabei ist die während der Trainingsphase gelernte Wahrscheinlichkeitsverteilung schwarz eingetragen, während die Kurve für die Zeitdauerwahrscheinlichkeiten der Testtrajektorien in rot eingezeichnet ist. Die Verteilungskurve in Abbildung 5.6e hat ursprünglich zwei Wahrscheinlichkeitsspitzen (eine beim Zeitdauerwert 3 und die andere beim Wert 21), was darauf hinweist, dass die Substruktur eine Rasterzelle beinhaltet, die von den Robotern sowohl zur Bearbeitung der Werkteile durch die entsprechende Maschine angefahren wird (längere Zeitdauern, hier: 21), aber auch als einfache Wegstrecke (kürzere Zeitdauer, hier: 3) genutzt wird. Nach dem Ausfall der Maschine fallen offensichtlich diese längeren Zeitdauern weg (siehe rote Kurve), da die betroffene Stelle nur noch als Wegstrecke genutzt wird.

Im zweiten Beispiel dagegen flacht die größere der beiden Wahrscheinlichkeitsspitzen im Laufzeitmodell (kurze Zeitdauer, hier: 3) nach dem Ausfall der Maschine ab (siehe rote Linie in Abbildung 5.6f), während die zweite Spitze (längere Zeitdauer, hier: 20) und die Wahrscheinlichkeitswerte für die Zeitdauerwerte darum herum ansteigen. Dieses Verhalten zeigt sich bei Substrukturen, die den Standort einer Maschinen gleichen Typs, wie den der defekte Maschinen, enthalten. Der weiterhin anfallende Bedarf für Bearbeitungsschritte diesen Typs wird auf diese Maschine umgelenkt, wodurch die entsprechende Rasterzelle nun öfter für die Bearbeitung von Werkteilen angefahren wird (höhere Wahrscheinlichkeiten für lange Zeitdauern), als als Wegstrecke (geringere Wahrscheinlichkeiten für kurze Zeitdauern) wie bisher. Zusätzlich ergeben sich längere Wartezeiten an der einzig verbleibenden Maschine diesen Typs, was sich in steigenden Wahrscheinlichkeitswerten für Zeitdauerwerte rechts des zweiten Spitze im Zeitdauermodell zeigt.

Insgesamt konnte also in diesem Abschnitt gezeigt werden, dass globale Ände-

rungen bezüglich der Maschinennutzung in einem SOIS mit mobilen Robotern nicht wie bisher über die *relative Nutzerhäufigkeit* der Substrukturen erkannt werden können (siehe Abschnitt 3.6.2), sondern stattdessen ein Abgleich der Zeitdauerwahrscheinlichkeitsverteilungen der Testtrajektorien mit den Zeitdauermodellen des Substrukturverzeichnisses durchgeführt werden muss.

Zusätzlich können bei der weiteren Analyse der so detektierten anomal genutzten Maschinen neben der Lokalisierung derselben, auch weitere Effekte, wie die Lastenumverteilung auf alternative Maschinen beobachtet werden.

5.1.4 Zusammenfassung

In diesem Abschnitt wurde das in Kapitel 3 vorgestellte Konzept zur Erkennung und Analyse von Anomalien in raumbezogenen Daten von SOIS für Bewegungsdaten von Objekten mit variabler Geschwindigkeit umgesetzt.

Dazu wurde zunächst ein neues Simulationsmodell eines SOIS eingeführt, in dem die mobilen Roboter nicht mehr nur ihre Routen, sondern eben auch ihre Geschwindigkeit selbst bestimmen können, um die Werkteile von einer Maschine zur nächsten zu transportieren (siehe Abschnitt 5.1.1). Um auch diese zusätzliche zeitliche Dimension zur Laufzeit überwachen zu können, wurde das Substrukturverzeichnis, wie es bisher bei Bewegungsdaten von Objekten mit konstanter Geschwindigkeit zum Einsatz gekommen ist (siehe Abschnitt 3.4), um eine zeitliche Komponente erweitert. Dafür wird für jede Substruktur ein Zeitdauermodell erstellt, indem eine Dichteschätzung über die einzelnen Zeitspannenbeobachtungen durchgeführt wird (siehe Abschnitt 5.1.2).

Im Rahmen der Evaluation wurde zunächst gezeigt, dass, wie im Falle von Bewegungsdaten von Objekten mit konstanter Geschwindigkeit, auch im Falle von Bewegungsdaten von Objekten mit variabler Geschwindigkeit mittels Pathlet Learning ein zur Anomalieerkennung geeignetes Substrukturverzeichnis erstellt werden kann, das die Qualitätskriterien der *Universalität* der Substrukturen, *Kompaktheit* des Verzeichnisses und *Effizienz* des Rekonstruktionsprozesses erfüllt (siehe Abschnitt 5.1.3.1).

Für die weitergehende Evaluation wurden in Abschnitt 5.1.3.2 und Abschnitt 5.1.3.3 sowohl Anomalien bezüglich der räumlichen Bewegung, als auch bezüglich des zeitlichen Ablaufs untersucht. Dafür wurden in beiden Fällen Situationen simuliert, in denen es zu anormalem Verhalten eines einzelnen Roboters kam, oder aber Ereignisse eintraten, die sich global auswirkten.

So konnte in Abschnitt 5.1.3.2.1 gezeigt werden, dass zur Erkennung von Einzeltrajektorien, die sich anormal bewegen, analog zu SOIS mit Fließbandarchitektur die Überdeckungsrate, die sich bei der Rekonstruktion der Testdaten ergibt, als Maß für die (Ab-)Normalität genutzt werden und eine weiterführende Analyse der detektieren Anomalie durch Visualisierung der nicht rekonstruierbaren Teilstücke erfolgen kann.

Für die Erkennung von Änderungen im globalen Bewegungsverhalten dagegen musste ein neues Verfahren eingeführt werden, bei dem die drei Bewegungspara-

parameter *Räumliche Lage*, *Richtung* und *Länge* für anormal genutzte Substrukturen ausgewertet werden (siehe Abschnitt 5.1.3.2.2). Änderungen im globalen Bewegungsverhalten der mobilen Roboter im SOIS können dann durch entsprechende Muster in den Bewegungsparametern erkannt und analysiert werden. Für den Fall einer Anomalie im zeitlichen Ablauf einer Einzeltrajektorie konnte schließlich in Abschnitt 5.1.3.3.1 gezeigt werden, dass durch den Abgleich der beobachteten Zeitspannen der Trajektorie mit den Laufzeitmodellen der zur Rekonstruktion verwendeten Substrukturen nun auch derartige Geschwindigkeits-Anomalien erkannt werden können. Eine weitergehende Analyse der detektierten Anomalie ist auch hier durch Visualisierung der entsprechenden Lokalitätsinformationen möglich.

Im Falle einer Anomalie, die globale Auswirkungen bezüglich des zeitlichen Verlaufs der mobilen Roboter hat, wie zum Beispiel der Ausfall einer Maschine, konnte gezeigt werden, dass die reine Beobachtung der relativen Nutzerhäufigkeit der Substrukturen, wie im Falle von Objekten mit konstanter Geschwindigkeit, nicht mehr ausreicht (siehe Abschnitt 5.1.3.3.2). Stattdessen muss ein Abgleich der Wahrscheinlichkeitsverteilung der Zeitdauern der Testtrajektorien mit den Zeitdauermodellen des Substrukturverzeichnisses durchgeführt werden. Die anschließende Analyse der detektierten Anomalien kann erneut über geeignete Visualisierung erfolgen.

Damit konnte also in diesem Abschnitt gezeigt werden, dass mit Hilfe der vorgestellten Erweiterung des Substrukturverzeichnisses das Konzept zur Erkennung und Analyse von Anomalien aus Kapitel 3 auch für Bewegungsdaten von Objekten mit variabler Geschwindigkeit umgesetzt werden kann.

5.2 Laufzeitüberwachung von Zustandsbeschreibungen

Bisher wurde das Konzept aus Abschnitt 3.1 für die Bewertung von Bewegungsdaten von mobilen Werkteilen (siehe Kapitel 3) oder Robotern (siehe Abschnitt 5.1) in einem SOIS umgesetzt. Wie in Abschnitt 2.1.3 bereits erwähnt, ergeben sich in SOIS aber auch noch andere Arten von raumbezogenen Daten, wie zum Beispiel räumliche Schichtmodelle, die im kognitiven System zur Darstellung des Systemzustands verwendet werden (siehe Abschnitt 2.1.3.3).

Auch diese Daten müssen auf Anomalie hin untersucht werden, insbesondere wenn sie - wie im Falle von einem mittels bestärkendem Lernens erzeugten kognitiven Systems - als Eingabe für das entsprechende CNN verwendet werden (siehe Abschnitt 2.1.3.3). Schließlich ergibt sich bei Neuronalen Netzen generell das Problem, dass diese keine Informationen darüber liefern, wie sicher beziehungsweise unsicher sie sich bei ihren Aussagen sind [Gal 16].

Dies ist insbesondere dann problematisch, wenn sich die Anfragen an das Neuronale Netz signifikant von den Eingabe-Ausgabe-Werten aus dem Training unterscheiden. In diesem Fall steigt das Risiko inkorrektur Ausgabewerte, weil

sich das Neuronale Netz „unsicher“ ist.

Dieses Problem kann umgangen werden, indem im Vorfeld bereits die Eingabewerte des Neuronalen Netzes auf Anomalien hin untersucht werden. Damit kann erkannt werden, wenn Eingabewerte signifikant von den Situationen aus der Trainingsphase abweichen und somit ein erhöhtes Risiko inkorrektur Ausgaben vorliegt.

Für die Qualitätssicherung in SOIS kann eine derartige Beobachtung dazu genutzt werden, um zu erkennen, wenn das bisherige Strategiemodell des kognitiven Systems nicht mehr zur Entscheidungsfindung geeignet ist, da sich die zu bewertenden Zustände zu stark von denen aus der Trainingsphase unterscheiden. Das Risiko von inkorrekten Aussagen des Neuronalen Netzes steigt, ohne dass dies vom Neuronalen Netz in Form von Sicherheits- oder Unsicherheitsangaben angezeigt wird.

Nachdem auch in diesem Fall die Herausforderung für die Erkennung von Anomalien in dem enorm großen Zustandsraum liegt, der sich durch den hohen Freiheitsgrad der sich bewegenden Objekte ergibt, und gleichzeitig ein Raumbezug vorliegt, bietet es sich an, das in Abschnitt 3.1 vorgestellte Konzept zur Erkennung und Analyse von Anomalien auch für raumbezogene Schichtmodelle umzusetzen. Dabei gilt es, geeignete Substrukturen zu extrahieren und in einem Verzeichnis abzulegen, um die (Ab-)Normalität von Zustandsbeschreibungen über ihre Rekonstruierbarkeit bewerten zu können.

5.2.1 Verwandte Arbeiten

Bezüglich verwandter Arbeiten für die Erkennung von Anomalien in raumbezogene Daten sei auf den Abschnitt 3.2 verwiesen.

Eine weitere Möglichkeit, um auszuschließen, dass inkorrekte Aussagen von Neuronalen Netzen erzeugt werden, da keine entsprechenden Trainingsdaten vorhanden waren, ist, derartige Unsicherheiten durch die Auswertung von Information, die sich beim Bearbeiten der Anfrage im Neuronalen Netz ergeben, zu erkennen. Im Bereich der Klassifikation von Bildern gibt es zum Beispiel ein Verfahren, bei dem die internen Datenrepräsentationen im Modell – im Speziellen die vorletzte Schicht des Netzes – analysiert werden, um anormale Bilder zu erkennen, die nicht korrekt klassifiziert werden können, da das Neuronale Netz nicht dafür trainiert wurde [Bend 16].

Allerdings funktioniert dieses Verfahren lediglich für Klassifikationsprobleme. Damit ist es nicht anwendbar auf CNN, welche entsprechend des Prinzips des bestärkenden Lernens ein Strategiemodell lernen, da es sich dabei um ein Regressionsproblem handelt.

Es gibt zwar alternative Methoden, die unabhängig von Klassifikation oder Regression die Unsicherheit eines Neuronalen Netzes über den internen Datenverarbeitungsprozess schätzen, diese benötigen allerdings ein spezielles Training. Der Ansatz von Gal und Ghahramani [Gal 16] beziehungsweise auch eine Variante [McCl 16], nutzen das Prinzip des Aussetzens, engl. „dropout“, das aller-

dings bereits zum Training des Neuronalen Netzes angewendet werden muss. Das Gleiche gilt für den Ansatz von Lakshminarayanan et al. [Laks 17], der die Idee von gegnerischem Training, engl. „adversarial training“, und Ensemblelernen, engl. „ensemble learning“, zur Unsicherheitsschätzung nutzt, was allerdings eine spezielles Training voraussetzt.

Insgesamt haben also alle diese Verfahren den großen Nachteil, dass sie nicht bei Regressionsproblemen wie dem Erlernen eines Strategiemodells funktionieren oder, dass ein spezielles Training notwendig ist, wodurch sie nicht auf bereits bestehende Neuronale Netze anwendbar sind.

Dazu kommt, dass derartig erzeugte Aussagen über die Sicherheit oder Unsicherheit nicht nachvollziehbar sind, da der Datenverarbeitungsprozess innerhalb von Neuronalen Netzen für den Menschen nicht ohne enormen Mehraufwand fassbar ist. Aus diesem Grund ist gerade für die Qualitätssicherung in SOIS der Anomalieerkennungsansatz auf den Eingabedaten zu bevorzugen, dessen Entscheidungsprozess auch für den Menschen nachvollziehbar ist.

5.2.2 Räumliche Schichtmodelle

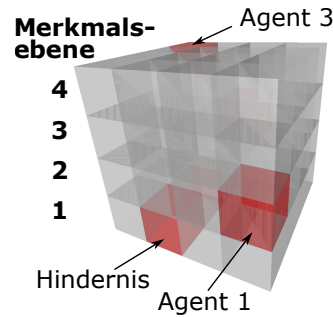
In [Silv 16, Gupt 17, Phan 18] werden räumliche Schichtmodelle zur Repräsentation von Zuständen in Single- und Multi-Agenten-Systemen verwendet. Silva et al. nutzen diese Form der Zustandsdarstellung zum Beispiel im kognitiven System ihres weltbekannten Programms „AlphaGo“, das zum ersten Mal KI-basiert einen menschlichen Go-Spieler schlagen konnte [Silv 16].

Gupta et al. dagegen wendeten räumliche Schichtmodelle zum ersten Mal auch auf ein Multi-Agenten-System an, in denen nun nicht nur die optimale Handlung für einen Agenten gesucht wird, sondern auch die Koordination von mehreren Agenten optimiert werden soll [Gupt 17].

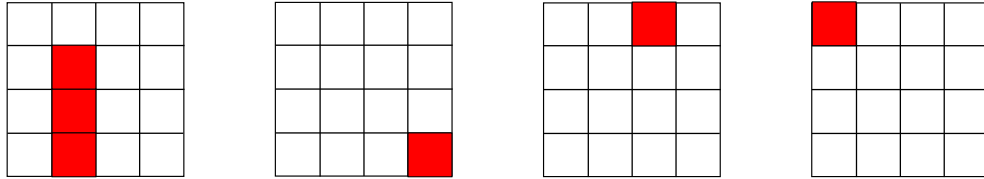
Darauf basierend zeigte Phan et al. in [Phan 18], dass räumliche Schichtmodelle auch im kognitiven System eines SOIS, das in diesem Fall einem Multi-Agenten-System entspricht, verwendet werden können. Wie in Abschnitt 2.1.3.3 bereits erwähnt, ist der große Vorteil dieser Art der Darstellung, dass dadurch die räumlichen Beziehungen zwischen den relevanten Ereignissen festgehalten werden können.

Ein einfaches Beispiel eines solchen Schichtmodells ist in Abbildung 5.7a zu sehen. Es handelt sich dabei um ein Schichtmodell entsprechend des einfachen Multi-Agenten-Szenarios aus [Gupt 17]. Der allgemeine Aufbau des Schichtmodells ist in Tabelle 5.1a beschrieben. Im dargestellten Beispiel bewegen sich drei Agenten (Merkalsebenen 2-4) in einer Umgebung, deren Grundfläche durch eine 4x4 Rasterzellenstruktur beschrieben wird und ein Hindernis (Merkmals-ebene 1) enthält.

In diesem Fall wird eine 1-aus-n-Kodierung, engl. „one-hot encoding“, verwendet. Das heißt, jeder Rasterzelle wird entweder der Wert 0 oder 1 zugewiesen. Dadurch kann die Lage eines Hindernisses darüber beschrieben werden, dass der entsprechenden Rasterzelle der Wert 1 zugewiesen wird (in Abbildung



(a) 3D-Darstellung eines Zustands im einfachen Multi-Agenten-Szenario.



(b) 2D-Ansicht von Merkmalsebene 1 (Lage des Hindernisses). (c) 2D-Ansicht von Merkmalsebene 2 (Position von Agent 1). (d) 2D-Ansicht von Merkmalsebene 3 (Position von Agent 2). (e) 2D-Ansicht von Merkmalsebene 4 (Position von Agent 3).

Abbildung 5.7: 2D- und 3D-Ansichten eines räumlichen Schichtmodells.

5.7a und Abbildung 5.7b rot markiert), wohingegen alle anderen Werte auf der Merkmalsebene 0 bleiben. Das gleiche gilt für die Agenten-Merkmalsebenen: zur Beschreibung der aktuellen Position des Agentens wird der Wert der entsprechende Rasterzelle auf 1 gesetzt.

Im Gegensatz dazu können räumliche Schichtmodelle, die im kognitiven System eines SOIS, in dem mobile Roboter Werkstücke entsprechend deren Produktspezifikationen zu passenden Maschinen fahren, verwendet werden, wesentlich komplexer werden (siehe Tabelle 5.1b, [Phan 18]).

In diesem Fall sind die Zahlenwerte der Rasterzellen nicht beschränkt auf $\{0, 1\}$ sondern können Werte in \mathbb{N} annehmen. Dazu werden die Maschinenanfragen und Agentenzustände lokal aggregiert. Sollten sich also zum Beispiel vier Roboter an ein und derselben Maschine einreihen, wird der Wert der Rasterzelle auf der entsprechenden Merkmalsebene auf 4 gesetzt.

Mit der Erweiterung von $\{0, 1\}$ auf \mathbb{N} erweitert sich damit auch der Zustandsraum, da die Anzahl der möglichen Zustände zunimmt. Daher wird im Folgenden sowohl für das einfache Multi-Agenten-Szenario, als auch für SOIS-Szenarien gezeigt, wie das Konzept aus Abschnitt 3.1 zu Erkennung und Analyse von Anomalien in den raumbezogenen Daten auch für räumliche Schichtmodelle umgesetzt werden kann.

Tabelle 5.1: Beispielstrukturen räumlicher Schichtmodelle.

Ebene	Merkmal	Beschreibung
$(n + 1)$	Agenten	Aktuelle Position von Agent a_n .
\vdots	\vdots	\vdots
2		Aktuelle Position von Agent a_1 .
1	Hindernisse	Lage von Hindernissen, die von den Agenten umfahren werden müssen.

(a) Einfaches Multi-Agenten-Szenario.

Ebene	Merkmal	Beschreibung
$(r + 5)$	Maschinenanfragen	Position (und Anzahl) von Agenten, die Maschine m_r anfragen.
\vdots	\vdots	\vdots
6		Position (und Anzahl) von Agenten, die Maschine m_1 anfragen.
5	Agentenzustände	Position jedes Agenten klassifiziert nach:
4		• <i>inkorrekt eingereicht</i>
3		• <i>korrekt eingereicht</i>
2		• <i>inkorrekt positioniert</i>
1	Maschinen	Position und Typ jeder Maschine.

(b) SOIS-Szenario.

5.2.3 Substrukturextraktion

Für die Umsetzung des Konzepts aus 3.1 für räumliche Schichtmodelle stellt sich zunächst die Frage, was geeignete Substrukturen sind, die aus den Trainingsdaten extrahiert und anschließend in Form eines Substrukturverzeichnis zur Anomalieerkennung eingesetzt werden können.

Im Falle von räumlichen Bewegungsdaten ergaben sich diese Substrukturen durch Zerteilung der Trainingstrajektorien (siehe Abschnitt 3.4 und Abschnitt 4.1.1). Versucht man nun räumliche Schichtmodelle ebenso in geeignete Teilstücke zu zerlegen, ist zu beachten, dass sich letztendlich durch die Schichtung der Merkmalsebenen für jede Rasterzelle ein Merkmalsvektor ergibt.

Da die Reihenfolge der Schichten und damit der Merkmale im Vektor vorgegeben sind, sollte diese Dimension nicht bei der Zerteilung der räumlichen Schichtmodelle berücksichtigt werden. Stattdessen geht es bei der Substrukturextraktion in räumlichen Schichtmodellen darum, anwendungsspezifische Ausprägungen von Merkmalsvektoren und mögliche räumlich Zusammenhänge derselben zu erkennen, die – den hohen Freiheitsgrades des Systems modellierend – zu validen Zuständen zusammengesetzt werden können.

Da weiterhin gilt, dass das sich ergebende Substrukturverzeichnis möglichst *kompakt* und gleichzeitig *effizient* bezüglich der Rekonstruktion von Testdaten sein soll, wird im Folgenden ein Verfahren vorgestellt, das aufbauend auf dem rekursiven Zerteilungsansatz des Pathlet Learning eine effiziente Substrukturextraktion in räumlichen Schichtmodellen ermöglicht (siehe Abschnitt 5.2.3.1).

Außerdem wird ein zusätzlicher Optimierungsschritt eingeführt, der die Verzeichnisgröße nachträglich noch weiter reduziert (siehe Abschnitt 5.2.3.2).

5.2.3.1 Substrukturextraktion mittels RCB-Baum

Die Idee des in Abschnitt 3.4 vorgestellte Pathlet Learning-Ansatzes ist es, für jede Trainingstrajektorie die „optimale“ Zerteilung zu finden, sodass am Ende durch Zusammenfügen aller dieser Teilstücke ein Verzeichnis entsteht, das möglichst wenige Einträge hat, aber trotzdem alle Trainingsdaten repräsentieren kann.

Wie bereits erwähnt, ergibt sich auf Grund der Merkmalsschichtung eine Einschränkung bezüglich der Zerteilung der räumlichen Schichtmodelle, sodass Zerteilungen in horizontaler Ebene ausgeschlossen werden können. Das Problem lässt sich also auf die Zerteilung der Grundfläche der Umgebung reduzieren.

Nichtsdestotrotz ist es durch die zusätzliche Dimension (2-dimensionale Grundfläche) nicht mehr praktikabel, wie im Falle der Trajektorien, die als 1-dimensionale Linien gesehen werden können, alle Möglichkeiten der Zerteilung rekursiv auszutesten, um die gemäß Gleichung 3.2 „optimale“ Lösung zu finden.

Stattdessen muss die Zerteilung der Grundfläche entsprechend einer geeigneten Heuristik erfolgen. Eine sehr einfache und damit auch schnell durchzuführende Partitionierungsheuristik ist die *rekursive Koordinaten-Bisektion*, engl. „recursive coordinate bisection“ (RCB).

Ziel der RCB ist es, eine Domäne rekursiv in Sub-Domänen zu unterteilen [Simo 91]. Dafür wird in jedem Rekursionsschritt die Domäne folgendermaßen in zwei Sub-Domänen unterteilt:

Algorithmus 2: Rekursive Koordinaten-Bisektion (RCB)

- 1 Bestimme Dimension mit längster Ausdehnung
 - 2 Sortiere die Datenpunkte entsprechend der ausgewählten Koordinaten-Richtung
 - 3 Weise je die Hälfte der Punkte einer Sub-Domäne zu
-

Der Algorithmus wird rekursiv gemäß des *Teile-und-herrsche*, engl. „divide and conquer“, Verfahrens ausgeführt, bis keine weitere Unterteilung mehr möglich ist oder ein anderweitig definiertes Abbruchkriterium erfüllt ist. Entsprechend erhält man nach k Rekursionsschritten 2^k Sub-Domänen.

Dieses Partitionierungsverfahren kann nun für die Zerteilung der Grundflächen von räumlichen Schichtmodellen verwendet werden, wobei die Rasterzellen als Datenpunkte aufgefasst werden, die zerteilt werden müssen. Die sich so bei der Ausführung des Algorithmus ergebenden Partitionen (Sub-Domänen) sind mögliche Kandidaten für das Substrukturverzeichnis.

In den Abbildungen 5.8a - 5.8d ist der Ablauf der RCB schematisch für das

einfache Multi-Agenten-Szenario mit einer Grundfläche von 4x4 Rasterzellen dargestellt. Die Richtung der ersten Zerteilung kann frei gewählt werden, da die Ausdehnung in x- und y-Richtung dieselbe ist (jeweils vier Rasterzellen). Im betrachteten Fall erfolgt die Zerteilung in x-Richtung, sodass die beiden Partitionen B und C entstehen (siehe Abbildung 5.8a).

Im nächsten Schritt werden diese beiden Teilflächen erneut unterteilt, diesmal allerdings in y-Richtung, da die Ausdehnung in diese Dimension mit vier Rasterzellen größer ist als in x-Richtung (zwei Rasterzellen). Dadurch ergeben sich die Partitionen D, E, F und G (siehe Abbildung 5.8b). Dieses Vorgehen wird gemäß Algorithmus 2 rekursiv wiederholt, bis keine weitere Zerteilung mehr möglich ist (siehe Abbildung 5.8d).

Für die Bewertung der möglichen Zerteilungen wird der Ablauf der RCB und die sich dabei ergebenden Partitionen als Graph dargestellt (siehe Abbildung 5.8e). Dieser entsteht, indem für jeden Rekursionsschritt zwei Kindknoten im Graphen ergänzt werden, die den zwei neuen Partitionen entsprechen.

Um nun die „optimale“ Zerteilung zu finden, wird jede Kante im Graphen mit einem Gewicht $f(p)$ versehen. Diese Gewichte sind eine rekursive Umformulierung von Gleichung 3.2:

$$f(p) = \begin{cases} \min\left(\frac{1}{|D(p)|}, \lambda + f(p_1) + f(p_2)\right), & \text{wenn } p = p_1 \cup p_2 \\ \frac{1}{|D(p)|}, & \text{sonst,} \end{cases} \quad (5.4)$$

wobei p_1 und p_2 die Kindknoten von Partition p sind und $D(p)$ die Menge an Trainingsdaten, die p ebenfalls enthalten.

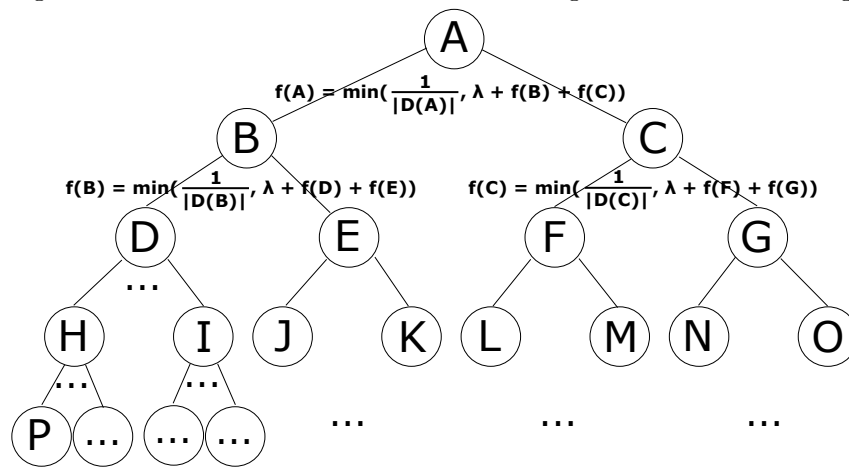
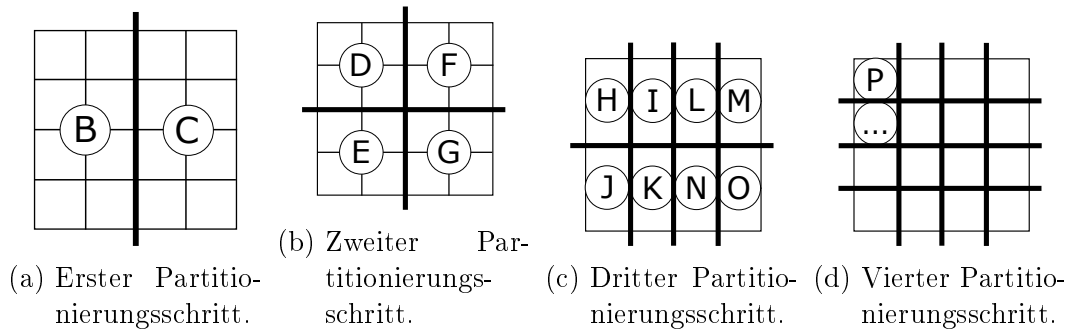
Ausgehend von den Blattknoten wird nun geprüft, ob

$$\lambda + f(p_1) + f(p_2) > f(p) \quad (5.5)$$

gilt. Ist dies der Fall, werden die beiden entsprechenden Partitionen aggregiert und die Prüfung auf dem nächsthöheren Level im Graphen fortgesetzt.

Damit erhält man eine Zerteilung der Grundfläche eines räumlichen Schichtmodells, die die Anforderungen des bedingten Optimierungsproblems für ein *kompaktes*, aber zugleich für die Rekonstruktion von Testdaten *effizientes* Verzeichnis erfüllt (siehe Abschnitt 3.4).

Zwar ist die Anwendung von RCB relativ restriktiv was die Zerteilung der Grundfläche angeht, da dadurch die Menge an Substrukturkandidaten auf die Partitionen limitiert wird, die sich durch bisektive Unterteilung der räumlichen Fläche ergeben und dadurch andere, unter Umständen nützlichere Partitionierungen von vornherein ausgeschlossen werden. Allerdings zeigen die Evaluationen im folgenden Abschnitt 5.2.4, dass RCB trotzdem gut zur Erzeugung von Substrukturverzeichnissen für die Erkennung und Analyse von Anomalien in räumlichen Schichtmodellen verwendet werden kann. Der Vorteil der schnellen und einfachen Berechnung überwiegt hier deutlich die Nachteile der Restriktion.



(e) RCB-Baum.

Abbildung 5.8: Vorgehen zur Zerlegung der Grundfläche eines räumlichen Schichtmodells, sowie der sich daraus ergebende RCB-Baum.

5.2.3.2 Optimierung der Verzeichnisgröße

Wie bereits erwähnt, kann in einem nachgelagerten Optimierungsschritt die Verzeichnisgröße zusätzlich reduziert werden, ohne dass dadurch eines der beiden anderen Evaluationskriterien – *Effizienz* des Rekonstruktionsprozesses oder *Universalität* – beeinträchtigt wird.

Die Idee dahinter ist, Redundanten zu erkennen, das heißt, Einträge zu identifizieren, deren Informationen so auch in anderen Substruktur enthalten sind. Diese wären damit nicht zwingend zur Rekonstruktion erforderlich, solange die diese überdeckenden Substrukturen stattdessen angewendet werden können.

Um herauszufinden, wann dies der Fall ist und tatsächlich redundante Einträge aus dem Verzeichnis entfernt werden können, wird eine Menge von Testdaten mit dem voll umfänglichen Verzeichnis rekonstruiert, wobei Substrukturen, die eine große Grundfläche abdecken, bevorzugt werden.

Das heißt, kann ein räumliches Schichtmodell der Testmenge bereits mit größeren Substrukturen rekonstruiert werden, kommen die kleineren, möglicherweise redundanten Einträge nicht mehr zum Einsatz. Dadurch sinkt sichtbar ihre Nutzerhäufigkeit und sie können entsprechend aus dem Verzeichnis entfernt werden. Dass die Auswirkungen dieses nachgelagerten Optimierungsschrittes auf die Gesamtqualität des Verfahrens vernachlässigbar sind, wird im folgenden Abschnitt 5.2.4 gezeigt.

5.2.4 Erkennung und Analyse von Anomalien

Im folgenden Abschnitt wird der vorgestellte Ansatz zur Erkennung und Analyse von Anomalien in räumlichen Schichtmodellen evaluiert. Dazu werden entsprechende Substrukturverzeichnisse sowohl für ein einfaches Multi-Agenten-Szenario mit 1-aus-n-Kodierung (siehe Abschnitt 5.2.4.1) als auch für komplexere SOIS-Szenarien (siehe Abschnitt 5.2.4.2) erstellt und gemäß der Evaluationskriterien *Universalität*, *Kompaktheit* des Verzeichnisses und *Effizienz* des Rekonstruktionsprozesses bewertet.

Darüber hinaus wird das Prozedere der Erkennung und Analyse von Anomalien beispielhaft für das einfache Multi-Agenten-Szenario mit entsprechenden Testdaten ausgeführt und ausführlich erläutert. Genauso wird das Wirken des Optimierungsschrittes an einem konkreten Beispiel nachvollzogen.

Alle zur Evaluation verwendeten Datensätze wurden synthetisch erzeugt, indem regelbasiert valide Zustandsbeschreibungen erzeugt wurden, ohne dass dafür ein System mit entsprechender kognitiver Komponente ausgeführt werden musste.

5.2.4.1 Einfaches Multi-Agenten-Szenario

In Anlehnung an [Gupt 17] wird der vorgestellte Ansatz zunächst für ein einfaches Multi-Agenten-Szenario umgesetzt (siehe Tabelle 5.2a). Dabei bewegen sich drei mobile Agenten in einer Umgebung, die eine Grundfläche von 10x10

Rasterzellen hat und insgesamt zwei Hindernisse enthält.

Grundsätzlich gilt, dass sich mehrere Agenten gleichzeitig auf derselben Rasterzelle befinden können, wohingegen sich die Position der Agenten zu keiner Zeit mit denen der Hindernisse überschneiden dürfen. Dementsprechend wäre ein Beispiel für einen anormalen Zustand, der durch den Anomalieerkennungsprozess detektiert werden sollte, wenn ein Agent sich auf der Position eines Hindernisses befindet.

Für die Erkennung derartiger anomaler Zustandsbeschreibungen wird ein geeignetes Substrukturverzeichnis benötigt. Dafür müssen zunächst geeignete Werte für die beiden Parameter *Anzahl an Trainingsdaten* und λ bestimmt werden, wofür wie in Abschnitt 3.4 für verschiedene Parameterkombination die entsprechenden Evaluationskriterien ausgewertet werden.

Folgerichtig sind in Abbildung 5.9 die Ergebnisse für die Größe des sich ergebenden Verzeichnisses (Anzahl an enthaltenen Substrukturen), die durchschnittliche Überdeckungsrate bei der Rekonstruktion von 1000 Testdaten und die durchschnittliche Anzahl an Substrukturen, die zur Rekonstruktion eines gegebenen räumlichen Schichtmodells benötigt werden, zu sehen. Damit sind nun Aussagen über die *Kompaktheit* und *Universalität* des Verzeichnisses, sowie die *Effizienz* des Rekonstruktionsprozesses möglich, um schließlich eine geeignete Parametrisierung zu erhalten.

Tabelle 5.9d bietet dabei eine zusätzliche Detailansicht der relevanten Werte der durchschnittlichen Überdeckungsrate und die mit * gekennzeichnete Ergebnisse sind die Auswertungen für die gemäß Abschnitt 5.2.3.2 optimierten Substrukturverzeichnisse.

Aus den Ergebnissen in Abbildung 5.9c und Tabelle 5.9d lässt sich schließen, dass für das vorliegende Szenario 20 Trainingsdaten ausreichen, um eine ausreichend hohe Überdeckungsrate für die Testdaten zu erzielen (0.99965, bzw. 0.99951 bei Verwendung des optimierten Verzeichnisses). Mit weniger Trainingsdaten (zum Beispiel 10 Zustandsbeschreibungen) ergibt sich kein zur Anomalieerkennung dienliches Substrukturverzeichnis (Überdeckungsrate = 0) und ein Mehr an Trainingsdaten (zum Beispiel 30 Zustandsbeschreibungen) erzeugt keine deutliche Steigerung der durchschnittlichen Überdeckungsrate, der den durch mehr Trainingsdaten verursachten erhöhten Rechenaufwand rechtfertigen würde.

Bezüglich λ ist festzustellen, dass die Ergebnisse, die mit $\lambda = 0.01$ erzielt werden, ein guter Kompromiss sind zwischen der *Effizienz* des Rekonstruktionsprozesses (durchschnittlich werden 21 Substrukturen zur Rekonstruktion eines räumlichen Schichtmodells verwendet, siehe Abbildung 5.9b) und der *Kompaktheit* des Verzeichnisses (30 Einträge nach dem Optimierungsschritt, siehe Abbildung 5.9a), während eine ausreichend große durchschnittlichen Überdeckungsrate erzielt wird (0.99951, siehe Tabelle 5.9d).

Interessant ist zu sehen, dass für $\lambda > 0.1$ die durchschnittliche Überdeckungsrate auf Werte von 0 abfällt und das unabhängig von der Anzahl der verwendeten Trainingsdaten. Dies liegt daran, dass $\lambda + f(p_1) + f(p_2)$ (vgl. Gleichung 5.5)

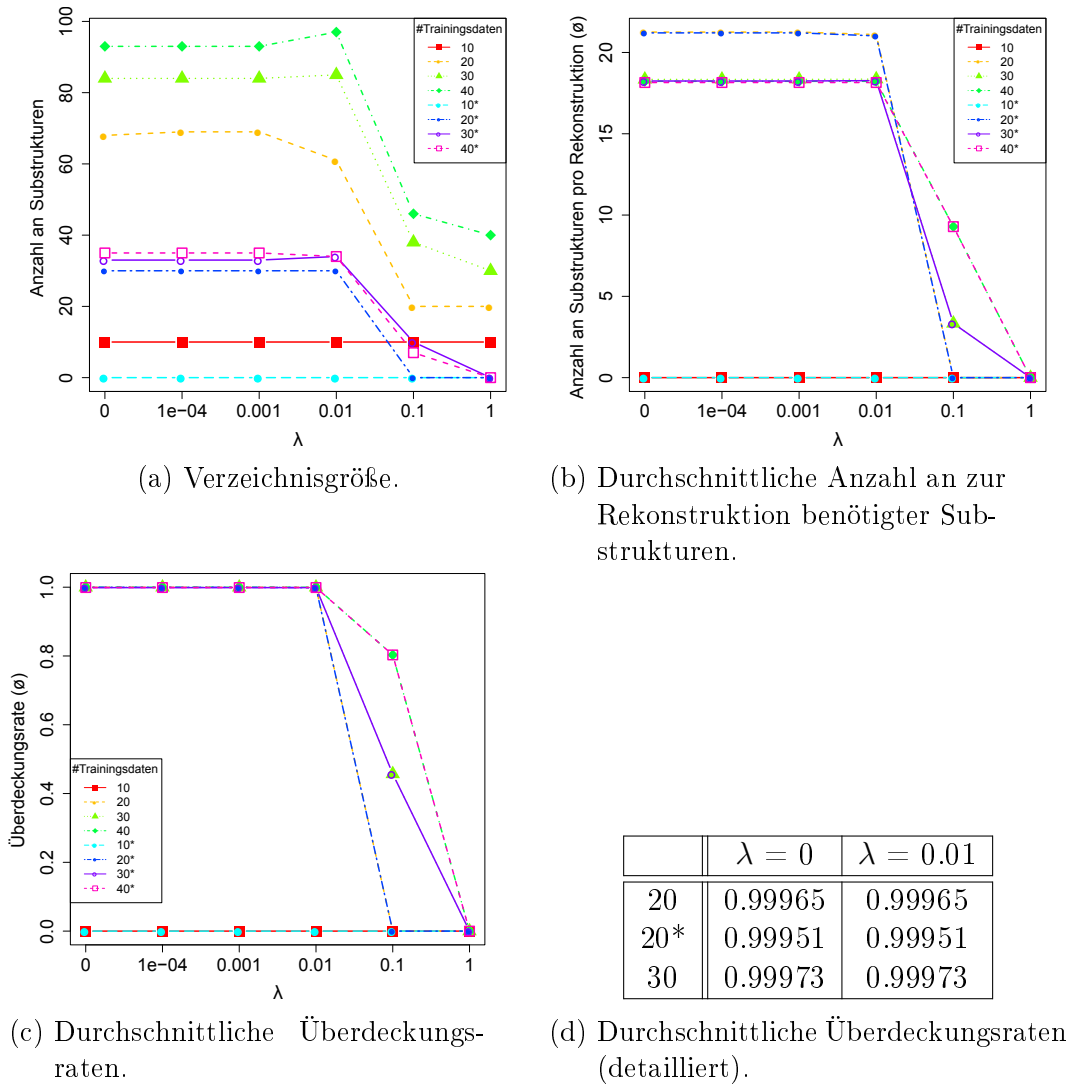


Abbildung 5.9: Evaluationsergebnisse für das einfache Multi-Agenten-Szenario.

auf allen Baumleveln stets größer ist als $f(p)$ und als Konsequenz damit die Knoten des RCB-Baums (die Partition) auf dem Weg von den Blattknoten zur Wurzel immer vollständig zum Ausgangsschichtmodell aggregiert werden. Entsprechend besteht das Substrukturverzeichnis in einem solchen Fall aus allen (nicht zerteilten) Trainingsdaten, aber eben keinen für die Anomalieerkennung nützlichen Einträgen.

Das heißt, unter Verwendung von 20 Trainingsdaten und $\lambda = 0.001$ kann für das vorliegende einfache Multi-Agenten-System ein Substrukturverzeichnis erzeugt werden, das eine ausreichend große *Universalität* bei der Rekonstruktion von 1000 Testzuständen zeigt und zugleich den besten Kompromiss zwischen *Kompaktheit* des Verzeichnisses und *Effizienz* des Rekonstruktionsprozesses darstellt.

Das Wirken des Optimierungsschrittes, wie er in Abschnitt 5.2.3.2 eingeführt wurde, soll im Folgenden anhand der sogenannten Freiraumeinträge illustriert werden. Freiraumeinträge sind dabei Substrukturen, deren Rasterzellen alle den Wert 0 haben. Damit beschreiben sie Flächen, auf denen sich aktuell sowohl kein Hindernis, als auch keine Agenten befinden. An diesen Freiraumeinträgen lassen sich Redundanzen gut erkennen.

Dazu sind in Abbildung 5.10 Freiraumeinträge des ursprünglich mittels RCB-Baum erzeugten Substrukturverzeichnisses visualisiert, wobei die Rasterzellen alle weiß sind, da sie den Wert 0 haben.

Insgesamt werden in diesem Fall durch den Optimierungsschritt vier dieser Freiraum-Substrukturen aus dem Verzeichnis entfernt (siehe Abbildung 5.10a). Sie sind nicht zwingend notwendig für die Rekonstruktion, da es offensichtlich andere Substrukturen im Verzeichnis gibt, die diese überdecken und stattdessen eingesetzt werden können (siehe Abbildung 5.10b).

So beschreibt zum Beispiel die dritte Substruktur in Abbildung 5.10a eine Freifläche von 1x2 Rasterzellen. Dieses Struktur wurde im Optimierungsschritt eliminiert, da die Freiflächen, die sich in den räumlichen Schichtmodellen der Testdaten ergeben, genauso durch Verzeichniseinträge rekonstruiert werden können, die eine größere Grundfläche abdecken (zum Beispiel die dritte Substruktur des optimierten Verzeichnisses in Abbildung 5.10b, die eine Grundfläche von 1x3 abdeckt).

Konkret reduziert der nachgelagerte Optimierungsschritt die Verzeichnisgröße im Falle von 20 Trainingsdaten und $\lambda = 0.01$ von 61 auf 30 Einträge (siehe Abbildung 5.9a). Auch bei größeren Mengen an Trainingsdaten ergibt sich eine ähnliche Reduktionen: Im Durchschnitt lässt sich die Verzeichnisgröße um 40% reduzieren, wobei die Werte für die durchschnittlichen Überdeckungsrate und damit die *Universalität* dadurch nicht negativ beeinflusst werden (siehe Abbildung 5.9a und Abbildung 5.9c).

Das heißt, dieser Optimierungsschritt kann verwendet werden, um ein mittels RCB-Baum erzeugtes Substrukturverzeichnis noch kompakter zu machen, ohne dass die Qualität des Verzeichnisses darunter leidet.

Zur Illustration des Prozederes zur Erkennung und Analyse von Anomalien in

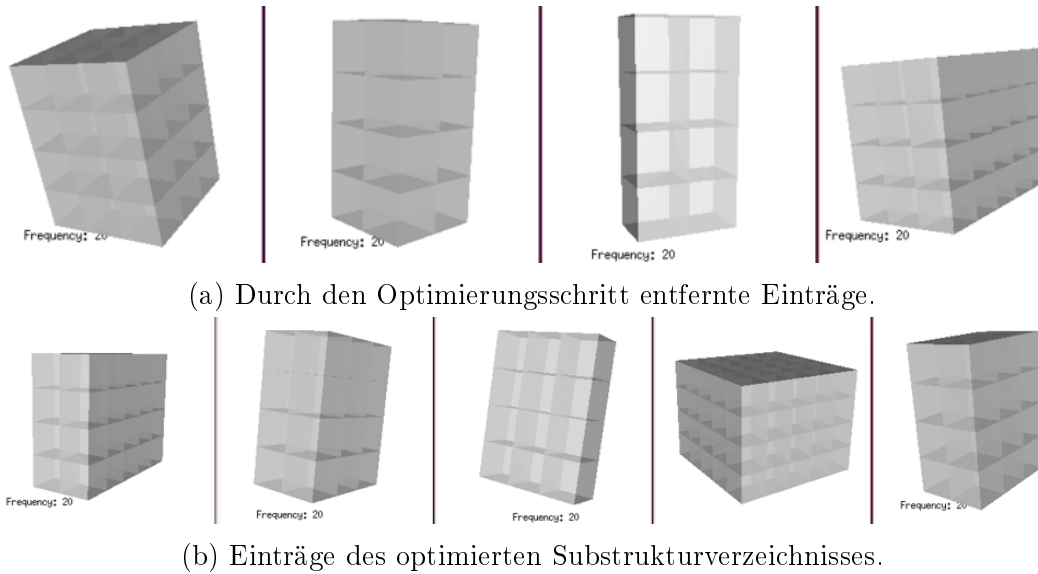


Abbildung 5.10: Freiraum-Verzeichniseinträge im einfachen Multi-Agenten-Szenario.

räumlichen Schichtmodellen wurden zwei verschiedene Testdatenmengen erzeugt. Für die erste Testmenge wurden den Rasterzellen im Schichtmodellen zufällig Werte von $\{0, 1\}$ zugewiesen; für die zweite Menge an Testdaten wurden den Rasterzellen zufällig Werte von $\{0, 1, 2\}$ zugewiesen. Damit ergeben sich auch anormale Zustände, die mit dem vorgestellten Ansatz detektiert werden können.

In Abbildung 5.11 sind beispielhaft nun zwei dieser Testzustände zusammen mit ihrem Ergebnis des Anomalieerkennungprozesses visualisiert. Dafür werden Rasterzellen mit Wert 0 wie gehabt weiß dargestellt. In Abbildung 5.11a und Abbildung 5.11b sind Rasterzellen mit Wert 1 türkis eingefärbt, wohingegen in Abbildung 5.11c und Abbildung 5.11d Rasterzellen mit Wert 1 grün und Rasterzellen mit Wert 2 dunkelblau eingefärbt sind.

Abbildung 5.11b zeigt die nicht rekonstruierbaren Teile des Schichtmodells für das erste Beispiel. Eine visuelle Analyse dieser Teilstücke bestätigt die Anomalie der untersuchten Zustandsbeschreibung, da die mit einem Rasterzellenwert von 1 auf Merkmalsebene 4 angegebene Position des dritten Agenten gleichzeitig die Position eines Hindernisses (Rasterzellenwert von 1 auf Merkmalsebene 1) ist.

Die nicht rekonstruierbaren Teile des zweiten Beispiels sind in Abbildung 5.11d zu sehen. Auch hier können die detektierten Anomalien visuell analysiert werden. Die Färbungen zeigen auf verschiedenen Merkmalsebenen Rasterzellenwerte von 2 an (dunkelblau). Allerdings sind auf Grund der 1-aus-n-Kodierung keine Rasterzellenwerte von 2 vorgesehen. Das heißt, alle Merkmalsvektoren, die einen Rasterzellenwert von 2 haben sind nicht rekonstruierbar und damit das korrekte Ergebnis der Anomalieerkennung.

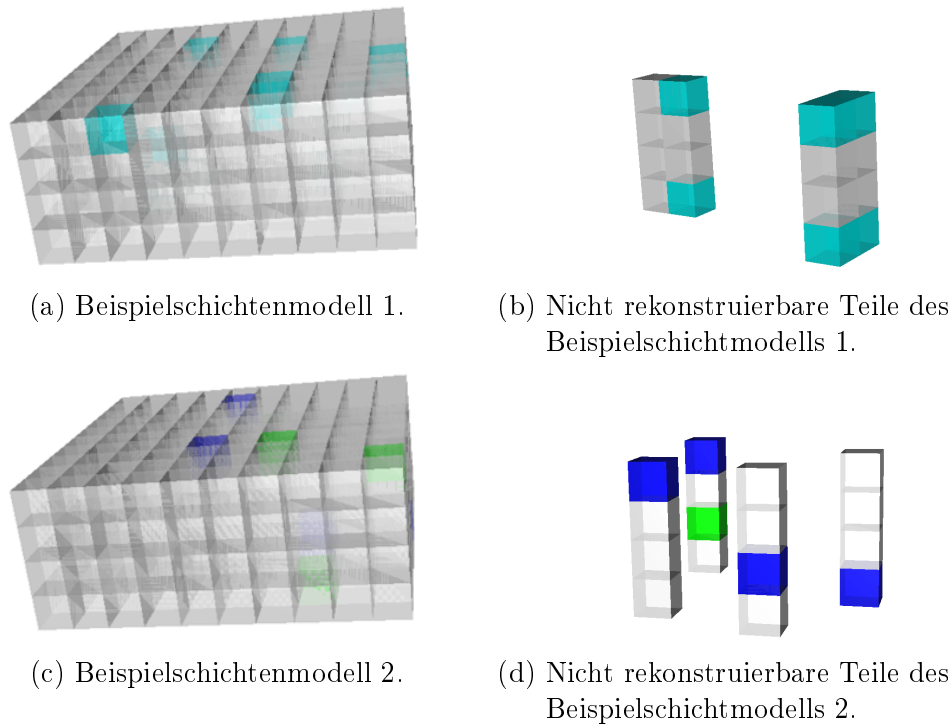


Abbildung 5.11: Analyse von detektierten Anomalien.

Am Beispiel des einfachen Multi-Agenten-Szenarios lässt sich zudem sehr schön zeigen, dass die Verwendung derartigen Substrukturen nicht nur deswegen sinnvoll ist, da damit anwendungsspezifisch räumliche Zusammenhänge verschiedener Merkmale abgebildet werden können, sondern auch aus Effizienzgründen. Dazu kann die ungünstigste, engl. „worst case“, Laufzeit für die Rekonstruktion vom vorgestellten Verfahren mit dem eines naiven Ansatzes, der jeden Merkmalsvektor einzeln auf Validität überprüft, verglichen werden.

Aus den Evaluationsergebnissen lässt sich für das mit 20 Trainingsdaten und $\lambda = 0.001$ erstellte Substrukturverzeichnis ablesen, dass zur Rekonstruktion eines gegebenen räumlichen Schichtmodells durchschnittlich 18-22 Substrukturen benötigt werden (siehe Abbildung 5.12b), wobei das Verzeichnis 30 Einträge enthält (siehe Abbildung 5.9a). Das heißt, im ungünstigsten Fall wenn man zur Rekonstruktion eines Testzustandes alle im Verzeichnis enthaltenen Einträge ausprobieren muss, sind gemäß der Formel

$$\text{Verzeichnisgröße} * \text{Anzahl an Ersetzungen} = 30 * 20 = 600 \text{ Versuche}$$

für die Rekonstruktion nötig. Im Gegensatz dazu bräuchte man beim naiven Ansatz, der nicht anwendungsspezifisch zusammenhängende Merkmalsvektoren, sondern jeden einzelnen der in diesem Fall (10x10 Rasterzellen =) 100

Tabelle 5.3: Valide Merkmalsvektoren für das einfache Multi-Agenten-System.

Beschreibung	Mermals-ebene 1	Mermals-ebene 2	Mermals-ebene 3	Mermals-ebene 4
Freiraum	0	0	0	0
Hindernis	1	0	0	0
nur Agent 1	0	1	0	0
nur Agent 2	0	0	1	0
nur Agent 3	0	0	0	1
Agent 1 und Agent 2	0	1	1	0
Agent 1 und Agent 3	0	1	0	1
Agent 2 und Agent	0	0	1	1

Merkmalsvektoren auf seine Validität überprüft insgesamt

$$\text{Verzeichnisgröße} * \text{Anzahl an Ersetzungen} = 8 * 100 = 800 \text{ Versuche,}$$

wobei die acht validen Merkmalsvektoren für dieses Szenario in Tabelle 5.3 aufgeführt sind.

Das heißt, die Verwendung des Substrukturen-Konzeptes zur Anomalieerkennung bei räumlichen Schichtmodellen hat einen echten Effizienzvorteil gegenüber einer naiven Einzelprüfung der Merkmalsvektoren, da die im schlechtesten Fall zu erwartende Anzahl an Abgleichen und damit die Laufzeitkomplexität reduziert werden kann.

Insgesamt konnte in diesem Abschnitt gezeigt werden, dass durch geeignete Parameterwahl Substrukturverzeichnisse mittels RCB-Baum erzeugt werden können, die die Evaluationskriterien der *Universalität*, *Kompaktheit* und *Effizienz* erfüllen. Damit können in einem derartigen Multi-Agenten-Szenario Anomalien gemäß des in Abschnitt 3.1 vorgestellten Konzeptes in räumlichen Schichtmodellen detektiert und anschließend auch analysiert werden.

Die Analyse erfolgt dabei über eine geeignete Visualisierung der nicht rekonstruierbaren Teilstücke des Schichtmodelles. Darüber hinaus konnte das Wirken des Optimierungsschrittes am Beispiel von Freiraumeinträgen nachvollzogen werden und ein anschaulicher Laufzeitvergleich gegenüber einer einfachen Merkmalsvektorvalidierung ausgeführt werden.

5.2.4.2 SOIS-Szenarien

In diesem Abschnitt wird die Anwendbarkeit des vorgestellten Ansatzes auf komplexere Multi-Agenten-Systeme wie SOIS gezeigt. Dafür wird die Multi-Agenten-Architektur für SOIS aus [Phan 18] übernommen. Die dabei verwendete Struktur der räumlichen Schichtmodelle zur Zustandsbeschreibung im kognitiven System ist in Tabelle 5.1b zu sehen und wurde in Abschnitt 5.2.2 bereits eingeführt.

Tabelle 5.4: SOIS-Szenarien

Szenario	Grundfläche	# Maschinen	# Agenten
1 (Basis)	10x10	5	8
2	10x10	5	16
3	10x10	10	8
4	20x20	5	8

Insgesamt wurde die in Abschnitt 5.2.3 vorgestellte Substrukturextraktion mittels RCB-Baum für vier verschiedene derartige SOIS-Szenarien evaluiert. Der Aufbau der einzelnen Szenarien ist in Tabelle 5.4 zu sehen.

Das Basisszenario ist dabei ein SOIS, dessen Grundfläche von 10x10 Rasterzellen beschrieben wird, auf denen fünf Maschinen platziert sind. Zusätzlich befinden sich insgesamt acht Agenten im System, die Aufgaben abarbeiten müssen, indem sie geeignete Maschinen anfahren.

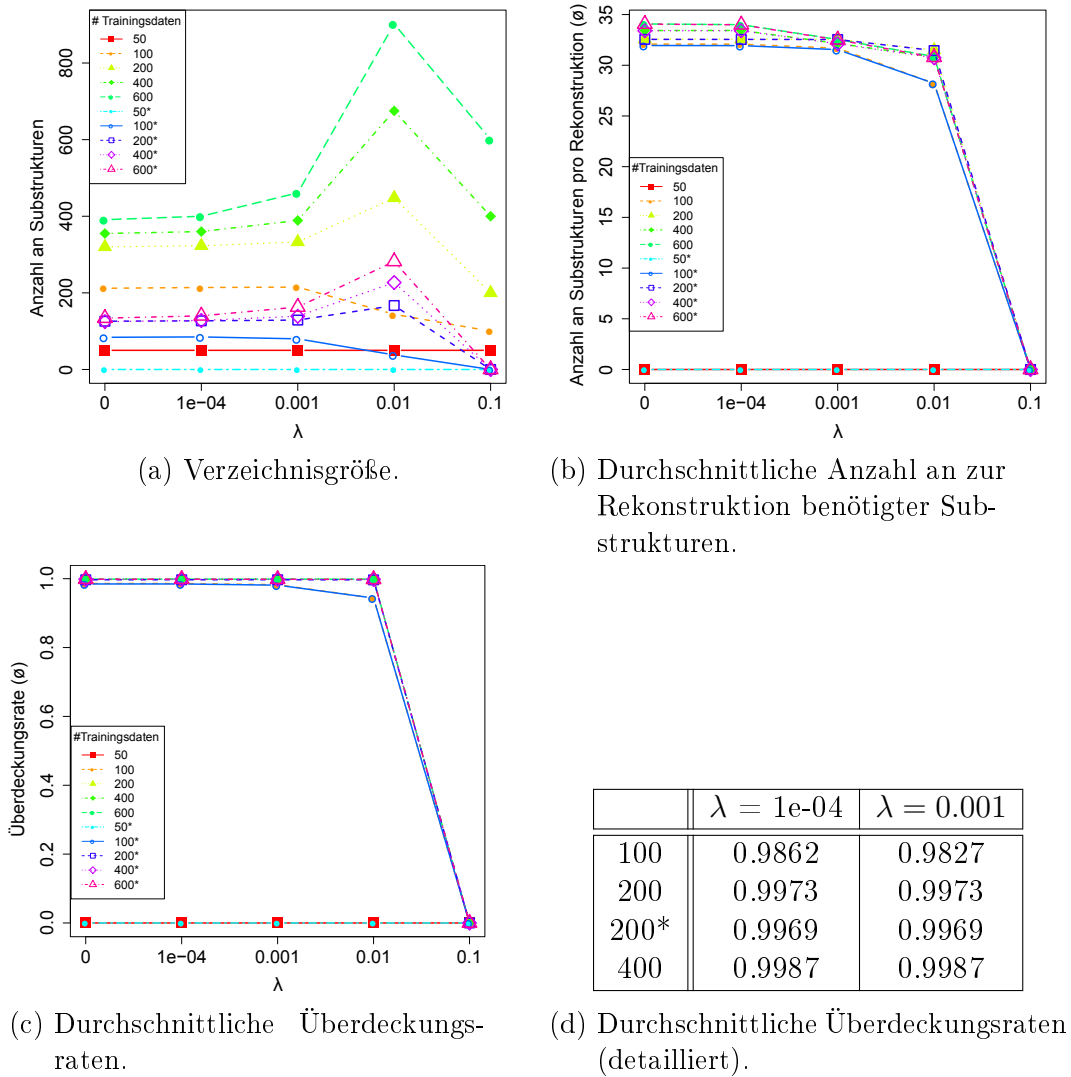
Analog zu den Experimenten bezüglich des einfachen Multi-Agenten-Szenarios in Abschnitt 5.2.4.1 werden im Folgenden zunächst für alle vier Szenarien für verschiedene Parametrisierungen Substrukturverzeichnisse erzeugt. Anschließend werden die relevanten Größen zur Bewertung der *Universalität* und *Kompaktheit* des erzeugten Verzeichnisses, sowie der *Effizienz* des Rekonstruktionsprozesses systematisch ausgewertet, um die Eignung der Verzeichnisse zur Anomalieerkennung zu prüfen.

Dementsprechend sind für das Basisszenario in Abbildung 5.12 die Verzeichnisgröße, durchschnittliche Anzahl an zur Rekonstruktion benötigten Substrukturen und durchschnittliche Überdeckungsrate für die Verzeichnisse zu sehen, die sich mit # Trainingsdaten = {100, 200, 300, 400} und $\lambda = \{0, 0.0001, 0.001, 0.01, 1\}$ ergeben, wobei in Tabelle 5.12d zusätzlich die durchschnittlichen Überdeckungsraten relevanter Parametrisierungen im Detail aufgeführt sind.

Die Ergebnisse zeigen, dass mit 200 Trainingsdaten und $\lambda = 0.001$ ein guter Kompromiss zwischen Verzeichnisgröße (333, bzw. 129 bei Verwendung des optimierten Verzeichnisses, siehe Abbildung 5.12a) und Effizienz des Rekonstruktionsprozesses (32.57, bzw. 32.54 bei Verwendung des optimierten Verzeichnisses, siehe Abbildung 5.12b) erzielt und dabei gleichzeitig eine ausreichend hohe (>0.99) durchschnittliche Überdeckungsrate erreicht werden kann (0.9973, bzw. 0.9969 bei Verwendung des optimierten Verzeichnisses, siehe Tabelle 5.12d).

Offensichtlich hat auch hier der nachträgliche Optimierungsschritt (siehe Abschnitt 5.2.3.2) keine signifikanten Negativauswirkungen auf die Qualität des erzeugten Substrukturverzeichnisses und kann daher auch an dieser Stelle zur weiteren Effizienzsteigerung eingesetzt werden.

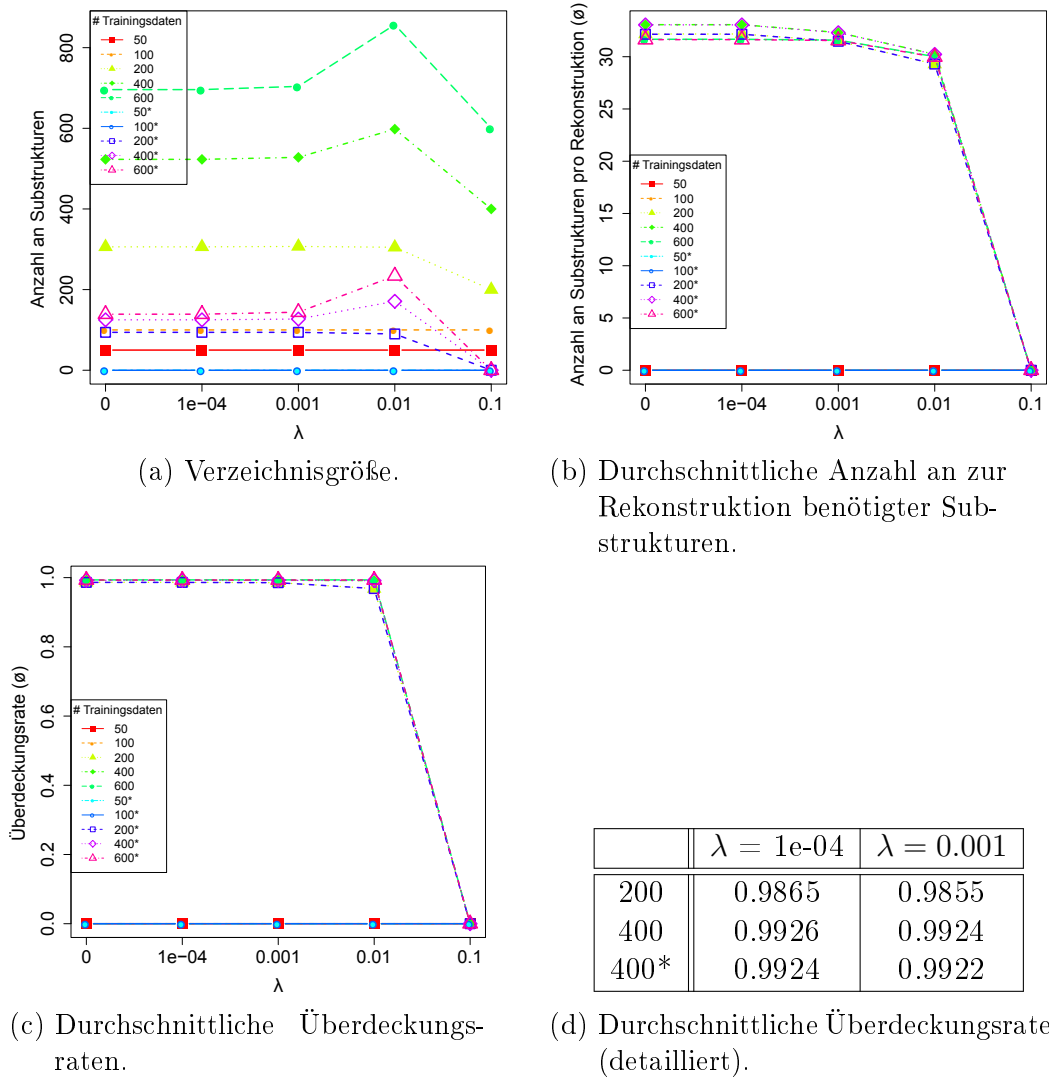
Im zweiten Szenario wurde ausgehend vom Basisszenario die Anzahl an Agenten von acht auf 16 verdoppelt (siehe Tabelle 5.4). Die relevanten Ergebnisgrößen sind in Abbildung 5.13 aufgeführt.



	$\lambda = 1e-04$	$\lambda = 0.001$
100	0.9862	0.9827
200	0.9973	0.9973
200*	0.9969	0.9969
400	0.9987	0.9987

(d) Durchschnittliche Überdeckungs-raten (detailliert).

Abbildung 5.12: Evaluationsergebnisse für das SOIS-Basisszenario.



	$\lambda = 1e-04$	$\lambda = 0.001$
200	0.9865	0.9855
400	0.9926	0.9924
400*	0.9924	0.9922

Abbildung 5.13: Evaluationsergebnisse für das SOIS-Szenario 2.

In diesem Fall muss die Anzahl an Trainingsdaten von 200 auf 400 erhöht werden, um Überdeckungsraten >0.99 zu erzielen (0.9924, bzw. 0.9922 bei Verwendung des optimierten Verzeichnisses, siehe Tabelle 5.13d). Der Grund dafür ist, dass durch die Erhöhung der Anzahl der Agenten der Zustandsraum (Anzahl an möglichen Zuständen) steigt und damit auch mehr Substrukturen zur Normalitätsmodellierung benötigt werden. So kann sich zum Beispiel durch ein Mehr an Agenten der Wertebereich der Rasterzellenwerte, die die Maschinenanfragen beziffern (Merkmalsebenen ≥ 6 , siehe Tabelle 5.1b), nach oben hin vergrößern, da sich vermutlich längere Warteschlangen an den Maschinen bilden, deren Kapazitäten nicht mit der steigenden Anzahl an Agenten gewachsen sind.

An dieser Stelle kommt der Unterschied zwischen räumlichen Schichtmodellen mit 1-aus-n-Kodierung und denen ohne Einschränkung des Wertebereichs zum Tragen. Im letzten Fall ergeben sich wesentlich mehr unterschiedliche Ausprägungen der Merkmalsvektoren, die jeweils als eigene Substrukturen im Verzeichnis hinterlegt werden müssen, um darüber das anwendungsspezifische Normalverhalten des Systems abzubilden.

Dies sind zum Beispiel eben jene Beobachtungen bezüglich der Maschinenauslastung. Das heißt, mit einem Verzeichnis, das für ein SOIS trainiert wurde, bei dem die Anfragewerte an die Maschinen tendenziell gering sind, kann bei der Laufzeitüberwachung mit Hilfe von Substrukturverzeichnissen erkannt werden, wenn auf einmal die Maschinen vermehrt angefragt werden. Dies kann entweder der Fall sein, wenn anormal viel Agenten in der Fabrik sind, diese anormale Aufträge abarbeiten müssen oder auch wenn eine Maschine ausfällt, sodass die Anfragen an alternative Maschinen anormal steigen.

Im Gegensatz zur Anzahl der Trainingsdaten muss der Wert von λ nicht angepasst werden, sondern kann wie im Basisszenario auf 0.001 gesetzt werden. Damit ergibt sich ein guter Kompromiss für eine *kompakte* Verzeichnisgröße (528, bzw. 127 bei Verwendung des optimierten Verzeichnisses, siehe Abbildung 5.13a) und gleichzeitig einen *effizienten* Rekonstruktionsprozess (32.32, bzw. 32.30 bei Verwendung des optimierten Verzeichnisses, siehe Abbildung 5.13b).

Im dritten Szenario ist ausgehend vom Basisszenario die Anzahl an Maschinen von fünf auf zehn verdoppelt worden. Die entsprechenden Evaluationsergebnisse für verschiedene Parametrisierung sind in Abbildung 5.14 zu sehen.

In diesem Fall muss die Anzahl an Trainingsdaten wesentlich stärker, nämlich von 200 auf 4000, angehoben werden, um weiterhin eine ausreichend hohe durchschnittliche Überdeckungsrate zu erzielen (0.9914, bzw. 0.9910 bei Verwendung des optimierten Verzeichnisses, siehe Tabelle 5.14d). Eine derart signifikante Änderung bezüglich der Menge an Trainingsdaten wird dadurch bedingt, dass durch das Verdoppeln der Anzahl an Maschinen die Anzahl an Merkmalsebenen – in diesem Fall von zehn auf 15 – steigt (siehe Tabelle 5.1b). Dadurch vergrößert sich der Zustandsraum (Anzahl an möglichen Zuständen), weswegen dann konsequenterweise auch die Anzahl an Trainingsdaten erhöht

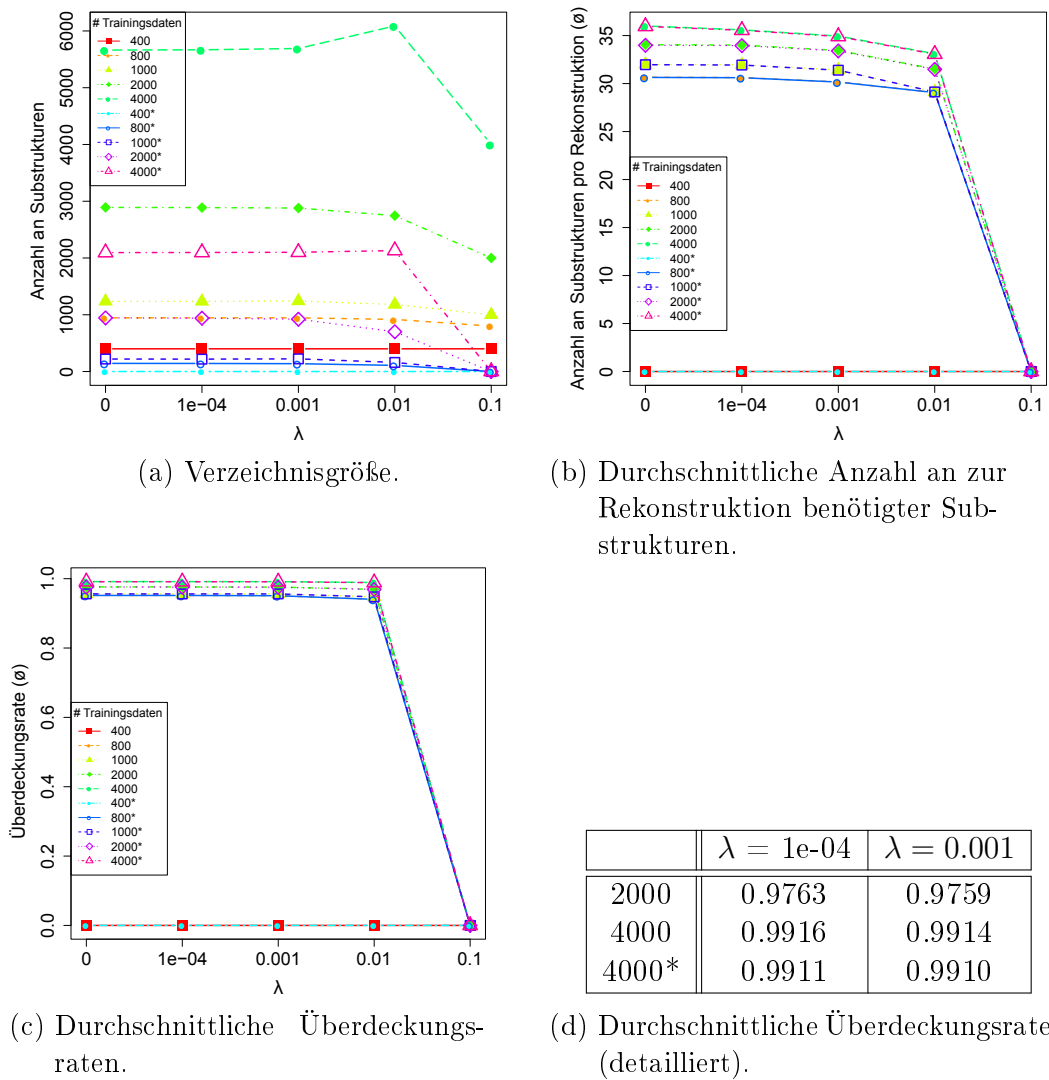


Abbildung 5.14: Evaluationsergebnisse für das SOIS-Szenario 3.

werden muss, um weiterhin das Normalverhalten der Agenten modellieren zu können.

Im Vergleich zum vorhergehenden zweiten Szenario fällt die Steigerung so deutlich aus, da es sich hier nicht nur um eine Erweiterung bezüglich des Wertebereichs von Rasterzellenbelegungen handelt, sondern die Merkmalsvektoren um weitere Dimensionen erweitert werden, deren normale Belegung zusammen mit den räumlichen Verbunden zusätzlich gelernt werden müssen.

Unabhängig davon kann λ unverändert auf 0.001 gesetzt werden, um einen guten Mittelweg zwischen geringer Verzeichnisgröße (5691, bzw. 2101 bei Verwendung des optimierten Verzeichnisses, siehe Abbildung 5.14a) bei gleichzeitig effizienter Rekonstruktion (35.6, bzw. 34.0 bei Verwendung des optimierten Verzeichnisses, siehe Abbildung 5.14b) sicherzustellen.

Im vierten Szenario wurde die Grundfläche des SOIS vervierfacht, sodass sie statt von $10 \times 10 = 100$ von $20 \times 20 = 400$ Rasterzellen repräsentiert wird. Die entsprechenden Ergebnisse sind in Abbildung 5.15 zu sehen.

In diesem Fall kann die Anzahl an Trainingsdaten im Vergleich zum Basis-szenario sogar von 200 auf 100 reduziert und trotzdem eine ausreichend hohe durchschnittliche Überdeckungsrate erzielt werden (0.9956 bzw. 0.9955 bei Verwendung des optimierten Verzeichnisses, siehe Tabelle 5.15d).

λ kann unverändert bei 0.001 bleiben, um die Evaluationskriterien *Kompaktheit* (250, bzw. 112 bei Verwendung des optimierten Verzeichnisses, siehe Abbildung 5.15a) und *Effizienz* (38.85, bzw. 38.8 bei Verwendung des optimierten Verzeichnisses, siehe Abbildung 5.15b) auszugleichen.

Eine interessante Beobachtung an dieser Stelle ist, dass in diesem Szenario die durchschnittliche Anzahl an Substrukturen, die zur Rekonstruktion eines Testzustandes benötigt werden, im Vergleich zu den Werten aus dem Basis-szenario lediglich relativ gering steigt (von 32.54 auf 38.79 bei Verwendung des optimierten Verzeichnisses, siehe Abbildung 5.15b), obwohl sich die zu rekonstruierende Grundfläche vervierfacht. Gleichzeitig enthält das Verzeichnis weniger Einträge als beim 10×10 -Szenario (112 statt 129 bei Verwendung der optimierten Verzeichnisse, siehe Abbildung 5.15a).

Beide Beobachtungen lassen den Schluss zu, dass die Substrukturen im Bezug auf die von ihnen abgedeckte Grundfläche größer geworden sind. Das ist möglich, da die Anzahl an Maschinen und Agenten im SOIS konstant bei fünf beziehungsweise acht geblieben ist. Lediglich der Raum zwischen ihnen wird größer, da sie über eine größere Fläche verstreut sind. Dementsprechend werden weniger, aber dafür größere Substrukturen gelernt, die entweder reine Freiräume oder Agenten-/Hindernispositionen mit Freiraum außen herum sind, wodurch die Anzahl an für die Rekonstruktion benötigten Verzeichniseinträge relativ konstant bleibt.

Damit bestätigt sich, dass tatsächlich anwendungsspezifisch Substrukturen extrahiert werden und nicht beliebige Teilstücke der räumlichen Schichtmodelle. Ansonsten würde die Anzahl an Substrukturen linear mit der Grundfläche der räumlichen Schichtmodelle steigen.

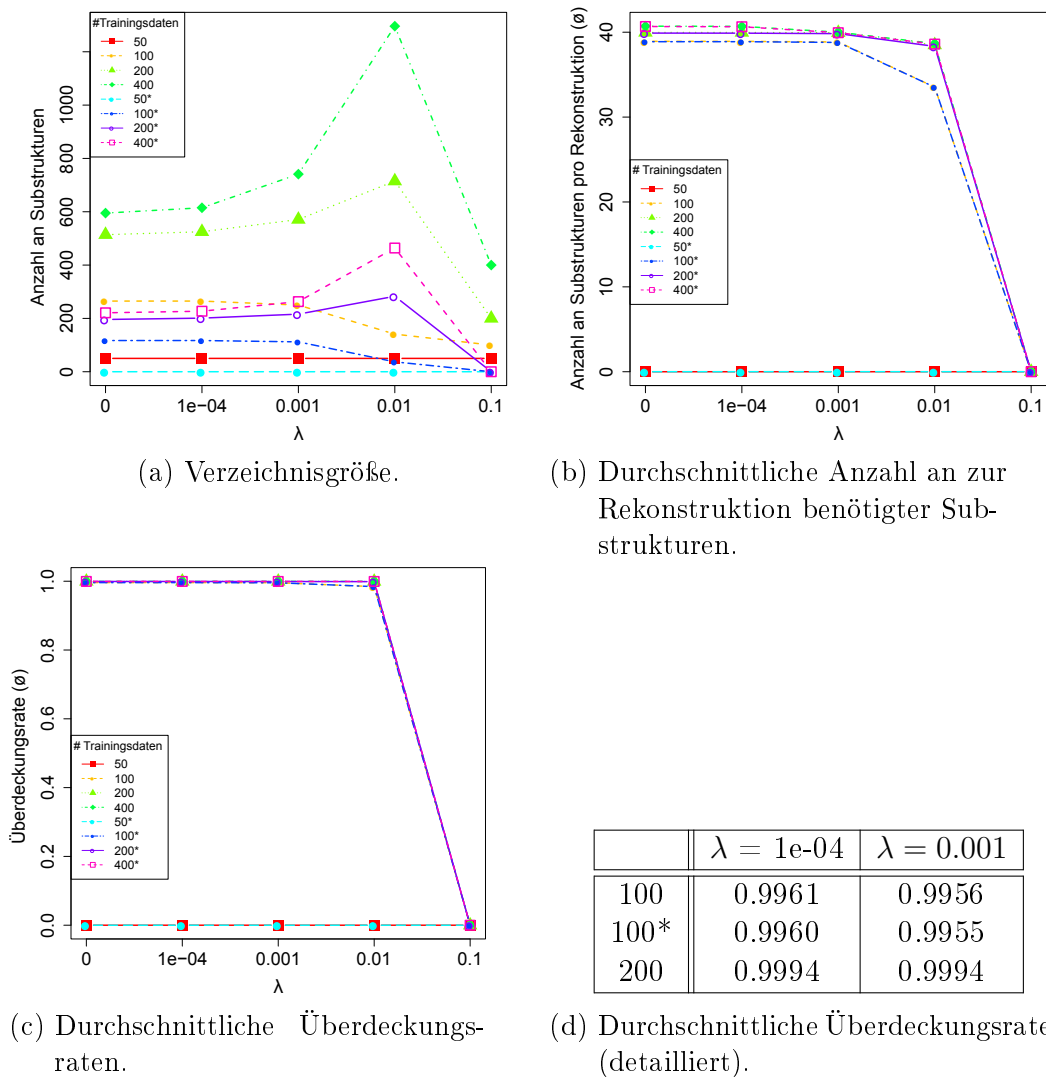


Abbildung 5.15: Evaluationsergebnisse für das SOIS-Szenario 4.

Zusätzlich kann auch hier festgehalten werden, dass unabhängig vom vorliegenden Szenario der in Abschnitt 5.2.3.2 vorgestellte Optimierungsschritt deutlich die Verzeichnisgröße reduziert ohne weitere Einbußen bei den Qualitätskriterien *Effizienz* oder *Universalität* zu verursachen. Im Durchschnitt kann damit nachträglich die Anzahl der Substrukturen im Verzeichnis um rund 40% gesenkt werden, während die durchschnittliche Überdeckungsrate nahezu unverändert bleibt (siehe Abbildungen 5.12-5.15).

Das heißt, dieser Optimierungsschritt kann auch in komplexeren Szenarien, wie einem SOIS, verwendet werden, um den Anomalieerkennungsprozess effizienter zu gestalten.

Insgesamt zeigen diese Experimente, dass der vorgestellte Ansatz zur Anomalieerkennung auch bei komplexeren räumlichen Schichtmodellen, wie sie sich in einem SOIS ergeben, anwendbar ist. Für jedes der vier Szenarien können geeignete Substrukturverzeichnisse erzeugt werden. Dafür wurden die entsprechenden Evaluationskriterien ausgewertet, um passende Parametrisierungen zu finden. Die richtige Anzahl an Trainingsdaten und ein korrekt gesetztes λ ergeben dann ein *universelles*, aber gleichzeitig *kompaktes* Verzeichnis, das *effizient* für die Anomalieerkennung eingesetzt werden kann.

5.2.5 Zusammenfassung

In diesem Abschnitt wurde das Konzept aus Kapitel 3 zur Erkennung und Analyse von Anomalien in SOIS für räumliche Schichtmodelle umgesetzt. Dazu wurde zunächst die Struktur dieser Zustandsbeschreibungen und entsprechende Anwendungen eingeführt (siehe Abschnitt 5.2.2).

Für die Umsetzung des Konzeptes aus Kapitel 3 wurde anschließend in Abschnitt 5.2.3.1 ausgehend von den Ansätzen des Pathlet Learning ein Substrukturextraktionsverfahren für räumliche Schichtmodellen vorgestellt.

Dabei wird der Zerteilungsprozess der Grundfläche des Schichtmodells gemäß Rekursiver Koordinaten-Bisektion in einem RCB-Baum abgebildet, dessen Kanten mit Gewichten gemäß der Approximation des bedingten Optimierungsproblems des Pathlet Learning-Ansatzes (siehe Gleichung 3.2) versehen werden. Die Substrukturen ergeben sich dann aus Zusammenfügen der entstandenen Partitionen, nachdem im RCB-Baum ausgehend von Blattknoten anhand der Gewichte die Einhaltung der Anforderung Kompaktheit und Effizienz geprüft wurden.

Darüber hinaus wird in Abschnitt 5.2.3.2 ein zusätzlicher Optimierungsschritt eingeführt, der Redundanzen in dem mittels RCB-Baum erzeugten Substrukturverzeichnis identifiziert und eliminiert.

Zur Evaluierung dieser Umsetzung des Konzepts aus Kapitel 3 für räumliche Schichtmodelle wurden in Abschnitt 5.2.4 Substrukturverzeichnisse für ein einfaches Multi-Agenten-System, sowie für vier SOIS-Szenarien erzeugt und ihre Eignung zur Anomalieerkennung entsprechend der Qualitätskriterien *Universalität* der Substrukturen, *Kompaktheit* des Verzeichnisses und *Effizienz* des

Rekonstruktionsprozesses bewertet.

Beim einfachen Multi-Agenten-System wurde darüber hinaus der Gesamtprozess der Erkennung und Analyse von Anomalien mit dem erzeugten Substrukturverzeichnis ausgeführt und die Ergebnisse umfänglich diskutiert.

Auch das Wirken des Optimierungsschrittes konnte am Beispiel von Freiraumeinträgen sehr anschaulich illustriert werden.

Auf Grund seiner einfachen Struktur konnte beim einfachen Multi-Agenten-System zudem ein anschaulicher Laufzeitvergleich des Substruktur-Ansatzes gegenüber einer einfachen Merkmalsvektor-Validierung ausgeführt werden, der den Effizienzvorteil des Konzepts aus Kapitel 3 bei der Erkennung von Anomalien in raumbezogenen Daten bestätigt.

Bei den SOIS-Szenarien konnten zudem interessante Beobachtungen gemacht werden, wie die Parameter auf das jeweilige Szenario abgestimmt werden müssen. Bei einer Verdoppelung der Anzahl der Agenten zum Beispiel muss die Anzahl an Trainingsdaten leicht erhöht werden, da sich der Wertebereich der Rasterzellenwerte, die die Warteschlange einer Maschine abbildet, unter Umständen erhöht.

Im Falle einer Verdoppelung der Anzahl der Maschinen muss dagegen die Anzahl an Trainingsdaten deutlich stärker erhöht werden, da der Zustandsraum durch zusätzliche Merkmalsebenen wesentlich mehr wächst als im vorangegebenen Fall.

Für das letzten Szenario wurde schließlich die Grundfläche vervierfacht, wodurch die Anzahl an Trainingsdaten sogar leicht reduziert werden konnte und gleichzeitig die durchschnittliche Anzahl an Substrukturen pro Rekonstruktion nur geringfügig zunahm. Dass die Anzahl der Substrukturen nicht linear mit der Grundfläche der räumliche Schichtmodelle steigt, bestätigt, dass tatsächlich anwendungsspezifische Substrukturen extrahiert werden, die bei mehr Freifläche einfach größere Substrukturen lernen, die entweder reine Freiräume oder Agenten-/Hindernispositionen mit Freiraum darum herum sind.

Insgesamt konnte damit in diesem Abschnitt also gezeigt werden, dass mit Hilfe des vorgestellten Substrukturextraktionsverfahrens mittels RCB-Baum das Konzept zur Erkennung und Analyse von Anomalien aus Kapitel 3 auch für räumliche Schichtmodelle umgesetzt werden kann.

5.3 Zusammenfassung

In diesem Kapitel wurde das Konzept aus Kapitel 3 zur Erkennung und Analyse von Anomalien mit Hilfe von Substrukturverzeichnissen für komplexere raumbezogene Daten von SOIS, wie Bewegungsdaten von Objekten mit variabler Geschwindigkeit (siehe Abschnitt 5.1) und räumliche Schichtmodelle (siehe Abschnitt 5.2) umgesetzt.

Für den Fall von Bewegungsdaten von Objekten mit variabler Geschwindigkeit konnte gezeigt werden, dass das bisherige Verfahren zur Substrukturextraktion mittels Pathlet Learning weiterhin zur Erstellung geeigneter Verzeichnisse ver-

wendet werden kann (siehe Abschnitt 5.1.3). Allerdings müssen für die Überwachung der Geschwindigkeit die Substrukturverzeichnisse um eine zeitliche Komponente erweitert werden, sodass für jeden Verzeichniseintrag zusätzlich ein Laufzeitmodell vorliegt (siehe Abschnitt 5.1.2).

Des Weiteren wurden neue Verfahren zur Erkennung bestimmter Anomalietypen entwickelt. Entsprechend wurden in Abschnitt 5.1.3 unter anderem Bewegungsparameter zur Erkennung und Analyse globaler Änderungen im Bewegungsverhalten der mobilen Roboter eingeführt, sowie zur Erkennung und Analyse globaler Änderungen mit Auswirkungen auf die Geschwindigkeit der Roboter ein statistischer Abgleich der Zeitdauerwahrscheinlichkeiten.

Die Evaluation des Verfahrens zeigte, dass mit der vorgestellten Erweiterung neben Anomalien bezüglich der räumlichen Bewegung, nun auch Anomalien bezüglich des zeitlichen Ablaufs detektiert werden können und das sowohl im Falle von anormalen Verhalten eines einzelnen mobilen Roboters, als auch im Falle globaler Verhaltensänderungen.

Für die Umsetzung des Konzeptes aus Kapitel 3 für räumliche Schichtmodelle musste ein komplett neuartiges Verfahren zur Extraktion geeigneterer Substrukturen entwickelt werden. Das in Abschnitt 5.2.3 vorgestellte Verfahren nutzt das Prinzip der „optimalen“ Zerteilung von Testdaten des Pathlet Learning, allerdings übertragen auf die Grundfläche der räumlichen Schichtmodelle und eingeschränkt auf Kombinationen von RCB-Partitionen (siehe Abschnitt 5.2.3.1).

Daneben wurde ein zusätzlicher Optimierungsschritt eingeführt, der nachträglich bei der Substrukturextraktion entstandene Redundanzen identifiziert und eliminiert 5.2.3.2.

Das vorgestellte Verfahren wurde sowohl für ein einfaches Multi-Agenten-System (siehe Abschnitt 5.2.4.1), als auch vier verschiedene SOIS-Szenarien (siehe Abschnitt 5.2.4.2) evaluiert. Die Ergebnisse zeigen, dass in allen Fällen durch geeignete Wahl der Parameter Substrukturverzeichnisse zur Erkennung und Analyse von Anomalien in räumlichen Schichtmodellen erzeugt werden können.

Damit wurde in diesem Kapitel das Konzept aus Kapitel 3 zur Erkennung und Analyse von Anomalien mit Hilfe von Substrukturverzeichnissen für zwei weitere Datentypen mit Raumbezug umgesetzt, die in SOIS erfasst und verarbeitet werden. Im Zuge dessen ermöglichen die vorgestellten Verfahren, dass auch diese beiden Datentypen nun trotz des enorm großen Zustandsraums im Rahmen des Qualitätssicherungsprozesses zur Laufzeit überwacht werden können.

6 Zusammenfassung und Ausblick

6.1 Zusammenfassung

In der vorliegenden Arbeit wurde das Problem der Erkennung und Analyse von Anomalien in SOIS diskutiert. Im Speziellen ging es um das Detektieren von Anomalien in Daten, die einen räumlichen Bezug haben. Dies umfasst sowohl Bewegungsdaten, die die Positionen von mobilen Werkteilen oder Robotern nachzeichnen, als auch räumliche Schichtmodelle, die zur Zustandsbeschreibung verwendet werden.

Die große Herausforderung dabei ist, dass sich auf Grund der Autonomie eines SOIS ein enorm großer Zustandsraum ergibt. Um diesen hohen Freiheitsgrad des Systems in einem Normalitätsmodell zur Anomalieerkennung abzubilden, wurde in Kapitel 3 ein Konzept zur Erkennung und Analyse von Anomalien in SOIS mit Hilfe von Substrukturverzeichnissen vorgestellt.

Die Idee dabei ist, dass die Flexibilität des Prozessverlaufs in einem SOIS über das Zusammensetzen bereits beobachteter Teilstrukturen nachgebildet werden kann. Das heißt, bei der Laufzeitüberwachung des Systems werden im Rahmen der Anomalieerkennung die Testdaten aus den zur Verfügung stehenden Substrukturen rekonstruiert. Informationen, die dabei anfallen, wie etwa der Grad der Rekonstruierbarkeit, können zur Erkennung und Analyse von Anomalien ausgewertet werden.

In Kapitel 3 wurde dieses Konzept zunächst für Bewegungsdaten von Objekten mit konstanter Geschwindigkeit, die sich in einem SOIS mit Fließbandarchitektur ergeben, umgesetzt. Für die Extraktion geeigneter Substrukturen wurde Pathlet Learning genutzt, damit ein möglichst *kompaktes*, aber auch im Rekonstruktionsprozess *effizientes* Verzeichnis entsteht. Es konnte gezeigt werden, dass sich damit Substrukturverzeichnisse erstellen lassen, sodass durch Auswertung der bei der Rekonstruktion der Bewegungsdaten anfallenden Informationen verschiedene für die Qualitätssicherung in SOIS relevante Anomalien erkannt werden können.

Durch Auswertung der implizit in den Substrukturen enthaltenen Informationen, wie räumliche Lage und Bewegungsrichtung, sind zudem weiterführende Analysen der detektierten Anomalien möglich.

Insgesamt konnte gezeigt werden, dass durch Umsetzung des vorgestellten Kon-

zeptes zum einen einzelne Werkteile identifiziert werden können, die sich anormal in der Fabrik bewegen, aber auch globale anormale Ereignisse, wie der Ausfall einer Maschine.

Aber auch ein anormaler Prozessfortschritt, der sich zum Beispiel ergibt, wenn ein Werkteil in eine Schleife gerät und ununterbrochen dieselben Schritte ausführt, sowie anormale Änderungen in der Lastenverteilung und die Neuartigkeit von Anomalien werden mit Hilfe dieses Verfahrens trotz des enorm großen Zustandsraumes erfasst.

Für eine möglichst effiziente technische Realisierung dieses Verfahrens wurden in Kapitel 4 zudem die Themen Skalierbarkeit des Substrukturprozesses und Onlinefähigkeit des Rekonstruktionsprozesses aufgegriffen und jeweils eine entsprechende Modullösung erarbeitet.

In ersterem Fall wurde ein alternatives Verfahren zur Substrukturextraktion vorgestellt, das zwar im Vergleich zu dem Pathlet Learning-basierten Ansatz geringfügige Einbußen in der Effizienz des Rekonstruktionsprozesses mit sich bringt. Dafür steigt die Rechenkomplexität lediglich linear mit der Anzahl der Koordinatenpunkte pro Trainingstrajektorie und nicht quadratisch wie beim Pathlet Learning.

Das heißt, im Falle von sehr punktreichen Trajektorien ist der in diesem Kapitel vorgestellte Substrukturextraktionsprozess mittels SEQUITUR eine gut skalierbare Alternative zum Pathlet Learning.

Die zweite Modullösung umfasst ein Verfahren zum inversen online Substrukturabgleich. Damit wird zum einen ein effizienter Rekonstruktionsprozess zur Laufzeit ermöglicht, indem die vorhandenen Substrukturen in Kantenlisten indexiert werden und darüber sukzessive geeignete Kandidaten für eingehende Testtrajektorien gefunden werden können.

Zum anderen können bereits zur Laufzeit Zwischenergebnisse über die Rekonstruierbarkeit der Testtrajektorie abgefragt werden, was für die Qualitätssicherung in einem SOIS von zentraler Bedeutung ist. Damit ermöglichen die beiden in diesem Kapitel vorgestellten Modullösungen eine effiziente Realisierung des Verfahrens aus Kapitel 3 zur Laufzeitüberwachung von SOIS.

Die Übertragbarkeit des in Kapitel 3 vorgestellten Konzeptes auf weitere raumbezogene Daten von SOIS wird schließlich in Kapitel 5 gezeigt. Dafür werden Umsetzungen für Bewegungsdaten von Objekten mit variabler Geschwindigkeit, sowie räumliche Schichtmodelle vorgestellt.

Im Falle von Bewegungsdaten von Objekten mit variabler Geschwindigkeit muss das Substrukturverzeichnis um eine zeitliche Komponente ergänzt werden, sodass für jeden Verzeichniseintrag noch ein zusätzliches Laufzeitmodell vorhanden ist.

Außerdem müssen die Verfahren zur Auswertung der Rekonstruktionsinformationen angepasst werden, da nun neben Anomalien bezüglich der räumlichen Bewegung, auch Anomalien bezüglich des zeitlichen Ablaufs auftreten und erkannt werden müssen.

Insgesamt konnte gezeigt werden, dass durch die Erweiterung des Substruk-

turverzeichnis und entsprechende Auswertungen der bei der Rekonstruktion anfallenden Informationen sowohl einzelne mobile Roboter identifiziert werden können, die sich entweder bezüglich ihrer räumlichen Bewegung oder bezüglich ihrer Geschwindigkeit anormal verhalten, als auch Änderungen im globalen Bewegungsverhalten oder der Ausfall einer Maschine, der eine globale Änderung bezüglich der zeitlichen Abläufe auslöst, detektiert werden können.

Für die Umsetzung des Konzeptes aus Kapitel 3 für räumliche Schichtmodelle musste zunächst ein Substrukturextraktionsverfahren entwickelt werden, das analog zur Idee des Pathlet Learning ein kompaktes, aber im Rekonstruktionsprozess effizientes Verzeichnis zur Anomalieerkennung erzeugt, wobei dafür diesmal eben eine 2-dimensionale Grundfläche und nicht die 1-dimensionalen Routen der Objekte zerteilt werden müssen.

Insgesamt konnte dann aber erneut gezeigt werden, dass auch in diesem Fall das Konzept aus Kapitel 3 das Problem des enorm großen Zustandsraumes bei der Erkennung und Analyse von Anomalien in SOIS mit Hilfe von Substrukturverzeichnissen für raumbezogene Daten löst.

Die vorliegende Arbeit liefert damit einen wichtigen Beitrag zum Thema Qualitätssicherung in SOIS, da die vorgestellten Lösungen trotz des hohen Freiheitsgrades des Systems eine Systemüberwachung zur Laufzeit ermöglichen und damit als Bestandteil eines Gesamtqualitätssicherungsprozesses eingesetzt werden können.

6.2 Ausblick

Abschließend soll noch ein Ausblick auf mögliche weiterführende Arbeiten zur Weiterentwicklung des in der vorliegenden Arbeit vorgestellten Konzeptes zur Erkennung und Analyse von Anomalien in SOIS mit Hilfe von Substrukturverzeichnissen gegeben werden.

So könnten unter anderem Aktualisierungsstrategien, engl. „update strategies“, für die Substrukturverzeichnisse entwickelt werden. Die Herausforderungen dabei wären zum einen die Entscheidung, wann eine derartige Aktualisierung auf Grund von Änderungen im System nötig wird. Zum anderen muss dann die Frage geklärt werden, ob für eine derartige Aktualisierung das Verzeichnis von Grund auf neu gelernt werden muss, oder ob bei Erkennung von Neuartigkeit (siehe Abschnitt 3.6.5) das Hinzunehmen einzelner Verzeichniseinträge ausreicht.

Des Weiteren könnten im Rahmen weiterführender Arbeiten das vorgestellte Konzept für weitere (nicht raumbezogene) Datentypen umgesetzt werden, damit auch diese im Falle eines enorm großen Zustandsraumes zuverlässig auf ihre (Ab-)Normalität hin untersucht werden können.

Interessante Anwendungsfälle wären hier sicherlich die Prozessanalyse, engl. „process mining“, aber auch ganz allgemein alle den Ablaufprozess aufzeichnenden Daten, engl. „logging“, in autonomen Systemen. Dazu müssten analog zu den Erweiterungen aus Kapitel 5 datentypspezifische Verfahren zur Sub-

strukturextraktion, sowie geeignete Mechanismen für die Auswertung der bei der Rekonstruktion anfallenden Informationen entwickelt werden.

Damit erwächst automatisch ein weiteres sehr interessantes Feld für weiterführende Arbeiten, das sich mit der effizienten technischen Realisierung der jeweiligen Verfahren beschäftigt. So könnten wie in Kapitel 4 für die jeweiligen Umsetzungen des Konzepts entsprechende vereinfachende Ausschlussverfahren, enlg. „pruning“, oder Indexstrukturen entwickelt werden, um eine effiziente Implementierung zu ermöglichen.

Literaturverzeichnis

- [Abe 06] N. Abe, B. Zadrozny, J. Langford. “Outlier detection by active learning”. In: *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 504–509, ACM, 2006.
- [arca 11] arcatech - National Academy of Science and Engineering, Ed. *Cyber-Physical Systems. Driving force for innovation in mobility, health, energy and production (acatech POSITION PAPER)*, Kap. 2, p. 15. Springer Verlag, 2011.
- [Barr 11] J. A. Barria, S. Thajchayapong. “Detection and classification of traffic anomalies using microscopic traffic variables”. *IEEE Transactions on Intelligent Transportation Systems*, Vol. 12, Nr. 3, pp. 695–704, 2011.
- [Baue 17] T. Bauernhansl. “Die vierte industrielle Revolution–Der Weg in ein wertschaffendes Produktionsparadigma”. In: *Handbuch Industrie 4.0 Bd. 4*, pp. 1–31, Springer, 2017.
- [Bend 16] A. Bendale, T. E. Boult. “Towards open set deep networks”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1563–1572, 2016.
- [Burt 17] S. Burton, L. Gauerhof, C. Heinzemann. “Making the case for safety of machine learning in highly automated driving”. In: *International Conference on Computer Safety, Reliability, and Security*, pp. 5–16, Springer, 2017.
- [Chaw 12] S. Chawla, Y. Zheng, J. Hu. “Inferring the root cause in road traffic anomalies”. In: *2012 IEEE 12th International Conference on Data Mining*, pp. 141–150, IEEE, 2012.
- [Chen 11] C. Chen, D. Zhang, P. S. Castro, N. Li, L. Sun, S. Li. “Real-time detection of anomalous taxi trajectories from GPS traces”. In: *International Conference on Mobile and Ubiquitous Systems: Computing, Networking, and Services*, pp. 63–74, Springer, 2011.
- [Chen 13] C. Chen, H. Su, Q. Huang, L. Zhang, L. Guibas. “Pathlet learning for compressing and planning trajectories”. In: *Proceedings*

- of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pp. 392–395, ACM, 2013.
- [De A 05] V. T. De Almeida, R. H. Güting. “Indexing the trajectories of moving objects in networks”. *GeoInformatica*, Vol. 9, Nr. 1, pp. 33–60, 2005.
- [De L 13] R. De Lemos, H. Giese, H. A. Müller, M. Shaw, J. Andersson, M. Litoiu, B. Schmerl, G. Tamura, N. M. Villegas, T. Vogel *et al.* “Software engineering for self-adaptive systems: A second research roadmap”. In: *Software Engineering for Self-Adaptive Systems II*, pp. 1–32, Springer, 2013.
- [Deng 11] K. Deng, K. Xie, K. Zheng, X. Zhou. *Trajectory Indexing and Retrieval*, pp. 35–60. Springer New York, New York, NY, 2011.
- [Dijk 59] E. W. Dijkstra. “A note on two problems in connexion with graphs”. *Numerische mathematik*, Vol. 1, Nr. 1, pp. 269–271, 1959.
- [Eber 15] B. Eberhardinger, G. Anders, H. Seebach, F. Siefert, W. Reif. “A Framework for Testing Self-organisation Algorithms”. *Softwaretechnik-Trends*, Vol. 35, Nr. 1, 2015.
- [Eber 17] B. Eberhardinger, G. Anders, H. Seebach, F. Siefert, A. Knapp, W. Reif. “An approach for isolated testing of self-organization algorithms”. In: *Software Engineering for Self-Adaptive Systems III. Assurances*, pp. 188–222, Springer, 2017.
- [Eder 13] K. I. Eder, N. M. Villegas, F. Trollmann, P. Pelliccione, H. A. Müller, D. Schneider, L. Grunske, B. Rumpe, M. Litoiu, A. Perini10 *et al.* “Assurance using models at runtime for self-adaptive software systems”. *State-of-the-Art Survey on Models at Runtime. Lecture Notes in Computer Science. Springer*, 2013.
- [Fren 03] E. Frentzos. “Indexing objects moving on fixed networks”. In: *International Symposium on Spatial and Temporal Databases*, pp. 289–305, Springer, 2003.
- [Gabo 16] T. Gabor, L. Belzner, M. Kiermeier, M. T. Beck, A. Neitz. “A simulation-based architecture for smart cyber-physical systems”. In: *2016 IEEE International Conference on Autonomic Computing (ICAC)*, pp. 374–379, IEEE, 2016.
- [Gal 16] Y. Gal, Z. Ghahramani. “Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning”. In:

- Proceedings of the 33rd International Conference on Machine Learning (ICML)*, 2016.
- [Ge 11] Y. Ge, H. Xiong, C. Liu, Z.-H. Zhou. “A taxi driving fraud detection system”. In: *2011 IEEE 11th International Conference on Data Mining*, pp. 181–190, IEEE, 2011.
- [Ge 13] Z. Ge, Z. Song, F. Gao. “Review of recent research on data-based process monitoring”. *Industrial & Engineering Chemistry Research*, Vol. 52, Nr. 10, pp. 3543–3562, 2013.
- [Gupt 17] J. K. Gupta, M. Egorov, M. Kochenderfer. “Cooperative multi-agent control using deep reinforcement learning”. In: *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems*, pp. 66–83, Springer, 2017.
- [IEEE610 90] “IEEE Standard Glossary of Software Engineering Terminology”. Standard, The Institute of Electrical and Electronics Engineers, New York, USA, 1990.
- [Kage 11] H. Kagermann, W.-D. Lukas, W. Wahlster. “Industrie 4.0: Mit dem Internet der Dinge auf dem Weg zur 4. industriellen Revolution”. *VDI Nachrichten*, Nr. 13, p. 2, 2011.
- [Kage 12] H. Kagermann, W. Wahlster, J. Helbig. “Im Fokus: Zukunftsprojekt Industrie 4.0 mit Handlungsempfehlungen zur Umsetzung sowie Vorstellung der weiteren von der Promotorengruppe behandelten Initiativen”. Bericht der Promotorengruppe Kommunikation, Forschungsunion im Stifterverband für die Deutsche Wirtschaft e.V., 2012.
- [Kier 17a] M. Kiermeier. “Online Reverse Subpattern Matching for Reproducing Trajectories from Sub-paths”. In: *17th International Conference on Computational Science and Its Applications (ICCSA), 2017*, pp. 635–646, Springer, 2017.
- [Kier 17b] M. Kiermeier, H. Sauer, J. Wieghardt. “Monitoring self-organizing industrial systems using sub-trajectory dictionaries”. In: *IEEE 15th International Conference on Industrial Informatics (INDIN), 2017*, pp. 665–670, IEEE, 2017.
- [Kier 17c] M. Kiermeier, M. Werner, C. Linnhoff-Popien, H. Sauer, J. Wieghardt. “Anomaly detection in self-organizing industrial systems using pathlets”. In: *IEEE 18th International Conference on Industrial Technology (ICIT), 2017*, pp. 1226–1231, IEEE, 2017.

- [Kier 17d] M. Kiermeier, M. Werner, H. Sauer, J. Wieghardt. “Building scalable models for anomaly detection in self-organizing industrial systems”. In: *IEEE 15th International Conference on Industrial Informatics (INDIN), 2017*, pp. 245–250, IEEE, 2017.
- [Kier 18a] M. Kiermeier, S. Feld. “Visual Analytics for Root Cause Analysis in Self-Organizing Industrial Systems”. In: *IEEE 16th International Conference on Industrial Informatics (INDIN), 2018*, pp. 315–320, IEEE, 2018.
- [Kier 18b] M. Kiermeier, S. Feld, T. Phan, C. Linnhoff-Popien. “Anomaly Detection in Spatial Layer Models of Autonomous Agents”. In: *18th International Conference on Intelligent Data Engineering and Automated Learning (IDEAL), 2018*, pp. 156–163, Springer, 2018.
- [Kier 18c] M. Kiermeier, T. Phan, H. Sauer, J. Wieghardt. “Monitoring Autonomous Agents in Self-Organizing Industrial Systems”. In: *IEEE 16th International Conference on Industrial Informatics (INDIN), 2018*, pp. 653–658, IEEE, 2018.
- [Koid 15] S. Koide, Y. Tadokoro, T. Yoshimura. “SNT-index: Spatio-temporal index for vehicular trajectories on a road network based on substring matching”. In: *Proceedings of the 1st International ACM SIGSPATIAL Workshop on Smart Cities and Urban Analytics*, pp. 1–8, ACM, 2015.
- [Koop 16] P. Koopman, M. Wagner. “Challenges in autonomous vehicle testing and validation”. *SAE International Journal of Transportation Safety*, Vol. 4, Nr. 1, pp. 15–24, 2016.
- [Koop 18] P. Koopman, M. Wagner. “Toward a Framework for Highly Automated Vehicle Safety Validation”. In: *SAE World Congress*, 2018.
- [Kriz 12] A. Krizhevsky, I. Sutskever, G. E. Hinton. “Imagenet classification with deep convolutional neural networks”. In: *Advances in neural information processing systems*, pp. 1097–1105, 2012.
- [Krog 14] B. Krogh, N. Pelekis, Y. Theodoridis, K. Torp. “Path-based queries on trajectory data”. In: *Proceedings of the 22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pp. 341–350, ACM, 2014.
- [Laks 17] B. Lakshminarayanan, A. Pritzel, C. Blundell. “Simple and scalable predictive uncertainty estimation using deep ensembles”. In: *Advances in Neural Information Processing Systems*, pp. 6405–6416, 2017.

- [Lee 08] J.-G. Lee, J. Han, X. Li. “Trajectory outlier detection: A partition-and-detect framework”. In: *2008 IEEE 24th International Conference on Data Engineering*, pp. 140–149, IEEE, 2008.
- [Liao 10] Z. Liao, Y. Yu, B. Chen. “Anomaly detection in GPS data based on visual analytics”. In: *Visual Analytics Science and Technology (VAST), 2010 IEEE Symposium on*, pp. 51–58, IEEE, 2010.
- [Liu 11] W. Liu, Y. Zheng, S. Chawla, J. Yuan, X. Xing. “Discovering spatio-temporal causal interactions in traffic data streams”. In: *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 1010–1018, ACM, 2011.
- [Liu 14] S. Liu, L. M. Ni, R. Krishnan. “Fraud detection from taxis’ driving behaviors”. *IEEE Transactions on Vehicular Technology*, Vol. 63, Nr. 1, pp. 464–472, 2014.
- [Lope 11] R. H. Lopes. “Kolmogorov-smirnov test”. In: *International encyclopedia of statistical science*, pp. 718–720, Springer, 2011.
- [McCl 16] P. McClure, N. Kriegeskorte. “Robustly representing uncertainty through sampling in deep neural networks”. *arXiv preprint arXiv:1611.01639*, 2016.
- [Nevi 97] C. G. Nevill-Manning, I. H. Witten. “Identifying Hierarchical Structure in Sequences: A linear-time algorithm”. *Journal of Artificial Intelligence Research*, Vol. 7, pp. 67–82, 1997.
- [Pan 13] B. Pan, Y. Zheng, D. Wilkie, C. Shahabi. “Crowd sensing of traffic anomalies based on human mobility and social media”. In: *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pp. 344–353, ACM, 2013.
- [Pana 05] L. Panait, S. Luke. “Cooperative multi-agent learning: The state of the art”. *Autonomous agents and multi-agent systems*, Vol. 11, Nr. 3, pp. 387–434, 2005.
- [Pang 11] L. X. Pang, S. Chawla, W. Liu, Y. Zheng. “On mining anomalous patterns in road traffic streams”. In: *International Conference on Advanced Data Mining and Applications*, pp. 237–251, Springer, 2011.
- [Pang 13] L. X. Pang, S. Chawla, W. Liu, Y. Zheng. “On detection of emerging anomalous traffic patterns using GPS data”. *Data & Knowledge Engineering*, Vol. 87, pp. 357–373, 2013.

- [Pfos 05] D. Pfoser, C. S. Jensen. “Trajectory indexing using movement constraints”. *GeoInformatica*, Vol. 9, Nr. 2, pp. 93–115, 2005.
- [Phan 18] T. Phan, L. Belzner, T. Gabor, K. Schmid. “Leveraging Statistical Multi-Agent Online Planning with Emergent Value Function Approximation”. In: *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems (AA-MAS)*, pp. 730–738, 2018.
- [Pime 14] M. A. Pimentel, D. A. Clifton, L. Clifton, L. Tarassenko. “A review of novelty detection”. *Signal Processing*, Vol. 99, pp. 215–249, 2014.
- [Pnue 07] A. Pnueli. “Roam: Rule-and motif-based anomaly detection in massive moving object data sets”. 2007.
- [Roh 11a] G.-P. Roh, S.-w. Hwang. “TPM: Supporting pattern matching queries for road-network trajectory data”. In: *Proceedings of the 14th International Conference on Extending Database Technology*, pp. 554–557, ACM, 2011.
- [Roh 11b] G.-P. Roh, J.-W. Roh, B.-K. Yi *et al.* “Supporting pattern-matching queries over trajectories on road networks”. *IEEE Transactions on Knowledge and Data Engineering*, Vol. 23, Nr. 11, pp. 1753–1758, 2011.
- [Russ 10] S. Russell, P. Norvig. *Artificial Intelligence - A Modern Approach*. Prentice Hall, 2010.
- [Sand 11] I. Sandu Popa, K. Zeitouni, V. Oria, D. Barth, S. Vial. “Indexing in-network trajectory flows”. *The VLDB Journal—The International Journal on Very Large Data Bases*, Vol. 20, Nr. 5, pp. 643–669, 2011.
- [Schn 17] D. Schneider, M. Trapp, J. Dörr, S. Dukanovic, T. Henkel, R. Khondoker, C. Krauß, S. Mauthofer, D. Scheuermann, D. Zelle. “Umfassende Sicherheit”. *Informatik-Spektrum*, Vol. 40, Nr. 5, pp. 419–429, 2017.
- [Seru 05] G. D. M. Serugendo, M.-P. Gleizes, A. Karageorgos. “Self-organization in multi-agent systems”. *The Knowledge Engineering Review*, Vol. 20, Nr. 2, pp. 165–189, 2005.
- [Siem 17] Siemens AG. “The Digitalization Productivity Bonus – Welchen Mehrwert bietet die Digitalisierung für Fertigungsunternehmen”. Whitepaper, Siemens Financial Services, 2017.

- [Sill 08] R. R. Sillito, R. B. Fisher. “Semi-supervised Learning for Anomalous Trajectory Detection.”. In: *BMVC*, pp. 035–1, 2008.
- [Silv 16] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot *et al.* “Mastering the game of Go with deep neural networks and tree search”. *Nature*, Vol. 529, Nr. 7587, pp. 484–489, 2016.
- [Simo 91] H. D. Simon. “Partitioning of unstructured problems for parallel processing”. *Computing systems in engineering*, Vol. 2, Nr. 2-3, pp. 135–148, 1991.
- [Skani 18] S. Skansi. *Introduction to Deep Learning: From Logical Calculus to Artificial Intelligence*. Springer, 2018.
- [Tamu 13] G. Tamura, N. M. Villegas, H. A. Müller, J. P. Sousa, B. Becker, G. Karsai, S. Mankovskii, M. Pezzè, W. Schäfer, L. Tahvildari *et al.* “Towards practical runtime verification and validation of self-adaptive software systems”. In: *Software Engineering for Self-Adaptive Systems II*, pp. 108–132, Springer, 2013.
- [Trit 16] I. Tritchkov, H. Goetz. “Verification and Validation of Decentralized, Self-Organizing Cyber-Physical Production Systems: A Blueprint Process for Testing Cyber-Physical Production Systems with Self-Properties”. In: *IEEE International Workshops on Foundations and Applications of Self* Systems*, pp. 112–117, IEEE, 2016.
- [Venk 03a] V. Venkatasubramanian, R. Rengaswamy, S. N. Kavuri. “A review of process fault detection and diagnosis: Part I: Quantitative model-based methods”. *Computers & chemical engineering*, Vol. 27, Nr. 3, pp. 239–311, 2003.
- [Venk 03b] V. Venkatasubramanian, R. Rengaswamy, S. N. Kavuri. “A review of process fault detection and diagnosis: Part II: Qualitative models and search strategies”. *Computers & chemical engineering*, Vol. 27, Nr. 3, pp. 313–326, 2003.
- [Venk 03c] V. Venkatasubramanian, R. Rengaswamy, S. N. Kavuri, K. Yin. “A review of process fault detection and diagnosis: Part III: Process history based methods”. *Computers & chemical engineering*, Vol. 27, Nr. 3, pp. 327–346, 2003.
- [Voge 17] B. Vogel-Heuser. “Herausforderungen und Anforderungen aus Sicht der IT und der Automatisierungstechnik”. In: *Handbuch Industrie 4.0 Bd. 4*, pp. 33–44, Springer, 2017.

- [Yuan 11] G. Yuan, S. Xia, L. Zhang, Y. Zhou, C. Ji. “Trajectory outlier detection algorithm based on structural features”. *Journal of Computational Information Systems*, Vol. 7, Nr. 11, pp. 4137–4144, 2011.
- [Zhan 11] D. Zhang, N. Li, Z.-H. Zhou, C. Chen, L. Sun, S. Li. “iBAT: detecting anomalous taxi trajectories from GPS traces”. In: *Proceedings of the 13th international conference on Ubiquitous computing*, pp. 99–108, ACM, 2011.
- [Zhan 12] J. Zhang. “Smarter outlier detection and deeper understanding of large-scale taxi trip records: a case study of NYC”. In: *Proceedings of the ACM SIGKDD International Workshop on Urban Computing*, pp. 157–162, ACM, 2012.
- [Zhen 11a] Y. Zheng, Y. Liu, J. Yuan, X. Xie. “Urban computing with taxicabs”. In: *Proceedings of the 13th international conference on Ubiquitous computing*, pp. 89–98, ACM, 2011.
- [Zhen 11b] Y. Zheng, X. Zhou. *Computing with spatial trajectories*. Springer Science & Business Media, 2011.