



TECHNISCHE
UNIVERSITÄT
WIEN
Vienna University of Technology

Diplomarbeit

Anwendung und Bewertung von Bayes'schen Netzwerken zur Prognose von ungeplanten Stillständen im Bereich der Halbleiterfertigung

ausgeführt zum Zwecke der Erlangung des akademischen Grades eines

Diplom-Ingenieurs

unter der Leitung von

Univ.-Prof. Dr.-Ing. Dipl. Wirt.-Ing. Prof. eh. Dr. h.c. Wilfried Sihn

(E330 Institut für Managementwissenschaften, Bereich: Betriebstechnik und Systemplanung)

und mit der Unterstützung der fachlichen Betreuerin

Dipl.-Ing. Tanja Nemeth

(E330 Institut für Managementwissenschaften, Bereich: Betriebstechnik und Systemplanung,
Fraunhofer Austria Research GmbH)

eingereicht an der Technischen Universität Wien

Fakultät für Maschinenwesen und Betriebswissenschaften

von

Fabian Biebl

Wien, im Dezember 2018

Fabian Biebl



TECHNISCHE
UNIVERSITÄT
WIEN
Vienna University of Technology

Ich habe zur Kenntnis genommen, dass ich zur Drucklegung meiner Arbeit unter der Bezeichnung

Diplomarbeit

nur mit Bewilligung der Prüfungskommission berechtigt bin.

Ich erkläre weiters Eides statt, dass ich meine Diplomarbeit nach den anerkannten Grundsätzen für wissenschaftliche Abhandlungen selbstständig ausgeführt habe und alle verwendeten Hilfsmittel, insbesondere die zugrunde gelegte Literatur, genannt habe.

Weiters erkläre ich, dass ich dieses Diplomarbeitsthema bisher weder im In- noch Ausland (einer Beurteilerin/einem Beurteiler zur Begutachtung) in irgendeiner Form als Prüfungsarbeit vorgelegt habe und dass diese Arbeit mit der vom Begutachter beurteilten Arbeit übereinstimmt.

Wien, im Dezember 2018

Fabian Biebl

Danksagung

An dieser Stelle möchte ich mich im Besonderen bei meiner Betreuerin Dipl.-Ing. Tanja Nemeth bedanken. Ohne die exzellente inhaltliche und fachliche Unterstützung wäre das Erstellen dieser Arbeit, in dieser Form, wohl nicht möglich gewesen. Frau Nemeth nahm sich bei Fragen regelmäßig Zeit diese zu besprechen und erlaubte es mir durch ihre differenzierte Sichtweise in gewissen Gebieten, über den sprichwörtlichen Tellerrand hinauszusehen. Auch in Zeiten in denen ich, aufgrund von Rückschlägen bei der Erstellung der Arbeit, große Sorgen bezüglich der erfolgreichen Fertigstellung hatte, schaffte es Frau Nemeth mich durch ihre motivierende Art und Weise wieder auf den richtigen Pfad zu leiten.

Herrn Univ.-Prof. Dr.-Ing. Dipl. Wirt.-Ing. Prof. eh. Dr. h.c. Wilfried Sihm möchte ich dafür danken, dass er die Verfassung der Arbeit an seinem Lehrstuhl bewilligt hat.

Herrn Dr.-techn. Andreas Jäger, MSc., MBA möchte ich dafür Dank sagen, dass er mich, im Zuge meiner Tätigkeit für die Fraunhofer Austria GmbH., sowohl im wissenschaftlichen als auch im technisch/operativen Bereich allzeit gefördert hat. Ohne ihn hätte sich mein brennendes Interesse und meine Begeisterung für das Themengebiet der Industrial Data Science wohl nie entwickelt.

Dem Unternehmen und den beteiligten Mitarbeitern, die mich mit den, für die Erstellung der Arbeit, benötigten Daten versorgt haben, möchte ich für diese Form der Kooperation danken. Gerade in der heutigen Zeit, in der Unternehmen beim Teilen von Daten zur außerordentlichen Vorsicht angehalten sind, ist dies wahrlich keine Selbstverständlichkeit.

Zu guter Letzt möchte ich mich bei meiner Familie und dabei insbesondere bei meinen Eltern und meiner Lebensgefährtin bedanken, die mich während meiner Studienzeit sowohl finanziell als auch seelisch unterstützt haben. Durch ihr Zutun war es möglich unangenehmen Situationen zu trotzen und das Ziel nicht aus den Augen zu verlieren.

Kurzfassung

Das Ziel dieser Arbeit besteht darin, ein Modell zu entwickeln, welches zur Prognose von ungeplanten Stillständen in der Halbleiterfertigung verwendet werden kann. Das Prognosemodell soll dabei mittels der Methodik der „Bayes’schen Netzwerke“ (BN) umgesetzt werden, um die Möglichkeit der Realisierung zu examinieren. Darüber hinaus sollen im Anschluss Aussagen bezüglich der Effizienz der Methode, beziehungsweise bezüglich der Eignung/Nicht-Eignung für den präsentierten Use-Case getroffen werden.

Zunächst wird sich der theoretische Teil dieser Arbeit mit Grundbegriffen der Instandhaltung und dem derzeitigen State-of-the-Art befassen. Dabei wird besonderer Fokus auf das Forschungsgebiet der wissensbasierten Instandhaltung und die Bereiche der prädiktiven und präskriptiven Instandhaltung gelegt. Anschließend widmet sich das Kapitel Generierung und Aufbereitung von Daten dem Thema der Wissenserschließung und Entscheidungsfindung und der Thematik der Daten- und Informationsqualität. Darauffolgend wird der CRISP-DM Zyklus als standardisierte Methode des Data-Mining präsentiert. Zuletzt beschäftigt sich der theoretische Teil dieser Arbeit mit Bayes’schen Netzen. Hierfür werden zunächst die mathematischen Grundlagen und Zusammenhänge erläutert. Darauf aufbauend wird die Erweiterung zu dynamischen Bayes’schen Netzen (DBN) erklärt und zu guter Letzt Beispiele präsentiert, in denen die Methodik, im Bereich der Instandhaltung bereits angewendet wurde.

Für die Erstellung des BN zur Ausfallsprognose werden die Unternehmensdaten eines österreichischen Herstellers für Halbleitererzeugnisse herangezogen. Nach durchgeführter Sichtung und Analyse der Daten wird ein erstes BN erstellt, welches mittels komponentenspezifischer Einflüsse auf das Ausfallverhalten, Prognosen erstellen soll. Da dieser Ansatz aufgrund unzureichender Datenqualität nicht weiterverfolgt werden konnte, wird ein zweites Netz entworfen, welches die Prognose aufgrund von Zusammenhängen zwischen auftretenden Maschinentalarmen und der Time to Failure (TTF) erstellt. Dieses wurde nach erfolgreicher Umsetzung zu einem dynamischen Bayes’schen Netz erweitert, um sich ändernden Umgebungseinflüssen anpassen zu können.

Bei der Durchführung der Arbeit wurde die Möglichkeit, Bayes’sche Netze für den Bereich der prädiktiven Instandhaltung zu erstellen und anzuwenden, nachgewiesen. Dabei zeigt sich, dass die Erstellung und Erprobung äußerst viel Zeit in Anspruch nimmt und die Effizienz bezüglich Input und Output folglich, im Vergleich zu anderen Methoden, gering erscheint. Jedoch zeigt sich, dass der Ansatz der Bayes’schen Netze im Bereich der präskriptiven Instandhaltung zur Fehlerdiagnose und Ableitung geeigneter Instandhaltungsmaßnahmen einen bedeutenden Mehrwert liefern kann.

Abstract

The primary objective of this thesis aims on developing a model, which is capable to predict unplaned downtimes in semi-conductor manufacturing. The creation of the forecasting model should therefore be conducted via „Bayesian Networks“ to examine the possibilty of creating a prediction model by means of this methodology. Furthermore, statements should be made regarding the efficiency, respectively the applicability/non-applicability, of the used approach for the presented use-case.

First of all, the theoretical part of this thesis will describe the fundamentals and the state-of-the-art of maintenance in production systems. Particular emphasis is thereby placed on the research area of knowledge-based maintenance and the areas of predictive and prescriptive maintenance. Subsequently the chapter generating and preparation of data is dedicated to the topic of knowledge development and decision-making and the topic of data and information quality. After that, the CRISP-DM cycle is presented as a standardized method of data mining. Finally the theoretical part of this thesis will deal with the topic of bayesian networks. Therefor the mathematical foundations and relationships are explained and the enhancement to dynamic bayesian networks is illustrated. Last but not least, examples are presented in which the methodology has already been applied in the field of maintenance.

To develop the BN, with the purpose of predicting unscheduled downtimes, company data of an Austrian manufaturer for semi-conductor produtcs is used. After the process of data cleaning, a first BN is created, which should be able to conduct forecasts via component-specific influences on the failure behaviour. Unfortunately this approach could not be pursued due to insufficient data quality. Hence, a second network was created which should operate on the basis of coherencies between machine alarms and the time to failure. After the successful creation of the model, the network is enhanced to a dynamic bayesian network to be capable to adapt to changing environmental influences.

During the conduction of this thesis, the possibility to create and apply bayesian networks in the field of predictive maintenance, was demonstrated. In the process one can determine, that the developing and testing requires a lot of effort. Therefore the efficiency in terms of input and output, compared to other methods, seems small. However, the approach bayesian networks can provide a significant added value in the field of prescriptive maintenance through fault diagnosis and the derivation of appropriate maintenance measures.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Ausgangssituation und Problemstellung	1
1.2	Forschungsfragen.....	3
1.3	Aufbau und Struktur der Arbeit	3
2	Theoretische Grundlagen.....	5
2.1	Instandhaltung und Instandhaltungsstrategien	5
2.1.1	Definition Instandhaltung	5
2.1.2	Ziele der Instandhaltung	6
2.1.3	Instandhaltungsstrategien	7
2.2	State-of-the-Art im Bereich der Instandhaltung	11
2.2.1	Wissensbasierte Instandhaltung	12
2.2.2	Prädiktive Instandhaltung.....	13
2.2.3	Präskriptive Instandhaltung.....	17
3	Generierung und Aufbereitung von Daten.....	20
3.1	Wissenserschließung und Entscheidungsfindung	20
3.2	Daten- und Informationsqualität	21
3.2.1	Datenqualität.....	22
3.2.2	Informationsqualität.....	25
3.3	Methodik zur Generierung und Aufbereitung von Daten.....	27
3.3.1	Definition und Hintergrund ”	27
3.3.2	CRISP-DM Reference Modell ”	28
3.3.3	CRISP-DM User-Guide	31
3.3.4	Erweiterung der CRISP-DM Methodik	32
4	Bayes’sche Netze	34
4.1	Grundlegende Definitionen und Eigenschaften	34
4.2	Dynamische Bayes’sche Netze	38
4.3	Bayes’sche Netze in der Instandhaltung	39
5	Erstellung eines Bayes’schen Netzwerkes zur Ausfallprognose in der Halbleiterfertigung	42
5.1	Beschreibung des Use-Case/Ausgangssituation.....	42
5.1.1	Ausgangssituation.....	42

5.1.2	Grundlagen der Halbleiterätzung	42
5.1.3	Anlagenaufbau	43
5.1.4	Kammernaufbau	45
5.1.5	E-Chuck Modul	45
5.1.6	Instandhaltungsabläufe und -planung	48
5.2	Beschreibung des Datensets	48
5.2.1	Fertigungsspezifische Kennzahlen	49
5.2.2	Alarme	49
5.2.3	Limitverletzungen (LV)	50
5.2.4	Instandhaltungsmaßnahmenbuch	51
5.2.5	Statusmeldungen	52
5.2.6	Reinigungsdaten	52
5.2.7	SAP	53
5.2.8	Unterteilung des Datensatzes	54
5.3	Modellierung des ersten Bayes'schen Netzes	54
5.3.1	Variablen Identifikation	54
5.3.2	Definition der BN Netzstruktur	55
5.3.3	Definition der bedingten Wahrscheinlichkeiten	55
5.4	Vorgehen bei der Erstellung des zweiten Netzes	56
5.4.1	Variablen Identifikation	56
5.4.2	Definition der Netzstruktur	60
5.4.3	Definition der bedingten Wahrscheinlichkeiten	62
5.4.4	Umsetzung des statischen Netzwerkes in R	82
5.5	Dynamisches Bayes'sches Netz	84
5.5.1	Erweiterung der Wahrscheinlichkeitstabellen	85
5.6	Validierung des Modells	97
5.6.1	Ergebnisse des statischen Netzwerkes	97
5.6.2	Ergebnisse des dynamischen Netzwerkes	98
6	Ergebnisdiskussion	100
6.1	Ergebnisse durchgeführter Analysen	100
6.2	Ergebnisse in Bezug auf die Instandhaltungsplanung	100
6.3	Vergleich der erzielten Ergebnisse mit anderen Vorgehensmethodiken	101

6.3.1	Vergleich mit Ergebnissen aus Literatur	101
6.3.2	Vergleich mit Ergebnissen eines Forschungsprojektes mit selbigem Use-Case.....	101
6.4	SWOT Analyse	102
6.5	Ergebnisse in Bezug auf die Forschungsfragen	103
6.6	Ergebnisse in Bezug auf die Problemstellung	104
6.7	Conclusio.....	104
7	Ausblick.....	106
8	Anhang.....	107
8.1	R-Skripten	107
8.1.1	Aufbereitung Limitverletzungen und TTF	107
8.1.2	Aufbereitung Alarme und TTF.....	108
8.1.3	Einlesen der prozessspezifischen Kennzahlen	110
8.1.4	Bestimmung TTF für Biasvoltage, Clampvoltage und Generator Power (LV und Alarme).....	111
8.1.5	Berechnung der Auftretenswahrscheinlichkeiten für Biasvoltage, Clampvoltage und Generator Power	115
8.1.6	Aufbereitung Reinigungsdaten.....	119
8.1.7	Erstellung Entscheidungsbaum VTT RFH <100h	120
8.1.8	Erstellung Entscheidungsbaum VTT RFH 100h < x < 300h.....	122
8.1.9	Erstellung Entscheidungsbaum VTT RFH > 300h	125
8.1.10	Erstellung Entscheidungsbaum ESC_RFH < 1000h	129
8.1.11	Erstellung Entscheidungsbaum ESC_RFH 1000h < x < 2500h.....	130
8.1.12	Erstellung Entscheidungsbaum ESC_RFH > 2500h	134
8.1.13	Erstellung und fitten des Bayes'Netz.....	137
9	Literaturverzeichnis	143
10	Abbildungsverzeichnis	151
11	Formelverzeichnis.....	154
12	Tabellenverzeichnis	155
13	Abkürzungsverzeichnis	156

Anmerkungen zum Sprachgebrauch:

Im Weiteren erfolgen die allgemeinen Bezeichnungen von Personen aus Gründen der Lesbarkeit und Übersichtlichkeit in konventioneller Sprachform. Mit allen verwendeten Personenbezeichnungen sind stets beide Geschlechter gemeint, außer es wird ausdrücklich auf weibliche oder männliche Personen hingewiesen bzw. geht entsprechendes aus dem Kontext hervor.

1 Einleitung

1.1 Ausgangssituation und Problemstellung

Das steigende Verlangen von Kunden nach maßgeschneiderten und individualisierten Lösungen stellt die Produktion der Zukunft vor neue komplexe Herausforderungen. Für produzierende Unternehmen wird es zunehmend erforderlich, stetig neue Technologien zu entwickeln und in das tägliche Produktionsgeschehen zu implementieren, um auf dem hoch wettbewerbsorientierten Markt weiterhin konkurrenzfähig sein zu können.¹

Derzeitige Entwicklungen im Bereich der Produktionstechnologie, die nach der Etablierung sogenannter „Smart Factories“ streben, in denen sämtliche Prozesse automatisiert und computergesteuert ablaufen verlangen zusätzlich zu nahezu autonom ablaufenden Produktionsschritten hohe Maschinenverfügbarkeitszeiten.²

Um diesen Forderungen gerecht zu werden, reicht es nicht mehr die klassischen Instandhaltungsstrategien zu verfolgen, die sich hauptsächlich mit den Fragen „Was ist passiert“ (deskriptive Instandhaltung) und „Warum ist es passiert“ (diagnostische Instandhaltung) beschäftigen.³ Vielmehr müssen zukünftige Stillstände prognostiziert werden, um geeignete Instandhaltungsmaßnahmen, bei denen es nur einen geringen Anteil an menschlicher Intervention bedarf, abzuleiten und diese in eine ganzheitliche Produktionsplanung zu integrieren.⁴ Unterstrichen wird die Forderung nach neuartigen Instandhaltungsstrategien durch Studien, deren Folge nach Instandhaltungsmaßnahmen an Maschinen und Anlagen in der amerikanischen Industrie jährliche Kosten in Höhe von 200 Milliarden US-Dollar verursachen. Daraus lässt sich ableiten, dass die Kosten für die Instandhaltung von Maschinen und Anlagen zwischen 15 und 60% der gesamten Produktionskosten darstellen.⁵

Prädiktive und präskriptive Instandhaltungsstrategien bieten hierzu einen geeigneten Ansatz und ermöglichen es, Unterbrechungen in der Wertschöpfungskette vorzubeugen und folglich Kosten zu sparen.⁶ Voraussetzung für die Implementierung dieser modernen Instandhaltungsansätze stellt das Vorhandensein einer validen Datenbasis dar, da die Datenqualität die Effizienz und Genauigkeit prognostischer Instandhaltungsmodelle wesentlich beeinflusst. Diese kann in weiterer Folge nicht nur für die Vorhersage von Ausfällen sondern auch für das Abgeben von Empfehlungen bezüglich durchzuführender Instandhaltungsmaßnahmen, genutzt werden.⁷

¹ vgl. Schumacher et al., 2016, S.161

² vgl. Glawar et al., 2018, S.46

³ vgl. Ansari et al., 2017, S.1

⁴ vgl. Karim et al., 2016, S.215

⁵ vgl. Jun and Kim, 2017, S.332

⁶ vgl. Abu-Samah et al., 2015, S.844ff

⁷ vgl. Matyas et al., 2017, S.462f

Fortschritte in der Entwicklung der genannten Instandhaltungsstrategien wurden bereits vielfach und intensiv in der Literatur diskutiert. Dennoch oder gerade deshalb stehen viele Unternehmen vor der Schwierigkeit, eine geeignete Vorgehensweise, welche auf für ihre unternehmensspezifischen Rahmenbedingungen zugeschnitten ist, auszuwählen und somit messbare Vorteile im Bereich der Instandhaltung zu erzielen.⁸ Dies liegt häufig an der Überforderung der Unternehmen, aus der zunehmenden Menge an Daten prozessrelevante Informationen zu extrahieren und daraus passende Strategien abzuleiten. Hierbei spielt auch das Thema der Daten- und Informationsqualität eine zentrale Rolle. Dies liegt darin begründet, dass ein Großteil der in der Literatur präsentierten Ansätze, zur Prognose ungeplanter Stillstände, datengetrieben und folglich auf manuelle Investigation und Identifikation von Fehlercharakteristiken und auf darauf aufbauendem „Feature-Engineering“ angewiesen sind. Demnach ist die Anwendung spezifischer Methodiken nur unter der Voraussetzung möglich, dass der Grad der Daten und Informationsqualität, bestimmten Anforderungen genügt. Jedoch stellt sich der derzeitige Qualitätsgrad der Daten in Unternehmen häufig als ungenügend heraus.

Zur Lösung dieser Vielzahl an Problemstellungen ist es somit erforderlich, den Informationsprozess in der Instandhaltung klar zu strukturieren und an die aktuellen Gegebenheiten anzupassen.

Karim et. al empfiehlt die Gliederung dieses Prozesses in folgende Schritte:⁹

- Data Acquisition
- Data Transition
- Data Fusion
- Data Analysis
- Information Visualisation

Während die ersten drei Schritte hauptsächlich die Aufbereitung und Verarbeitung der Daten beinhalten soll der vierte Schritt der Analyse und dem damit verbundenen Wissenszuwachs dienen. Derzeit existieren mehrere Methoden zur Datenanalyse und Modellbildung, die die Prognostizierung von Stillständen zum Ziel haben. Wie bereits erwähnt gestaltet es sich, gerade aufgrund dieser Vielzahl an Methoden, für Unternehmen schwierig die für den Betrieb passende zu identifizieren.

Diese Arbeit hat es sich zum Ziel gesetzt, Sensordaten, die von Produktionsmaschinen eines namhaften österreichischen Herstellers von Halbleitererzeugnissen extrahiert wurden, nach den genannten Aspekten aufzubereiten und zu analysieren, um folglich Ausfälle von Equipment prognostizieren zu können.

Im Zuge dessen sollen die Daten nach der zuvor präsentierten Logik aufbereitet und Parameter identifiziert werden, die Einflüsse auf das Ausfallverhalten aufweisen.

⁸ vgl. Tiddens et al., 2015, S.171

⁹ vgl. Karim et al., 2016, S.215

Darüber hinaus wird auf Basis der erlangten Informationen ein Bayes'sches Netzwerk erzeugt, welches in der Lage sein soll Maschinenstillstände zuverlässig vorherzusagen.

Die Ergebnisse der ausgewählten Methode sollen in weiterer Folge, den Resultaten anderer Analyseverfahren gegenübergestellt werden, um Abweichungen oder auch Gemeinsamkeiten identifizieren und bewerten zu können.

Eine abschließende SWOT-Analyse soll Auskunft über die Eignung/Nicht-Eignung bzw. der Effizienz der Vorgehensweise geben, um demzufolge die Auswahl der geeigneten Methode für die Halbleiterindustrie zu unterstützen.

1.2 Forschungsfragen

Abgeleitet aus der dargestellten Problemstellung ergeben sich die folgenden Forschungsfragen:

- Forschungsfrage: Können mit Hilfe eines geeignet parametrisierten und erprobten Bayes'schen Netzwerkes, ungeplante Ausfälle, des für die Fertigung von Halbleiterprodukten benötigten Equipments, prognostiziert werden, sodass ungeplanten Stillständen vorgebeugt werden kann und folglich eine funktionierende Wertschöpfung gewährleistet wird?
- Forschungsfrage: Wie kann die Effizienz des entwickelten Ansatzes im Vergleich zu anderen Methoden zur Ausfallprognose beurteilt werden?
- Forschungsfrage: Lassen sich allgemeine Aussagen über Eignung/Nicht-Eignung der verglichenen Methoden für die Halbleiterindustrie ableiten?

1.3 Aufbau und Struktur der Arbeit

Das Kapitel theoretische Grundlagen soll anhand einer Literaturrecherche den derzeitigen State-of-the-Art im Bereich der Instandhaltung aufzeigen. Das darauffolgende Kapitel, welches sich mit der Generierung und Aufbereitung von Daten befasst, soll im weiteren Aufschluss über die Wichtigkeit einer soliden Daten- und Informationsbasis geben. Durch die darauf anknüpfende Erforschung grundlegender Begriffe in Bezug auf Bayes'sche Netze soll theoretisches Wissen aufgebaut werden, welches das Fundament für die praktische Umsetzung und Modellierung bildet. Zusätzlich soll aufgezeigt werden, in welchen Bereichen der Instandhaltung, diese Methodik bereits angewandt wurde.

Um das Modell zu erstellen werden die Daten von einem Halbleiter fertigenden Unternehmen in Österreich herangezogen, gesichtet und analysiert. Darauf aufbauend werden die zu verwendenden Variablen definiert, Abhängigkeiten bestimmt und

Wahrscheinlichkeiten der einzelnen Knoten berechnet. Darauffolgend findet eine Erweiterung des Modells statt, um ein dynamisches BN zu erstellen, welches in der Lage ist, temporäre Einflüsse zu berücksichtigen. Abschließend wird das Modell validiert und die Bewertung der Methodik diskutiert.

2 Theoretische Grundlagen

Dieses Kapitel widmet sich den theoretischen Grundlagen des Bereichs Instandhaltung. Zunächst werden allgemeine Definitionen und Begriffe erklärt. In Folge dessen gibt eine Präsentation des State-of-the-Art Aufschluss über den derzeitigen Stand und auch Entwicklungen im Bereich der Instandhaltung.

2.1 Instandhaltung und Instandhaltungsstrategien

Der Fachbereich der Instandhaltung galt lange Zeit als eher unwichtig und bekam demnach sowohl im industriellen, als auch im Bereich der Forschung wenig Aufmerksamkeit geschenkt.¹⁰ Durch die Erhöhung der Komplexität technischer Systeme und Anlagen stiegen die Instandhaltungskosten aufgrund der Verfolgung von reaktiven Instandhaltungsstrategien stark an, was letztendlich zu einem Paradigmenwechsel führte.¹¹

2.1.1 Definition Instandhaltung

Instandhaltung konstituiert sich als ein Verteidigungssystem gegen Schäden und wird laut DIN 31051 folgend definiert:

„Kombination aller technischen und administrativen Maßnahmen des Managements während des Lebenszyklus einer Betrachtungseinheit zur Erhaltung des funktionsfähigen Zustands oder der Rückführung in diesen, sodass sie die geforderte Funktion erfüllen kann.“¹²

Des Weiteren kann der Bereich Instandhaltung in die Maßnahmen Wartung, Inspektion, Instandsetzung und Verbesserungen eingeteilt werden (siehe Abbildung 1).¹³

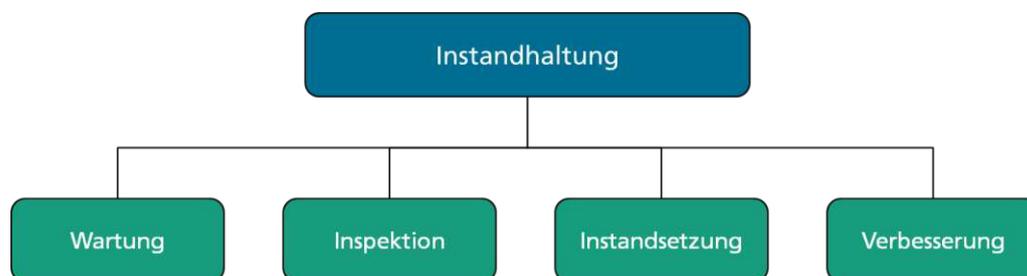


Abbildung 1: Unterteilung der Instandhaltung nach DIN 31051¹⁴

¹⁰ vgl. Acatech, 2015, S.17ff

¹¹ vgl. Glawar et al., 2018, S.49

¹² DIN EN 31051:2012, 2012, S.4

¹³ vgl. ebenda

¹⁴ vgl. ebenda

Im Folgenden sollen auf die oben genannten Kategorien eingegangen werden, die in DIN 31051 definiert werden:¹⁵

- **Wartung:** „Maßnahmen zur Verzögerung des vorhandenen Abnutzungsvorrats.“
- **Inspektion:** „Maßnahmen zur Feststellung und Beurteilung des Istzustandes einer Einheit einschließlich der Bestimmung der Ursachen der Abnutzung und dem Ableiten der notwendigen Konsequenzen für eine künftige Nutzung.“
- **Instandsetzung:** „Physische Maßnahme, die ausgeführt wird, um die Funktion einer fehlerhaften Einheit wiederherzustellen.“
- **Verbesserung:** „Kombination aller technischen und administrativen Maßnahmen sowie Maßnahmen des Managements zur Steigerung der Zuverlässigkeit und/oder Instandhaltbarkeit und/oder Sicherheit einer Einheit, ohne ihre ursprüngliche Funktion zu ändern.“

2.1.2 Ziele der Instandhaltung

In DIN 13306 ist der Begriff Instandhaltungsziele folgend festgelegt:

„Zugewiesene und akzeptierte Ziele für die Instandhaltungstätigkeiten.“¹⁶

Wie die meisten Unternehmensbereiche hat auch die Instandhaltung das Ziel, die betrieblichen Gesamtkosten weitestgehend zu minimieren. Folglich ist es notwendig, unter Zuhilfenahme einer geeigneten Strategie ein Optimum zwischen den Kosten vorbeugender Maßnahmen und Maschinenausfallkosten zu finden.¹⁷ Dieser äußerst wichtige Zusammenhang wird in Abbildung 2 gezeigt.

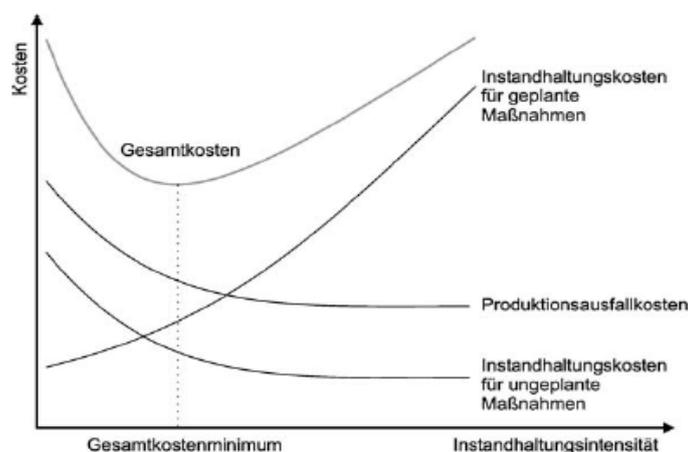


Abbildung 2: Trade-Off zwischen Instandhaltungskosten und -intensität¹⁸

¹⁵ DIN EN 31051:2012, S.5f

¹⁶ DIN EN 13306:2018-02, 2018, S.9

¹⁷ vgl. Matyas, 2016, S.48

¹⁸ vgl. Palm, 1981, S.258

Folglich resultiert, dass jedes Unternehmen die Ziele selbst bestimmen muss. Das Instandhaltungsmanagement hat dabei die Verantwortung, die richtigen Strategien zum Erreichen dieser Ziele zu wählen. Es gilt jedoch zwischen den Zielvorgaben und den gewählten Strategien, die auch aufgrund von sich ändernden Umgebungsbedingungen angepasst werden können, zu unterscheiden.¹⁹

Somit kann man die Hauptziele des Bereiches Instandhaltung folgend untergliedern:²⁰

- Gewährleistung der Sicherheit und Anlagenverfügbarkeit
- Kostenminimierung bzw. Gewinnmaximierung

Unter Gewährleistung der Sicherheit wird dabei sowohl das Vermeiden von ungeplanten Stillstandszeiten als auch die Vermeidung von Unfällen, die zu Umwelt- oder auch Personenschäden führen können, verstanden.²¹

2.1.3 Instandhaltungsstrategien

In der Literatur finden sich allgemein mehrere Definitionen zum Thema Instandhaltungsstrategie. Laut „Deutscher Industrienorm“ DIN EN 13306 ist der Begriff der Instandhaltungsstrategie wie folgt definiert:

„Vorgehensweise des Managements zur Erreichung der Instandhaltungsziele.“²²

Auf diese Definition aufbauend folgt schließlich, dass unter dem Begriff der Instandhaltungsstrategie ein Regelwerk verstanden werden kann, das klar festlegt in welchem Umfang an welchem Objekt spezifische Maßnahmen in welcher zeitlichen Folge durchzuführen sind.²³

2.1.3.1 Unterteilung der Instandhaltungsstrategien

Die Strukturierung verschiedener Instandhaltungsstrategien kann aufgrund ihrer Planmäßigkeit und der zugehörigen Vorbeugungswirkung erfolgen. Grundsätzlich erfolgt die Unterteilung der Strategien in planmäßig oder nicht planmäßig. Im Bereich der planmäßigen Instandhaltung gilt es zudem, die beiden Grundstrategien der geplanten Ausfallstrategie und der Strategie der planmäßig vorbeugenden Instandhaltung zu unterscheiden.²⁴ Diese werden in der Literatur auch häufig als reaktive und präventive Instandhaltungsstrategie bezeichnet (siehe Abbildung 3).

¹⁹ vgl. Warnecke and Bauer, 1992, S.33ff

²⁰ vgl. Biedermann, 2008, S.13

²¹ vgl. Matyas, 2016, S.120f

²² DIN EN 13306:2018-02, 2018, S.9

²³ vgl. Prüß and Nebl, 2006, S.730

²⁴ vgl. Slaby and Krasselt, 1998, S.94 ff

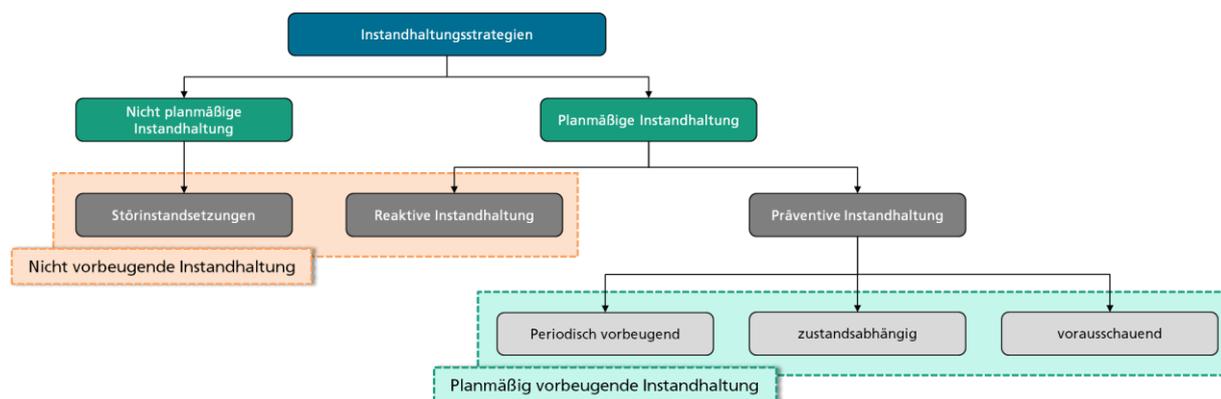


Abbildung 3: Unterscheidung von Instandhaltungsstrategien^{25,26}

▪ Reaktive Instandhaltungsstrategie

Die Strategie der reaktiven Instandhaltung sieht eine Reaktion bzw. eine Maßnahme vor, nachdem bereits ein Ausfall durch das Erreichen einer festgelegten Schadensgrenze, einer Komponente erfolgt ist. Zu beachten gilt, dass keine zwischenzeitlichen Wartungen oder auch Inspektionen vorhergesehen sind. Reaktive Strategien führen in der Regel zu den höchsten Ausfallzeiten und -kosten und bergen aufgrund der Unplanbarkeit den Nachteil, dass keine garantierten Aussagen bezüglich der Anlagenverfügbarkeit getroffen werden können. Demzufolge ist diese Strategie nur geeignet, wenn das betrachtete System Redundanzen aufweist oder genug Ressourcen verfügbar sind und somit außerordentlich schnell auf Ausfälle reagiert werden kann.²⁷

▪ Präventive Instandhaltungsstrategie

Die präventive Instandhaltungsstrategie verfolgt den Ansatz, Bauteile auszutauschen bevor diese einen Ausfall verursachen. Hierbei kann in

- periodisch vorbeugend,
- zustandsabhängig,
- und vorausschauend

unterschieden werden.²⁸

Diese drei Strategien, die auch unter dem Terminus der planmäßig vorbeugenden Instandhaltung bekannt sind, sollen im Weiteren kurz erläutert werden.

Periodisch vorbeugende Instandhaltung: Unter periodisch vorbeugender Instandhaltung versteht man das Überholen und Austauschen bestimmter Bauteile in Abhängigkeit des erreichten Bauteilalters (Lebensdauer). Dies geschieht unabhängig

²⁵ vgl. ebenda, 1998, S. 95

²⁶ vgl. Schenk, 2010, S.27

²⁷ vgl. Schenk, 2010, S.27f

²⁸ vgl. Matyas, 2016, S.120

vom tatsächlichen Zustand der Komponente, woraus abgeleitet werden kann, dass sich diese Herangehensweise besonders eignet, wenn im Zuge von Ausfällen Auswirkungen auf Sicherheit und Umwelt befürchtet werden. Auch wenn die approximierte Lebensdauer bekannt ist und der Großteil der anderen Maschinen- oder Anlagenteile ebenfalls bis zu diesem Zeitpunkt funktionstüchtig bleibt, kann auf diese Strategie zugegriffen werden.^{29,30}

Zustandsabhängige Instandhaltung: Diese Strategie, die auch unter „condition monitoring“ bekannt ist, basiert auf der periodischen oder kontinuierlichen Ermittlung des technischen Zustands der Anlage.³¹ Die abgeleiteten Maßnahmen orientieren sich folglich am Abnutzungsvorrat der betrachteten Anlage, wobei davon ausgegangen wird, dass ein Großteil von Funktionsstörungen nicht abrupt, sondern über einen gewissen Zeitraum entstehen und sich dabei durch Warnsignale ankündigen. Die in Abbildung 4 gezeigte Grafik visualisiert die PF-Kurve, die den Zusammenhang zwischen Zustand und zeitlichem Verlauf darstellt. P markiert dabei den Punkt, an dem erkannt wird, dass sich eine Störung entwickelt, F den Zeitpunkt an dem eine Störung auftritt.³²

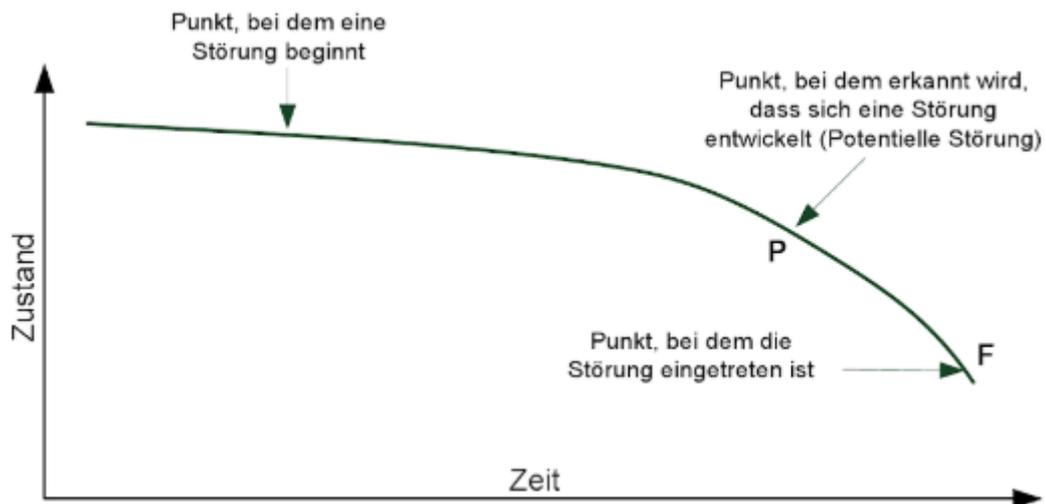


Abbildung 4: PF-Kurve³³

Vorausschauende Instandhaltung: Vorausschauende Instandhaltung adressiert die Problematik verdeckter Störungen, die im Zuge der zustandsorientierten Instandhaltung unerkannt bleiben. Demnach zielt diese Strategie darauf ab, Störungen zu verhindern bevor sie auftreten. Bei durchzuführenden Maßnahmen muss die technische Machbarkeit und der Kosten/Nutzen-Faktor bewertet werden. Abbildung 5

²⁹ vgl. Matyas, 2002, S.14

³⁰ vgl. Spickenheuer, 1978, S.337ff

³¹ vgl. Sturm and Förster, 1990, S.18ff

³² vgl. Matyas, 2016, S.125

³³ vgl. ebenda

zeigt den Ansatz zur Auswahl von Instandhaltungsmaßnahmen für eine vorausschauende Instandhaltung nach Moubray.³⁴

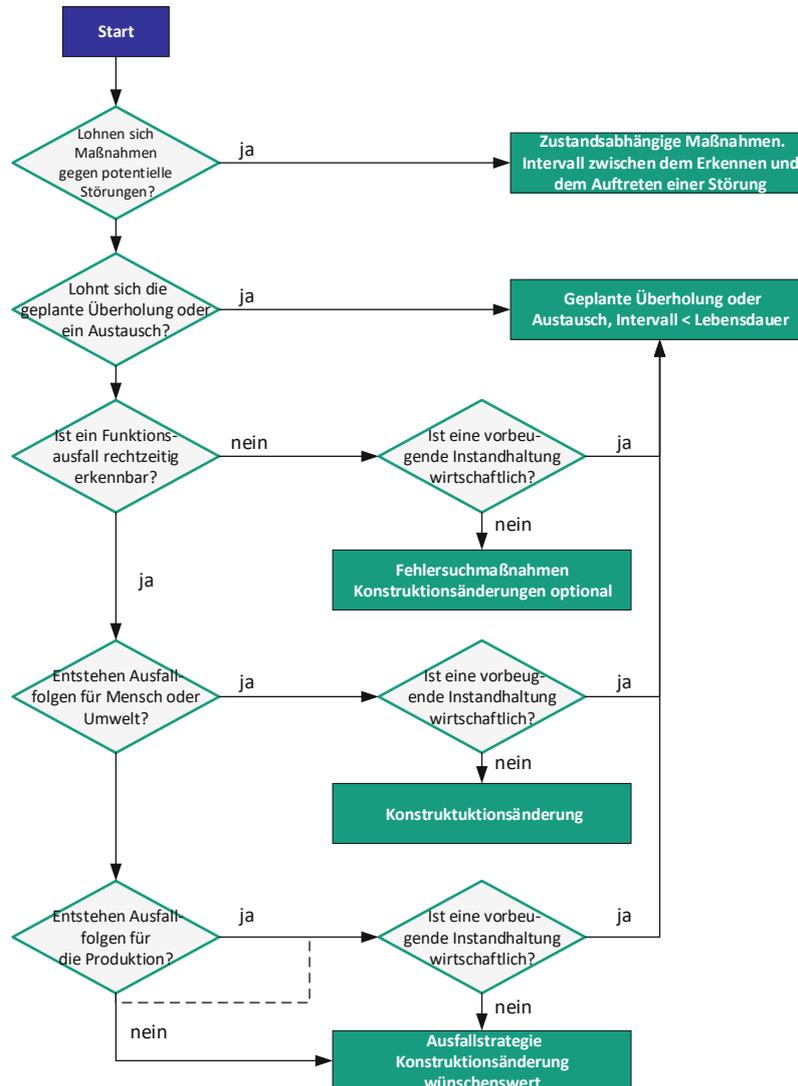


Abbildung 5: Empfohlene Auswahl von Instandhaltungsmaßnahmen für die vorausschauende Instandhaltung³⁵

2.1.3.2 Planung der Instandhaltungsstrategie

Wie bereits erwähnt existieren in der Literatur mehrere Definitionen zum Thema Instandhaltungsstrategie, woraus folgt, dass auch im Bereich der Planung verschiedene Ansätze verfolgt und beschrieben werden. Demnach ergibt sich, dass die Planung der Instandhaltungsstrategie ein breites Spektrum in der betrieblichen Praxis abbildet.³⁶

Biedermann bezeichnet die Instandhaltungsziele als Grundlage für die abzuleitenden Strategien. Um diese zu erreichen, muss eine strategische Instandhaltungsplanung

³⁴ Moubray and Kugler, 1996

³⁵ vgl. ebenda

³⁶ vgl. Hölbfer, 2014, S.36

erfolgen, um das optimale Instandhaltungsprogramm auswählen zu können. Somit bildet die Strategieplanung die Grundlage der operativen Durchführungsplanung. Der zugehörige Regelkreis der Instandhaltungsplanung wird in Abbildung 6 gezeigt.³⁷

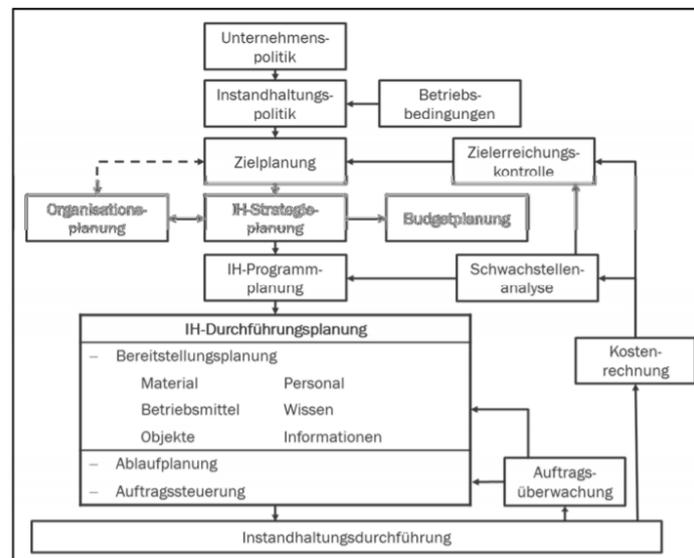


Abbildung 6: Regelkreis der Instandhaltung³⁸

Matyas, K. empfiehlt bei der Einführung eines umfassenden Instandhaltungskonzepts einen optimalen Mix aus Ausfallbehebung, vorbeugender Instandhaltung und vorausschauender Instandhaltung, wobei die Art der Instandhaltungsstrategie auf Basis einer umfangreichen Analyse, bei der unterschiedliche Kriterien bewertet werden, festgelegt wird.³⁹

2.2 State-of-the-Art im Bereich der Instandhaltung

Im Laufe der letzten 80 Jahren hat sich die Instandhaltung aufgrund neuer technischer Errungenschaften und Wissenszuwachs in vielen Teilbereichen stark weiterentwickelt. War es in den 1940er Jahren noch üblich, ausschließlich Instandhaltungsmaßnahmen vorzusehen sobald ein ungeplanter Ausfall eintrat, werden heutzutage fast ausschließlich die Ansätze der „prädiktiven“ und „präskriptiven“ Instandhaltung verfolgt. Abbildung 7 visualisiert die Entwicklung der Instandhaltungstechniken der letzten 80 Jahre.

³⁷ vgl. Biedermann, 1990, S.320

³⁸ ebenda, S.320

³⁹ vgl. Matyas, 2016, S.120

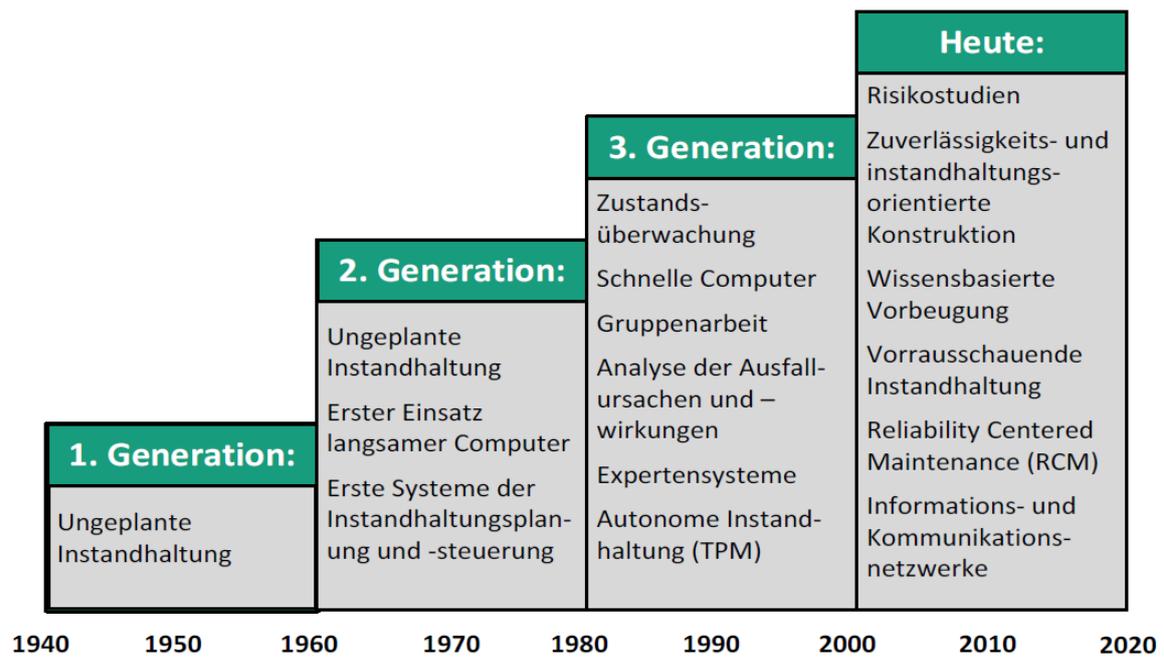


Abbildung 7: Entwicklung der Instandhaltungstechniken⁴⁰

Die Ansätze der prädiktiven und präskriptiven Instandhaltung, welche zum Bereich der vorausschauenden Instandhaltungsstrategien zählen, stehen derzeit, sowohl im Bereich der theoretischen als auch angewandten Forschung im Mittelpunkt und spiegeln somit den aktuellen State-of-the-Art wider. Demnach sollen diese beiden Ansätze in den nachfolgenden Kapiteln genauer beleuchtet werden. Zusätzlich wird anfangs ein Einblick in die Wissensbasierte Instandhaltung gegeben.

2.2.1 Wissensbasierte Instandhaltung

Der Übergang der Produktion in Richtung Industrie 4.0 und die Entwicklung „Cyber Physical Production Systems“ hat auch im Bereich der Instandhaltung zu einem Paradigmenwechsel von deskriptiver zu prädiktiver und präskriptiver Instandhaltung geführt, die im Bereich der „Wissensbasierten Instandhaltung“ (KBM Knowledge Based Maintenance) den höchsten Reife- und Komplexitätsgrad darstellen. KBM zufolge reicht die detaillierte Betrachtung relevanter Komponenten nicht aus, um Prognosen über Instandhaltungszeitpunkte und damit verbundenen Kosten zu generieren. Folglich muss, um korrekte und zuverlässige Aussagen über zukünftige Events treffen zu können, eine ganzheitliche Betrachtung von Produktionsprozessen vorausgesetzt werden.^{41,42} Daraus konstituiert sich das zentrale Ziel der „Wissensbasierten Instandhaltung“ in der Entwicklung eines generischen Konzepts um Instandhaltungsprozesse durch die ganzheitliche Betrachtung von Systemzuständen,

⁴⁰ Moubray and Kugler, 1996

⁴¹ vgl. Zülch et al., 2005, S.187ff

⁴² vgl. Ansari et al., 2017, S.1ff

der Organisation und Prozessen, zu optimieren.⁴³ Abbildung 8 zeigt die Entwicklungsstufen von KBM.

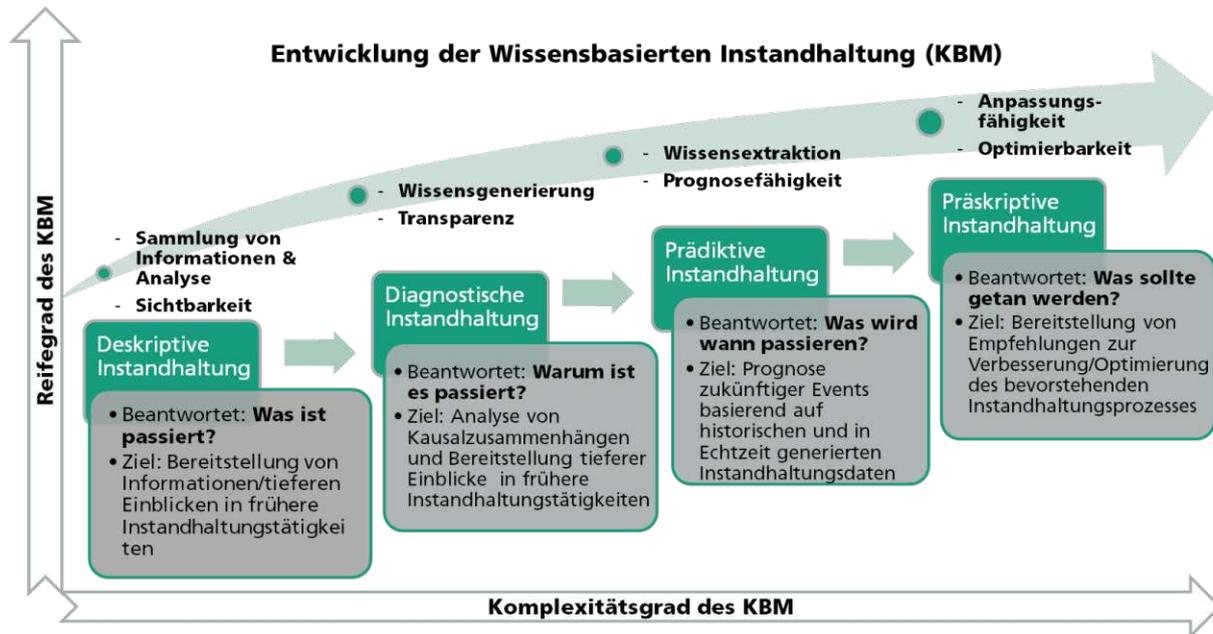


Abbildung 8: Entwicklung der Wissensbasierten Instandhaltung⁴⁴

In der Literatur wird in Bezug auf KBM zwischen mehreren Vorgehensweisen unterschieden, die in 2 Hauptgruppen unterteilt werden können:⁴⁵

- **Knowledge-Driven Ansätze:** Bei dieser Gruppe handelt es sich um wissensgesteuerte Ansätze (d.h. es werden eine Reihe von Methoden zum Data-Mining, maschinellen Lernen etc. angewendet) und anhand von diagnostischen, prädiktiven oder präskriptiven Instandhaltungsansätzen inklusive Instandhaltungsplanung, -überwachung und -aktivitäten, evaluiert.
- **Integrated Knowledge-Intensive Approaches or Reference Models:** Zu dieser Gruppe gehören Ansätze, die das Ziel der multiplen Datenverknüpfung und -kombination verschiedener Datenquellen und Wissensquellen verfolgen, um somit die Intelligenz des Instandhaltungssystems zu erhöhen.

2.2.2 Prädiktive Instandhaltung

Das Ziel der prädiktiven Instandhaltung stellt die Prognose zukünftiger Events dar, wobei die Vorhersage auf der Verarbeitung von Daten (historische und Echtzeitdaten) in dafür angefertigten „Predictor-Modellen“ beruht. Diese Art der Instandhaltung wird häufig als „Smart-Maintenance“ oder auch „Maintenance 4.0“ bezeichnet.⁴⁶ Prinzipiell ist die prädiktive Instandhaltung dem Bereich der vorausschauenden Instandhaltung

⁴³ vgl. Pawellek, 2013, S.9f

⁴⁴ vgl. Ansari et al., 2017, S.2

⁴⁵ vgl. ebenda

⁴⁶ vgl. Mobley, 2002, S.4ff

zuzuordnen, bei dem sowohl der aktuelle Zustand der Maschine als auch der momentane Produktionsplan und die damit verbundenen Belastungen berücksichtigt werden.⁴⁷ Auf Basis dieser Informationen wird versucht, einen optimalen Wartungsplan zu erstellen, bei dem diverse Softwarelösungen dazu verwendet werden, Parameter zu optimieren und zusätzlich als Entscheidungsunterstützung zu fungieren.⁴⁸

Eine große Schwierigkeit bei der Erstellung einer ganzheitlichen, abteilungsübergreifenden Instandhaltungsplanung stellt die Interaktion der drei Bereiche Qualitätsmanagement, Instandhaltung und Produktionsplanung- und -steuerung dar.⁴⁹ Die starke Verknüpfung der genannten Bereiche und das daraus resultierende Spannungsfeld wurden mehrmals in der Literatur nachgewiesen.⁵⁰ Folglich berücksichtigen viele Planungsmodelle nicht alle drei Bereiche, weshalb eine Unterscheidung in zwei Gruppen erfolgt:⁵¹

- **Interaktive Modelle:** Diese bestreben die Optimierung einer definierten Zielfunktion unter der Berücksichtigung weiterer Funktionen. Die Erfordernisse der Nebenfunktionen stellen dabei Restriktionen für das Modell dar.
- **Integrierte Modelle:** Diese Modelle zielen auf die Optimierung zwei oder mehrerer Elemente zur selben Zeit ab.

In Abbildung 9 wird das Spannungsfeld der drei Bereiche schematisch zusammengefasst.

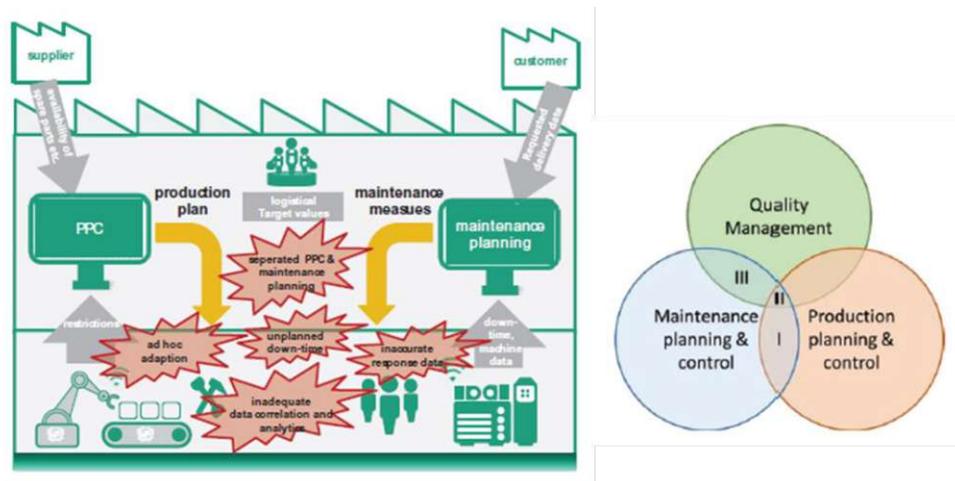


Abbildung 9: Problematik einer abteilungsübergreifenden Instandhaltungsplanung⁵²

⁴⁷ vgl. Hoppenstedt et al., 2017, S.21f

⁴⁸ vgl. Reichel et al., 2018, S.356ff

⁴⁹ vgl. Glawar et al., 2018, S.46ff

⁵⁰ vgl. Aghezzaf et al., 2016, S.190ff

⁵¹ vgl. Hadidi et al., 2012, S.25ff

⁵² vgl. Glawar et al., 2018, S.47

Eine weitere Herausforderung bei der Etablierung eines ganzheitlichen Instandhaltungsplanungssystems, welches die Kosten- und Aufwandsminimierung zum Ziel hat, stellt die Verknüpfung verschiedenster Software-Pakete dar. Meist gibt es nur unzureichende Anbindungen und noch viel häufiger müssen neue Schnittstellen geschaffen werden, die eine weitreichende Vernetzung der IT-Landschaft ermöglichen. Dementsprechend können die Systeme zur Zustandsüberwachung, Planung und Durchführung von Instandhaltungsmaßnahmen meist nicht ganzheitlich verwendet werden. Um dieser Problematik gegenüber treten zu können, wurde im Rahmen eines europäischen Forschungsprojektes mit dem Namen „SUPREME - Sustainable Predictive Maintenance for Manufacturing Equipment“ eine Architektur für zukünftige Instandhaltungssysteme entworfen (siehe Abbildung 10).⁵³

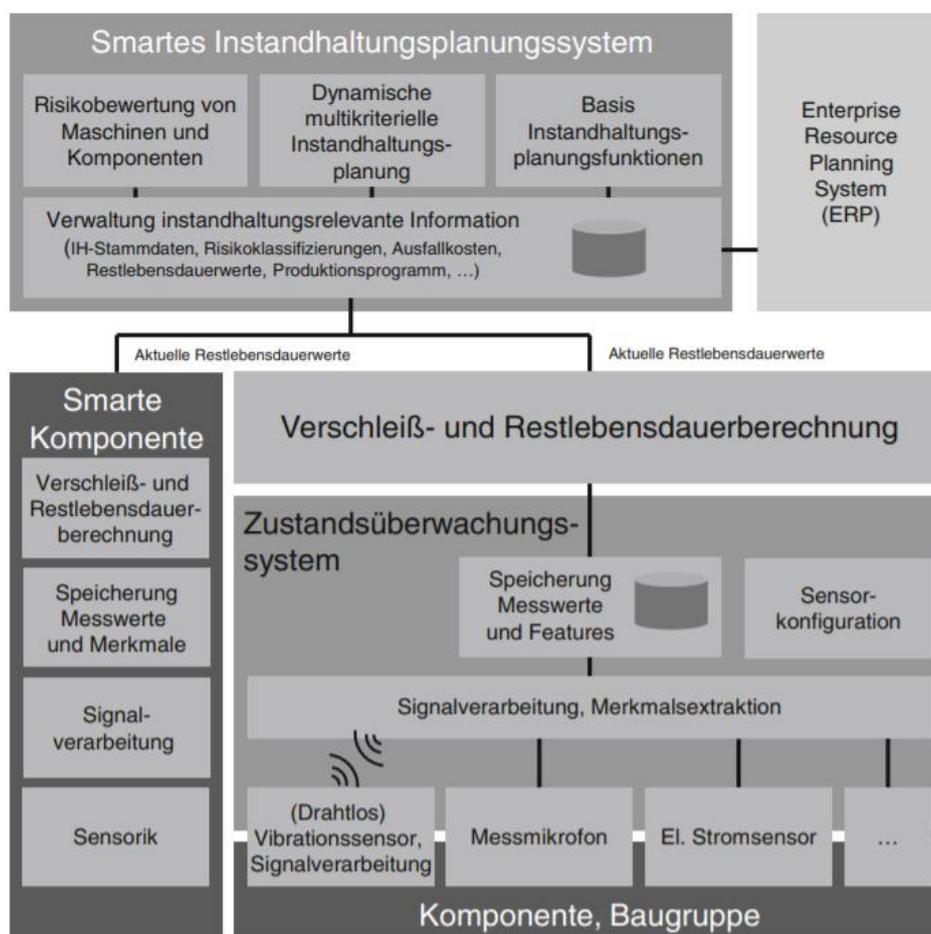


Abbildung 10: Architektur eines zukünftigen Instandhaltungsplanungssystems⁵⁴

- **Instandhaltung 4.0 – Entwicklung eines Instandhaltungsleitstands**

Im Zuge eines an der TU Wien durchgeführten Projektes wurde eine Herangehensweise entwickelt, die eine Verknüpfung der Maschinen-, Prozess-, und Produktebene ermöglicht, um somit die Basis eines integrativen

⁵³ vgl. Vogel-Heuser et al., 2017, S.81ff

⁵⁴ vgl. ebenda, 2017, S.83

Instandhaltungsansatzes darstellt. Die Vorgehensmethode bei der Entwicklung dieses neuen Ansatzes wird in Tabelle 1 dargestellt:

Tabelle 1: Methodik des Instandhaltungsleitstands⁵⁵

Schritt 1: Framework Entwicklung
<ul style="list-style-type: none"> Abbildung der Produktionsanlagen auf Komponentenebene mit zugehörigen Lastprofilen Identifikation und Übertragung von Produktionszustand einer Realanlage auf entwickelte Systematik
Schritt 2: Datenanalyse- und Simulationsstudie
<ul style="list-style-type: none"> Auswahl und Vorbereitung der historischen Daten (Maschinen-, Produkt-, Prozessdaten), Condition Monitoring Daten und Belastungsdaten Untersuchung von instandhaltungsrelevanten Daten, um Rückschlüsse auf Last-induzierte Abnutzung und Qualitätsabweichungen zu erhalten
Schritt 3: Identifikation von Belastungseinflüssen
<ul style="list-style-type: none"> Klassifikation und Korrelation unterschiedlicher Belastungsszenarien Ableitung, Validierung und Verdichtung der Belastungseinflüsse
Schritt 4: Entwicklung eines integrativen IH-Ansatzes
<ul style="list-style-type: none"> Ableitung von allgemein gültigen Regeln Implementierung und Testlauf der Reaktionsmodelle im Leitstand

Ergebnis dieses Forschungsprojektes ist ein Instandhaltungsleitstand, der in der Lage ist, auf Basis von Qualitätsdaten, des aktuellen Produktionsplans und in der Software hinterlegten Regeln, die mittels eines Korrelations- und Verdichtungsmodell generiert wurden, verlässliche Prognosen bezüglich zukünftiger Ausfallzeitpunkte zu erstellen (siehe Abbildung 11).⁵⁶

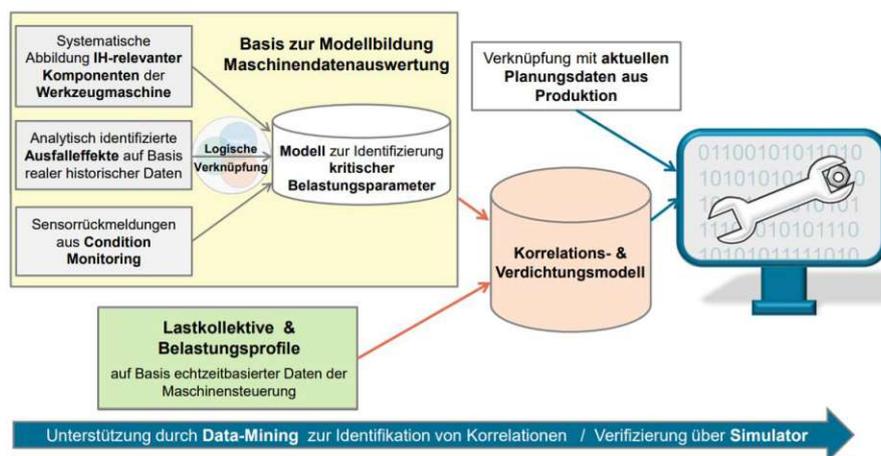


Abbildung 11: Verknüpfung von Daten in der „Instandhaltung 4.0“⁵⁷

⁵⁵ vgl. Nemeth et al., 2015, S.570

⁵⁶ vgl. ebenda, 2015, S.569-573

⁵⁷ vgl. Matyas, 2016, S.141

2.2.3 Präskriptive Instandhaltung

Im Gegensatz zum Ansatz der prädiktiven Instandhaltung, dessen Hauptziel die Prognose zukünftiger Ausfälle ist, beschäftigt sich die präskriptive oder auch vorschreibende Instandhaltung mit der Frage „Was soll getan werden?“, um Empfehlungen zur Verbesserung und Optimierung anstehender Instandhaltungsprozesse bereitstellen zu können.⁵⁸ Demzufolge reicht das Konzept der präskriptiven Instandhaltung weit über die Prognose zukünftiger Ausfälle hinaus. Durch die Anwendung moderner Sensorsysteme und das vorhandene Expertenwissen der Prozesstechniker in Kombination mit, auf historischen oder Echtzeitdaten basierender, prädiktiver Datenanalyse, können nicht nur Aussagen bezüglich des nächsten Ausfallzeitpunkts getroffen werden, sondern zusätzlich Empfehlungen für die optimale Maßnahme abgegeben werden.^{59,60} Die Anwendung von präskriptiver Instandhaltung führt somit dazu, dass Anlagen in Kollaboration mit ihren Operatoren proaktiv an ihrer eigenen Instandhaltung partizipieren.⁶¹ Im Folgenden werden einige Ansätze zur Umsetzung von präskriptiver Instandhaltung präsentiert.

- **Prescriptive Maintenance Model PriMa⁶²**

Das von Ansari et al. vorgeschlagene Rahmenwerk, zur erfolgreichen Implementierung von präskriptiver Instandhaltung besteht aus der horizontalen und vertikalen Integration mehrerer miteinander verlinkter Systeme. In der untersten Ebene steht ein Datawarehouse, welches mit Prozess-, Produkt- und Maschinendaten angereichert wird. Diese drei Dimensionen sind vergleichbar mit dem horizontalen und vertikalen Datenstrom von „Cyber Physical Production Systems“ (CPPS). Während die horizontale Struktur des Modells die Datensammlung abbildet, gestaltet die vertikale Struktur die Verknüpfung und Auswertung der erhobenen Daten. Mittels der in der mittleren Ebene abgebildeten Meta-Analytic & Learning Toolbox werden die gesammelten Daten aggregiert und korreliert. Dies fließt wiederum eine Stufe höher in den Bereich der „Predictive Analytics“ ein, welche die Basis für das PriMa Empfehlungs-Cockpit darstellt. Eine Feedback Schleife, die als Rückkopplung zwischen der Meta-Analytic & Learning Toolbox und der Predictive Analytics Ebene fungiert, soll im Weiteren dazu dienen dynamisch lernende Netzwerke zu erstellen, die den Entscheidungsfindungsprozess mittels neu dazugewonnenem Wissens, unterstützen und verbessern.

⁵⁸ vgl. Ansari et al., 2017, S.1

⁵⁹ vgl. ebenda

⁶⁰ vgl. Fischer, 2015

⁶¹ vgl. Khoshafian and Rostetter, 2015, S.6f

⁶² vgl. Ansari et al., 2017, S.4ff

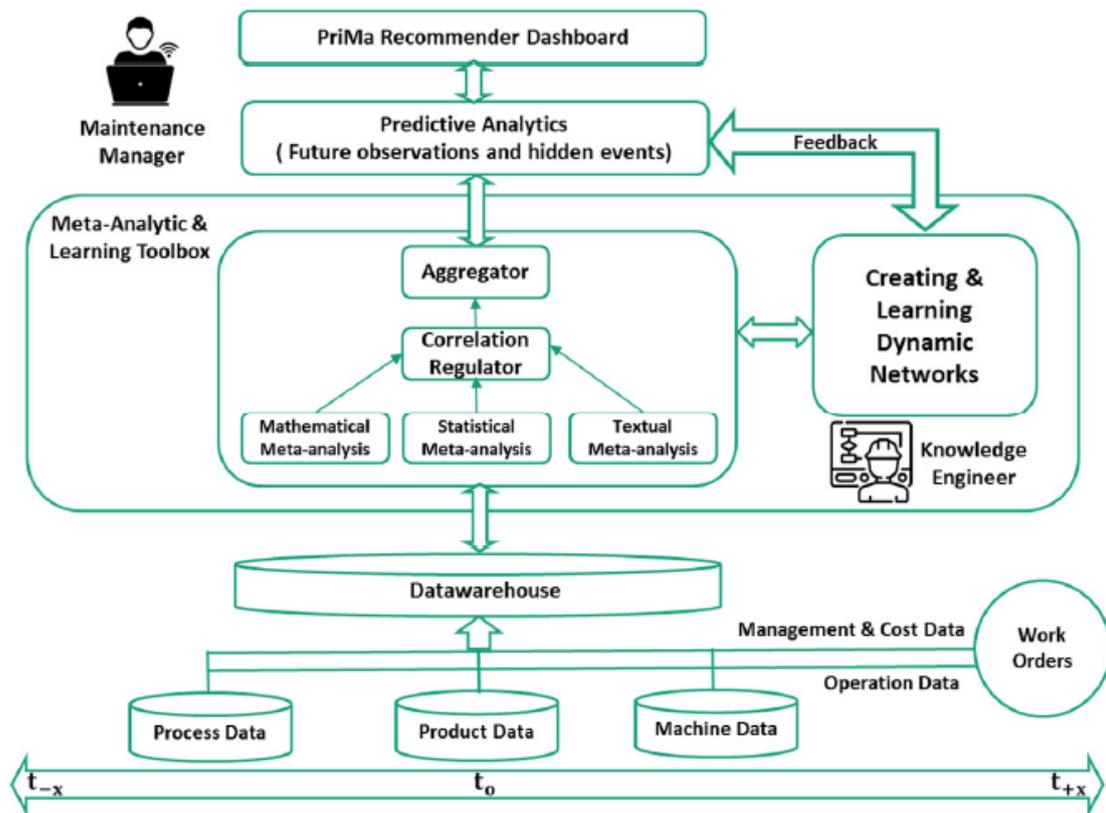


Abbildung 12: PriMa Modell für präskriptive Instandhaltung⁶³

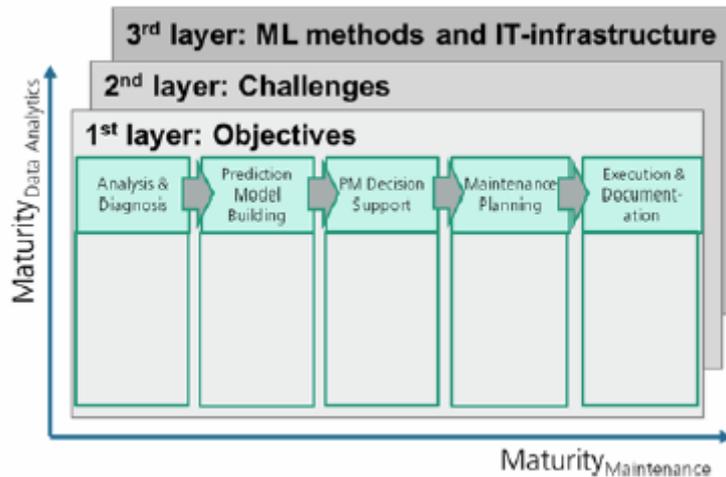
- **PriMa-X Referenzmodell⁶⁴**

Das PriMa-X Referenzmodell, welches von Nemeth et al. entwickelt wurde, konstituiert sich als eine drei schichtige Matrix, die Ziele und Herausforderungen adressiert, sowie Machine-Learning Methoden und notwendige IT-Strukturen aufschlüsselt, die sich als essentiell für eine erfolgreiche Implementierung von präskriptiver Instandhaltung in bestehende Produktionssysteme darstellen. Jeder Layer des Modells ist in die folgenden Phasen untergliedert (siehe Abbildung 13):

- Analyse und Diagnose
- Prediction Modell Building
- PM Decision Support
- Maintenance Planning
- Execution und Documentation

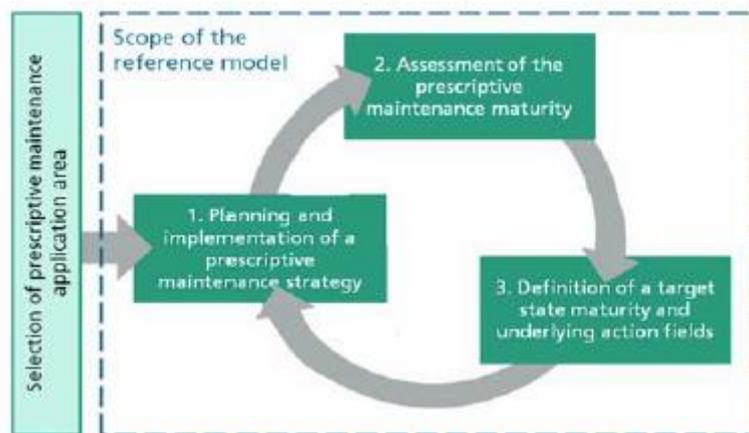
⁶³ vgl. ebenda, S.4

⁶⁴ vgl. Nemeth et al., 2018, S.1039-1044

Abbildung 13: PriMa-X Portfolio Matrix⁶⁵

Das Referenzmodell baut auf drei iterativen Schritten auf (siehe Abbildung 14):

- Planen und implementieren einer präskriptiven Instandhaltungsstrategie
- Beurteilung des Reifegrades der präskriptiven Instandhaltung
- Festlegung des Zielzustandes bezüglich des zukünftigen Reifegrads und Definition der zugehörigen Handlungsfelder

Abbildung 14: Iterativer Zyklus des PriMa-X Referenzmodells⁶⁶

Somit stellt das entwickelte Referenzmodell einen umfassenden Ansatz zur systematischen Realisierung einer präskriptiven Instandhaltungsstrategie dar, der sowohl den operativen, als auch den Management Bereich eines Produktionssystems adressiert.

⁶⁵ vgl. ebenda, S.1042

⁶⁶ vgl. ebenda, S.1042

3 Generierung und Aufbereitung von Daten

Im folgenden Kapitel soll das wichtige Thema Wissenserschließung auf Basis von Daten behandelt werden. Des Weiteren erfolgt ein Exkurs zum Thema Datenqualität und Informationsqualität. Abschließend wird eine standardisierte Methodik zur Generierung und Aufbereitung von Daten präsentiert (CRISP-DM Methodik).

3.1 Wissenserschließung und Entscheidungsfindung

Unter Wissenserschließung (häufig unter dem englischen Namen „knowledge discovery“ bekannt) versteht man die nicht triviale Extraktion von impliziten, zuvor unbekanntem und potentiell brauchbaren Informationen aus Daten.⁶⁷ Aus der Literatur kann man eruieren, dass „Knowledge Discovery Systems“ eine Vielzahl von Informationen auf Basis von Mustererkennung generieren können, die sich für den End-User jedoch häufig als unbrauchbar herausstellen. Ein Ansatz zur Lösung dieser Problematik stellt das Messen des Interessanztheitsgrades von spezifischen Mustern auf Basis der Struktur und der hinterlegten Daten im Entdeckungsprozess dar.⁶⁸ Die Fähigkeit schnell und effizient die Ursachen für Fehler bestimmen und darauf aufbauend optimale Instandhaltungsmaßnahmen, ohne der Notwendigkeit menschlicher Intervention, vorzuschlagen zu können, wird in zukünftigen Produktionsumgebungen unerlässlich sein. Folglich gestaltet es sich vor allem für komplexe Anlagen als essentiell, ausreichend Informationen in Form von Daten zu sammeln um auf Basis von Auswertungen Entscheidungen treffen zu können. Problematisch ist hierbei die Tatsache, dass die benötigten Informationen meist nicht kumuliert in einem System vorliegen, sondern über mehrere, aufgrund von fehlenden Schnittstellen oft schwer zugänglichen Speicherorten verteilt sind.⁶⁹ Die Zusammenführung von Daten aus verschiedenen Systemen mit dem Ziel Schlussfolgerungen bezüglich spezifischer Zusammenhänge treffen zu können, ist unter dem Namen „Data Fusion“ bekannt und stellt eine zentrale Komponente bei der Verarbeitung von Daten aus mehreren Quellen dar.^{70,71} Karim et al. empfiehlt zur generischen Erschließung und Entdeckung von Wissen den in Abbildung 15 visualisierten, iterativen Prozess, der sich in folgende Schritte gliedert:⁷²

- Data Acquisition
- Data Transition
- Data Fusion

⁶⁷ vgl. Frawley, 1992, S.58

⁶⁸ vgl. Silberschatz and Tuzhilin, 1996, S.970

⁶⁹ vgl. Karim et al., 2016, S.215

⁷⁰ vgl. Liggins et al., 2009, Chapter 22.2

⁷¹ vgl. Karim et al., 2016, S.215

⁷² vgl. ebenda, S.215f

- Data Analysis
- Information Visualization

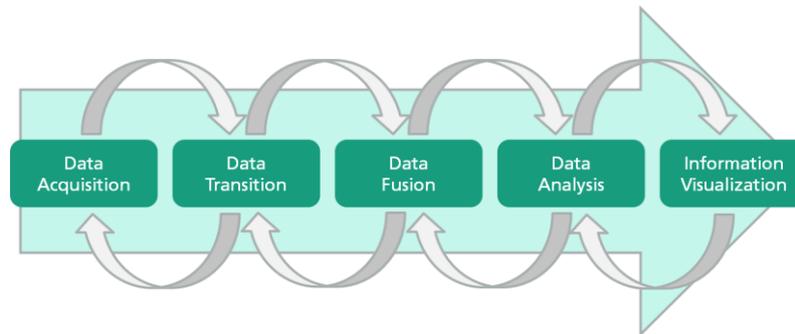


Abbildung 15: Instandhaltungs-Informations-Prozess⁷³

3.2 Daten- und Informationsqualität

Aufgrund der Tatsache, dass Daten die Grundlage für Informationen darstellen, resultiert ein kausaler Zusammenhang und ein enges Beieinanderliegen zwischen den Termini Daten- und Informationsqualität (DQ und IQ). Sowohl in Literatur als auch im täglichen Gebrauch wird häufig auf die Unterscheidung der beiden Begrifflichkeiten verzichtet, obwohl eine synonyme Verwendung streng genommen zu Ungenauigkeiten führen kann. Während Daten Fakten oder Observationen ohne Kontext repräsentieren und folglich nicht aussagekräftig sind, resultieren Informationen daraus, Daten mittels aussagekräftigen Kontext zu verknüpfen.⁷⁴ Damit stellen Daten, die in diesem Zusammenhang auch oft Rohdaten genannt werden, den Rohstoff zur Generierung von Informationen und somit ein überaus wertvolles Gut dar (siehe Abbildung 16).^{75,76}

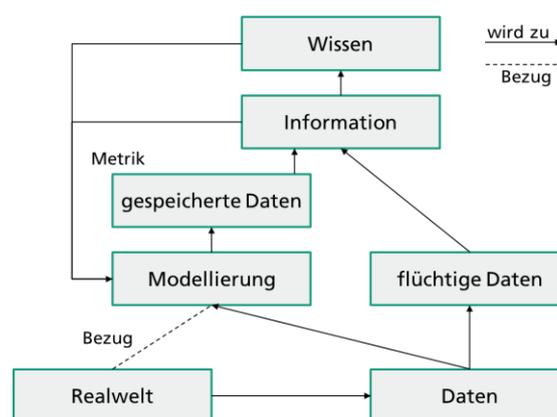


Abbildung 16: Erweitertes Begriffsschema nach Schucan^{77,78}

⁷³ vgl. ebenda, S.2

⁷⁴ vgl. Zack, 1999, S.46

⁷⁵ vgl. Batini et al., 2009, S.10

⁷⁶ vgl. Winkler, 2004, S.1

⁷⁷ vgl. Würthele, 2003, S.19

⁷⁸ vgl. Schucan, 1999, S.24

3.2.1 Datenqualität

In der Literatur können für den Begriff Datenqualität mehrere Definitionen aufgefunden werden, wobei eine allgemeingültige oder normierte Definition ausbleibt.⁷⁹ Bauer und Günzel zum Beispiel bezeichnen die Qualität als Verhältnis zwischen erreichter und erwarteter Leistung, in der die Daten sämtliche Eigenschaften aufweisen, damit alle vom Anwender definierten Anforderungen erfüllt werden können.⁸⁰ Datenqualität gestaltet sich somit als multidimensionales Konzept. Vergleichbar mit physischen Produkten ist der finale Qualitätsgrad von Daten von vorgelagerten Prozessen abhängig, mittels denen sie entworfen und erstellt worden sind.⁸¹ Wang und Wand definieren die Qualität von Daten folgendermaßen: „*The Quality of data depends on the design and production processes involved in generating the data. To design for better quality, it is necessary first to understand what quality means and how it is measured.*“⁸²

▪ Merkmale der Datenqualität

Wang und Strong befolgen bei der Festlegung des Begriffes Datenqualität ebenfalls den bereits beschriebenen Ansatz, Daten als zu erstellendes und zu bearbeitendes Produkt zu verstehen, welches bestimmten Qualitätsanforderungen entsprechen muss. Mit Hilfe dieses Ansatzes wurde ein konzeptionelles Rahmenwerk bezüglich Datenqualität entwickelt (siehe Abbildung 17).⁸³

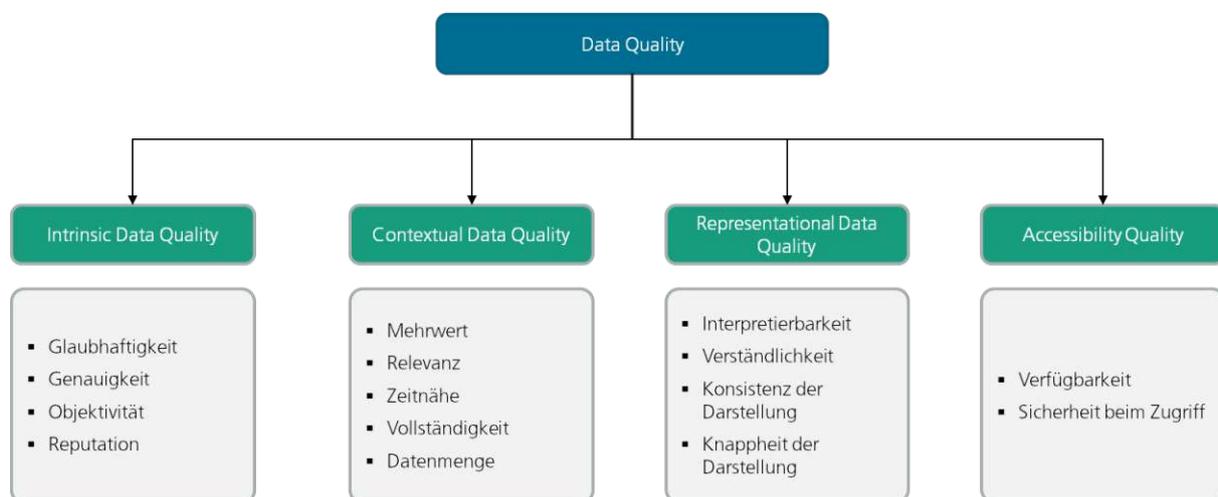


Abbildung 17: Datenqualitätsmerkmale nach Wang and Strong⁸⁴

⁷⁹ vgl. Helmig and Hollmann, 2009, S.95ff

⁸⁰ vgl. Bauer and Günzel, 2013

⁸¹ vgl. Wang and Strong, 1996, S.87ff

⁸² vgl. ebenda S.89

⁸³ vgl. Wang and Strong, 1996, S.18ff

⁸⁴ vgl. Wang and Strong, 1996, S.20

▪ Probleme bei der Gewährleistung von Datenqualität

Um die im Rahmenwerk definierten Qualitätsmerkmale einhalten zu können sind meist sogenannte „Data Cleaning“ Prozesse notwendig, um Fehler, Redundanzen oder andere Inkonsistenzen aus den Daten entfernen zu können. Hierbei gilt, dass mit zunehmender Anzahl miteinander zu verknüpfender oder integrierender Datensysteme der Aufwand des „Cleaning“ stark ansteigt. Rahm und Do teilen Datenfehler nach der in Abbildung 18 visualisierten Logik ein. In dieser wird zusätzlich unterschieden, ob Fehler in einzelnen Datenquellen vorhanden sind, oder ob diese aufgrund der Verknüpfung mit anderen Datensystemen entstanden sind.^{85, 86}

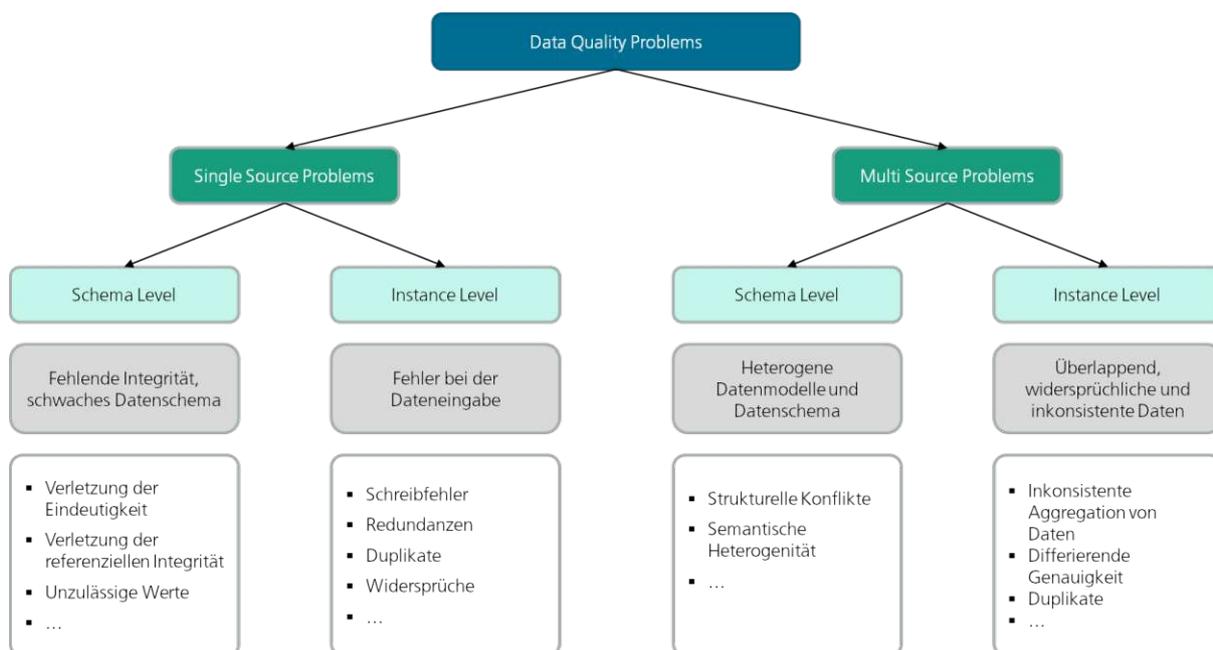


Abbildung 18: Klassifikation von Problemen der Datenqualität in Datenquellen⁸⁷

▪ Datenqualitätsradar nach Würthele⁸⁸

Im Zuge seiner Doktorarbeit mit dem Titel „Datenqualitätsmetrik für Informationsprozess: Datenqualitätsmanagement mittels ganzheitlicher Messung der Datenqualität“ entwarf Würthele ein Radar, welches in der Lage ist, die Datenqualität einer Unternehmung zu einem bestimmten Zeitpunkt t_x , zu bewerten. Hierzu erfolgt die Definition der Datenqualität als: „Mehrdimensionales Maß für die Eignung von Daten, den an ihre Erfassung/Generierung gebundenen Zweck zu erfüllen. Diese Eignung kann sich über die Zeit ändern, wenn sich die Bedürfnisse ändern.“⁸⁹

⁸⁵ vgl. Rahm and Do, 2000, S.2ff

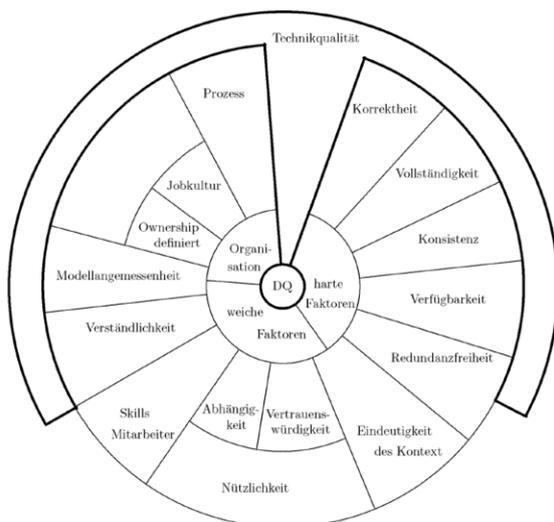
⁸⁶ vgl. Naumann, 2007, S.27ff

⁸⁷ vgl. Rahm and Do, 2000, S.3

⁸⁸ vgl. Würthele, 2003

⁸⁹ vgl. ebenda, S.21

Die kreisrunde Form des Radars ist in verschiedene Sektoren unterteilt, die unterschiedliche Qualitätsmerkmale beschreiben. Merkmale die unmittelbar nebeneinanderliegen stellen dabei Attribute dar, die eine enge Beziehung untereinander aufweisen. Zwei Drittel der Sektoren werden von der sogenannten „Technikqualität“ umfasst, die den äußersten Sektor bildet. Den Kern des Radars stellt die „Datenqualität als Ganzes“ dar, die sich im Zentrum befindet und sich in harte, weiche und organisatorische Bestandteile aufgliedern lässt. Besonderer Fokus wird in dem Modell auf die Technikqualität gelegt. Dies wird anhand des Eisbergmodells begründet. In Informatiksystemen, in denen schlechte Technikqualität aber gute Datenqualität vorliegt, werden Gefahren wie bei einem Eisberg zunächst nicht wahrgenommen (alles oberhalb der Wasseroberfläche ist in Ordnung). Mit dem Terminus Technikqualität wird dabei die langfristige gesicherte Datenqualitäts-Unterstützung verstanden. Ist diese ungenügend vorhanden, kann die Verfügbarkeit der Daten nicht gewährleistet werden. Die Verfügbarkeit der Daten ist somit direkt von der technischen Lösung abhängig.



DQ	Datenqualität
Harte Faktoren	Faktoren, die sich durch exakte Regeln prüfen und messen lassen
Weiche Faktoren	Faktoren, deren Bewertung nicht objektiv möglich ist
Organisation	Faktoren, die stark von Aufbau- und Ablauforganisation sowie von der konkreten organisatorischen Ausgestaltung des Arbeitslebens im jeweiligen Betrieb abhängig sind
Korrektheit	Gibt an, ob eine Wertausprägung richtig ist
Vollständigkeit	Gibt an, ob zu allen relevanten Datenelementen vom Leerwert verschiedene Wertausprägungen verbunden sind
Konsistenz	Gibt an, ob mehrere für sich betrachtet brauchbare Datenelemente auch im Zusammenhang korrekt sind
Verfügbarkeit	Gibt an, ob Daten innerhalb geforderter Zeit zur Verfügung stehen
Redundanzfreiheit	Gibt an, in welchem Maße ein System frei von Redundanzen ist. Redundanzen sind eine Quelle für Inkonsistenz
Eindeutigkeit des Kontexts	Korrektheit von Datenelementen alleine genügt nicht - sie müssen auch im richtigen Kontext und kontextgerecht angezeigt werden
Vertrauenswürdigkeit	Subjektives Maß für die Glaubwürdigkeit eines Systems oder einzelner oder mehrerer Datenelemente daraus
Abhängigkeit	Maß für die Notwendigkeit, dass einzelne oder mehrere Datenelemente für eine Aufgabe / einen Prozess benötigt werden
Nützlichkeit	Kombination von Abhängigkeit und Vertrauenswürdigkeit
Skills Mitarbeiter	Eignung des Fach- und Methodenwissens der Mitarbeiter gemessen an den Anforderungen an einen Standard-Benutzer
Verständlichkeit	Eignung der Datenpräsentation für Standard-Benutzer (Missverständene Daten haben die selbe Wirkung wie falsche Daten)
Modellangemessenheit	Maß der Harmonie zwischen Geschäftsprozess-Modell, Modell des Benutzers und Modell des Informatiksystems
Ownership definiert	Gibt an, ob zu jedem Datenelement ein eindeutiger Owner definiert und verantwortlich ist
Jobkultur	Gibt an, ob alle Mitarbeiter für die Belange der Datenqualität sensibilisiert sind und die negativen Folgen einer schlechten DQ in ihrem Umfeld kennen und ob sie „Qualität leben“
Prozess	Gibt an, ob klar geregelte Prozesse eine geordnete Bewirtschaftung von Daten ermöglichen.
Technikqualität	Gibt an wie gut die gewählte technische Lösung einschließlich der definierten ablauforganisatorischen Prozesse in der Lage ist, eine gute Datenqualität zu gewährleisten

Abbildung 19: Datenqualitätsradar nach Würthele⁹⁰

⁹⁰ vgl. ebenda, S.31

3.2.2 Informationsqualität

Der Begriff Informationsqualität wird von Hinrichs definiert als: „*Grad, in dem... Merkmale eines Datenproduktes Anforderungen genügen*“. ⁹¹ Dabei ist mit der Notation „Merkmal eines Datenproduktes“ beispielsweise die übersichtliche Illustration oder auch die Übereinstimmung der Datenbasis mit der Realität gemeint. ⁹²

In vielen Unternehmen hat sich die nachhaltige Verbesserung und Sicherstellung der Informationsqualität zu einem wichtigen Thema entwickelt. Dabei hat sich herauskristallisiert, dass die erfolgreiche Umsetzung eines IQ-Managements nur möglich ist, wenn eine Bewertung der Informationsqualität erfolgt. ⁹³

▪ 15 IQ Dimensionen und die zugehörigen IQ Kategorien

Die von Wang und Strong durchgeführte Studie mit dem Ziel der Beschreibung der Qualität von Informationen in standardisierten Informationsqualität Dimensionen gilt als eines der meistzitierten Werke zu diesem Thema und bildet im Weiteren die fundamentale Basis für viele IQ-Managementkonzepte. Da die Verbreitung dieser Begrifflichkeiten hauptsächlich im englischen Sprachraum zu finden ist, entwarfen Rohweder et al. eine auf Wang and Strong's Definition aufbauende deutschsprachige Version der IQ-Dimensionen. Zusätzlich wurde das Problem von Redundanzen der Begrifflichkeiten untereinander durch technische Begriffsbestimmung adressiert und gelöst. Folglich resultieren die in Tabelle 2 ausgewiesenen Dimensionen der Informationsqualität. Jede dieser definierten Dimensionen wird als kritischer Faktor für das erfolgreiche Operieren eines IQ- Systems verstanden. Somit kann die Behauptung aufgestellt werden, dass die Funktionsfähigkeit eines Informationssystems erst gegeben ist, wenn alle Dimensionen eine ausreichende Qualität aufweisen. ^{94,95} Im Gegenzug dazu kann geringe Qualität von Informationen anhand der Definition von Eppler erkannt werden:

„Low quality information is incomplete, inaccurate, obscure, useless, prolix (or wordy), inconsistent, false, obsolete, delivered in an inconvenient, late, undeterminable and rigid way, via an infrastructure that is inaccessible, exposed to manipulation and other security risks, cumbersome or costly to maintain, and slow.“ ⁹⁶

⁹¹ vgl. Hinrichs, 2001, S.26

⁹² vgl. Rohweder et al., 2008, S.25

⁹³ vgl. Lee, 2006, S.133f

⁹⁴ vgl. Wang and Strong, 1996, S.22f

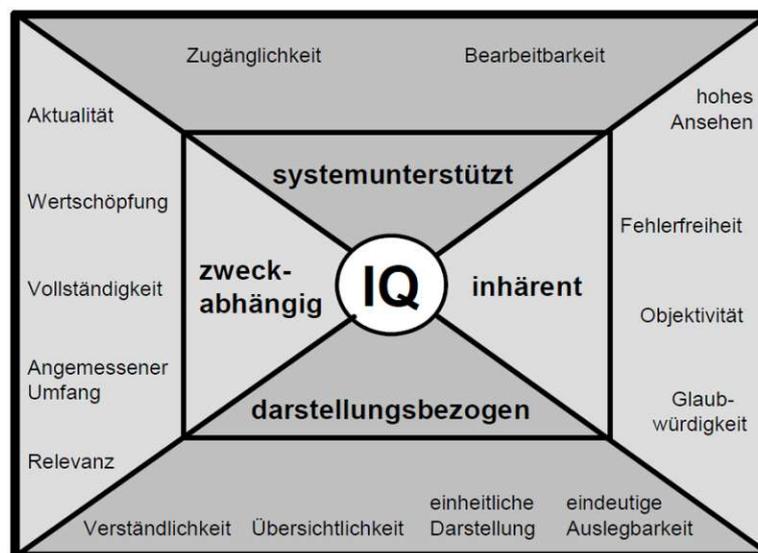
⁹⁵ vgl. Rohweder et al., 2008, S.29ff

⁹⁶ Eppler, 2006, S.76

Tabelle 2: 15 Dimensionen der Informationsqualität⁹⁷

Zugänglichkeit	Einheitliche Darstellung	Relevanz
Angemessener Umfang	Bearbeitbarkeit	Hohes Ansehen
Glaubwürdigkeit	Fehlerfreiheit	Aktualität
Vollständigkeit	Eindeutige Auslegbarkeit	Verständlichkeit
Übersichtlichkeit	Objektivität	Wertschöpfung

Um eine Kategorisierung der 15 Dimensionen durchführen zu können, wurde eine Studie durchgeführt, in der IT-Anwender gebeten wurden die IQ-Dimensionen in Gruppen zusammenzufassen. In 80% der Fälle wurde diese Untergliederung nach der in Abbildung 20 visualisierten Logik vorgenommen.⁹⁸

Abbildung 20: Einteilung der 15 IQ Dimensionen in 4 IQ Kategorien⁹⁹

Abschließend ist zu nennen, dass sich Informationsqualität als hochkomplexes Thema darstellt, bei dem meist der Ansatz verfolgt wird, die Gebrauchstauglichkeit von Informationen zu bewerten (fit-for-use Konzept). Es muss jedoch explizit darauf hingewiesen werden, dass es trotz vieler in der Literatur präsentierter Ansätze zum Thema IQ keine allgemeingültige Herangehensweise geben kann. Dies resultiert aus der Tatsache, dass jedes Unternehmen und Organisation aufgrund der Vielzahl von Geschäftsfeldern andere Ansprüche und Anforderungen an Informationen stellt.^{100,101}

⁹⁷ vgl. Rohweder et al., 2008, S.28

⁹⁸ vgl. ebenda

⁹⁹ ebenda, S.49

¹⁰⁰ vgl. Eppler, 2006, S.17ff

¹⁰¹ vgl. Rohweder et al., 2008, S.44

3.3 Methodik zur Generierung und Aufbereitung von Daten (CRISP-DM Methodik)

3.3.1 Definition und Hintergrund ^{102,103,104}

Die Entwicklung der CRISP-DM Methodik erfolgte 1996 im Rahmen eines EU-Förderprojekts. CRISP-DM bedeutet „Cross Industry Standard Process for Data Mining“. Als standardisierte Vorgehensweise zur Bearbeitung von Data Mining Projekten wird sie in Form eines hierarchischen Prozessmodells beschrieben, das die folgenden Abstraktionsebenen umfasst:

- Phases
- Generic Tasks
- Specialized Tasks
- Process Instances

In der obersten Ebene wird der „Data-Mining“ (DM) Prozess in einzelne Phasen untergliedert, welche aus mehreren „Generischen Aufgaben“ bestehen. Diese „Generischen Aufgaben“, die auf der zweiten Abstraktionsebene zu finden sind, sind insofern generalisiert, sodass sie den gesamten Prozess des Data-Minings und alle damit verbundenen Anwendungsfälle abdecken. Die dritte Ebene, die sich aus „Speziellen Aufgaben“ zusammensetzt, wird verwendet um zu beschreiben, wie generische Aufgaben in speziellen Situationen bewältigt werden sollten. In der vierten und letzten Ebene sind die „Prozessinstanzen“ zu finden, die eine Aufzeichnung von Tätigkeiten, Entscheidungen und Ergebnissen eines DM Projektes darstellen.

Es ist jedenfalls zu beachten, dass die CRISP-DM Methodik zwischen dem Referenzmodell und dem „User-Guide“ unterscheidet. Während das Referenzmodell die Phasen, Aufgaben, Inputs und Outputs eines Data-Mining Projektes präsentiert, bietet der „User-Guide“ detaillierte Unterstützung für jede Phase und Aufgabe an und zeigt folglich auf, wie ein DM-Projekt durchgeführt werden sollte (siehe Abbildung 21). In den folgenden Unterkapiteln wird sowohl auf das Referenzmodell als auch auf den User Guide eingegangen.

¹⁰² vgl. Chapman et al., 1999, S.6ff

¹⁰³ vgl. Wirth and Hipp, 2000, S.3ff

¹⁰⁴ vgl. Gabriel et al., 2011, S.124ff

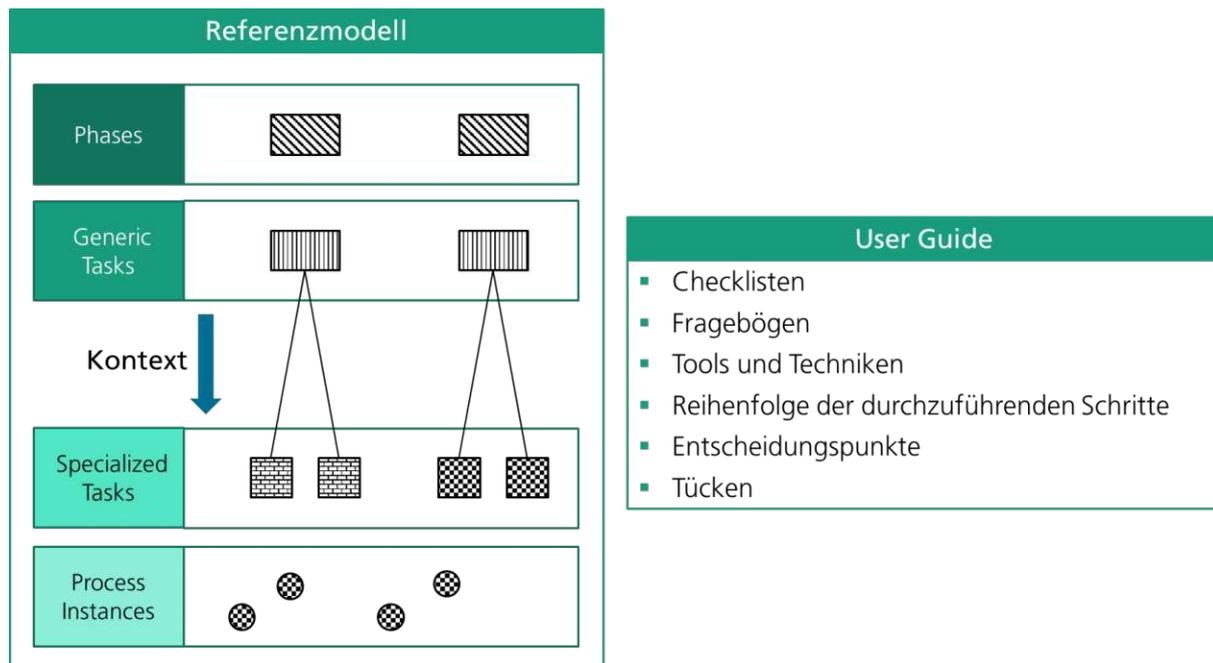


Abbildung 21: 4 Ebenen der CRISP-DM Methodik für Data-Mining¹⁰⁵

3.3.2 CRISP-DM Reference Modell ^{106,107,108}

Das CRISP-DM Referenz Modell kann als Lebenszyklus eines Data-Mining Projektes interpretiert werden, in dem im Allgemeinen sechs Phasen unterschieden werden:

- Business Understanding
- Data Understanding
- Data Preparation
- Modelling
- Evaluation
- Deployment

Die einzelnen Phasen werden sequenziell angeordnet und mit Rückkopplungen versehen (siehe Abbildung 22). Im Folgenden soll auf diese Phasen, die als Teilschritte verstanden werden können, im Detail eingegangen werden.

¹⁰⁵ vgl. Wirth and Hipp, 2000, S.4

¹⁰⁶ vgl. Chapman et al., 1999, S.10ff

¹⁰⁷ vgl. Wirth and Hipp, 2000, S.4ff

¹⁰⁸ vgl. Gabriel et al., 2011, S.125ff

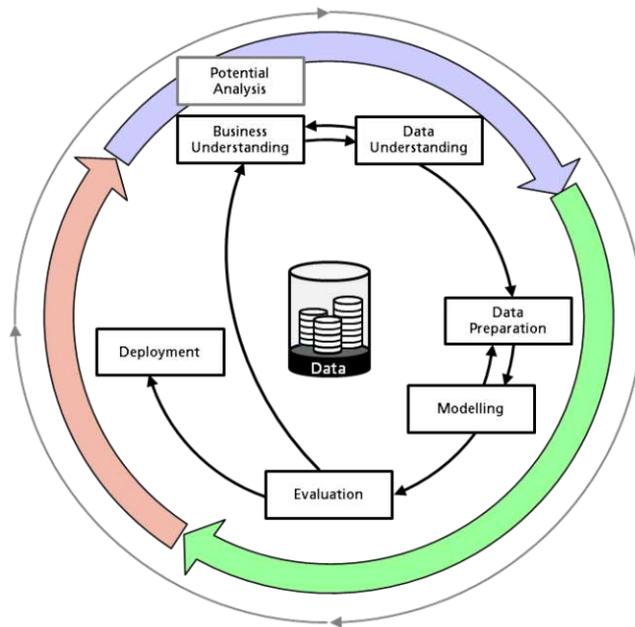


Abbildung 22: Phasen des CRISP-DM-Prozessmodells¹⁰⁹

Phasen der CRISP-DM Methodik

Business Understanding

Die erste Phase der Methodik dient der Definition von Zielen, der Ableitung der damit verbundenen Aufgabenstellung und Vorgehensweise und der Erstellung eines Projektplans.

Data Understanding

Die Phase Data Understanding dient dazu, die zu verwendenden Daten zu sichten, zu bewerten und auszuwählen und gliedert sich in Sub-Phasen, die im Folgenden rudimentär beschrieben werden.

Sammlung und Beschreibung der Daten

Im Zuge dieses Sub-Prozesses werden die zu verwendenden Daten aus internen und gegebenenfalls aus externen Datenquellen extrahiert und gesammelt, um sie anschließend zu beschreiben. Häufig werden hierfür Metadaten verwendet, die Rückschlüsse auf die Größe des Datenbestandes und der Formate, in denen die Daten abgelegt wurden, zulassen.

Analysieren des vorhandenen Datenbestands

Sobald die Phase der Datensammlung abgeschlossen ist, wird empfohlen Verfahren der deskriptiven Statistik anzuwenden bzw. anhand von Visualisierungstechniken erste Graphiken zu erstellen. Zusätzlich sollte in dieser Sub-Phase eine erste

¹⁰⁹ vgl. ebenda, S.125

Verifikation der Datenqualität erfolgen. Dies erweist sich als besonders wichtig, da die Qualität der erhobenen Daten die Aussagekraft der Analyseergebnisse maßgeblich beeinflusst.

Auswahl der Daten

Die Auswahl der Daten stellt den zentralen Punkt der Phase Data Understanding dar. Da die auszuwählenden Daten vom jeweiligen Ziel des Projektes (siehe Phase Business Understanding) abhängen, existiert hierfür keine standardisierte Vorgehensmethodik.

Data Preparation

Die Data Preparation Phase umfasst alle durchzuführenden Aktivitäten, die zur Generierung eines finalen Datensets dienen, welches zur Erstellung und im Weiteren auch zur Erprobung des angestrebten Modells (z.B. Prognosemodell) gebraucht wird. Der Bedarf zur Aufbereitung bestehender Datensätze resultiert meist aus der von vielen Data-Mining-Werkzeugen vorausgesetzten, spezifischen Datenbeschaffenheit, die für die erfolgreiche Durchführung von Analysen von Nöten ist. Es gilt zu beachten, dass ein konsistentes hochqualitatives Datenset als essentiell für den Erfolg von Data-Mining Projekten gilt, wodurch sich die Notwendigkeit der äußerst sorgfältigen Durchführung dieser Phase begründet. Folglich gestaltet sich diese Phase laut Expertenmeinungen zur aufwändigsten, bei der bis zu 80% der für das DM Projekt zur Verfügung gestellten Ressourcen (zeitlich, technisch, personell) aufgewendet werden müssen.

Modeling

Nachdem in der vorangehenden Phase der Datenvorbereitung Daten in ein verwertbares Format transformiert wurden, werden in der Modeling Phase verschiedenste Methoden angewendet, die zum Kategorisieren, Klassifizieren, Prognostizieren der Daten und für Korrelationsanalysen dienen. Das Anwenden verschiedener Ansätze zur Lösung ein und desselben spezifischen Data-Mining Problems ist hierbei nicht untypisch. In dieser Phase ist die Entstehung neuer, in vorherigen Schritten nicht bedachten Aufgabenstellungen nicht ungewöhnlich, woraus die Verknüpfung mit der Phase „Data Preparation“ resultiert und somit die Rückkopplung im Referenzmodell (siehe Abbildung 22) begründet wird.

Evaluation

Die Phase Evaluation dient zur kritischen Begutachtung des bisherigen Projektverlaufs, der Bewertung der Ergebnisse (Data-Mining Ergebnisse) und der Überprüfung der Qualität (Bewertung des gesamten Data Mining Prozesses), wobei der Fokus vor allem auf letzteren beiden, als Kernaktivitäten dieser Phase geltenden,

Prozessen liegen sollte. Wichtig ist es hierbei das gesamte Modell einer gründlichen Überprüfung zu unterziehen, bei der alle Schritte, die im Zuge des DM Projektes durchgeführt wurden, miteinbezogen werden, um somit sicherzustellen zu können, dass alle festgelegten Geschäftsziele ausreichend erfüllt werden. Als Schlüsselziel gilt es Kenntnis darüber zu erlangen, ob alle Belange geschäftlicher Natur angemessen berücksichtigt wurden.

Deployment

Nach der Erstellung des Modells muss das erlangte Wissen aufbereitet werden, um dem Kunden die Anwendung zu ermöglichen. Dies kann in Form von Protokollen, Anleitungen oder in Form von wiederholbaren Data-Mining Prozessen geschehen. Es ist keine Seltenheit, dass die Deployment Phase größtenteils vom Anwender und nicht vom Datenanalysten durchgeführt wird. Wichtig ist hierbei, dass bereits im Vorfeld berücksichtigt wird, welche Schritte durchgeführt werden müssen, um eine erfolgreiche Anwendung der Modelle zu gewährleisten.

Zusammengefasst werden die Aufgaben und die damit verbundenen Outputs des CRISP-DM Referenzmodells in Abbildung 23 dargestellt.

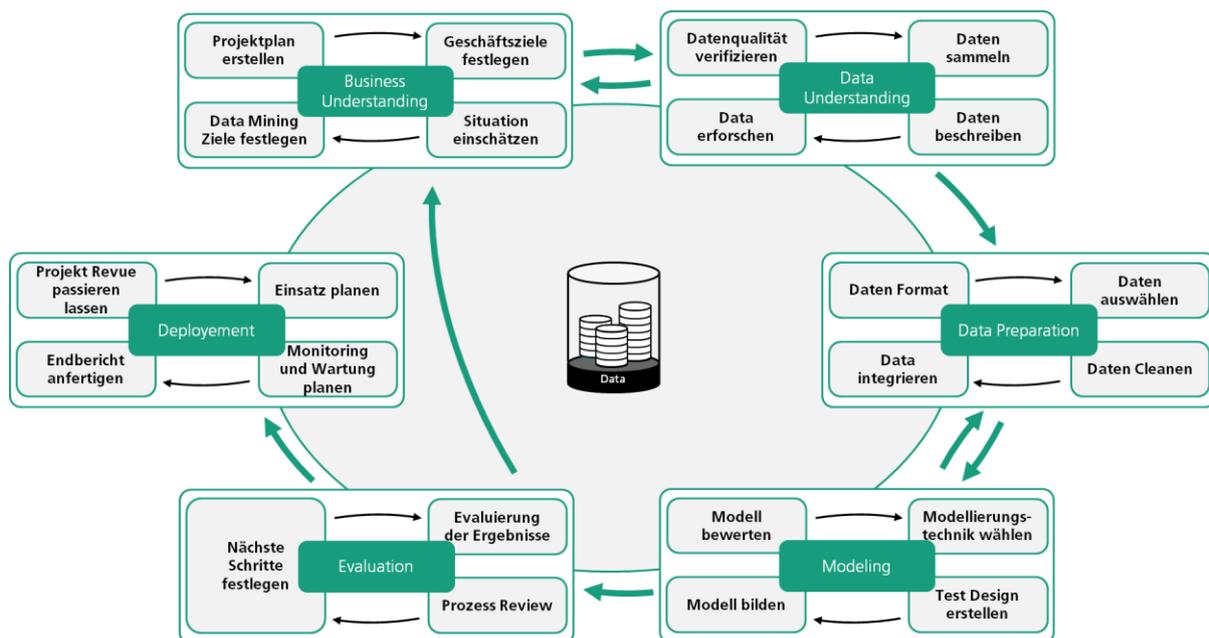


Abbildung 23: CRISP-DM Referenzmodell Phasen mit zugeordneten Aufgaben¹¹⁰

3.3.3 CRISP-DM User-Guide¹¹¹

Im Unterschied zum CRISP DM Referenzmodell kann man den zugehörigen User-Guide als Anleitung zur Durchführung verstehen. Folglich werden nicht nur die Phasen und ihre theoretischen In- und Outputs beschrieben, sondern Vorgehensweisen und

¹¹⁰ vgl. Wirth and Hipp, 2000, S.6

¹¹¹ vgl. Chapman et al., 1999, S.30ff

Checklisten zu den einzelnen Prozessen präsentiert. Diese gliedern sich in die Kategorien:

- Aufgabe
- Aktivitäten
- Output
- Gute Idee
- Tücken

Auf die ausführliche Beschreibung des DM User Guides wird in dieser Arbeit verzichtet und auf vertiefende Literatur verwiesen.

3.3.4 Erweiterung der CRISP-DM Methodik

Die CRISP-DM Methodik wurde, wie bereits erwähnt, in den späten 90er Jahren entwickelt. Folglich wurden seither einige Methoden präsentiert, die sich um eine Weiterentwicklung der ursprünglichen Version bemühen. Eine der bekanntesten stellt hierbei die von IBM entwickelte Methode „ASUM DM“ dar, die im Jahr 2015 vorgestellt wurde. Im Folgenden soll der besagte Ansatz kurz erläutert werden.

▪ ASUM-DM¹¹²

Der ASUM-DM Ansatz sieht eine Auftrennung des bisherig verwendeten CRISP-DM Zyklus in einen „Deployment-Cycle“ und einen „Development Cycle“ vor. Während der Development Zyklus nahezu unverändert bleibt, wird die Phase Deployment um einige Schritte erweitert (siehe Abbildung 24). IBM begründet die Erweiterung des Modells damit, dass der „Development-Zyklus“ immer als vollständig analytischer Teil bei Data-Mining Projekten gesehen wurde und somit der meiste Aufwand in die ersten Phasen des Zykluses einfluss. Daraus resultierte häufig eine Vernachlässigung der Phase „Deployment“, weshalb Unternehmen häufig nicht den gesamten geschäftlichen Nutzen ihrer analytischen Bemühungen ausschöpfen konnten. Im neu entwickelten „Deployment-Zyklus“ erscheint die Industrialisierung der Analyse nicht so glamourös, stellt aber die Phase dar, bei der die „PS auf die Straße gebracht werden“.

¹¹² IBM, IT Best Kept Secret is Optimization, https://www.ibm.com/developerworks/community/blogs/jfp/entry/An_ode_to_the_analytics_grease_monkeys_analytics_deployment_ROI?lang=en (Gelesen am: 28.10.2018 13.36)

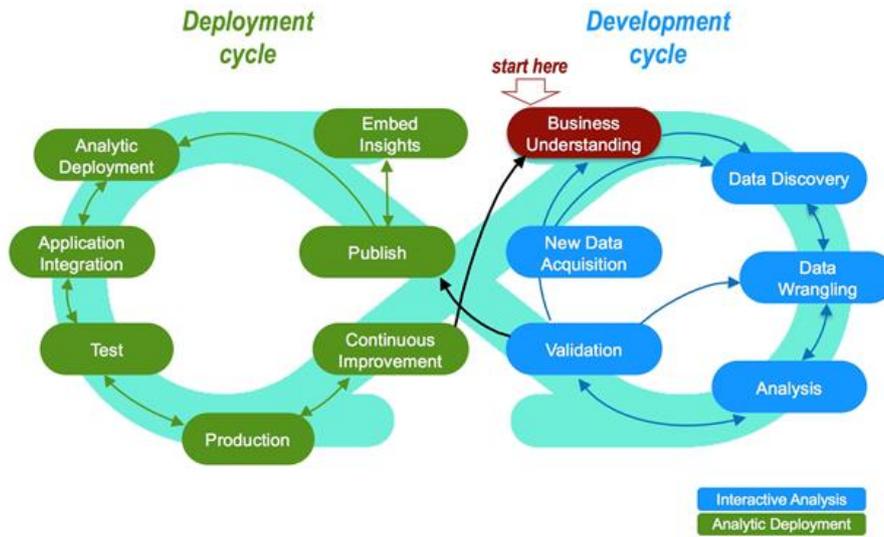


Abbildung 24: ASUM-DM Zyklus von IBM¹¹³

¹¹³ ebenda, 2015

4 Bayes'sche Netze

Dieses Kapitel dient dazu die theoretischen Grundlagen von Bayes'schen Netzen (BN) zu vermitteln. Zuerst werden grundlegende mathematische Definitionen und Eigenschaften präsentiert. Danach soll anhand von zwei Beispielen aus der Literatur der Stellenwert und der State-of-the-Art von BNs im Bereich der Instandhaltung präsentiert werden.

4.1 Grundlegende Definitionen und Eigenschaften

Unter einem Bayes'schen Netz versteht man einen azyklischen, gerichteten Graphen, der es ermöglicht, Aussagen zu Wahrscheinlichkeitsverteilungen und bedingten Unabhängigkeiten von Zufallsvariablen zu treffen. Die Knoten repräsentieren hierbei die Systemvariablen, während die Kanten die Abhängigkeiten oder die Ursache-Wirkungsbeziehungen zwischen den Variablen beschreiben.^{114,115}

Um die Semantik und Grundlagen verstehen zu können, werden im Folgenden einige Begriffe der Graphentheorie eingeführt:

„**Definition 24.16 (Gerichtete Kante)** Sei $G = (V, E)$ ein (einfacher) Graph. Eine Kante $e = (A, B) \in E$ heißt gerichtete Kante, falls gilt:

$$(A, B) \in E \Rightarrow (B, A) \notin E$$

Eine solche Kante zeige von A nach B , geschrieben $A \rightarrow B$. Der Knoten A heißt Elternknoten von B , während B der Kindknoten von A ist.“¹¹⁶

„**Definition 24.25 (Azyklischer, gerichteter Graph (DAG))** Ein gerichteter Graph $G = (V, E)$ wird azyklisch genannt, wenn für jeden Pfad $X_1 \rightarrow \dots \rightarrow X_k$ in G gilt:¹¹⁷

$$X_k \rightarrow X_1 \notin E$$

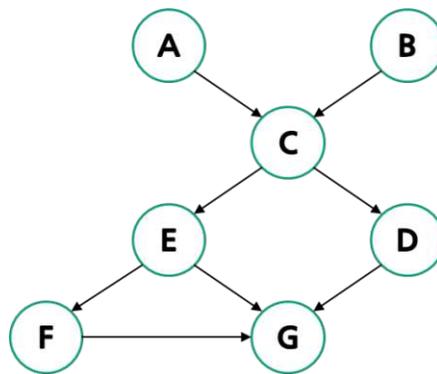


Abbildung 25: Beispiel für ein Bayes'sches Netz¹¹⁸

¹¹⁴ vgl. Weber et al., 2012, S.4

¹¹⁵ vgl. Heckerman, 1995, S.197

¹¹⁶ Kruse et al., 2015, S.420

¹¹⁷ ebenda, S.423

¹¹⁸ vgl. Jensen and Nielsen, 2007, S. 34

In einem BN wird jeder Variablen ein Knoten x_i zugewiesen. Knoten die untereinander Kausalitäten aufweisen, werden mittels sogenannter Kanten miteinander verbunden. Die Verbindung erfolgt dabei immer entlang der Ursache-Wirkungs-Folge. Das bedeutet, dass der Kindknoten x_i mit seinen Elternknoten ($parents(x_i)$) mittels Pfeilen verbunden werden, wobei der Pfeil in Richtung des Kindknotens x_i zeigt.¹¹⁹ Zu jeder Variablen existiert eine zugehörige bedingte Wahrscheinlichkeitstabelle. Diese setzt sich aus spezifischen Zuständen und den zugehörigen Wahrscheinlichkeitswerten, dass besagter Zustand eintritt, zusammen. Während allen Wurzelknoten, die keine Eltern aufweisen, zugrundeliegende Wahrscheinlichkeitswerte zugeordnet werden, resultieren die Wahrscheinlichkeitswerte aller anderen Knoten $P(x_i | parents(x_i))$ aus allen möglichen Zustandskombinationen der zugehörigen Elternknoten.^{120,121} Um die gesamte gemeinsame Verteilungsfunktion berechnen zu können, wird Formel 1 verwendet.¹²²

$$P(x_1, x_2, \dots, x_n) = \prod_{i=1}^n P(x_i | parents(x_i))$$

Formel 1: Gemeinsame Verteilungsfunktion¹²³

Das in Abbildung 25 präsentierte Netz wäre laut Formel 1 somit in folgender Form zu interpretieren (siehe Formel 2):

$$P(A, B, C, D, E, F, G) = P(A) P(B) P(C|A, B) P(E|C) P(D|C) P(F|E) P(G|D, E, F)$$

Formel 2: Gesamte Verteilungsfunktion anhand des in Abbildung 25 präsentierten Beispiels¹²⁴

Somit folgt die nachstehende Definition die BNs mathematisch beschreibt:

„Definition 12.25 (Bayes'sches Netzwerk): Sei $\mathbf{V} = \{x_1, \dots, x_n\}$ eine Menge von Aussagenvariablen, sei P eine gemeinsame Verteilung über \mathbf{V} , und sei $G = \langle \mathbf{V}, \varepsilon \rangle$ ein DAG. Für jedes $x_i \in \mathbf{V}$ bezeichne $pa(x_i) \subseteq \mathbf{V}$ die Menge aller Eltern(knoten), $de(x_i) \subseteq \mathbf{V}$ die Menge aller Nachkommen und $nd(x_i) \subseteq \mathbf{V}$ die Menge aller Nicht-Nachkommen von x_i .

$\beta = \langle \mathbf{V}, \varepsilon, P \rangle$ wird Bayessches Netzwerk (Bayesian network) genannt, wenn für jede Variable x_i gilt

$$x_i \perp\!\!\!\perp_P nd(x_i) | pa(x_i)$$

¹¹⁹ vgl. Pearl, 1988, S.51

¹²⁰ vgl. Pearl, 1986, S.244ff

¹²¹ vgl. Nawaz et al., 2014, S.253f

¹²² vgl. Pearl, 1988, S.51

¹²³ vgl. ebenda, S.51

¹²⁴ vgl. Jensen and Nielsen, 2007, S. 34

Wenn also jede Variable x_i bedingt unabhängig ist von ihren Nicht-Nachkommen $nd(x_i)$ bei gegebenen Werten ihrer Elternknoten $pa(x_i)$.¹²⁵

Laut der Unabhängigkeitsvoraussetzungen eines BN wird folglich jeder Knoten nur von seinen Elternknoten beeinflusst und vor anderen Knoten abgeschirmt, sofern es sich bei genannten Knoten nicht um Kindknoten des betrachteten Knotens handelt.

Sobald das Netzwerk gebildet wurde und die Abhängigkeiten durch Struktur und Wahrscheinlichkeitstabellen definiert wurden, können mittels des Satzes von Bayes, auf denen BNs im Allgemeinen aufbauen, Inferenzen im Netz gebildet werden. Inferenz bedeutet dabei das Berechnen der Wahrscheinlichkeiten spezifischer Variablen, unter dem Vorliegen exakter Werte von anderen Variablen (Evidenzen).¹²⁶

$$P(B|A) = \frac{P(A|B) * P(B)}{P(B)}$$

Formel 3: Satz von Bayes¹²⁷

▪ **Erstellung von Bayes'schen Netzen**¹²⁸

In der Literatur existieren unterschiedliche Ansätze zur Erstellung von BNs. Kraaijeveld et al. zum Beispiel definieren drei Möglichkeiten, zwischen denen beim Aufbau gewählt werden kann:

- a) Erstellung der Struktur und Parameter aus Daten mittels geeigneter Lernalgorithmen
- b) Domänenexperte erstellt Netz auf Basis von Expertenwissen
- c) Kombination von Expertenwissen und Lernen aus Daten

Trotz der guten Performance derzeitig existierender diagnostischer BNs wird der Methode häufig kritisch begegnet. Dies liegt hauptsächlich an dem hohen Aufwand, der mit der Erstellung der Modelle verbunden ist.

▪ **Vorgehensmethode nach Naïm**¹²⁹

Da im Zuge des Kapitel 5 bei der Erstellung des BN nach der von Naim et. al. vorgeschlagenen Methodik, die eine Kombination von Expertenwissen und Lernen aus Daten vorsieht, vorgegangen wird, soll diese nun kurz erläutert werden (siehe Abbildung 26).

¹²⁵ Beierle and Kern-Isberner, 2008, S. 383f

¹²⁶ vgl. Nawaz et al., 2014, S.254

¹²⁷ Bayes, 1763

¹²⁸ vgl. Kraaijeveld et al., 2008, S.2

¹²⁹ vgl. Naïm, 2008, S.209

4.3 Bayes'sche Netze in der Instandhaltung

Dieses Kapitel bereitet zwei der Literatur entnommene Beispiele auf, in denen Bayes'sche Netze im Themengebiet der Instandhaltung angewendet wurden.

▪ Fault Diagnosis in Semiconductor Etch Equipment Using Bayesian Networks¹³⁷

Nawaz et al. präsentierten 2014 einen Ansatz zur Fehlerdiagnose von Anlagen, die zur Halbleiterätzung verwendet werden. Hierfür wurde zunächst eine Untergliederung in Sub-Module durchgeführt. In einem weiteren Schritt wurde ein Ursache-Wirkungs-Diagramm (Abwandlung des bekannten Ishikawa-Fischgräten Diagramms) erstellt, welches potentielle Fehler beim Prozess den jeweiligen Modulen zuordnen sollte. Darauf aufbauend wird die Struktur des Bayes'schen Netzes entworfen, in dem drei Ebenen definiert werden:

- **Top Layer:** Root Causes. Diese beschreiben die Fehler, die aus dem Fischgrätendiagramm abgeleitet wurden. Da diese modulspezifisch sind, stellen sie die Elternknoten der zweiten Ebene dar.
- **Mid Layer:** In dieser Ebene werden die definierten Module eingetragen. Elternknoten dieser Ebene stellen, wie bereits erwähnt, die zugehörigen potentiellen Fehler dar.
- **Bottom Layer:** In dieser Ebene werden die Datenparameter eingetragen, die im Falle einer Störung auftreten können.

Das aus diesen Schritten gebildete Netzwerk wird in Abbildung 28 gezeigt.

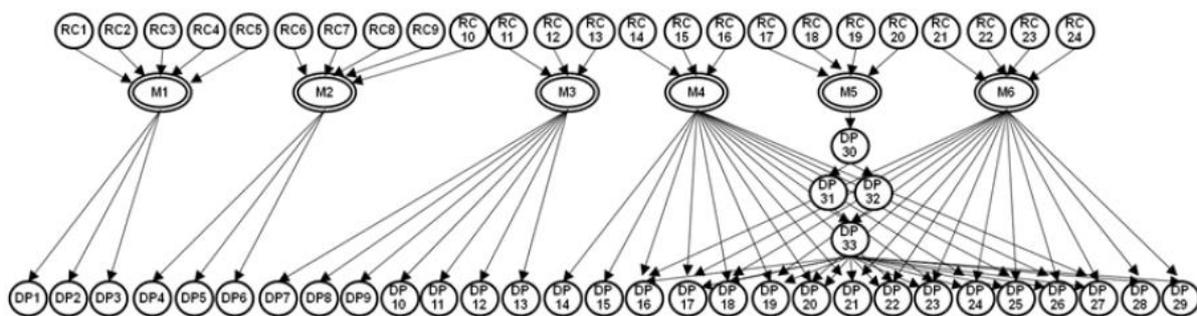


Abbildung 28: Bayes'sches Netz zur Fehlerdiagnose in der Halbleiterätzung¹³⁸

Aufbauend auf dieses Fundament wurden die Knoten mit den zugehörigen Wahrscheinlichkeiten gefüllt und im Anschluss darauf Tests durchgeführt. Dafür wurde die Anlage speziell präpariert, indem Fehler eingebaut wurden um die Genauigkeit der Fehlerdiagnose validieren zu können. Die Ergebnisse dieser Tests sind Abbildung 29

¹³⁷ vgl. Nawaz et al., 2014

¹³⁸ vgl. ebenda S.5

zu entnehmen. Der Knoten mit dem höchsten Wert stellt hierbei jeweils den Knoten dar, der den zugehörigen Fehler mit der höchsten Wahrscheinlichkeit verursacht hat.

	Power loss in Coaxial Cable	Coaxial Cable Loose Connection	Matcher Problem	MFC Mis-Calibration	Gas Line Leakage	Gas Cylinder Empty	Gas Control Valve Failure	Chamber Wall Deposition	Particles	Chamber Leak	Gasket or O-ring Leakage
Exp.1	0.47	0.15	0.24	0.57	0.36	0.14	0.23	0.22	0.33	0.46	0.12
Exp.2	0.37	0.12	0.22	0.44	0.31	0.11	0.21	0.22	0.32	0.43	0.11
Exp.3	0.39	0.13	0.22	0.57	0.36	0.14	0.23	0.21	0.32	0.36	0.11
Exp.4	0.48	0.15	0.24	0.44	0.31	0.11	0.21	0.22	0.34	0.47	0.12
Exp.5	0.53	0.16	0.25	0.50	0.34	0.12	0.22	0.24	0.36	0.56	0.13

Abbildung 29: Ergebnisse der Fehlerdiagnose¹³⁹

Somit wurde eine Methodik entwickelt, die im Falle einer Störung dem Instandhaltungspersonal eine Empfehlung bezüglich der Fehlerbehebung geben kann. Folglich kann dieser Ansatz dem Bereich der präskriptiven Instandhaltung zugeordnet werden.

- **Equipment Health Factor (EHF) prediction for complex semiconductor manufacturing facility¹⁴⁰**

Bouaziz et al entwickelten 2013 eine Methodik zur Bestimmung des sogenannten EHF um den zukünftigen Zustand von Ätzmaschinen in der Halbleiterfertigung zu prognostizieren.

Dazu wurden drei übergeordnete Komponenten der Maschine betrachtet:

Abnutzung des Reaktors: Die Abnutzung des Reaktors wurde anhand der Dicke des Quarzes und des Zeitpunkts der letzten Wartung bestimmt.

Abnutzung des Roboters: Die Abnutzung bzw. der EHF des Roboters kann auf Basis des Wissens über die letzten Instandhaltungsmaßnahmen ermittelt werden.

Abnutzung der Storage Chamber: Die Abnutzung dieser Komponente ist von den durchgeführten Instandhaltungen und der Anzahl der Alarme, die das System aufgrund von Unregelmäßigkeiten an den Benutzer sendet, abhängig.

Da das Ziel des Forschungsprojektes die Prognose des zukünftigen Maschinenzustands darstellt, musste die zeitliche Komponente in das System miteinfließen und das Bayes'sche Netzwerk zu einem dynamischen BN erweitert werden. In einem finalen Schritt wurden die drei betrachteten Komponenten in einer globalen EHF verbunden (siehe Formel 4 und Abbildung 30).

¹³⁹ vgl. ebenda, S.8

¹⁴⁰ vgl. Bouaziz and Zamaï, 2012

$$EHF = \sum W_i * P\left(\frac{FM_i}{Observations}\right)$$

W_i ... Gewichtetes Risiko der drei Module (auf Basis von Expertenwissens)

$P\left(\frac{FM_i}{Observations}\right)$..Wahrscheinlichkeiten der einzelnen Module

Formel 4: Kalkulation des globalen EHF¹⁴¹

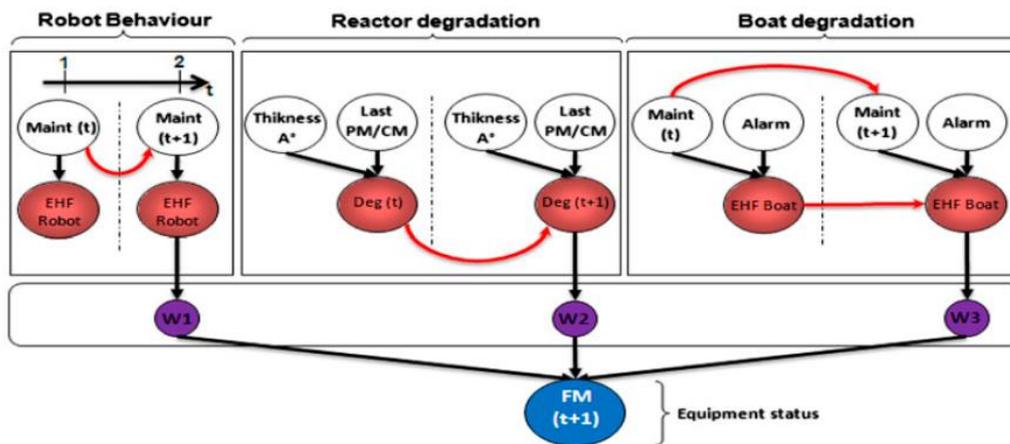


Abbildung 30: Ermittlung des Equipment Health Factor¹⁴²

Die präsentierte Methodik wurde im Zuge eines Feldversuchs erprobt. Die zugehörigen Ergebnisse können aus Tabelle 3 entnommen werden. Dabei sei zu erwähnen, dass die Höhe des Zahlenwertes des EHF die Abnutzung und somit das Risiko eines Ausfalls beschreibt. Während das Risiko für einen Ausfall in Run 1 noch als sehr gering prognostiziert wurde, ist es nach Run 500 schon bei 51%.

Tabelle 3: Prognoseergebnisse¹⁴³

Time	P (EHFrobot)	P (EHFreactor)	P (EHFboat)	P (EHFequipment)
Run1	.001	.005	.003	.003
Run20	.002	.011	.008	.007
Run35	.010	.015	.015	.013
Run50	.025	.039	.050	.036
Run100	.050	.090	.250	.100
Run200	.150	.230	.315	.215
Run300	.295	.455	.490	.404
Run400	.315	.555	.500	.463
Run500	.350	.610	.530	.510

Der präsentierte Ansatz zur Bestimmung des EHF kann somit als Unterstützung für die Planung von Instandhaltungsmaßnahmen dienen, bevor es zu ungeplanten Stillständen kommt. Folglich kann er dem Bereich prädiktiven Instandhaltung zugeschrieben werden.

¹⁴¹ vgl. ebenda, S.1009

¹⁴² vgl. ebenda, S.1009

¹⁴³ vgl. ebenda, S.1009

5 Erstellung eines Bayes'schen Netzwerkes zur Ausfallprognose in der Halbleiterfertigung

Das folgende Kapitel wird sich mit der Beschreibung des Use-Cases, den zur Verfügung gestellten Daten und der Erstellung und Erprobung des Bayes'schen Netzes zur Ausfallprognose beschäftigen.

5.1 Beschreibung des Use-Case/Ausgangssituation

5.1.1 Ausgangssituation

Die Halbleiterherstellende Industrie charakterisiert sich durch komplexe Produktionsprozesse und einem weltumfassenden wettbewerbsintensiven Markt.¹⁴⁴ Folglich ist es für Hersteller essentiell, eine effektive und effiziente Produktion aufrecht zu erhalten und für eine hohe Prozessstabilität und Anlagenverfügbarkeit Sorge zu tragen.

Diese Arbeit wird im Zuge eines Forschungsprojektes bei einem namhaften Hersteller von Halbleitererzeugnissen mit Sitz in Österreich durchgeführt, bei dem aktuell ein negatives Verhältnis zwischen geplanten und ungeplanten Ausfallzeiten vorliegt. Dies resultiert in einer verminderten Reaktionsfähigkeit und erhöhten Instandhaltungskosten. Um die produktiven Zeiten der Maschinen so lange wie möglich aufrecht zu erhalten und ungeplante Störfälle zu vermeiden, wird versucht mittels eines „Bayes'schen Netzes“ ein Prognosemodell für zukünftige Ausfälle zu entwerfen. In weiterer Folge soll dieses getestet, mit anderen Modellen verglichen und im Abschluss erörtert werden, welche Stärken und Schwächen der genannte Ansatz hat. Zusätzlich soll die Tauglichkeit hinsichtlich einer erfolgreichen Implementierung in die bestehende Instandhaltungsplanung diskutiert werden.

5.1.2 Grundlagen der Halbleiterätzung

In der Fertigung von Halbleitererzeugnissen spielt die genaue Überführung von festgelegten Mustern auf das Substrat eine zentrale Rolle. Dies ist erforderlich, um die im Vorhinein festgelegten physikalischen Eigenschaften der elektronischen Bauteile, die später aus den Wafern gewonnen werden, zu gewährleisten. Zur Herstellung dieser Muster kann auf folgende Art und Weise vorgegangen werden:¹⁴⁵

- Aufbringen von Schichten
- Modifikation von Schichten
- Entfernen von Schichten

¹⁴⁴ vgl. Bouaziz and Zamaï, 2012, S.1005

¹⁴⁵ vgl. Shearn et al., 2010, S.80

Unter Halbleiterätzen versteht man das durch chemische und/oder physikalische Prozesse initiierte Entfernen definierter Schichten eines Wafers.¹⁴⁶ Da bei der für den Use-Case betrachteten Maschine das sogenannte „Reactive-Ion-Etching“ (RIE), welches der Gruppe der Dry Etch Plasma Verfahren zuzuordnen ist, angewendet wird, soll im Weiteren nur auf diese spezielle Technik eingegangen werden.

Beim „Reactive-Ion-Etching“ (RIE) handelt es sich um einen der wichtigsten Prozessschritte bei der Fertigung von Halbleitererzeugnissen, bei dem Ionen dazu verwendet werden, Schichten aus dem zu bearbeitenden Wafer zu entfernen.¹⁴⁷ Der Prozess verläuft im Vakuum und muss in Reinräumen durchgeführt werden, um Wafer-Kontamination vorzubeugen, die negative Auswirkungen auf das Endprodukt haben würde. Zunächst wird der Wafer mittels Roboter auf einem sogenannten Chuck (in diesem Falle Elektro-Static-Chuck siehe Kapitel 5.1.5) positioniert, welcher diesen durch eine elektrostatisch aufgebrachte Kraft fixiert. Zwei parallel zueinander angeordnete, im oberen und unteren Teil der Kammern positionierte Platten dienen als Elektroden und sind mit einem Radio-Frequency (RF) Generator verbunden. Dieser wird benötigt, um die in die Kammer geführten Gase anzuregen um diese in den vierten, unter dem Namen Plasma bekannten Aggregatzustand zu versetzen. Durch die Applikation unterschiedlicher Spannungen zwischen der oberen und unteren Elektrode wird eine Bias-Spannung (auch Biasvoltage genannt) provoziert, die zu einer Beschleunigung der im Plasma entstandenen, positiv geladenen Ionen in Richtung Wafer führt.¹⁴⁸ Die beschleunigten Ionen „bombardieren“ in weiterer Folge den Wafer und eliminieren somit die zu entfernenden Schichten aus dem Substrat.¹⁴⁹

5.1.3 Anlagenaufbau

Als Referenzanlage wurde die Maschine 3_Oxid_LAM herangezogen, die zur 2300 Etch Product Family der Firma Lam Research gehört und das „Inductive Coupled Plasma RIE“ Verfahren zur Ätzung der Halbleiterelemente verwendet.

Die Anlage besteht im Allgemeinen aus einem Cluster mehrerer Prozesskammern und einem dazugehörigen Mainframe. Da die Kammern im Aufbau und Funktion nahezu ident sind, wird sich diese Arbeit mit Kammer P3 beschäftigen. Der Aufbau und der Prozessablauf wird in Abbildung 31 dargestellt.

¹⁴⁶ vgl. May and Spanos, 2006

¹⁴⁷ vgl. Lazar et al., 2006, S.389ff

¹⁴⁸ vgl. Novotn, 1988, S.689f

¹⁴⁹ vgl. Nawaz et al., 2014, S.254f

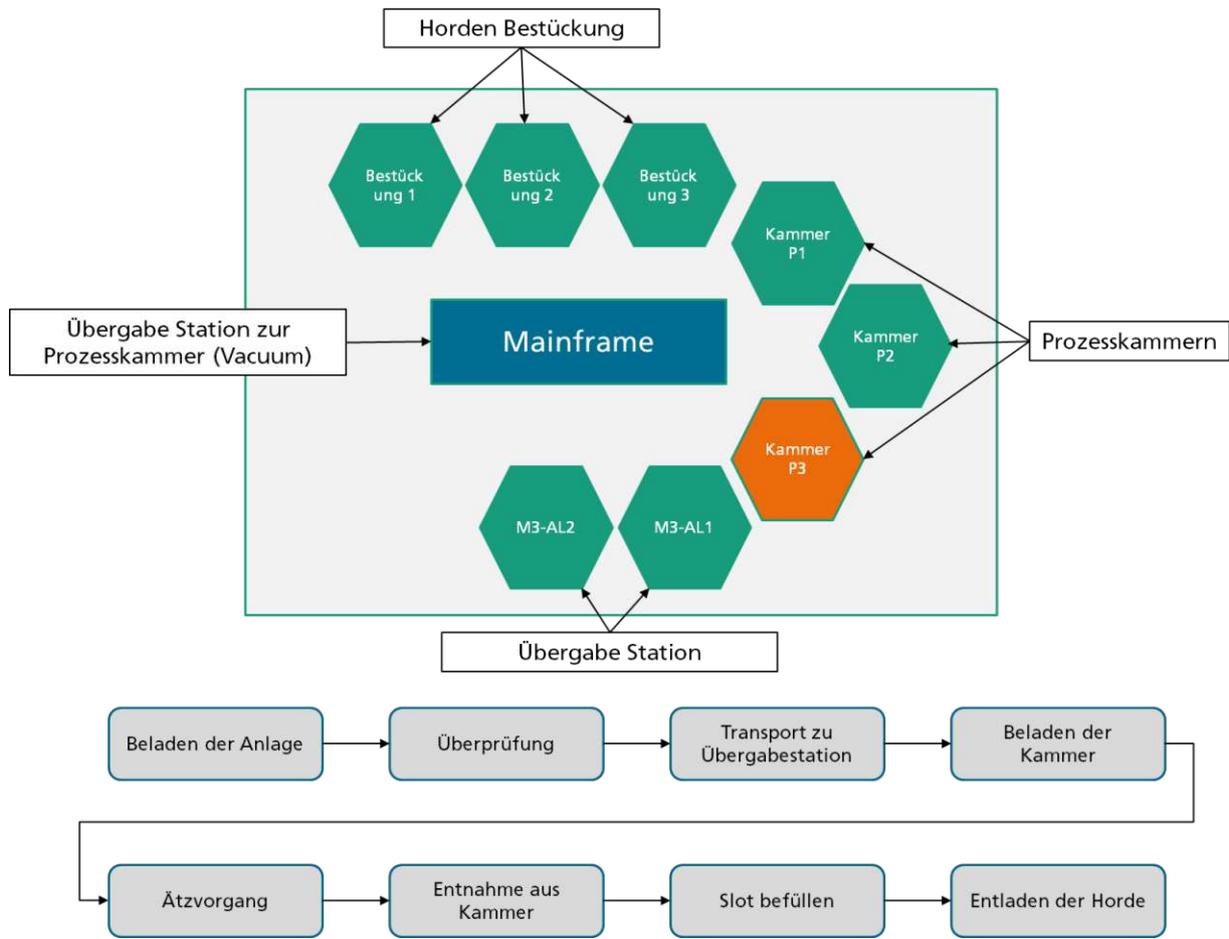


Abbildung 31: Prozesskammern der Referenzmaschine M3 und zugehöriger Prozessablauf

In einem ersten Schritt werden die Wafer, die in Horden organisiert sind, zur Anlage transportiert und in diese geladen. Danach erfolgt eine Überprüfung, bei der ein Abgleich bezüglich eingestelltem Rezept und der Horden-Nummer (Identifikationsnummer der zu ätzenden Wafer) durchgeführt wird. In weiterer Folge werden die Wafer zur Übergabestation geleitet. Da der Ätzzvorgang im Vakuum erfolgen muss, beinhalten diese Übergabestationen sogenannte „Airlocks“, die als „Schleuse“ verstanden werden können. Auf das Evakuieren anschließend werden die Kammern mittels eines im Vakuum arbeitenden Roboters mit den zu ätzenden Wafern bestückt. Darauffolgend wird der Ätzzvorgang gestartet, der je nach Rezept unterschiedliche Zeit in Anspruch nimmt. Nach erfolgreicher Durchführung erfolgt der Transport zur zweiten Übergabestation, in dem die Wafer den Vakuumbereich wieder durch eine Schleuse verlassen. Die Entladung wird durch das Ablegen der Wafer in für sie vorgesehene Slots vollzogen.

5.1.4 Kammernaufbau

Der Aufbau der Ätzkammer kann in 6 Module untergliedert werden welche wie folgt wären:¹⁵⁰

- Vakuummodul
- Gas Delivery Modul
- Transfer Modul
- E-Chuck Modul
- RF Power Modul
- Prozess Modul

Abbildung 32 zeigt den rudimentären Aufbau der Ätzkammer der Referenzanlage.

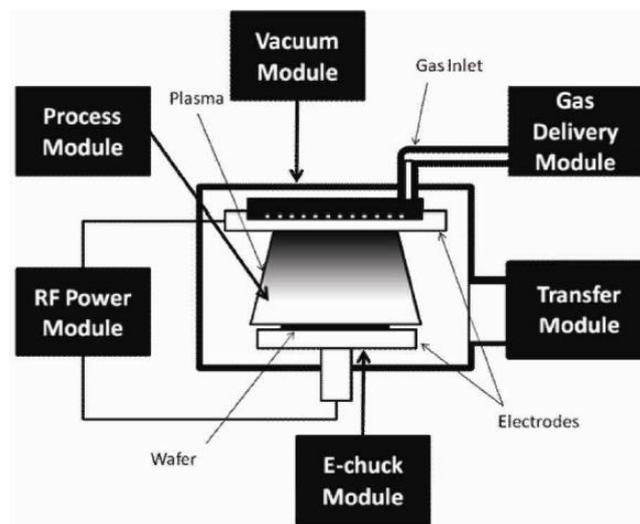


Abbildung 32: Aufbau einer Prozesskammer¹⁵¹

Da die verschiedenen Module und die zugehörigen Parameter einen Einfluss auf mehrere Bereiche der Kammer und auch Maschine nehmen können, ist zu betonen, dass diese Unterteilung hauptsächlich zum besseren Verständnis und der somit erleichterten Zuordnung verschiedener Prozessparameter dient.

5.1.5 E-Chuck Modul

Im weiteren Verlauf dieser Arbeit wird aufgrund des hohen Komplexitätsgrades, den der Maschinenaufbau darstellt, der Fokus bei der Erstellung des Bayes'schen Netzwerkes und dem Versuch der Prognostizierung von Ausfällen auf das Modul E-Chuck gelegt. Folglich soll dieses Unterkapitel dazu dienen, den Aufbau und die Funktionsweise dieser wichtigen Baugruppe zu erläutern.

¹⁵⁰ vgl. Nawaz et al., 2014, S.255

¹⁵¹ ebenda

Allgemeines:

Der Elektro-Static-Chuck (ESC) stellt im Prozess des Plasmaätzens eine sehr wichtige Baugruppe dar, die primär für die Fixierung des Wafers während des Ätzprozesses dient. Auch die geerdete Elektrode, die beim Ätzvorgang die entstandenen Ionen anzieht und beschleunigt, ist in diesem Modul integriert. Für optimale Ergebnisse beim Ätzvorgang ist es essentiell, dass der Wafer homogen und gleichmäßig auf der Oberfläche aufliegt. Anfänglich wurden hierfür Systeme verwendet, bei denen die Spannkraft mechanisch aufgebracht wird. Diese haben jedoch den großen Nachteil, dass die Spannsysteme nur am Rand des Wafers positioniert werden können, wodurch ein einwandfreies Aufliegen des Wafers nicht immer gewährleistet werden kann. Außerdem birgt das mechanische Aufbringen von Kraft auf die meist dünnen Wafer die Gefahr, diese zu beschädigen.¹⁵² Zusätzlich muss auch das Problem der bereits angesprochenen Wafer-Kontamination berücksichtigt werden. Mechanische Spannsysteme neigen laut durchgeführter Studien dazu, signifikante Mengen an Partikel und Abrieb zu erzeugen, die eine Verschmutzung zur Folge haben können.¹⁵³ Da der Ätzvorgang beim Plasmaätzen im Vakuum erfolgen muss, können auch Vakuumspannsysteme für diese Problemstellungen keine Abhilfe schaffen.¹⁵⁴ Infolgedessen wurden Elektro-Static-Chucks, die ein uniformes und nahezu berührungsloses Einspannen von Wafern ermöglichen, entwickelt.¹⁵⁵

Aufbau:

Eine Hauptkomponente des ESC stellt die Elektrode dar, die in einem Stapel dielektrischer Schichten, die üblicherweise aus Aluminium-, Silizium oder titanbasierten Keramiklegierungen bestehen, eingebettet wird. Auf der Oberfläche dieses Stapels sind Nuten vorhanden, die dafür sorgen, dass das zur Kühlung benötigte Gas gleichmäßig um den Wafer zirkulieren kann. Des Weiteren sind auf der Oberfläche elektrisch verstellbare Stifte vorhanden, auf denen der Wafer vor dem Festspannen (Chuck) bzw. nach dem Loslassen (Dechuck) aufliegt. Als weitere Komponente des ESC gilt ein eigens für dieses Modul zuständiges Gaszuführungssystem, welches das zur Waferkühlung vorgesehene Gas in die Ätzkammer zu- und ableitet. Die Energieversorgung des E-Chuck wird über eine Radio-Frequenz-Energiequelle (RF-Power) gewährleistet. Um den ESC ebenfalls vor Überhitzung schützen zu können, dient ein zusätzliches unabhängiges, mit einem speziellen Kühlmittel durchströmtes Kühlsystem. In Abbildung 33 wird der Aufbau des ESC visualisiert.¹⁵⁶

¹⁵² vgl. Yatsuzuka et al., 2001, S.399

¹⁵³ vgl. Kroninger et al., 1992, S.94f

¹⁵⁴ vgl. Hatakeyama and Asano, 1998, S.1733ff

¹⁵⁵ vgl. Asano et al., 2002, S.840ff

¹⁵⁶ vgl. Sun et al., 2015, S.3f

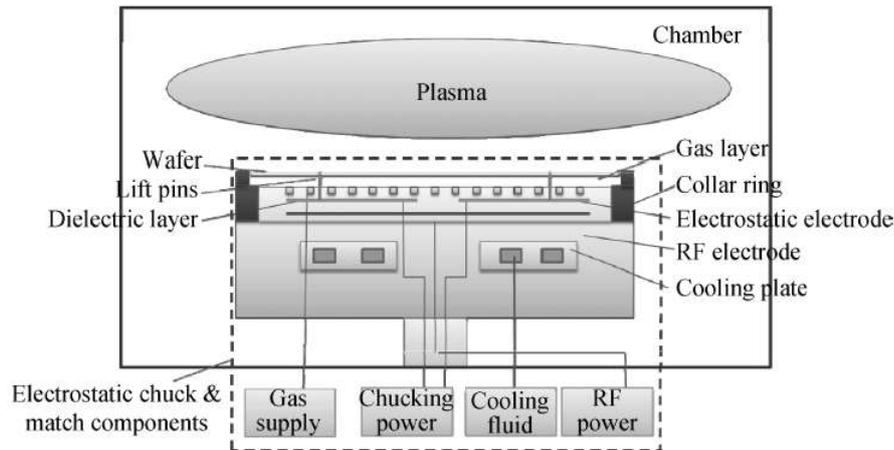


Abbildung 33: Aufbau des ESC Moduls¹⁵⁷

Funktionsweise:

Bevor ein neuer Wafer auf dem Chuck platziert werden kann, werden die elektrisch verstellbaren Stifte automatisch ausgefahren, auf denen der zu bearbeitende Wafer mittels eines Transportroboters platziert wird. Liegt der Wafer auf seiner Position, werden die Stifte wieder eingefahren, wodurch der Wafer auf dem zuvor beschriebenen dielektrischen Stapel liegt, wobei das Vorhandensein eines kleinen Spaltes zwischen Oberfläche und Wafer zu erwähnen ist. Dieser Spalt der im Fachjargon auch „Gas Layer“ genannt wird, resultiert aufgrund der vorhandenen Kühlnuten. Sobald der Wafer in Position gebracht wurde, erfolgt die Aktivierung der RF-Power, jedoch zunächst in einem niedrigen Bereich, um den Wafer vor anfänglicher Überhitzung zu schützen. Die verbaute Siliziumelektrode sorgt aufgrund der speziell konfigurierten Polarität (unter Aufbringen einer Clampvoltage), dass der Wafer festgehalten wird. Die Gaszufuhr wird geöffnet und das zur Kühlung verwendete Helium strömt in den Spalt zwischen Wafer und Stapeloberfläche. Im Anschluss darauf erfolgt das Anlegen der Biasspannung, die das Ionen-Bombardement des Wafers durch die im Plasma entstandenen geladenen Ionen auslöst (siehe Beschreibung Halbleiterätzung Kapitel 5.1.2). Sobald der Ätzvorgang durch das Erreichen des im Vorhinein definierten Endpunkt als abgeschlossen gilt, wird die Spannungszufuhr beendet. Dies impliziert, dass der Wafer nicht länger von der Elektrode angezogen wird. Das erneute Ausfahren der Stifte bewirkt die Rückführung des Wafers in seine Ausgangsstellung, von der aus die Entnahmesequenz des Wafers aus der Kammer, mittels Roboter erfolgt.^{158,159}

¹⁵⁷ vgl. Sun et al., 2015, S.3

¹⁵⁸ vgl. Asano et al., 2002, S.840f

¹⁵⁹ vgl. Yatsuzuka et al., 2001, S.401ff

5.1.6 Instandhaltungsabläufe und -planung

Die derzeitige Instandhaltungsstrategie des Unternehmens stellt eine Kombination aus Ausfallbehebung und periodischer Instandhaltung dar. Diese soll nun kurz aus der Perspektive einer regulären Schicht erklärt werden.

Das Schichtmodell sieht ein 3 Schichtsystem mit jeweils 8h vor (06:00-14:00, 14:00-22:00, 22:00-06:00, 365 Tage im Jahr). Zu Beginn jeder Schicht erfolgen sogenannte „Schichtmeetings“, in denen sich das Instandhaltungspersonal der kommenden und gehenden Schicht treffen, um sich bezüglich durchgeführter Maßnahmen, noch durchzuführender Instandhaltungen oder auch Anomalien im Produktionsbetrieb auszutauschen. In weiterer Folge teilt der für die Schicht zuständige Leiter der Instandhaltung das Personal den Maschinen zu, an denen Maßnahmen durchzuführen sind. Bei dieser Einteilung wird zwischen neuem, regulärem und Personal mit speziellen Ausbildungen unterschieden. Während neues Personal eher zu notwendigen periodischen Instandhaltungen wie zum Beispiel der Reinigung der Kammern herangezogen werden, sind reguläre Instandhalter mit der Ausfallbehebung der Maschinen beschäftigt. Personal mit speziellen Fähigkeiten können als eine Art „Task-Force“ verstanden werden, die bei Fehlern, deren Behebung ein großes Maß an Expertenwissen und erweiterten Kompetenzen voraussetzen, zum Einsatz kommen. Zur weiteren Unterstützung der Abteilung sind täglich Mitarbeiter der Firmen, die spezielle Komponenten der Anlagen herstellen (z.B. Pumpen), in Bereitschaft. Im Zuge von Interviews mit den Abteilungsleitern der Instandhaltung wurde eruiert, dass im Tagesgeschäft das Personal dauerhaft ausgelastet ist. Die Zuteilung zu den instandzusetzenden Maschinen erfolgt dabei meist nach Priorität bezüglich des Produktionsplans. Demzufolge wäre ein Modell oder Tool, welches prognostiziert bei welcher Maschine/Kammer als nächstes ein Ausfall zu erwarten ist, sehr hilfreich um zum richtigen Zeitpunkt die richtigen Maßnahmen zu veranlassen.

5.2 Beschreibung des Datensets

Die Datenerfassung erfolgt in verschiedenen Systemen die teilweise über Schnittstellen miteinander verbunden sind. Durch die Vielzahl voneinander unabhängig agierender Systeme, die Informationen in unterschiedlichen Formaten in verschiedenen Datenbanken abspeichern, gestaltet sich die Datenanalyse und die Generierung eines Datensets, welches für die Prognose von Ausfällen verwendet werden kann, aufwändig. Um die Daten für die Analyse und Modellbildung aufbereiten zu können, musste zunächst ein Extrakt erstellt werden. Dies geschieht mittels der Zuhilfenahme eines Python Skripts, welches die Dateien aus ihren ursprünglichen Formaten in CSV (comma separated values) Form konvertiert. Die nachfolgenden Beschreibungen der unterschiedlichen Datensysteme setzen diese Umwandlung bereits voraus.

5.2.1 Fertigungsspezifische Kennzahlen

Bei den fertigungsspezifischen Kennzahlen handelt es sich um prozessspezifische Kennzahlen. Sie beinhalten Informationen über:

- In welcher Kammer das Produkt gefertigt wurde
- Mehrere Zeitstempel (z.B. wann ein Los gestartet wurde)
- Typ des zu fertigen Produktes
 - Rezepttyp
 - Basis Type
- Einstellparameter für den Ätzprozess
 - Gaskonfigurationen
 - Spannungen
 - Frequenzen
 - ...

Es gilt zu beachten, dass die Einstellparameter für den Ätzprozess rezeptabhängig sind. Dies kann man daran erkennen, dass der Parametername mit dem zugehörigen Prozessschritt des zugehörigen Rezepts erweitert wird. Daraus resultiert eine Datenbank mit 495 verschiedenen Variablen, weswegen auf eine beispielhafte Visualisierung verzichtet wird.

5.2.2 Alarme

Alarme, die von der Maschine automatisch beim auftreten spezifischer Unregelmäßigkeiten im Produktionsprozess ausgelöst werden, werden im Datenbanksystem „Alarme“ gespeichert.

Es werden Informationen über:

- Timestamps (Zeitpunkt zu dem der Alarm auftritt)
- Alarmtext (Automatisch generierter Hinweis)
- AlarmID (eine dem Alarmtyp zugehörige Nummer)
- Alarmname
- Alarmkategorie
 - Critical
 - Warning
 - Error
 - Information
 - Other

hinterlegt.

Tabelle 4 visualisiert die Struktur der Alarmlogdaten.

Tabelle 4: Struktur der APC_Alarmlogs

Zeitpunkt	Text	ID	Name	Kategorie
07.07.2017 10:39:37	Biasvoltage Alarm	310010	Biasvoltage über Schwellwert	Warning
28.08.2017 09:24:58	Druck Ventil	310006	Fehler	Error

Es gilt zu berücksichtigen, dass die Alarmer nicht rezeptabhängig sind und folglich nur von maschinenspezifischen Einflüssen abhängen.

5.2.3 Limitverletzungen (LV)

Zu jedem fertigungsbasierten Einstellparameter aus den Kennzahlen (siehe Kapitel 5.2.1) wurden vom Unternehmen Toleranzgrenzen definiert, die nicht unter und auch überschritten werden dürfen. Im Falle von Verletzungen dieser Grenzwerte, werden Limitverletzungen ausgelöst. Diese können als zusätzliche Alarmlogs verstanden werden, wobei unter anderem folgende Informationen gespeichert werden:

- Betroffene Kennzahl
- Untere Toleranzgrenze (UTG)
- Aufgenommener Vergleichswert (VW)
- Obere Toleranzgrenze (OTG)
- Equipment (auf welche Maschine und in welcher Kammer trat Verletzung auf)
- Start des Loses
- Ende des Loses
- Kritikalität (1-6)
- Weitere Informationen (z.B. Rezept, Kommentare etc.)

Tabelle 5 zeigt einen Ausschnitt der Datalogs, die durch Limitverletzungen ausgelöst werden.

Tabelle 5: Ausschnitt Limitverletzungen

Betroffene Kennzahl	OTG	VW	UTG	Equipment	Start des Loses	Ende des Loses	Kritikalität
PWR_27MHZ	0	103	100	Maschine 3 P3	03.07.2017 14:06	03.07.2017 14:12	EMAIL
BIASVOLTAGE	0	105	100	Maschine 3 P3	05.08.2017 00:12	05.08.2017 00:24	EMAIL
PWR_2MHZ_REF	0	90	80	Maschine 3 P3	04.10.2017 11:02	04.10.2017 11:09	Sperren des Loses

Eine der wichtigsten Informationen stellt die Kritikalität der Limitverletzung dar. Dies resultiert daraus, dass abhängig von der Kritikalität unterschiedliche Maßnahmen veranlasst werden (automatisch und/oder manuell).

Im Folgenden sind die unterschiedlichen Kritikalitätsstufen und die damit verbundenen Konsequenzen aufgelistet:

- **E-Mail (Kritikalität 1):** Automatische Generierung einer Nachricht, die per Mail an die Prozesstechnik weitergeleitet wird.
- **Messenger (Kritikalität 2):** Der Maschinenführer erhält eine Nachricht, dass eine Toleranzgrenze verletzt wurde. Diese wird mittels POP-UP Fenster am Computer des Operators visualisiert.
- **Sperrung des Loses (Kritikalität 3):** Das aktuell zu fertigende Los wird gesperrt und einer Sonderüberprüfung unterzogen um festzustellen, ob es sich um Ausschuss handelt.
- **Blockierung Prozess (Kritikalität 4):** Die gesamte Anlage wird auf Stillstand gebucht, wodurch kein neues Los mehr gefertigt werden kann. Das sich bereits in der Fertigung befindliche Los wird jedoch noch finalisiert.
- **Blockierung Kammer (Kritikalität 5):** Die betroffene Kammer wird auf Stillstand gebucht, wodurch in ihr kein neues Los mehr gefertigt werden kann.
- **Equipment Stop (Kritikalität 6):** Anlage und betroffene Kammer werden heruntergefahren. Das sich gerade in der Fertigung befindliche Los wird ebenfalls gestoppt.

5.2.4 Instandhaltungsmaßnahmenbuch

Das Instandhaltungsmaßnahmenbuch dient als Dokumentationstool für Instandhalter. In diesem können Vermerke zu spezifischen Reparaturen angelegt werden. Zusätzlich besteht die Möglichkeit zu dokumentieren, welche Maßnahmen sich positiv oder auch negativ bei der Wiederinstandsetzung der Maschine ausgewirkt haben.

In diesem System werden Informationen über:

- Equipment,
- Zeitpunkt eines Events,
- der zum Event zugehörige GUI_Buchungstext,
- Art des Events,
- Aktion die Instandhalter gesetzt hat,
- Name des Instandhalters und
- Zeitstempel wann das Event behoben wurde

gesammelt abgespeichert und automatisch in das System der Statusmeldungen übernommen (siehe Kapitel 5.2.5). Hierbei muss beachtet werden, dass die Anzahl der Zeichen in den Statusmeldungen limitiert ist und folglich längere Texte abgeschnitten werden. Daraus folgt, dass die detaillierte Dokumentation zur Durchführung von Instandhaltungsmaßnahmen und Fehlerbehebung ausschließlich in

diesem System vorliegt. Tabelle 6 zeigt einen kurzen Auszug des Instandhaltungsmaßnahmenbuchs.

Tabelle 6: Exemplarischer Auszug aus dem Instandhaltungsmaßnahmenbuchs

EQUIPMENT	TIME_STAMP	GUI_BUCHUNGSTEXT	DOWN_EVENT	AKTION	AUTHOR	CREATED	CONTACT_PERSON
L2303-C	27.01.2015 02:33	BIASVOLTAGE_ESC	LV	Austausch ESC	Max Mustermann	27.01.2015 03:44	
L2303-C	28.01.2015 12:01	PM3TemperatureControlESCTargetCoolantFlowBelowAllowedLevelAlarm	ST	Überprüfung Kühlleitung ESC	Martina Musterfrau	28.01.2015 12:01	

Ein entscheidender Nachteil dieses Systems kristallisierte sich bei der Analyse der zur Verfügung gestellten Daten heraus: Da die Buchungen manuell durchgeführt und keine standardisierten Textbausteine verwendet werden, ist der Informationsgehalt der Einträge oftmals sehr gering. Dies liegt vor allem an der Buchungsqualität (eventuell auch Buchungsdisziplin) in Form von fehlenden und/oder unvollständigen Texten, nicht vereinheitlichter Bezeichnung von Bauteilen etc. (siehe Kapitel Modellierung des ersten Bayes'schen Netzes).

5.2.5 Statusmeldungen

Dieses System fungiert primär zur Erfassung von Maschinenzuständen, welche in verschiedene Kategorien eingeteilt werden können. Da diese Kategorisierung sehr umfangreich ist, werden nachfolgend nur die für diese Arbeit relevanten Zustände aufgeschlüsselt. Diese wären wie folgt:

- Maschine produktiv
- Limitverletzung
- Alarm
- Maschine hat einen Ausfall

Diese Statusmeldungen werden überwiegend manuell vom Operator gebucht. Nur im Falle von Limitverletzungen oder Alarmen, die vom System getriggert werden kann es zu einer automatischen „Down-Buchung“ des Mainframes kommen, was wiederum einem Stillstand der gesamten Anlage entspricht.

5.2.6 Reinigungsdaten

Da sich der Prozess der Halbleiterfertigung, wie bereits in Kapitel 5.1.1 beschrieben, sehr komplex gestaltet und einige Komponenten fertigungsbedingten Verschleißeffekten ausgesetzt sind, die die Qualität des Produktes nachhaltig beeinflussen, müssen regelmäßig Reinigungen und Wartungen besagter Module und Bauteile durchgeführt werden.

Um diese Reinigungen dokumentieren zu können wird eine Datenbank verwendet, in der alle Reinigungen und die Einsatzstunden der Verschleißkomponenten erfasst werden.

Folgende Informationen werden gesammelt:

- **Tool/Equipment:** Beschreibt die Kammer, in der die Reinigung durchgeführt wurde
- **Startzeit der Reinigung:** Zeitpunkt zu dem die Reinigung gestartet wurde
- **Endzeit der Reinigung:** Zeitpunkt, zu dem die Reinigung beendet wurde
- **Reinigungsgrund:** Begründung, warum eine Reinigung durchgeführt werden muss
- **Einsatzstunden von Verschleißkomponenten (RFh):** Verstrichene Zeit [h] seit letzter Wartung
 - VTT RFh: Verstrichene Zeit seit der letzten Komplettreinigung der Kammer
 - Quarz_Window: Einsatzzeit der Komponente Quarz
 - ESC_RFh: Einsatzzeit des Moduls E-Chuck
 - Nozzle_RFh: Einsatzzeit der Komponente Düse
 - Pendulum_RFh: Einsatzzeit der Komponente Pendulum
 - OES_Sichtfenster: Einsatzzeit OES_Sichtfenster
- **Leakrate vor VTT:** Leckrate vor einer Komplettreinigung
- **Leakrate nach VTT:** Leckrate nach einer Komplettreinigung

Tabelle 7 zeigt einen beispielhaften Auszug der Reinigungsdaten.

Tabelle 7: Auszug der Reinigungsdaten

Tool	Start-Zeit	End-Zeit	Reinigungs- grund	VTT RFh	Quarz RFh	ESC_RFh	Nozzle_RFh	Pend-ulum RFh	OES_--RFh	LR_vor_VT T	LR_nach VTT
L2301-A	23.03. 2017 02:47	23.03. 2017 20:00	DD nicht ok	557	1054	526	1078	1078	2495	0,22	0,39
L2301-A	27.04. 2017 06:24	27.04. 2017 04:50	RFh erreich t	778	1564	1147	1814	1780	3250	0,26	0,75

5.2.7 SAP

Sobald an einer Anlage eine Störung gebucht wird, wird automatisch ein zugehöriger Eintrag im SAP System angelegt. Die erstellten Aufträge dienen lediglich als Kostensammler, da das System nur zur Buchung von Ersatzteilen und Werkzeugen

verwendet wird. Automatisch erstellte Einträge werden nach der Durchführung der Instandhaltungsmaßnahme auch nicht quittiert, sondern kumuliert einmal im Quartal von einem Administrator abgeschlossen. Da dieses System demzufolge keinen Informationsgehalt für die Prädiktion von Ausfällen besitzt, wird es im Zuge der Modellbildung nicht verwendet und wird hier nur der Vollständigkeit halber angeführt.

5.2.8 Unterteilung des Datensatzes

Aufgrund der Tatsache, dass sich die Zeitreihen der zur Verfügung gestellten Daten hinsichtlich der Timestamps nicht vollständig überdecken, wurde mit der Schnittmenge der Daten gearbeitet, die in allen Systemen vorliegt. Somit wurde der Startpunkt mit 30.06.2017 und Endpunkt mit 31.12.2017 festgelegt. Um das Modell nach der Erstellung auf dessen Prognosefähigkeit validieren zu können, ist ein Aufteilen des Datensatzes erforderlich. Im Zuge dieser Arbeit wurden 75% der Daten zur Modellierung und 25% zur Validierung verwendet (Empfehlung Said et al.).¹⁶⁰ Somit startet das Trainingsdatensatz mit dem 30.06.2017 00:00:00 und endet mit dem 14.11.2017 00:00:00. Das Datensatz, das zur Validierung des erstellten Modells verwendet wird startet folglich mit dem 14.11.2017 00:00:01 und endet mit 31.12.2017 23:59:59.

5.3 Modellierung des ersten Bayes'schen Netzes

Im Folgenden soll die Vorgehensweise zur Erstellung des Netzes gezeigt werden. Diese orientiert sich an der bereits in Kapitel 4.1 präsentierten, von Naim et al. vorgeschlagenen „Vorgehensweise zur Erstellung von Bayes'schen Netzen“.¹⁶¹ Wie bereits in Kapitel 5.1.5 beschrieben, wird sich die Modellierung des Netzes mit dem Modul E-Chuck befassen. Der Vollständigkeit halber wird jedoch auch eine Empfehlung bezüglich der Durchführung entlang der gesamten Kammer abgegeben.

5.3.1 Variablen Identifikation

Zunächst mussten die Variablen bestimmt werden, die einen Einfluss auf das Ausfallverhalten der Kammer haben. Wie bereits in Kapitel 5.1.3 beschrieben, kann die Ätzkammer in 6 Module untergliedert werden, die sich jeweils aus den zugehörigen Komponenten und Submodulen zusammensetzen. Da meist bauteilspezifische Effekte Einfluss auf das Ausfallverhalten nehmen, wurde zunächst versucht, ein Modell auf dieser Basis aufzubauen.

¹⁶⁰ SAID et al., 2014, S.7

¹⁶¹ Na'im, 2008, S.209

5.3.2 Definition der BN Netzstruktur

Nach der Definition der Variablen, wird die Struktur des Netzes gebildet. Abbildung 34 zeigt diesen ersten Ansatz.

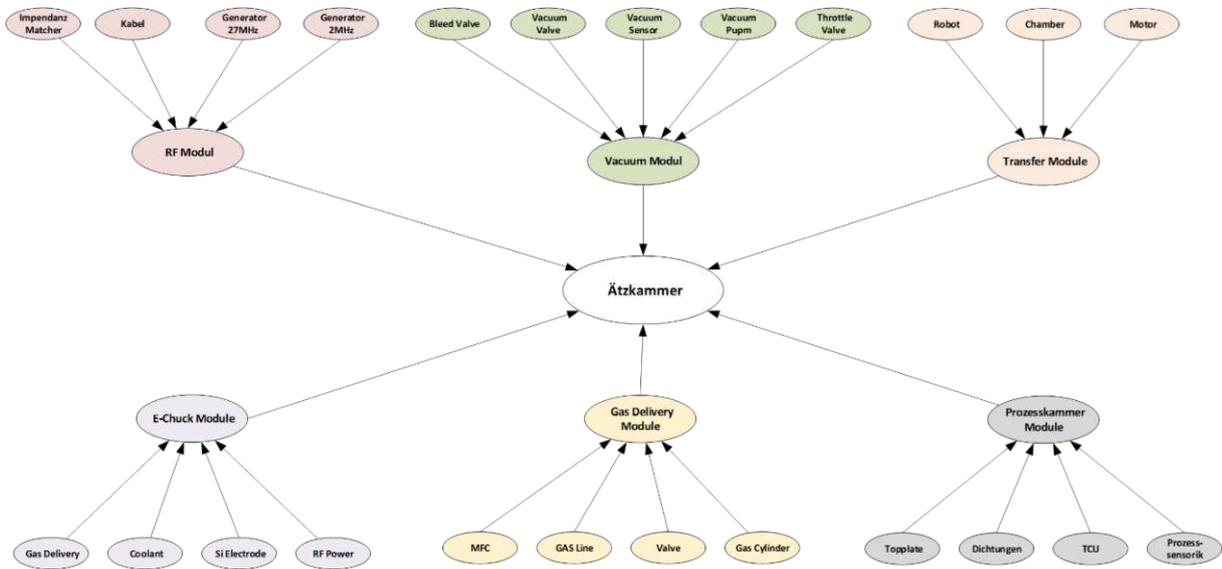


Abbildung 34: Erstmodellierung des Bayes'schen Netz für die gesamte Ätzkammer

Aufgrund der Tatsache, dass der Fokus dieser Arbeit auf dem Modul E-Chuck liegt, wird in Abbildung 35 ein Entwurf für dieses Subnetzwerk visualisiert.

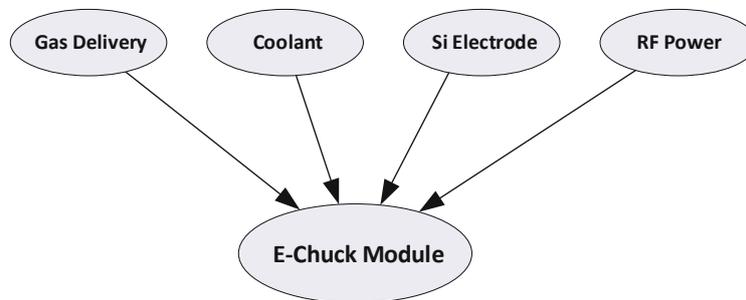


Abbildung 35: Netzwerk E-Chuck

Gespeicherte Informationen über die einzelnen Komponenten, wie zum Beispiel Austausch, Verschleiß etc. sollten dazu benutzt werden, eine Prognose über die restliche verbleibende Lebensdauer des Moduls E-Chuck anzustellen.

5.3.3 Definition der bedingten Wahrscheinlichkeiten

Nach der Definition der BN Netzstruktur sollen mit Hilfe der zur Verfügung gestellten Daten die Effekte und die bedingten Wahrscheinlichkeiten der Knoten kalkuliert werden. Aufgrund der Unvollständigkeit der Einträge im Instandhaltungsmaßnahmenbuch bezüglich verwendeter Ersatzteile bei der Durchführung von Instandhaltungsmaßnahmen konnte eine korrekte Zuordnung der für Ausfälle verantwortliche Bauteile (Gas Delivery, Coolant, Si-Elektrode, RF Power

und zugehörige Unterbaugruppen) nicht durchgeführt werden. Folglich konnte eine sachgemäße Bestimmung der Wahrscheinlichkeitswerte und die spätere Ausfallsprognose nicht in dieser Form durchgeführt werden. Demzufolge wird ein zweites Netzwerk entworfen, auf das in den nachfolgenden Kapiteln eingegangen wird.

5.4 Vorgehen bei der Erstellung des zweiten Netzes

Wie bereits beim ersten Netz gezeigt wird wieder nach der Logik:

- Variablen Identifikation
- Definition der BN Struktur
- Definition der bedingten Wahrscheinlichkeiten

vorgegangen.

5.4.1 Variablen Identifikation

Aufgrund der zuvor beschriebenen fehlenden Datenqualität wurde in einem weiteren Schritt versucht die Ausfallprognose anhand der Daten, die im Datenset der „Fertigungsspezifischen Kennzahlen“ aufscheinen, durchzuführen. Hierfür wurde es zunächst erforderlich, den jeweiligen Modulen die passenden Kennzahlen zuzuordnen. Es gilt zu beachten, dass die Namen der im Datenset vorliegenden Kennzahlen geändert und einige zur einfacheren Darstellung zusammengefasst wurden. Zusätzlich wurde eine weitere Unterteilung in „RF Power“, „Gas Delivery“ und „Coolant“ vorgenommen, um die Kennzahlen in Bereichen gruppieren zu können. In Abbildung 36 wird diese Klassifizierung anhand des E-Chucks gezeigt.

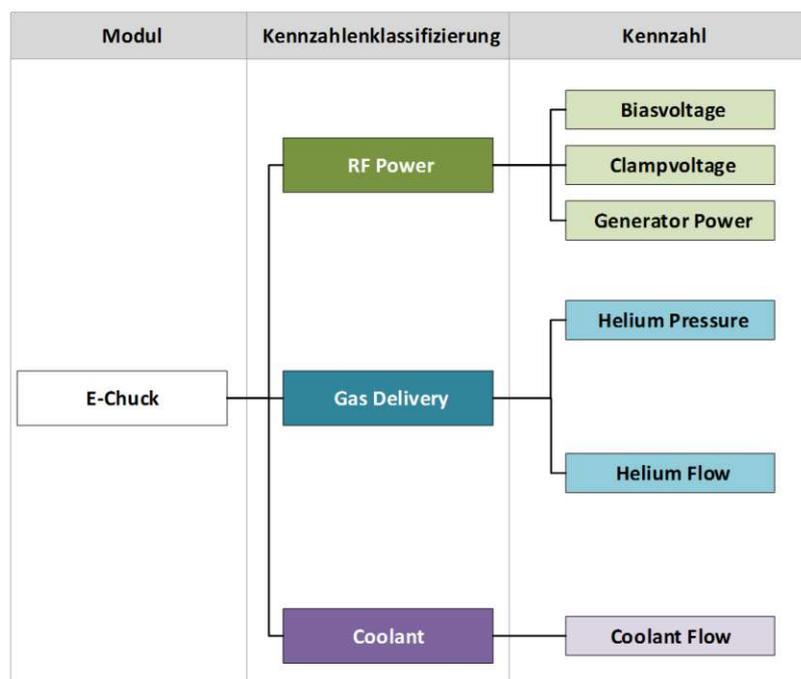


Abbildung 36: Zuordnung der Kennzahl zu Modulen und Klassifizierung anhand des E-Chucks

Im Zuge von Interviews mit Prozesstechnikern und Instandhaltungspersonal wurde eruiert, dass aus Erfahrung Limitverletzungen und Maschinenalarne als Ausfall beeinflussende Events gelten. Aufgrund des größeren Datenumfangs und dem höheren Maß an Datenqualität wurde folglich versucht, einen Zusammenhang zwischen Auftreten von Alarmen bzw. Limitverletzungen und der „Time-to-Failure“ (TTF) herzustellen. Aufgrund der Erwartungshaltung, dass die Einsatzzeiten der Kammer bzw. der Module ebenfalls miteinfließt, wurden auch die Reinigungsdaten miteinbezogen.

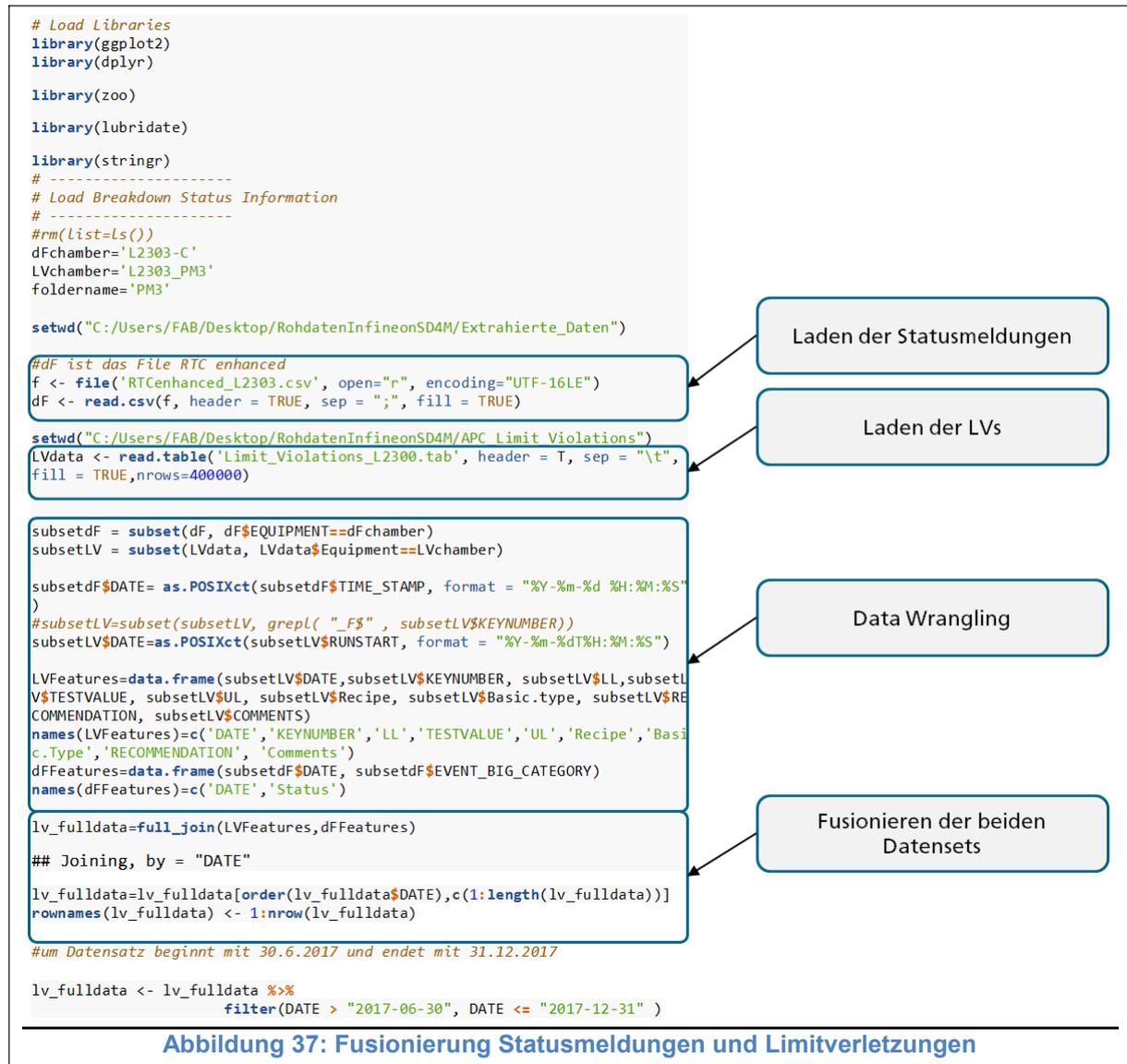
Für die Datenaufbereitung und -analyse wurde die statistische Programmiersprache R verwendet. Diese ist aufgrund der großen Anzahl an frei verfügbaren Packages (spezifische Erweiterungen für verschiedenste Einsatzzwecke) im Bereich der Technik weit verbreitet. Auch bei der Erstellung des Bayes'schen Netzes und den zugehörigen Analysen und Auswertungen wird R verwendet. In den nachfolgenden Kapiteln werden der Vollständigkeit halber nur einzelne Code-Fragmente präsentiert. Der gesamte Code, der zur Erstellung und Durchführung der Analyse erstellt wurde ist im Anhang (Kapitel 8) ausgewiesen.

5.4.1.1 Berechnung der „Time to Failure“ für Limitverletzungen (TTF)

Da das Ziel des Modells die Prognose von Ausfällen in einem gewissen zukünftigen Zeitfenster darstellt, mussten zunächst die Daten der verschiedenen Systeme verknüpft werden. Da die Informationen über den Maschinenstatus (siehe Kapitel 5.2.5) und dem Auftreten von Limitverletzungen bzw. Alarmen in unterschiedlichen Datenbanken abgespeichert werden, musste zunächst ein R Script geschrieben werden, welches diese beiden Informationen verknüpft. Hierfür wurde wie folgt vorgegangen:

- **Verknüpfung von Limitverletzungen und Statusmeldungen:**

Zunächst wurden die Daten aus den beiden Systemen in R geladen und insoweit manipuliert, dass nur zur späteren Verarbeitung benötigte Informationen enthalten sind. In weiterer Folge wurden die beiden Datensets über den Zeitstempel zusammengeführt (siehe Abbildung 37).



▪ **Berechnung der Time to Failure für Limitverletzungen:**

Nach der erfolgreichen Verknüpfung der Datensysteme musste die TTF berechnet werden. Hierfür wurde zunächst eine zusätzliche Spalte eingeführt, die für den Fall, dass die Kammer den Zustand „Breakdown“ oder „Limitviolation“ annimmt, den Wert „STOP“ beinhaltet. In weiterer Folge wurde mittels einer for-Schleife für jede Spalte die Differenz zwischen Datum der Spalte und Datum des nächsten Stillstandes eingetragen (siehe Abbildung 38).

```
#Lv_fullldata=Lv_fullldata[c(2110:10000),]

#neue spalte einfügen
lv_fullldata$StatusStop=NA
#wenn in status Limitviolation oder breakdown angezeigt wird dann wird die
#neue spalte mit STOP markiert (für Berechnung)
lv_fullldata$StatusStop[which(lv_fullldata$Status=='LimitViolation')]='STOP'
lv_fullldata$StatusStop[which(lv_fullldata$Status=='BREAKDOWN')]='STOP'

LVinds=which(lv_fullldata$StatusStop=='STOP')

lv_fullldata$TimeDiff=0

lv_fullldata$TimeDiff[1:LVinds[1]]=lv_fullldata$DATE[LVinds[1]]-
lv_fullldata$DATE[1:LVinds[1]]

for (jj in 2:length(LVinds)){
  lv_fullldata$TimeDiff[(LVinds[jj-
1]+1):LVinds[jj]]=lv_fullldata$DATE[LVinds[jj]]-lv_fullldata$DATE[(LVinds[jj-
1]+1):LVinds[jj]]
}
#Lv_fullldata$TimeDiff=as.factor(lv_fullldata$TimeDiff)
lv_fullldata$TimeDiff=lv_fullldata$TimeDiff/(60*60) #Display in Hours
```

Erstellen von Hilfsspalten die im Falle eines Breakdowns „STOP“ beinhalten

Befüllen der TTF Spalte über for-Schleife

Abbildung 38: Berechnung der TTF Zeiten

Mit diesen generierten Informationen konnte die TTF anhand von Boxplots visualisiert werden (siehe Abbildung 39). Es ist sofort ersichtlich, dass die Ausfallzeiten zwischen verschiedenen Arten von Limitverletzungen schwanken. Dies lässt sich mit der bereits zuvor beschriebenen Einteilung der LV in Kritikalitäten erklären. Während einige Limitverletzungen zu einem sofortigen Stillstand führen, dauert es bei anderen länger, bis Auswirkungen bemerkbar werden. Aus Datenschutzgründen wurden die genauen Namen der LV in der Abbildung geschwärzt.

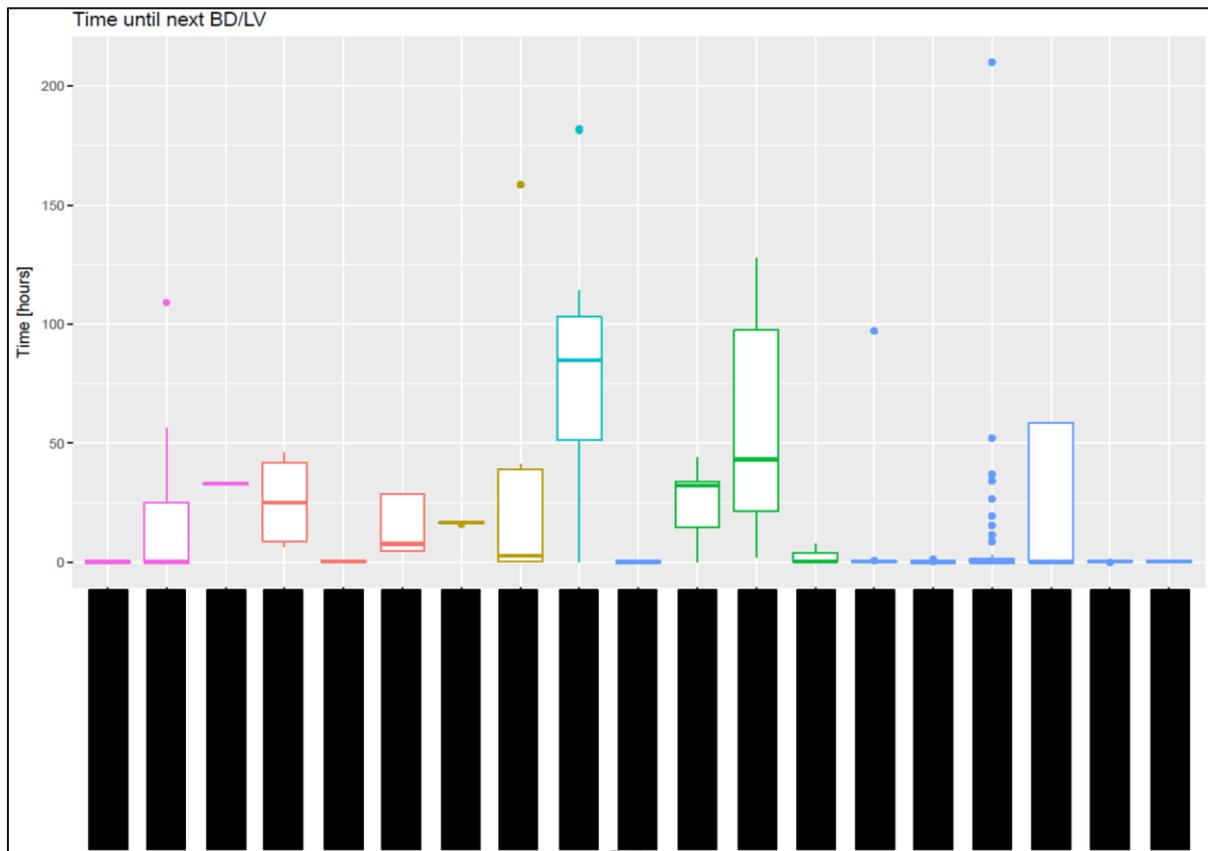


Abbildung 39: Auszug einiger Limitverletzungen und zugehörige TTF

5.4.1.2 Berechnung der TTF für Alarme

Bei der Ermittlung der Time to Failure für die auftretenden Alarme wurde nach derselben Art und Weise vorgegangen wie bei der Kalkulation der Limitverletzungen.

- **Verknüpfung von Alarmen und Statusmeldungen:**

Zunächst mussten wieder die Statusmeldungen und die Alarmlogs miteinander verknüpft werden. Da diese Verknüpfung analog zu der in Kapitel 5.4.1.1 präsentierten Methode von statten geht, wird das hierzu erstellte R-Script nicht gesondert ausgewiesen.

- **Berechnung der TTF für Alarme:**

Auch die Berechnung der TTF wurde bereits in Kapitel X erläutert. Folglich dient Abbildung 40 zur Darstellung des Ergebnisses für einige Alarme.

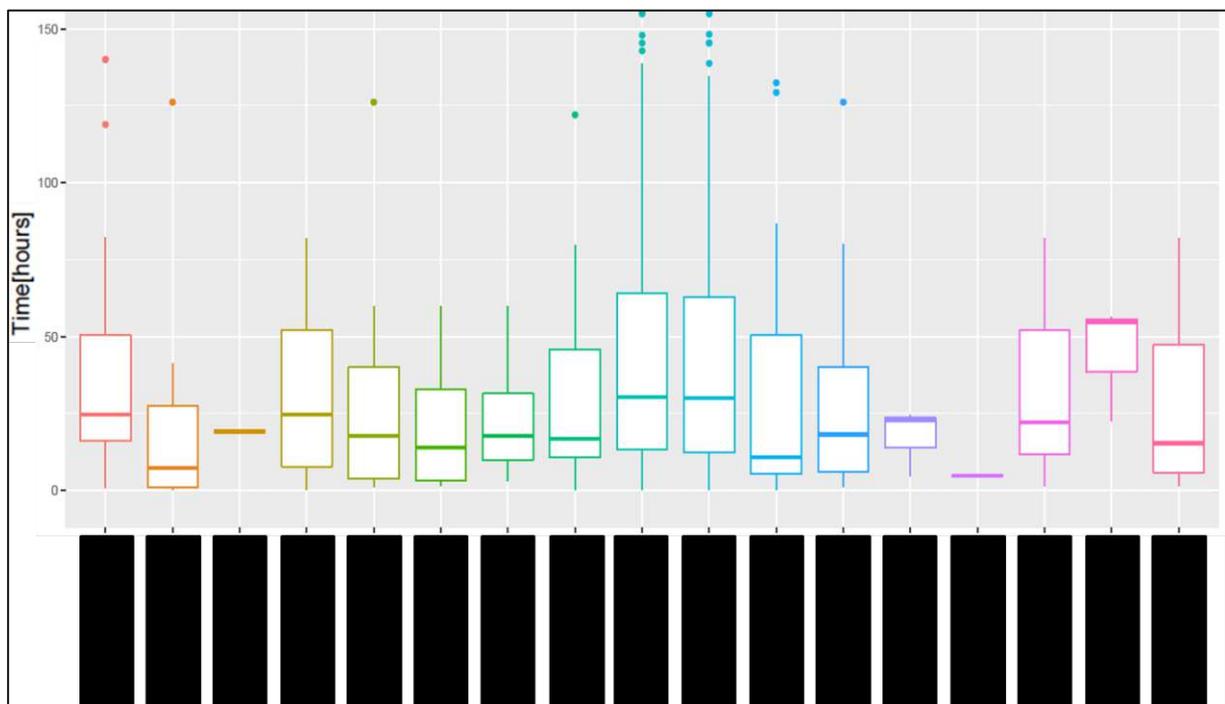


Abbildung 40: Auszug einiger Alarme und zugehörige TTF

5.4.2 Definition der Netzstruktur

Da im Zuge der Aufbereitung der Daten ein Einfluss spezifischer Limitverletzungen und Alarme auf die verbleibende Uptime der Maschine nachgewiesen werden konnte, wurde auf Basis dieser Erkenntnisse begonnen, ein Bayes'sches Netz aufzubauen. Zumal die Verknüpfung der einzelnen Knoten auf Expertenwissen basiert, erfolgte die Erstellung des Netzes auf manuellem Weg.

In Abbildung 41 wird ein erster Entwurf dieses Netzes dargestellt. Die Ebenen, in die das BN untergliedert wurde, sollen in weiterer Folge erklärt werden. Die spezifische Einteilung der Zustände je Knoten wird in späteren Kapiteln aufgezeigt.

• **Ebene 1**

In der ersten Ebene befinden sich die Elternknoten ESC RFH (Einsatzstunden des gesamten E-Chuck Modul) und VTT RFH (Einsatzstunden der Maschine seit der letzten Reinigung der Kammer). Dadurch, dass diese Parameter laut dem Expertenwissen der Prozesstechniker auf alle Komponenten Einfluss ausüben, haben sie eine Kante zu allen Knoten in der darauffolgenden Ebene.

• **Ebene 2**

In Ebene 2 befinden sich die Kennzahlen, die im Falle von Alarmen oder Limitverletzungen die Maschinenausfallzeit beeinflussen. Hierbei wurde die selbe, bereits präsentierte Unterteilung der Kennzahlen in Klassen angewandt, welche die nachkommende Ebene 3 darstellt. Es gilt zu beachten, dass auf die Unterteilung der Kennzahlen in Limitverletzungen und Alarme verzichtet wurde, um eine übersichtlichere Darstellung zu ermöglichen.

• **Ebene 3**

Ebene 3 besteht aus der Untergliederung des Moduls E-Chuck und beinhaltet die Knoten RF Power, Coolant und Gas Delivery. Zu jedem dieser Knoten wird ein Bereich der TTF prognostiziert. Die Ausfallprognose für das Modul E-Chuck erfolgt dann als Kombination dieser drei Submodule (Beispiel hierfür siehe Kapitel 5.6).

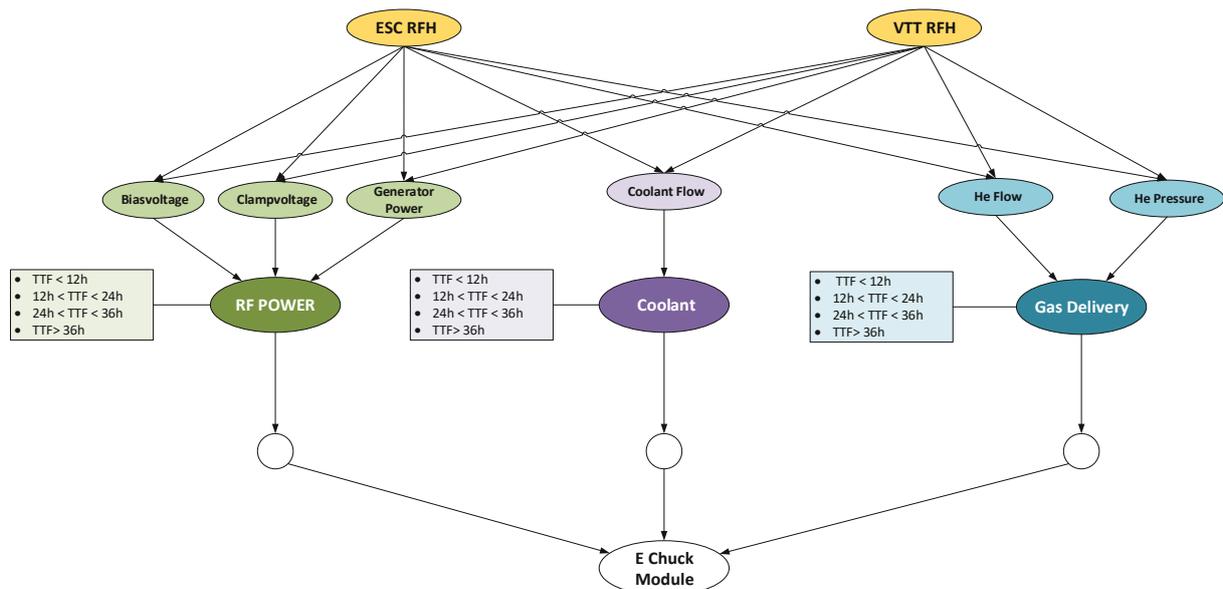


Abbildung 41: Bayes'Netz zur Ausfallprognose für das Modul E-Chuck

5.4.3 Definition der bedingten Wahrscheinlichkeiten

Aufgrund des großen Aufwandes im Bereich der Datenanalyse und -aufbereitung sollen im Zuge dieser Arbeit nicht alle der definierten Submodule des E-Chucks betrachtet werden, sondern ausschließlich der Bereich der RF Power. Dieses wird in Abbildung 42 visualisiert. Die detaillierte Beschreibung der einzelnen Knoten wird in den nachfolgenden Unterkapiteln durchgeführt.

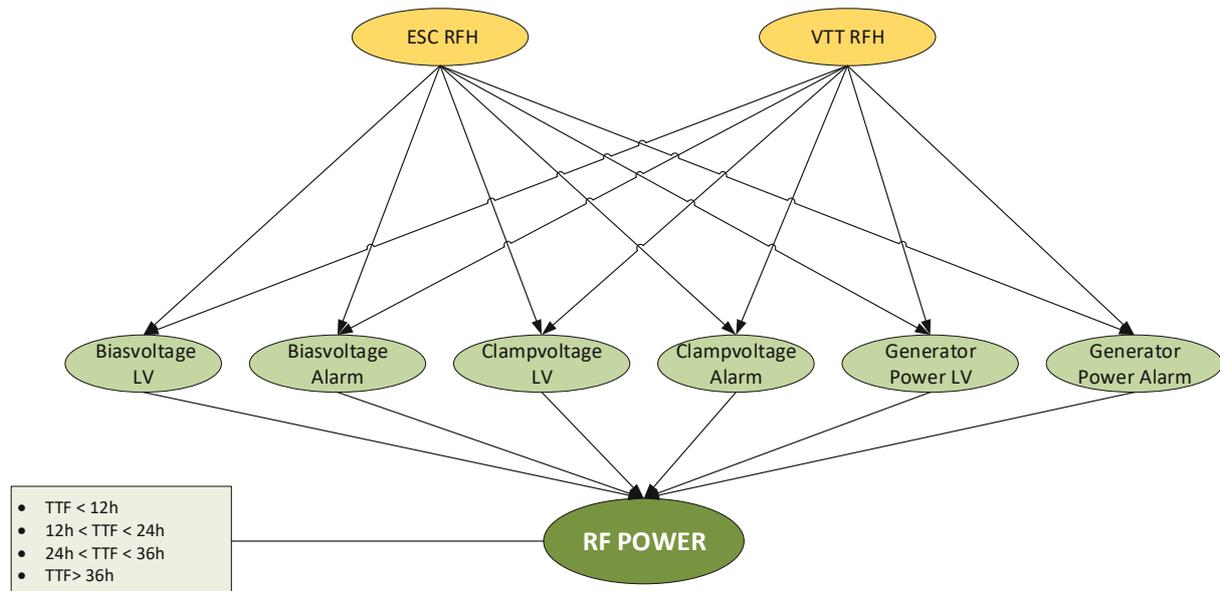


Abbildung 42: Submodell des E-Chucks

Um die bedingte Wahrscheinlichkeitstabelle befüllen zu können, müssen jeweils alle Zustände berücksichtigt werden, die der Knoten einnehmen kann. Diese Zustände stellen die Zeilen in der Tabelle dar und ergeben sich als Zustandskombination der Elternknoten (siehe Formel 5).

$$\text{Spaltenanzahl des Knotens} = \prod \text{Zustände der Elternknoten}$$

Formel 5: Spaltenkombination der einzelnen Knoten

Wie man aus der gezeigten Formel entnehmen kann ist es essentiell, die Anzahl der Zustände so gering wie möglich zu halten, da ansonsten der Komplexitätsgrad dramatisch ansteigt.

5.4.3.1 Knoten der 1.Ebene

Im Folgenden soll die Befüllung der Wahrscheinlichkeitstabellen der Knoten der ersten Ebene erfolgen.

- **Knoten ESC_RFH**

Bei diesem Knoten handelt es sich um die bereits verstrichenen Einsatzstunden des E-Chuck Moduls. Der Einfluss dieses Knotens resultiert aufgrund der bereits

erwähnten fertigungsbedingten Verschleißeffekte, die auf die Komponenten der Maschine wirken. Informationen über die bereits geleisteten Einsatzstunden können aus den Reinigungsdaten entnommen werden, wobei zu beachten ist, dass die RFh bei Eingriffen manuell vom jeweils zuständigen Operator eingetragen werden. Dieser liest die von der Maschine angezeigten Stunden ab und trägt sie in die Datenbank ein. Demzufolge wurden in den Daten einige Male Ungenauigkeiten bzw. Unvollständigkeiten entdeckt, deren Ursache auf Übertragungsfehler zurückzuführen sein könnten.

Definieren der Zustände für Knoten ESC_RFH:

Da die Einsatzstunden des ESC eine kontinuierliche fortlaufende Variable darstellt, mussten diese zur Anwendung im Netzwerk zu einer diskreten umgewandelt werden. In Absprache mit Prozesstechnikern und auf Basis der vorhandenen Daten wurden drei Bereiche definiert:

Tabelle 8: Diskretisierung der kontinuierlichen Variable ESC_RFH

Bereich 1	ESC_RFH < 1000h
Bereich 2	ESC_RFH 1000h < x < 2500h
Bereich 3	ESC_RFH >2500h

Wahrscheinlichkeitstabelle ESC_RFH:

Für das statische Netzwerk wurde angenommen, dass alle drei Zustände mit derselben Wahrscheinlichkeit auftreten können, woraus sich die folgende Wahrscheinlichkeitstabelle ergibt (siehe Abbildung 43).

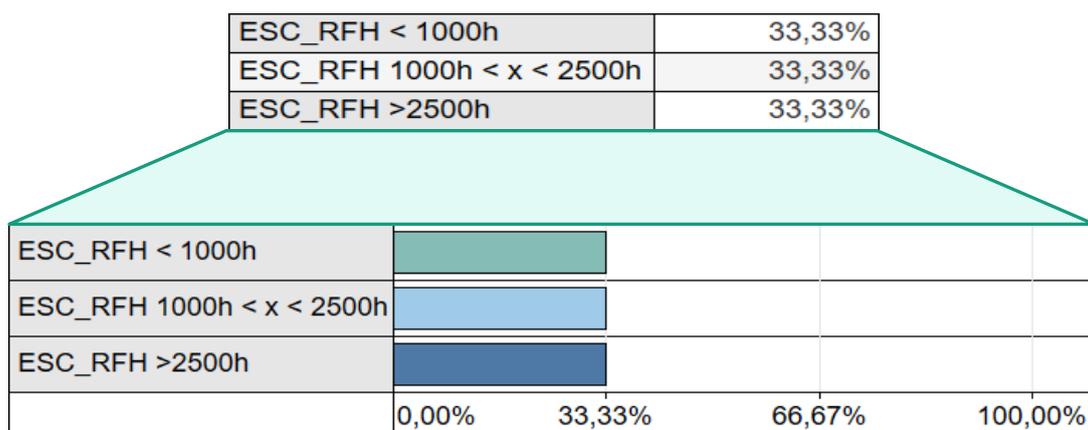


Abbildung 43: Bedingte Wahrscheinlichkeiten für Knoten ESC_RFH

Da es sich bei dem Knoten ESC_RFH ausschließlich um einen Elternknoten handelt, ergeben sich nur drei unabhängige Zustände. Die Auswirkungen dieser auf die zugehörigen Kindknoten, werden in den nachkommenden Kapiteln behandelt.

▪ **Knoten VTT**

In diesem Knoten werden die Einsatzstunden der Maschine abgebildet, die seit der letzten Komplettreinigung der Kammer vergangen sind. Diese Komplettreinigung muss in regelmäßigen Zeiträumen durchgeführt werden, wobei der maximale Zeitraum zwischen Reinigungen 380 Stunden nicht überschreiten darf. Die Informationen bezüglich dieser Kennzahl können ebenfalls aus den Reinigungsdaten generiert werden, weswegen die bereits beschriebene Problematik bezüglich der Unvollständigkeit bzw. Fehleranfälligkeit auch hier gilt.

Definieren der Zustände für Knoten VTT_RFH:

Auch bei diesem Knoten muss zunächst die anfänglich in kontinuierlicher Form vorliegende Information in ein diskretes Format umgewandelt werden. Dazu wurden wiederum drei Bereiche definiert (siehe Tabelle 9).

Tabelle 9: Diskretisierung der kontinuierlichen Variable VTT_RFH

Bereich 1	VTT < 100h
Bereich 2	VTT 100h < x < 300h
Bereich 3	VTT_RFH > 300h

Wahrscheinlichkeitstabelle VTT_RFH:

Bei der anfänglichen Umsetzung des statischen Netzwerkes wurde angenommen, dass jeder Zustand mit 33,33% Wahrscheinlichkeit auftreten wird. Da auch dieser Knoten als reiner Elternknoten fungiert, ergeben sich die folgenden, in Abbildung 44 gezeigten bedingten Wahrscheinlichkeiten.

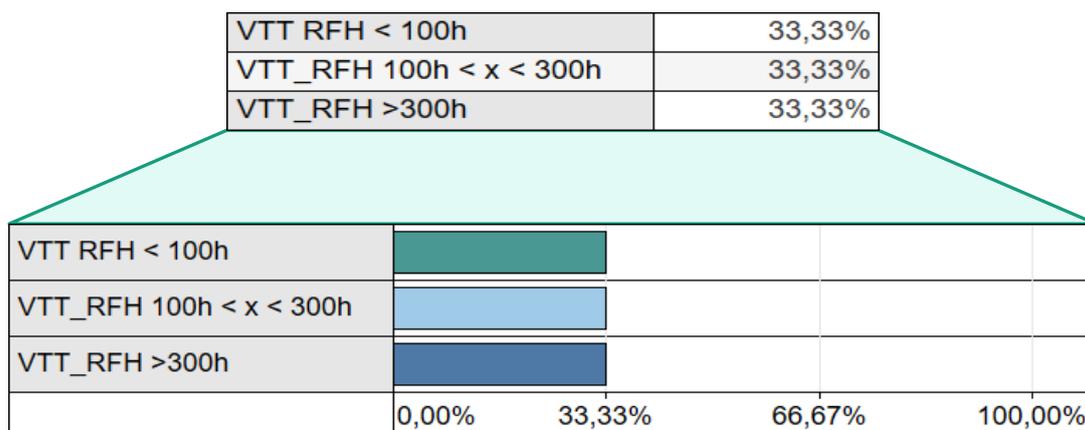


Abbildung 44: Bedingte Wahrscheinlichkeiten für Knoten VTT_RFH

5.4.3.2 Einfluss der Einsatzzeiten auf Limitverletzungen und Alarme

Bevor die Knoten der Ebene 2 beschrieben werden, soll der Einfluss der Ebene 1 auf die Ebene 2 erklärt werden.

Die Sichtung der Daten und der Input von Prozesstechnikern des Unternehmens ließ darauf schließen, dass das Auftreten von Limitverletzungen und Alarmen vor allem von der Einsatzzeit der Module und der Zeit seit der letzten Komplettreinigung abhängt. In Abbildung 45 wird dieser Zusammenhang anhand der Biasvoltage-Limitverletzungen und der Einsatzstunden des ESC-Moduls gezeigt. Die blauen Punkte markieren die Einsatzstunden, die roten Querlinien die Zeitpunkte zu denen Limitverletzungen aufgetreten sind. Wie man erkennen kann, sind im Bereich bis 1000h wenige Limitverletzungen zu erwarten. Ab 1000h ändert sich dieses Bild und erfährt seinen Höhepunkt ab etwa 2500h.

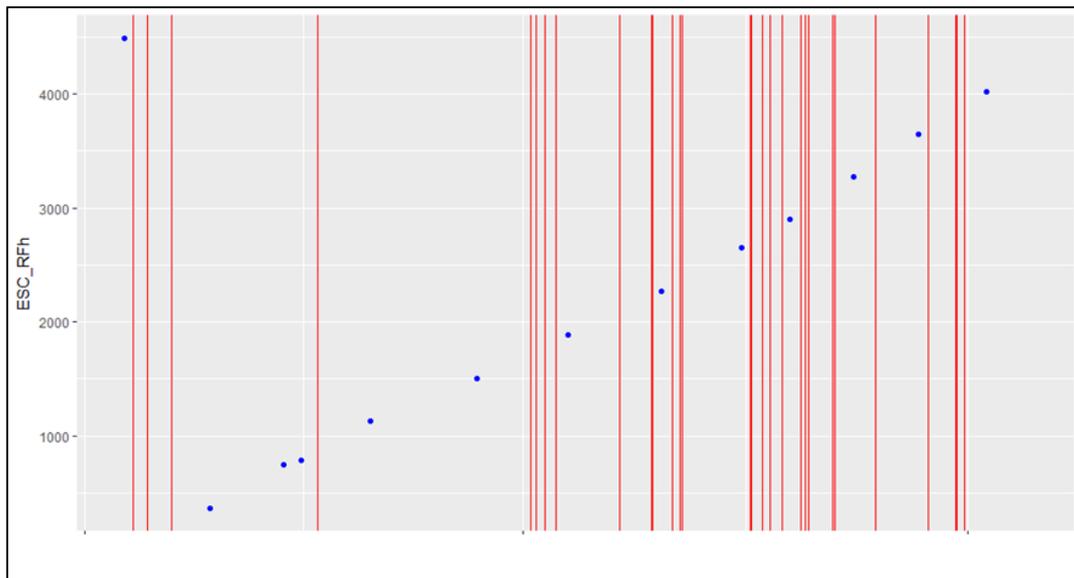


Abbildung 45: Einfluss der Einsatzzeit des E-Chuck Moduls auf das Auftreten von Biasvoltage Limitverletzungen

Ein Bezug zu weiteren Einflüssen auf Alarme und Limitverletzungen, zum Beispiel der von kritischen Komponenten oder auch Prozessparametern, konnte aufgrund der ungenügenden Datenqualität des Instandhaltungsmaßnahmenbuches nicht hergestellt werden.

5.4.3.3 Knoten der 2. Ebene

Nachdem das Vorgehen bei der Bestimmung der Zustände und Wahrscheinlichkeitstabellen bei allen Knoten der Ebene 2 gleich ist, wird diese in diesem Kapitel für alle zusammengefasst. Zunächst wird der betrachtete Knoten angeführt und der Einfluss der Kennzahl auf den Produktionsprozess erklärt. Danach erfolgt jeweils die Definition der Zustände und der zugehörigen Wahrscheinlichkeitstabellen.

In den zur Verfügung gestellten Daten konnten keine Informationen über z.B. Trends in den prozessspezifischen Kennzahlen, die auf zukünftige Limitverletzungen/Alarme schließen lassen, nachgewiesen werden. Daher wurde für die Berechnung der Wahrscheinlichkeit einer Limitverletzung/eines Alarms jeweils die Anzahl der

aufgetretenen Störungen in Bezug auf das gesamte Datenset summiert. Das „Lernen“ des Modells ist folglich auf das Auszählen der Trainingsdaten beschränkt. Aufgrund der spärlichen Datenmenge ist zu bedenken, dass es zu ungenauen Ergebnissen kommen kann. Es ist daher angedacht, die Ergebnisse im Zuge des Forschungsprojektes mit einem erweiterten Datenset zu validieren, sobald dieses verfügbar ist. Die Berechnung der Wahrscheinlichkeiten soll nun anhand vom Knoten Biasvoltage LV gezeigt werden.

Beispiel Biasvoltage LV (BLV):

$$P(BLV = true) = \frac{N(BLV = true)}{N(BLV = true) + N(BLV = false)} = 11,65\%$$

Formel 6: Berechnung der Auftretenswahrscheinlichkeit anhand des Knotens Biasvoltage Limitverletzung

Um den Einfluss des Verschleißes der Maschine (ESC_RFh und VTT_RFH siehe Kapitel 5.4.3.2) mitabbilden zu können, wurden Faktoren eingeführt, mit denen die ermittelten Wahrscheinlichkeiten, die von den Einsatzstunden abhängig sind, multipliziert wurden (siehe Wahrscheinlichkeitstabellen der Knoten).

▪ Knoten Biasvoltage LV

Die Biasvoltage (Bias-Spannung) kontrolliert die Energie, mit der die Ionen auf den Wafer treffen. Wie bereits erwähnt führt der Unterschied der Spannung zwischen den beiden Elektroden dazu, dass die im Plasma entstandenen Ionen aufgrund ihrer Polarität beschleunigt werden. Daraus resultiert, dass eine höhere Biasspannung zu einer höheren Energiedichte führt und folglich zu einer höheren Ätzrate führt. Dementsprechend kann eine zu hohe Biasspannung zu einer Verletzung der im Vorhinein definierten Prozesszeit führen.

Definieren der Zustände für Knoten Biasvoltage LV

Für diesen Knoten wurden zwei Zustände definiert, um die erforderliche diskrete Form vorliegen zu haben (siehe Tabelle 10).

Tabelle 10: Zustände Biasvoltage LV

Zustand 1	TRUE
Zustand 2	FALSE

Aufgrund der Abhängigkeit dieses Knotens von den beiden Elternknoten mit jeweils 3 Zuständen ergibt sich:

$$\text{Spaltenanzahl des Knotens Biasvoltage LV} = \prod \text{Zustände der Elternknoten}$$

$$\text{Spaltenanzahl des Knotens Biasvoltage LV} = 3 * 3 = 9$$

Formel 7: Spaltenzahl des Knotens Biasvoltage LV

Wahrscheinlichkeitstabelle Biasvoltage LV

Die Ergebnisse der Berechnung der Auftretenswahrscheinlichkeit und die Gewichtung in Abhängigkeit der bereits geleisteten Einsatzstunden kann Tabelle 11 entnommen werden.

Tabelle 11: Wahrscheinlichkeitstabelle Biasvoltage LV

Elternknoten ESC_RFH	Elternknoten VTT_RFH	Biasvoltage LV TRUE	Biasvoltage LV FALSE
< 1000h	< 100h	11,65%	88,35%
1000h < x < 2500h	<100h	34,95%	65,05%
>2500h	<100h	69,89%	30,11%
< 1000h	100h < x < 300h	23,30%	76,70%
1000h < x < 2500h	100h < x < 300h	58,24%	41,76%
>2500h	100h < x < 300h	93,19%	6,81%
< 1000h	>300h	46,59%	53,41%
1000h < x < 2500h	>300h	81,50%	18,50%
>2500h	>300h	99%	1%

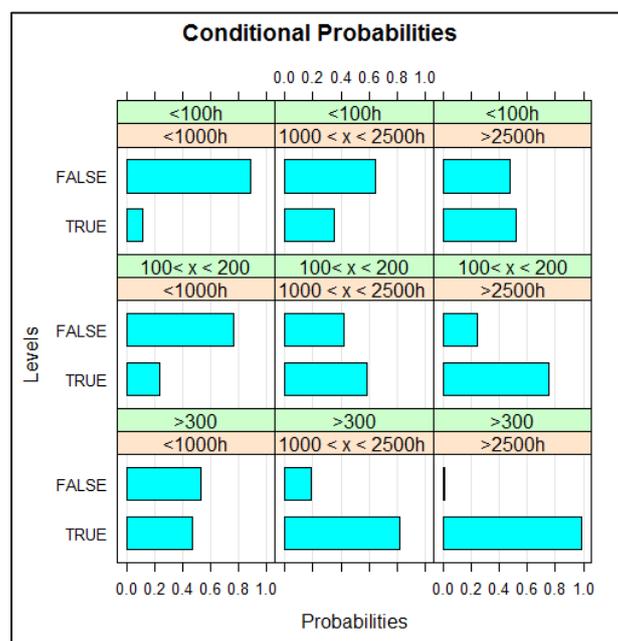


Abbildung 46: CPT Tabelle Biasvoltage LV

▪ **Knoten Biasvoltage Alarm**

Da der Einfluss der Biasvoltage auf den Produktionsprozess bereits erörtert wurde, soll hier im Weiteren darauf verzichtet werden.

Definieren der Zustände für Knoten Biasvoltage Alarm

Auch im Fall des Biasvoltage Alarm wird eine Diskretisierung in TRUE und FALSE durchgeführt.

Tabelle 12: Zustände Biasvoltage Alarm

Zustand 1	TRUE
Zustand 2	FALSE

Wahrscheinlichkeitstabelle Biasvoltage Alarm

Aufgrund der Abhängigkeit dieses Knotens von den beiden Elternknoten mit jeweils drei Zuständen ergibt sich:

$$\text{Spaltenanzahl des Knotens Biasvoltage Alarm} = \prod \text{Zustände der Elternknoten}$$

$$\text{Spaltenanzahl des Knotens Biasvoltage Alarm} = 3 * 3 = 9$$

Formel 8: Spaltenzahl des Knoten Biasvoltage Alarm

Da in den Trainingsdaten zu wenig Informationen bezüglich des Auftretens dieses Alarmtyps aufgezeichnet wurden, wurde die nachfolgende Wahrscheinlichkeitstabelle an die der Limitverletzungen angepasst.

Tabelle 13: Wahrscheinlichkeitstabelle Biasvoltage Alarm

Elternknoten ESC_RFH	Elternknoten VTT_RFH	Biasvoltage Alarm TRUE	Biasvoltage Alarm False
< 1000h	< 100h	11,65%	88,35%
1000h < x < 2500h	<100h	34,95%	65,05%
>2500h	<100h	69,89%	30,11%
< 1000h	100h < x < 300h	23,30%	76,70%
1000h < x < 2500h	100h < x < 300h	58,24%	41,76%
>2500h	100h < x < 300h	93,19%	6,81%
< 1000h	>300h	46,59%	53,41%
1000h < x < 2500h	>300h	81,50%	18,50%
>2500h	>300h	99%	1%

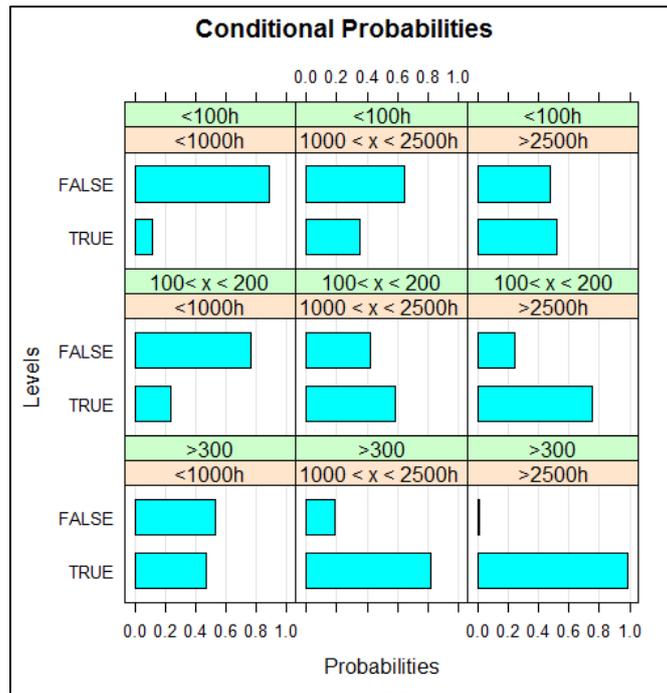


Abbildung 47: CPT Tabelle Biasvoltage Alarm

▪ **Knoten Clampvoltage LV**

Während des Produktionsprozesses wird wie bereits beschrieben der Wafer mittels einer elektrostatisch aufgebracht Kraft festgespannt. Um diese Kraft aufbringen zu können wird eine sogenannte Clampvoltage (Clampspannung) angelegt. Nach dem Produktionsprozess wird diese Spannung wieder abgeklemmt, wobei der Wafer in der Regel aufgrund spezieller Ladungseffekte noch für eine kurze Zeit auf der Oberfläche des Chucks „festklebt“ und einen sofortigen Weitertransport unmöglich macht. Bei falscher Einstellung oder Veränderung der Parameter ist somit ein Problem beim sogenannten „Dechuck-Prozess“ (Entspannen) möglich, der im schlimmsten Fall auch zu Waferbruch führen kann.¹⁶²

Definieren der Zustände für Knoten Clampvoltage LV

Wie in Tabelle 14 ersichtlich, erfolgte die Diskretisierung des Knotens anhand der Einteilung der Zustände in TRUE (tritt auf) und FALSE (tritt nicht auf).

Tabelle 14: Zustände Clampvoltage LV

Zustand 1	TRUE
Zustand 2	FALSE

¹⁶² Qin and McTeer, 2007

Wahrscheinlichkeitstabelle Clampvoltage LV

Aufgrund der Abhängigkeit dieses Knotens von den beiden Elternknoten mit jeweils 3 Zuständen ergibt sich:

$$\text{Spaltenanzahl des Knotens Clampvoltage LV} = \prod \text{Zustände der Elternknoten}$$

$$\text{Spaltenanzahl des Knotens Clampvoltage LV} = 3 * 3 = 9$$

Formel 9: Spaltenzahl des Knoten Clampvoltage LV

Tabelle 15: Wahrscheinlichkeitstabelle Clampvoltage LV

Elternknoten ESC_RFH	Elternknoten VTT_RFH	Clampvoltage LV True	Clampvoltage LV False
< 1000h	< 100h	0,24%	99,76%
1000h < x < 2500h	<100h	1,43%	98,57%
>2500h	<100h	2,16%	97,84%
< 1000h	100h < x < 300h	0,95%	99,05%
1000h < x < 2500h	100h < x < 300h	2,40%	97,60%
>2500h	100h < x < 300h	3,12%	96,88%
< 1000h	>300h	1,44%	98,56%
1000h < x < 2500h	>300h	2,88%	97,12%
>2500h	>300h	3,60%	96,40%

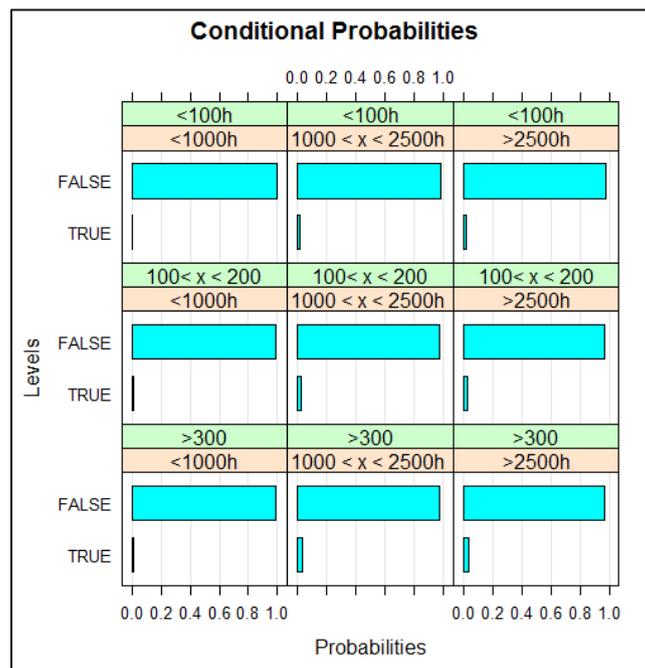


Abbildung 48: CPT Tabelle Clampvoltage LV

▪ **Knoten Clampvoltage Alarm**

Da der Nutzen und der Einfluss der Clampvoltage bereits im Zuge der Erörterung von Knoten Clampvoltage LV erfolgte, soll hier nicht nochmals im Detail darauf eingegangen werden.

Definieren der Zustände für Knoten Clampvoltage Alarm

Analog zu Knoten Clampvoltage LV erfolgte die Zuteilung der möglichen Zustände in TRUE und FALSE.

Tabelle 16: Zustände Clampvoltage Alarm

Zustand 1	TRUE
Zustand 2	FALSE

Wahrscheinlichkeitstabelle Clampvoltage Alarm

Aufgrund der Abhängigkeit dieses Knotens von den beiden Elternknoten mit jeweils 3 Zuständen ergibt sich:

$$\text{Spaltenanzahl des Knotens Clampvoltage Alarm} = \prod \text{Zustände der Elternknoten}$$

$$\text{Spaltenanzahl des Knotens Clampvoltage Alarm} = 3 * 3 = 9$$

Formel 10: Spaltenzahl des Knoten Clampvoltage Alarm

Tabelle 17: Tabelle Clampvoltage Alarm

Elternknoten ESC_RFH	Elternknoten VTT_RFH	Clampvoltage Alarm True	Clampvoltage Alarm False
< 1000h	< 100h	2,60%	97,40%
1000h < x < 2500h	<100h	15,57%	84,43%
>2500h	<100h	23,36%	76,64%
< 1000h	100h < x < 300h	10,38%	89,62%
1000h < x < 2500h	100h < x < 300h	25,96%	74,04%
>2500h	100h < x < 300h	33,75%	66,25%
< 1000h	>300h	15,57%	84,43%
1000h < x < 2500h	>300h	31,16%	68,84%
>2500h	>300h	38,94%	61,06%

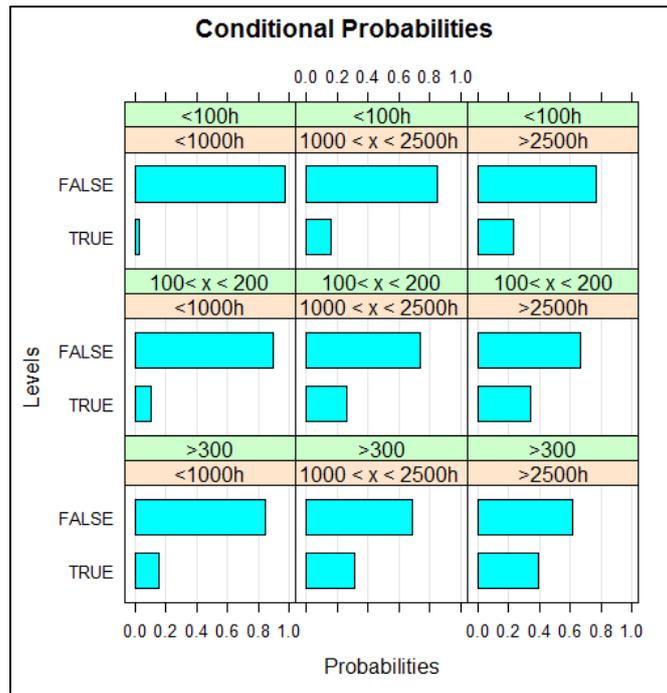


Abbildung 49: CPT Tabelle Clampvoltage Alarm

▪ **Knoten Generator Power LV**

In diesem Knoten werden alle Verletzungen und Anomalien abgebildet, die mit dem energieerzeugenden Generator und dem zugehörigen „Matching Network“ des E-Chuck Moduls in Zusammenhang stehen.

Definieren der Zustände für Knoten Generator Power LV

Die Zuteilung der Zustände erfolgte wie bereits bei den zuvor beschriebenen Knoten in TRUE und FALSE.

Tabelle 18: Zustände Generator Power LV

Zustand 1	TRUE
Zustand 2	FALSE

Wahrscheinlichkeitstabelle Generator Power LV

Bei der Bestimmung der Wahrscheinlichkeitstabelle wurde nach dem bereits präsentierten Schema vorgegangen. Auch die bereits erwähnten, von den Einsatzstunden abhängigen Faktoren, wurden angewendet.

Aufgrund der Abhängigkeit dieses Knotens von den beiden Elternknoten mit jeweils 3 Zuständen ergibt sich:

$$\text{Spaltenanzahl des Knotens Generator Power LV} = \prod \text{Zustände der Elternknoten}$$

$$\text{Spaltenanzahl des Knotens Generator Power LV} = 3 * 3 = 9$$

Formel 11: Spaltenzahl des Knoten Generator Power LV

Tabelle 19: Wahrscheinlichkeitstabelle Generator Power LV

Elternknoten ESC_RFH	Elternknoten VTT_RFH	Generator Power LV True	Generator Power LV False
< 1000h	< 100h	0,54%	99,46%
1000h < x < 2500h	<100h	3,24%	96,76%
>2500h	<100h	4,86%	95,14%
< 1000h	100h < x < 300h	2,16%	97,84%
1000h < x < 2500h	100h < x < 300h	5,39%	94,61%
>2500h	100h < x < 300h	7,01%	92,99%
< 1000h	>300h	3,24%	96,76%
1000h < x < 2500h	>300h	6,47%	93,53%
>2500h	>300h	8,09%	91,91%

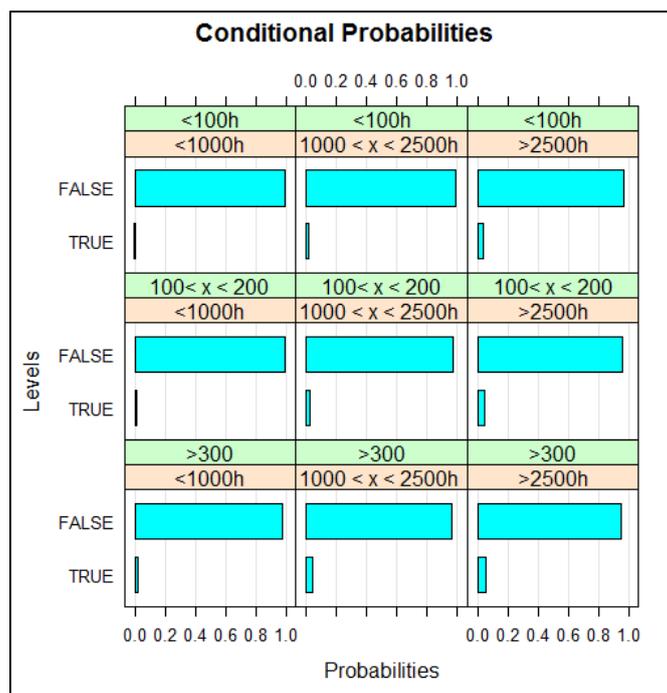


Abbildung 50: CPT Tabelle Generator LV

▪ **Knoten Generator Power Alarm**

Definieren der Zustände für Knoten Generator Power Alarm

Tabelle 20: Zustände Generator Power Alarm

Zustand 1	TRUE
Zustand 2	FALSE

Wahrscheinlichkeitstabelle Generator Power Alarm

Aufgrund der Abhängigkeit dieses Knotens von den beiden Elternknoten mit jeweils 3 Zuständen ergibt sich:

$$\text{Spaltenanzahl des Knotens Generator Power Alarm} = \prod \text{Zustände der Elternknoten}$$

$$\text{Spaltenanzahl des Knotens Generator Power Alarm} = 3 * 3 = 9$$

Formel 12: Spaltenzahl des Knoten Generator Power Alarm

Tabelle 21: Generator Power Alarm

Elternknoten ESC_RFH	Elternknoten VTT_RFH	Generator Power Alarm True	Generator Power Alarm False
< 1000h	< 100h	8,76%	91,24%
1000h < x < 2500h	<100h	26,28%	73,72%
>2500h	<100h	52,56%	47,44%
< 1000h	100h < x < 300h	17,52%	82,48%
1000h < x < 2500h	100h < x < 300h	43,81%	56,19%
>2500h	100h < x < 300h	70,09%	29,91%
< 1000h	>300h	35,04%	64,96%
1000h < x < 2500h	>300h	61,32%	38,68%
>2500h	>300h	87,61%	12,39%

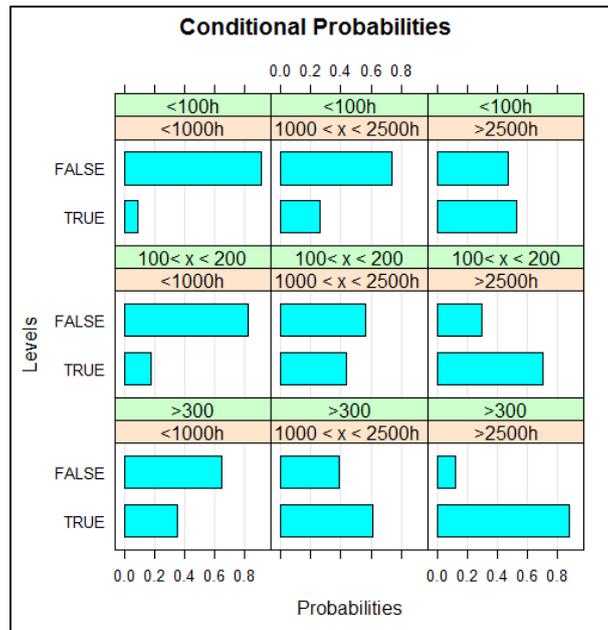


Abbildung 51: CPT Tabelle Generator Alarm

5.4.3.4 Knoten der 3.Ebene

In der dritten Ebene (des Submoduls) befindet sich ausschließlich der Knoten RF Power. Die Befüllung der zugehörigen bedingten Wahrscheinlichkeitstabelle wird in weiterer Folge erläutert.

- RF Power

Definieren der Zustände für Knoten RF Power

Für diesen Knoten, in dem die Wahrscheinlichkeit eines zukünftigen Ausfalls ausgegeben werden soll, wurden vier Zustände definiert (siehe Tabelle 22).

Tabelle 22: Zustände Knoten RF Power

Zustand 1	TTF < 12h
Zustand 2	12h < TTF < 24h
Zustand 3	24h < TTF < 36h
Zustand 4	TTF > 36h

Der Grund für diese Einteilung liegt im Schichtmodell des Unternehmens. Ein Ergebnis, das auf einen Ausfall in <12h hinweist ist ein Indikator dafür, dass noch innerhalb dieser oder in der ersten Hälfte der nächsten Schicht ein Breakdown auftritt.

Wahrscheinlichkeitstabelle RF Power

$$\text{Spaltenanzahl des Knotens RF Power} = \prod \text{Zustände der Elternknoten}$$

$$\text{Spaltenanzahl des Knotens RF Power} = 2 * 2 * 2 * 2 * 2 * 2 = 2^6 = 64$$

Formel 13: Spaltenzahl des Knoten RF Power

Tabelle 23 zeigt alle möglichen Zustandskombinationen der Elternknoten (Knoten der 2. Ebene).

Tabelle 23: Mögliche Zustandskombinationen der Elternknoten

Biasvoltage Alarm	Biasvoltage LV	Clampvoltage Alarm	Clampvoltage LV	Generator Power Alarm	Generator Power LV
True	True	True	True	True	True
False	True	True	True	True	True
True	False	True	True	True	True
False	False	True	True	True	True
True	True	False	True	True	True
False	True	False	True	True	True
True	False	False	True	True	True
False	False	False	True	True	True
True	True	True	False	True	True
False	True	True	False	True	True
True	False	True	False	True	True
False	False	True	False	True	True
True	True	False	False	True	True
False	True	False	False	True	True
True	False	False	False	True	True
False	False	False	False	True	True
True	True	True	True	False	True
False	True	True	True	False	True
True	False	True	True	False	True
False	False	True	True	False	True
True	True	False	True	False	True
False	True	False	True	False	True
True	False	False	True	False	True
False	False	False	True	False	True
True	True	True	False	False	True
False	True	True	False	False	True
True	False	True	False	False	True
False	False	True	False	False	True
True	True	False	False	False	True
False	True	False	False	False	True
True	False	False	False	False	True
False	False	False	False	False	True
True	True	True	True	True	False
False	True	True	True	True	False
True	False	True	True	True	False
False	False	True	True	True	False
True	True	False	True	True	False
False	True	False	True	True	False
True	False	False	True	True	False
False	False	False	True	True	False
True	True	True	False	True	False
False	True	True	False	True	False
True	False	True	False	True	False
False	False	True	False	True	False
True	True	False	False	True	False
False	True	False	False	True	False
True	False	False	False	True	False
False	False	False	False	True	False
True	True	True	True	False	False

Um eine Zuordnung zu den verschiedenen Zuständen des Knotens ermöglichen zu können, muss zunächst ein Schritt zurück zur Verknüpfung der Alarme und

Limitverletzungen gemacht werden. Wie in Kapitel 5.4 beschrieben, lösen spezifische Limitverletzungen und Alarme nach einem bestimmten Zeitraum Ausfälle aus. Im nachfolgenden werden die für diesen Use-Case erforderlichen aufgelistet.

TTF Werte für Biasvoltage:

Tabelle 24: TTF Biasvoltage Alarm

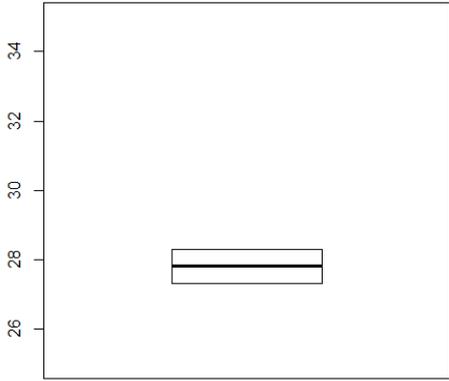
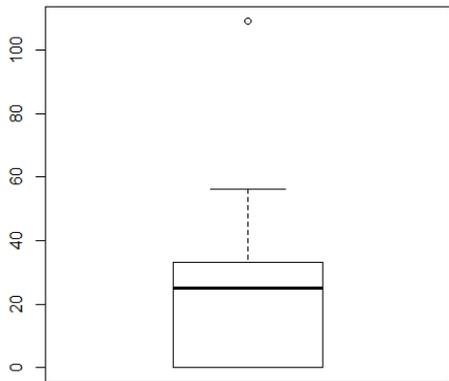
Biasvoltage Alarm	Oberes Quantil	28,30h	
	Mittelwert	27,82h	
	Median	27,82h	
	Unteres Quantil	27,34h	

Tabelle 25: TTF Biasvoltage LV

Biasvoltage LV	Oberes Quantil	32,40 h	
	Mittelwert	27,97h	
	Median	25,09h	
	Unteres Quantil	0,23h	

TTF Werte für Clampvoltage:

Tabelle 26: TTF Clampvoltage Alarm

Clampvoltage Alarm	Oberes Quantil	50,73h	
	Mittelwert	32,54h	
	Median	14,22h	
	Unteres Quantil	4,27h	

Tabelle 27: Clampvoltage LV

Clampvoltage LV	Oberes Quantil	34,81h	
	Mittelwert	24,36h	
	Median	23,27h	
	Unteres Quantil	13,37h	

TTF Werte für Generator Power:

Tabelle 28: TTF Generator Power Alarm

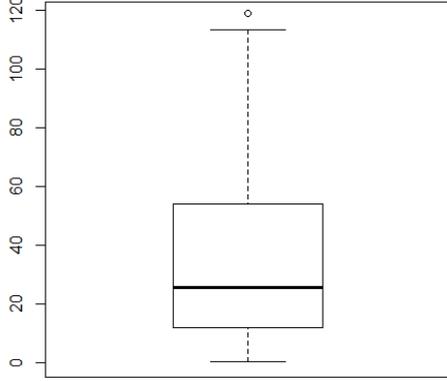
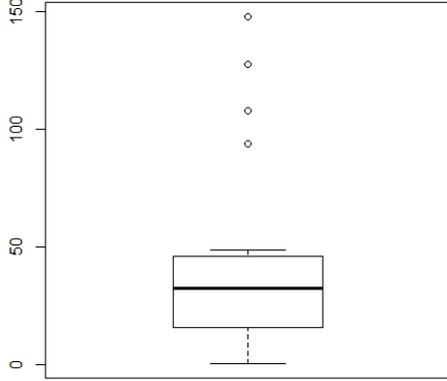
Generator Power Alarm	Oberes Quantil	46,18h	
	Mittelwert	43,53h	
	Median	32,34h	
	Unteres Quantil	15,68h	

Tabelle 29: TTF Generator Power LV

Generator Power Alarm	Oberes Quantil	46,18h	
	Mittelwert	43,52h	
	Median	32,34h	
	Unteres Quantil	15,68h	

Mit Hilfe dieser Informationen kann nun die Wahrscheinlichkeitstabelle für den RF Power Knoten erstellt werden. Der Vollständigkeit halber wird die Vorgehensweise anhand einiger Zeilen erläutert.

Zunächst wurden die möglichen Zustände der Knoten betrachtet. In Folge dessen wurden aus den TTF Zeiten (unteres Quantil, Mittelwert und oberes Quantil) jedes Knotens, der den Zustand True annimmt, der Mittelwert berechnet (siehe Abbildung 52).

Zustände Elternknoten						Mittelwerte der TTF Zeiten		
Biasvoltage Alarm	Biasvoltage LV	Clampvoltage Alarm	Clampvoltage LV	Generator Power Alarm	Generator Power LV	Uq	Mean	Oq
True	True	False	True	True	True	13,74	32,45	39,14
False	True	False	True	True	True	10,34	33,61	41,85
True	False	False	True	True	True	17,12	33,57	40,83
False	False	False	True	True	True	13,71	35,49	45,00

Abbildung 52: Ermittlung der TTF Mittelwerte

Danach erfolgte die Zuordnung der so ermittelte Werte, zu den im Knoten RF Power definierten Zuständen. Da die TTF Werte der einzelnen Knoten teilweise einer bestimmten Streuung unterliegen, wurde mittels einer Gewichtung versucht ein akkurateres Ergebnis zu ermöglichen. Hierfür wurden unteres und oberes Quantil mit 0,25 bewertet, der Mittelwert mit den restlich verbleibenden 0,50. Abbildung 53 visualisiert die eben erläuterten Schritte.

Uq	Mean	Oq	Bereich Uq				Bereich Mean				Bereich Oq				
			TTF<12	12<TTF<24	24<TTF<36	TTF>36	TTF<12	12<TTF<24	24<TTF<36	TTF>36	TTF<12	12<TTF<24	24<TTF<36	TTF>36	
13,74	32,45	39,14	0,00	0,25	0,00	0,00	0,00	0,00	0,50	0,00	0,00	0,00	0,00	0,00	0,25
10,34	33,61	41,85	0,25	0,00	0,00	0,00	0,00	0,00	0,50	0,00	0,00	0,00	0,00	0,00	0,25
17,12	33,57	40,83	0,00	0,25	0,00	0,00	0,00	0,00	0,50	0,00	0,00	0,00	0,00	0,00	0,25
13,71	35,49	45,00	0,00	0,25	0,00	0,00	0,00	0,00	0,50	0,00	0,00	0,00	0,00	0,00	0,25

Abbildung 53: Gewichtung der TTF Zeiten

In einem letzten Schritt wurden die jeweiligen Spalten der Bereiche addiert, um in weiterer Folge die Zustände des Knotens RF Modul zu bestimmen. Die Ergebnisse dieses finalen Schrittes können aus Abbildung 54 entnommen werden.

Biasvoltage Alarm	Biasvoltage LV	Clampvoltage Alarm	Clampvoltage LV	Generator Power Alarm	Generator Power LV	TTF<12	12<TTF<24	24<TTF<36	TTF>36
True	True	True	True	True	True	0,00	0,25	0,50	0,25
False	True	True	True	True	True	0,25	0,00	0,50	0,25
True	False	True	True	True	True	0,00	0,25	0,50	0,25
False	False	True	True	True	True	0,25	0,00	0,50	0,25
True	True	False	True	True	True	0,00	0,25	0,50	0,25
False	True	False	True	True	True	0,25	0,00	0,50	0,25
True	False	False	True	True	True	0,00	0,25	0,50	0,25
False	False	False	True	True	True	0,00	0,25	0,50	0,25
True	True	True	False	True	True	0,25	0,00	0,50	0,25
False	True	True	False	True	True	0,25	0,00	0,50	0,25
True	False	True	False	True	True	0,00	0,25	0,50	0,25
False	False	True	False	True	True	0,25	0,00	0,00	0,75
True	True	False	False	True	True	0,00	0,25	0,50	0,25
False	True	False	False	True	True	0,25	0,00	0,00	0,75
True	False	False	False	True	True	0,00	0,25	0,00	0,75
False	False	False	False	True	True	0,00	0,25	0,00	0,75
True	True	True	True	False	True	0,00	0,25	0,50	0,25
False	True	True	True	False	True	0,25	0,00	0,50	0,25
True	False	True	True	False	True	0,00	0,25	0,50	0,25
False	False	True	True	False	True	0,25	0,00	0,50	0,25
True	True	False	True	False	True	0,00	0,25	0,75	0,00
False	True	False	True	False	True	0,25	0,00	0,50	0,25
True	False	False	True	False	True	0,00	0,25	0,50	0,25
False	True	True	False	False	True	0,25	0,00	0,50	0,25
True	False	True	False	False	True	0,00	0,25	0,50	0,25
False	False	True	False	False	True	0,25	0,00	0,00	0,75
True	True	False	False	False	True	0,00	0,25	0,75	0,00
False	True	False	False	False	True	0,25	0,00	0,50	0,25
True	False	False	False	False	True	0,00	0,25	0,50	0,25
False	False	False	False	False	True	0,00	0,25	0,00	0,75
True	True	True	True	True	False	0,25	0,00	0,50	0,25
False	True	True	True	True	False	0,00	0,25	0,50	0,25
True	False	True	True	True	False	0,25	0,00	0,50	0,25
False	True	False	True	True	False	0,00	0,25	0,50	0,25
True	True	False	True	True	False	0,25	0,00	0,50	0,25
False	True	False	True	True	False	0,00	0,25	0,50	0,25
True	False	False	True	True	False	0,00	0,25	0,50	0,25
False	True	True	False	True	False	0,25	0,00	0,50	0,25
True	False	True	False	True	False	0,00	0,25	0,50	0,25
False	False	True	False	True	False	0,25	0,00	0,50	0,25
True	True	False	False	True	False	0,00	0,25	0,50	0,25
False	True	False	False	True	False	0,25	0,00	0,50	0,25
True	True	False	False	True	False	0,00	0,25	0,50	0,25
False	True	False	False	True	False	0,25	0,00	0,50	0,25
True	False	True	False	True	False	0,00	0,25	0,50	0,25
False	False	True	False	True	False	0,25	0,00	0,50	0,25
True	True	True	True	False	False	0,25	0,00	0,50	0,25
False	True	True	True	False	False	0,25	0,00	0,50	0,25
True	False	True	True	False	False	0,00	0,25	0,50	0,25
False	True	False	True	False	False	0,25	0,00	0,75	0,00
True	True	True	False	False	False	0,25	0,00	0,50	0,25
False	True	True	False	False	False	0,00	0,25	0,75	0,00
True	True	True	True	False	False	0,25	0,00	0,50	0,25
False	True	True	True	False	False	0,25	0,00	0,50	0,25
True	False	False	True	False	False	0,00	0,25	0,75	0,00
False	True	False	False	False	False	0,25	0,00	0,75	0,00
True	False	False	False	False	False	0,00	0,00	1,00	0,00
False	False	False	False	False	False	0,00	0,00	0,00	1,00

Abbildung 54: Wahrscheinlichkeitstabelle für Knoten RF Power

5.4.4 Umsetzung des statischen Netzwerkes in R

Die Umsetzung des Bayes'schen Netzwerkes wurde, wie bereits erwähnt mit der Software R realisiert. Hierfür wurde ein spezielles Paket namens „bnlearn“ verwendet, welches von Marco Scutari entwickelt wurde. Dieses erlaubt sowohl das automatische als auch manuelle Erstellen von Bayes'schen Netzwerken¹⁶³.

Im nachfolgenden soll die Umsetzung anhand einiger Code-Fragmente gezeigt werden (der gesamte Code befindet sich im Anhang).

5.4.4.1 Anlegen eines neuen Netzwerkes

Wie in Abbildung 55 gezeigt, wurde zunächst das Netzwerk als neues Netzwerk angelegt. Die Knoten werden dafür in eckigen Klammern gesetzt, senkrechte Striche definieren die Abhängigkeit des erstellten Knotens zu den Elternknoten.

```
#weiterer versuch zur erstellung des bayesian network
library(bnlearn)
##
## Attaching package: 'bnlearn'
## The following object is masked from 'package:stats':
##   sigma
library(readxl)

chuck_power_unit_2 <- model2network("[RFH][VTT][Biasvoltage_Alarm|RFH:VTT][Biasvoltage_LV|RFH:VTT][Clampvoltage_LV|RFH:VTT][Clampvoltage_Alarm|RFH:VTT][RF_POWER_LV|RFH:VTT][RF_POWER_Alarm|RFH:VTT][POWER_UNIT|Biasvoltage_LV:Biasvoltage_Alarm:Clampvoltage_LV:Clampvoltage_Alarm:RF_POWER_LV:RF_POWER_Alarm]")

graphviz.plot(chuck_power_unit_2, layout = "dot")
```

Abbildung 55: Erstellung des statischen Netzes in R

Als Beispiel:

Die Codezeile [Biasvoltage_Alarm | RFH:VTT] erstellt einen Knoten und stellt eine Verbindung zu den zugehörigen Elternknoten her (siehe Abbildung 56: Erklärung der Code Fragmente bei der Erstellung des Netzwerkes Abbildung 56).

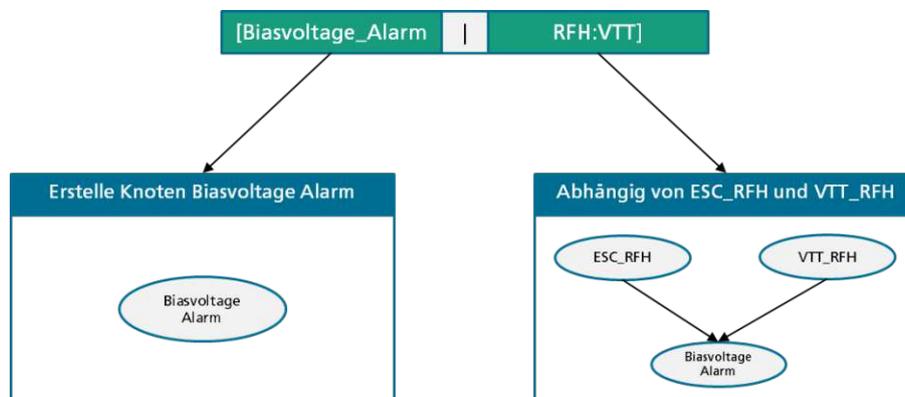


Abbildung 56: Erklärung der Code Fragmente bei der Erstellung des Netzwerkes

¹⁶³ (Nagarajan et al., 2013)

In einem nächsten Schritt mussten die Zustände definiert werden, die die Knoten einnehmen können (siehe Abbildung 57).

```
vtt=c("<100h", "100 < x < 200", ">300")
abnutzung_esc=c("<1000h", "1000 < x < 2500h ", ">2500h")
auftreten = c("TRUE", "FALSE")
TTF_RANGE= c("TTF<12", "12<TTF<24", "24<TTF<36", "TTF>36")
```

Abbildung 57: Definieren der Knotenzustände

5.4.4.2 Zuordnung der bedingten Wahrscheinlichkeitstabellen

Um die bereits in Kapitel 5.3.3 präsentierten Wahrscheinlichkeiten den jeweiligen Knoten zuzuordnen müssen Arrays erstellt werden, wobei die Anzahl der Dimensionen vom Zustand des Knotens und der möglichen Zustände der Elternknoten abhängt. Am Beispiel des Knoten RF Power (im Code unter dem Namen POWER_UNIT) soll dieses Vorgehen gezeigt werden.

Als erstes musste die bereits in Kapitel 5.4.3.4 vorgestellte, in Excel erstellte Wahrscheinlichkeitstabelle eingelesen werden. Danach wurde ein Array gebildet, welches sieben Dimensionen unterschiedlicher Länge beinhaltet.

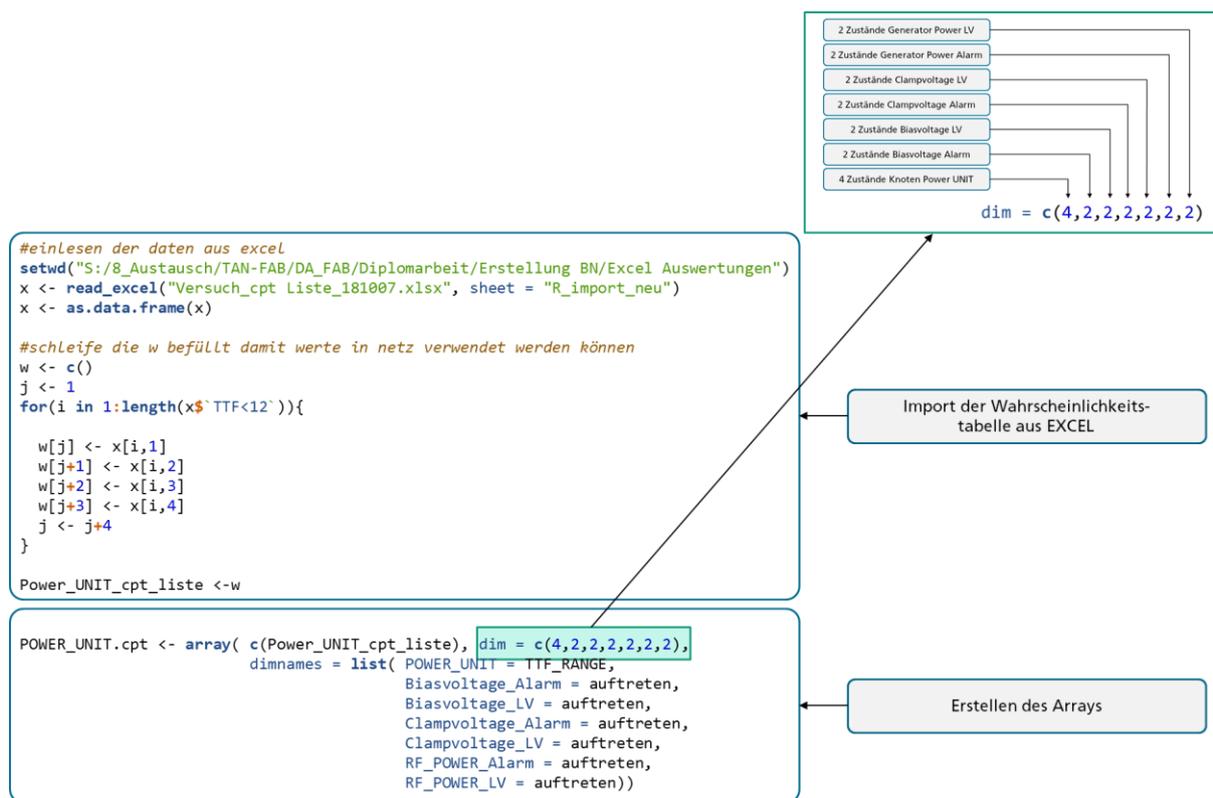


Abbildung 58: Erstellen der Wahrscheinlichkeitstabelle für den Knoten Power Unit

5.4.4.3 Fertigstellen des Netzes

Nach der erfolgreichen Zuordnung der bedingten Wahrscheinlichkeiten zu den Knoten muss in einem letzten Schritt das Modell „gefittet“ werden. Dies bedeutet, dass die

Wahrscheinlichkeitstabellen dem im ersten Schritt erstellten Netz zugewiesen werden (siehe Abbildung 59).

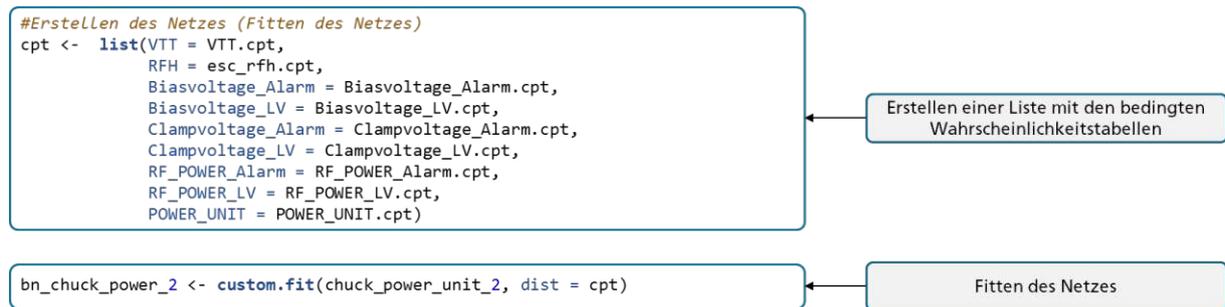


Abbildung 59: Fitten des Netzes

5.4.4.4 Anfragen stellen

Nachdem das Netz gefittet wurde, können Anfragen gestellt werden. Mittels dieser Anfragen können die Zustände des Submoduls RF Power unter bestimmten Bedingungen ermittelt werden. Die Vorgehensweise hierfür wird im Kapitel 5.6 Validierung gezeigt.

5.5 Dynamisches Bayes'sches Netz

Bei dem im Kapitel 5.4 präsentierten Modell zur Ausfallprognose handelt es sich um ein statisches Bayes'sches Netz, in dem der zeitliche Verlauf nicht berücksichtigt wird. Gerade im Bereich der Fertigung und insbesondere im Bereich der Halbleiterätzung verändern sich die Umgebungsbedingungen (z.B. Verschleiß der Maschine, Umwelteinflüsse etc.) kontiuierlich, woraus auch eine Änderung der Wahrscheinlichkeiten für Fehlerzustände resultiert. Daraus ergibt sich, dass die zeitliche Veränderung in Betracht gezogen werden muss, um eine dynamische Entscheidungsfindung bezüglich des Vorgehens (veranlassen von Instandhaltungsmaßnahmen, Personalkapazitätsplanung) zu ermöglichen¹⁶⁴.

Wie bereits auch im statischen Modell sind die Alarme und Limitverletzungen vom jeweiligen Zustand der Maschine abhängig, der in Form der Einsatzstunden des Moduls E-Chuck (ESC_RFH) und den Reinigungsintervallen (VTT_RFH) abgebildet wird. Im zukünftigen Zustand (t+1) hängen die Zustände der Knoten dieser beiden Kennzahlen einerseits vom Zustand im Zeitpunkt t und andererseits von der Anzahl der Wartungen bzw. Reinigungen ab, die zwischenzeitlich durchgeführt wurden.

Abbildung 60 visualisiert die zeitliche Veränderung und die damit verbundene Erweiterung der bedingten Wahrscheinlichkeitstabellen. Es gilt zu beachten, dass in dieser Abbildung nur einer der Kindknoten gezeigt wird (Biasvoltage LV). Dies wurde gemacht, um eine vereinfachte und übersichtliche Darstellung zu ermöglichen. Der

¹⁶⁴ (Bouaziz and Zamaï, 2012)

eingefügte Knoten X soll somit als Platzhalter für die restlichen nicht visualisierten Kindknoten fungieren.

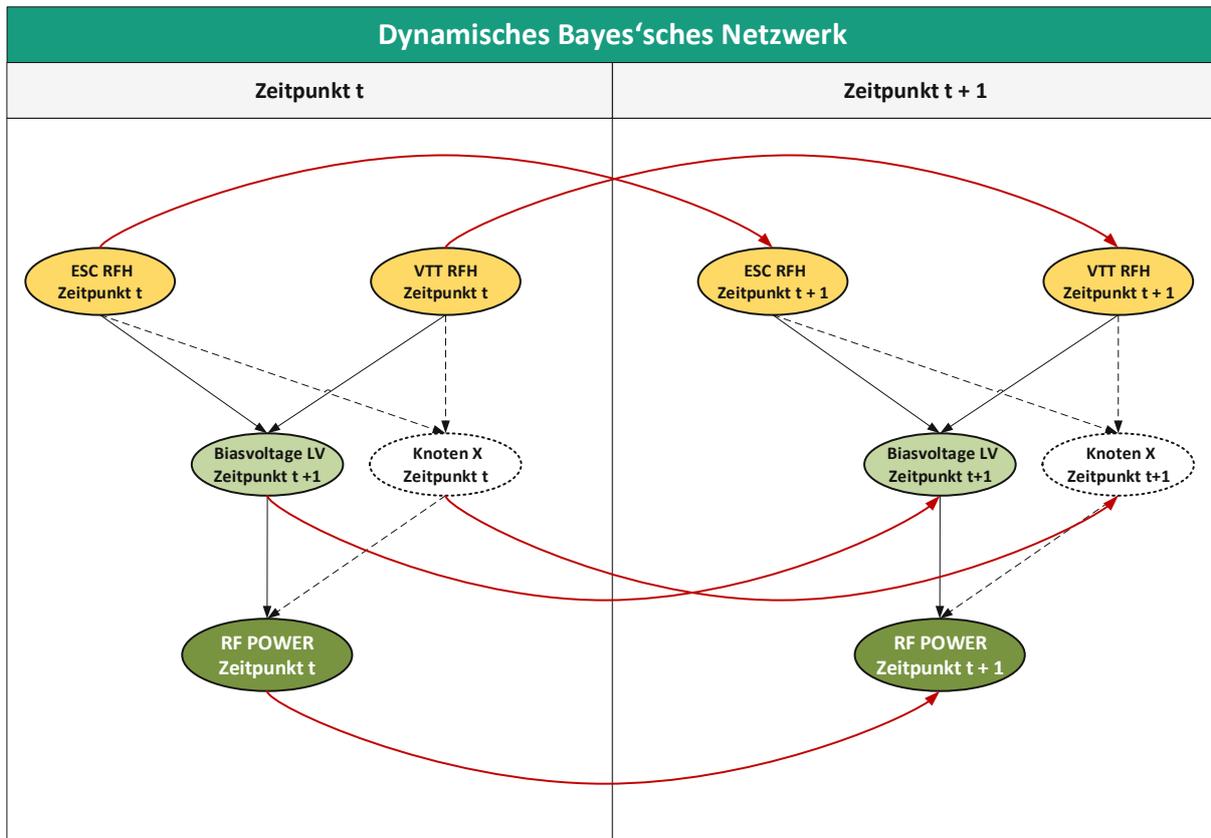


Abbildung 60: Dynamisches Bayes'sches Netz zur Ausfallprognose

Für das in dieser Arbeit präsentierte Modell soll der t+1 Zustand den 14.12.2017 darstellen. Dies liegt einerseits daran, dass dieser Punkt genau einen Monat hinter dem letzten im Trainingsdatenset berücksichtigten Datenpunkt liegt. Andererseits markiert dieser Punkt in etwa die Mitte des zur Validierung des Modells vorgesehen Subsets der Gesamtdaten.

5.5.1 Erweiterung der Wahrscheinlichkeitstabellen

Um den Verschleiß, dem die Maschine im Laufe der Zeit unterliegt, im Modell abbilden zu können, müssen die Wahrscheinlichkeitstabellen der Knoten ESC_RFH und VTT_RFH angepasst werden.

5.5.1.1 Generierung der erweiterten Wahrscheinlichkeitstabellen für VTT_RFH

Um den Zustand der Maschine im t+1 Zeitpunkt feststellen zu können, wird wie folgt vorgegangen: Es wird ermittelt, mit welcher Wahrscheinlichkeit die Maschine im Zustand t+1 eine bestimmte Anzahl an Einsatzstunden erreicht hat, unter der Voraussetzung im Zustand t einem definierten Zustand unterlegen zu sein.

Beispiel: Mit welcher Wahrscheinlichkeit betragen die Einsatzstunden seit der letzten Komplettreinigung in Zeitpunkt $t+1 < 100h$ unter der Voraussetzung, dass in Zeitpunkt t ebenfalls der Zustand $< 100h$ zugetroffen hat. (Darstellung in Tabelle $< 100h / < 100h$).

Die Ausweitung der möglichen Zustände für einen Zeitpunkt $t+1$ wird in Abbildung 61 gezeigt.

Zustand		P
t	< 100h	33,33%
	100h < x < 300h	33,33%
	>300h	33,33%

Zustand		P
$t+1$	< 100h / < 100h	XX,XX%
	100h < x < 300h / < 100h	XX,XX%
	>300h / < 100h	XX,XX%
	100h / 100h < x < 300h	XX,XX%
	100h < x < 300h / 100h < x < 300h	XX,XX%
	>300h / 100h < x < 300h	XX,XX%
	< 100h / > 300h	XX,XX%
	100h < x < 300h / > 300h	XX,XX%
	> 300h / > 300h	XX,XX%

Abbildung 61: Erweiterung der Wahrscheinlichkeitstabelle für VTT_RFH

Um zu eruieren, welche Anzahl an Einsatzstunden im Zeitpunkt $t+1$ seit der letzten Reinigung vergangen sind, mussten zunächst die Reinigungsdaten analysiert werden, um Informationen bezüglich der üblichen Reinigungsintervalle zu generieren.

Es war festzustellen, dass die Reinigungen im Median nach 368h durchgeführt werden. Zusätzlich wurde auch das untere und obere Quantil betrachtet (siehe Abbildung 62).

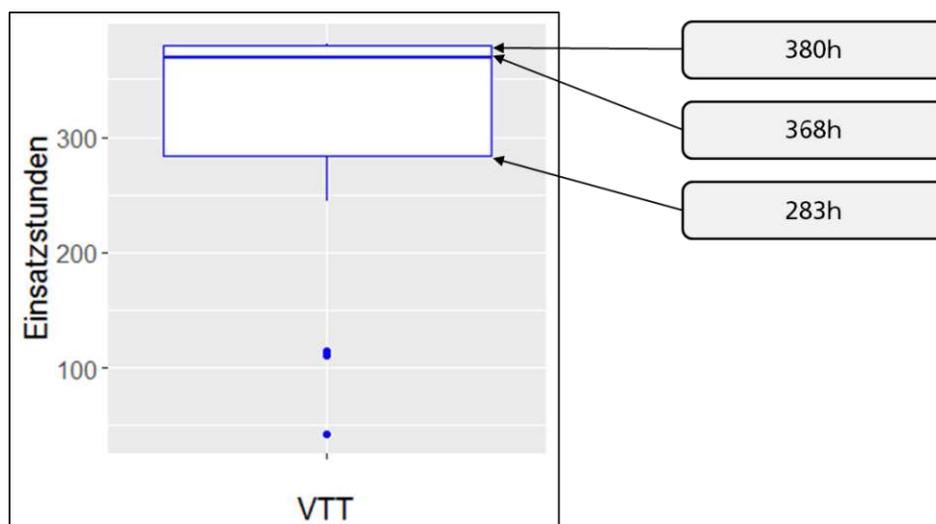


Abbildung 62: Boxplot Einsatzstunden VTT_RFH bis zur nächsten Reinigung

Wie bereits erwähnt soll der $t+1$ Zustand der 14.12.2017 darstellen und liegt somit 720h in der Zukunft. Um nun die Einsatzstunden des Moduls im $t+1$ Zeitpunkt feststellen zu können, wurden 3 Entscheidungsbäume angefertigt. Die Äste und Knoten bilden hierbei die Zeitpunkte, zu denen die Reinigungen durchgeführt wurden.

Um den Komplexitätsgrad des Baumes so gering wie möglich zu halten wurden die folgenden Annahmen getroffen:

- Mit 25% Wahrscheinlichkeit erfolgt die Reinigung nach 283 RFh (Unteres Quantil)
- Mit 50% Wahrscheinlichkeit erfolgt die Reinigung nach 368 RFh (Median)
- Mit 25% erfolgt die Reinigung nach 380 RFh (oberes Quantil bzw. Maximum)

5.5.1.1 Entscheidungsbaum für VTT_RFH < 100h in Zeitpunkt t:

Mit Hilfe dieses Entscheidungsbaumes werden die folgenden Zustände ermittelt:

- < 100h / <100h
- 100h < x < 300h / <100h
- >300h / <100h

Abbildung 63 zeigt den Aufbau dieses Entscheidungsbaumes für Zustand t < 100h

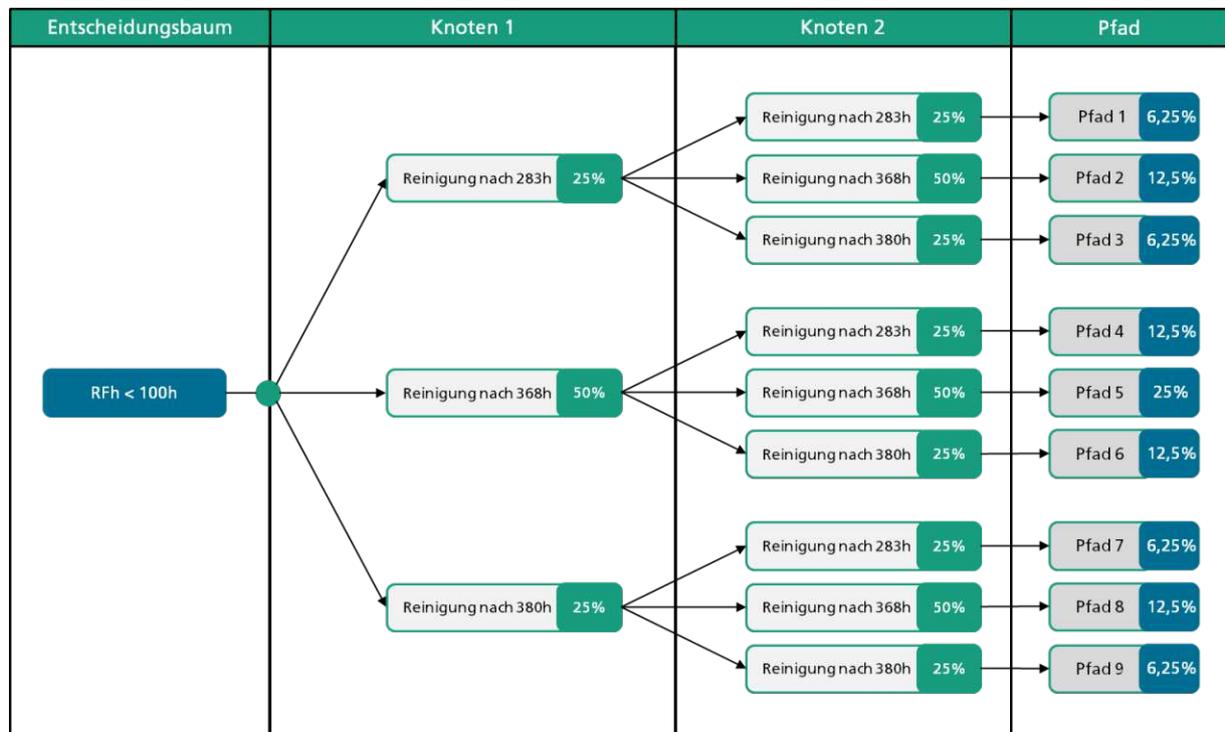


Abbildung 63: Entscheidungsbaum für Zustand t < 100h

Da im Zustand t theoretisch alle Zustände zwischen 0 und 99h möglich sind, muss dieser Baum 100-mal erstellt und durchgerechnet werden, um den Status zum Zeitpunkt t+1 zu ermitteln. Hierfür wurde ein R Script geschrieben, das jede dieser Möglichkeiten abbildet, gewichtet und in weiterer Folge das gewünschte Ergebnis liefert. In Tabelle 30 wird ein beispielhafter Auszug der 100 verschiedenen Entscheidungsbäume gezeigt.

Tabelle 30: Auszug Tabelle Entscheidungsbaum für VTT_RFH Zeitpunkt t <100h

Einsatzstunden Zeitpunkt t	Knoten bis nächster Wartung	Knoten 1	Wahrscheinlich- keit Knoten 1	Knoten 2	Wahrscheinlichkeit Knoten 2	Summe	Zeitpunkt t+1	Differenz	Zustand	Bereich	Gewichtung	Anteil
0	283	283	25%	283	25%	566	720	154	154	100 < x < 300	1%	0,000625
0	283	283	25%	368	50%	651	720	69	69	< 100h	1%	0,00125
0	283	283	25%	380	25%	663	720	57	57	<100h	1%	0,000625
0	368	368	50%	283	25%	651	720	69	69	<100h	1%	0,00125
0	368	368	50%	368	50%	736	720	-16	352	>300h	1%	0,0025
0	368	368	50%	380	25%	748	720	-28	352	>300h	1%	0,00125
0	380	380	25%	283	25%	663	720	57	57	<100h	1%	0,000625
0	380	380	25%	368	50%	748	720	-28	340	>300h	1%	0,00125
0	380	380	25%	380	25%	760	720	-40	340	>300h	1%	0,000625



50	233	283	25%	283	25%	516	720	204	204	100 < x < 300	1%	0,000625
50	233	283	25%	368	50%	601	720	119	119	100 < x < 300	1%	0,00125
50	233	283	25%	380	25%	613	720	107	107	100 < x < 300	1%	0,000625
50	318	368	50%	283	25%	601	720	119	119	100 < x < 300	1%	0,00125
50	318	368	50%	368	50%	686	720	34	34	<100h	1%	0,0025
50	318	368	50%	380	25%	698	720	22	22	<100h	1%	0,00125
50	330	380	25%	283	25%	613	720	107	107	100 < x < 300	1%	0,000625
50	330	380	25%	368	50%	698	720	22	22	<100h	1%	0,00125
50	330	380	25%	380	25%	710	720	10	10	<100h	1%	0,000625



Zum besseren Verständnis soll nun die Logik hinter den kombinierten Entscheidungsbäumen gezeigt werden:

- **Einsatzstunden Zeitpunkt t (Spalte 1):** In dieser Spalte werden die Einsatzstunden der Kammer zum Zeitpunkt t eingetragen.
- **Knoten bis zur nächsten Wartung (Spalte 2):** In dieser Spalte wird abhängig von Knoten 1 die Zeit bis zur nächsten Reinigung eingetragen.

Beispiel:

Einsatzstunden in Zeitpunkt $t = 50h$

Knoten 1 = 283h

Knoten bis zur nächsten Wartung = $283h - 50h = 233h$

- **Knoten 1 (Spalte 3):** Erste Reinigung (Wartung der Kammer) nach 283, 368 oder 380 Stunden
- **Wahrscheinlichkeit Knoten 1 (Spalte 4):** Wahrscheinlichkeit, dass jeweiliger Knoten gewählt wird (25%, 50%, 25% siehe Abbildung 63 Entscheidungsbaum)
- **Knoten 2 (Spalte 5):** Zweite Reinigung der Kammer nach 283, 368 oder 380 Stunden
- **Wahrscheinlichkeit Knoten 2 (Spalte 6):** Wahrscheinlichkeit, dass jeweiliger Pfad gewählt wird (25%, 50%, 25% siehe Abbildung 63 Entscheidungsbaum)

Beispiel:

Knoten 1: 283h (25% Wahrscheinlichkeit)

Knoten 2: 283h (25% Wahrscheinlichkeit)

Wahrscheinlichkeit für Pfad = $25\% * 25\% = 6,25\%$

- **Summe (Spalte 7):** Summe des Knoten bis zur nächsten Wartung und des Knoten 2.

Beispiel:

Knoten bis zur nächsten Wartung: 233h

Knoten 2: 283h

Summe = $233h + 283h = 516h$

- **Zeitpunkt t+1 (Spalte 8):** Anzahl der Stunden bis Zeitpunkt t+1. In diesem Anwendungsbeispiel 720h.
- **Differenz (Spalte 9):** Differenz zwischen Zeitpunkt t+1 und Summe.

Beispiel:

Zeitpunkt t+1: 720h

Summe: 516h

Differenz = $720 - 516h = 204h$

- **Zustand (Spalte 10):** Durch die unterschiedlichen Intervalle ist es möglich, dass die Summe über die 720h hinausgeht, woraus eine negative Differenz resultiert. Um den Zustand der Maschine bei 720h festzustellen, wurde diese Spalte eingeführt.

Beispiel:

Differenz: -16h

Summe: 736h
 Knoten 2: 380h
 Zustand = 380h – 16h = **364h**

- **Bereich (Spalte 11):** In der Spalte Bereich wird definiert, in welchem Zustand sich die Kammer, bezogen auf die Einsatzstunden, im Zeitpunkt t+1 befindet.

Beispiel:
 Zustand: 364h
 Bereich: **>300h**

- **Gewichtung (Spalte 12):** Nachdem der betrachtete Entscheidungsbaum 100-mal berücksichtigt werden muss (von Zustand t = 0h bis Zustand t = 99h), muss jedes Ergebnis gewichtet werden.
 Gewichtungsfaktor: **1%**

- **Anteil (Spalte 13):** In dieser Spalte wird die Wahrscheinlichkeit, dass ein Pfad gewählt wird, mit dem Gewichtungsfaktor belegt.

Beispiel:
 Wahrscheinlichkeit Pfad 1: 6,25%
 Gewichtung: 1%
 Anteil: 6,25% * 1% = **0,000625**

Nach der Erstellung des Entscheidungsbaumes in tabellarischer Form wurden die Zustände aufsummiert (siehe Abbildung 64).

```
#wahrscheinlichkeiten am 14.12 wenn mit zustand <100 begonnen wird
t_1_100_t_100 <-
sum(decision_tree_100_test$Anteil[which(decision_tree_100_test$Bereich
"<100h")])
t_1_100x300_t_100 <-
sum(decision_tree_100_test$Anteil[which(decision_tree_100_test$Bereich == "100< x <
300")])
t_1_300_t_100 <-
sum(decision_tree_100_test$Anteil[which(decision_tree_100_test$Bereich == ">300")])
```

Für den ersten Entscheidungsbaum liegt folgendes Ergebnis vor:

	Zustand	P
t + 1	< 100h / < 100h	55,88%
	100h < x < 300h / < 100h	30,25%
	>300h / < 100h	13,87%

Abbildung 64: Bereich der Einsatzstunden (VTT) der Maschine in Zeitpunkt t+1 unter der Voraussetzung, dass Einsatzstunden in Zeitpunkt t < 100h

5.5.1.1.2 Entscheidungsbaum für VTT_RFH $100h < x < 300h$ in Zeitpunkt t:

Mit Hilfe dieses Entscheidungsbaumes werden die folgenden Zustände ermittelt:

- $< 100h$ / $100h < x < 300h$
- $100h < x < 300h$ / $100h < x < 300h$
- $>300h$ / $100h < x < 300h$

Weil die Durchführung analog zu der in Kapitel 5.5.1.1.1 präsentierten erfolgt, werden in diesem Unterkapitel nur Ergebnisse bzw. ein Sonderfall bei der Erstellung kommuniziert.

Dieser Entscheidungsbaum muss aufgrund des Bereichs, der abgehandelt werden soll, 200-mal (von 100 bis 299 Einsatzstunden bei Zeitpunkt t) durchlaufen werden. Folglich ist zu berücksichtigen, dass ab dem Erreichen von 283 Einsatzstunden keine Möglichkeit mehr besteht, dass im ersten Knoten der Pfad „Wartung nach 283h“ gewählt wird. Deswegen muss der Entscheidungsbaum nochmals in 2 Teile untergliedert werden. Der erste bearbeitet den Bereich von 200h bis 282h, der zweite den von 283h bis 300h. Da ersterer derselben Logik wie dem in Kapitel 5.5.1.1.1 beschriebenen Baum folgt soll nur der zweite erklärt (siehe Abbildung 65) und im Anschluss nur Ergebnisse präsentiert werden.

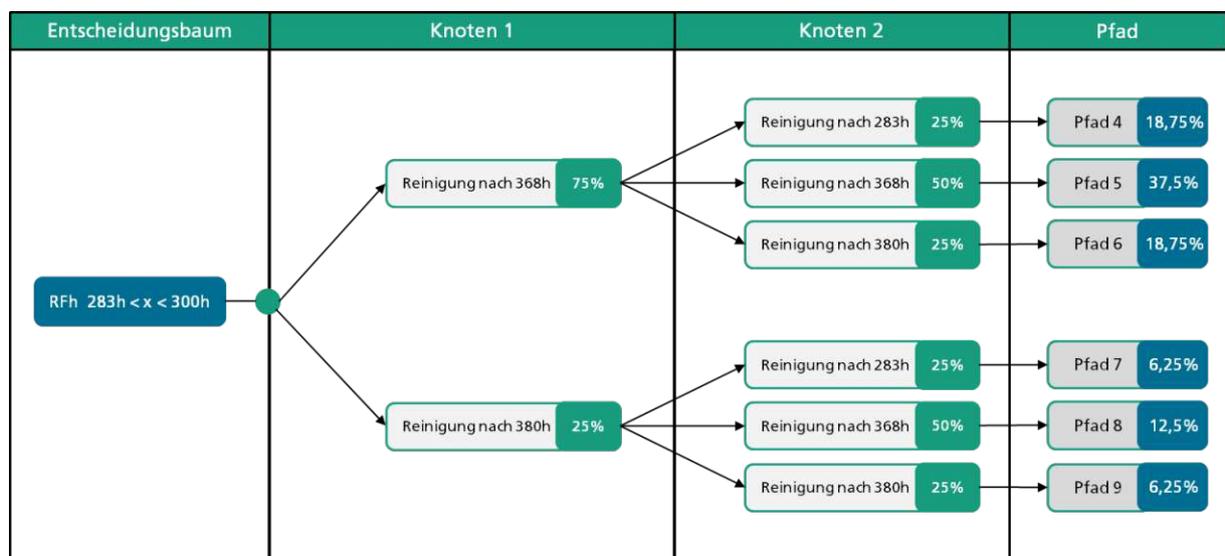


Abbildung 65: Zweiter Entscheidungsbaum für $100h < x < 300h$

Da im ersten Knoten nur zwischen den Zuständen Reinigung nach 368h und Reinigung nach 380h gewählt werden, ändern sich die Pfade und deren zugehörige Wahrscheinlichkeiten. Es wurde bestimmt, dass die Reinigung mit 75% nach 368h und mit 25% nach 380h stattfindet. Im darauffolgenden Knoten 2 sind wieder alle drei Pfade möglich.

Nach dem Kalkulieren beider Entscheidungsbäume müssen die Einzelergebnisse kombiniert werden. Die Ergebnisse des ersten Baumes wurden mit 91,5% (183 der

200 Möglichkeiten) gewichtet, der zweite mit den restlichen 8,5%. In weiterer Folge wurden die gewichteten Ergebnisse summiert (siehe Abbildung 66).

```
#für 100<x<300 wird nun gewichteter mittelwert errechnet (da zwei bäume)
#bis 283h mit 91,5% (183/200) gewichtet danach mit 8,5%

t_1_100_t_100x300_1 <- sum(decision_tree_100x300_1$Anteil[which(decision_tree_100x300_1$Bereich == "<100h")])*0.915
t_1_100x300_t_100x300_1 <- sum(decision_tree_100x300_1$Anteil[which(decision_tree_100x300_1$Bereich == "100< x < 300")])*0.915
t_1_300_t_100x300_1 <- sum(decision_tree_100x300_1$Anteil[which(decision_tree_100x300_1$Bereich == ">300")])*0.915

t_1_100_t_100x300_2 <- sum(decision_tree_100x300_2$Anteil[which(decision_tree_100x300_2$Bereich == "<100h")])*0.085
t_1_100x300_t_100x300_2 <- sum(decision_tree_100x300_2$Anteil[which(decision_tree_100x300_2$Bereich == "100< x < 300")])*0.085
t_1_300_t_100x300_2 <- sum(decision_tree_100x300_2$Anteil[which(decision_tree_100x300_2$Bereich == ">300")])*0.085

t_1_100_t_100x300 <- sum(t_1_100_t_100x300_1, t_1_100_t_100x300_2)
t_1_100x300_t_100x300 <- sum(t_1_100x300_t_100x300_1, t_1_100x300_t_100x300_2)
t_1_300_t_100x300 <- sum(t_1_300_t_100x300_1, t_1_300_t_100x300_2)
```

Abbildung 66: Ausschnitt des R Scripts zur Berechnung des zweiten Entscheidungsbaums

Für den zweiten Entscheidungsbaum liegt folgendes Ergebnis vor:

	Zustand	P
t + 1	100h / 100h < x < 300h	12,73%
	100h < x < 300h / 100h < x < 300h	77,00%
	>300h / 100h < x < 300h	10,27%

Abbildung 67: Bereich der Einsatzstunden (VTT) der Maschine in Zeitpunkt t+1 unter der Voraussetzung, dass Einsatzstunden in Zeitpunkt t $100h < x < 300h$

5.5.1.1.3 Entscheidungsbaum für VTT_RFH >300h in Zeitpunkt t:

Mit Hilfe dieses Entscheidungsbaumes werden die folgenden Zustände ermittelt:

- < 100h / >300h
- 100h < x < 300h / >300h
- >300h / >300h

Auch der Entscheidungsbaum 3 musste in zwei Teile untergliedert werden, da ab dem Zustand 368 Einsatzstunden zum Zeitpunkt t, in Knoten 1, nur noch der Pfad in Richtung „Reinigung nach 380h“ Sinn ergibt. Die Vorgehensweise bei dieser Untergliederung geschieht analog zu der in Kapitel 5.5.1.1.2 beschriebenen, weshalb in diesem Unterkapitel nur Ergebnisse präsentiert werden.

t + 1	Zustand	P
	< 100h / > 300h	35,48%
	100h < x < 300h / > 300h	25,41%
	> 300h / > 300h	39,11%

Abbildung 68: Bereich der Einsatzstunden (VTT) der Maschine in Zeitpunkt t+1 unter der Voraussetzung, dass Einsatzstunden in Zeitpunkt t >300h

5.5.1.1.4 Erweiterte Wahrscheinlichkeitstabelle für VTT

Um zur gesamten erweiterten Wahrscheinlichkeitstabelle kommen zu können müssen die Ergebnisse der drei einzelnen Bäume noch mit 33,33% (Auftrittswahrscheinlichkeit der Zustände in Zeitpunkt t) multipliziert werden. Die Ergebnisse können aus Abbildung 69 übernommen werden.

t	Zustand	P
	< 100h	33,33%
	100h < x < 300h	33,33%
	>300h	33,33%



t + 1	Zustand	P
	< 100h / < 100h	18,63%
	100h < x < 300h / < 100h	10,08%
	>300h / < 100h	4,63%
	100h / 100h < x < 300h	4,24%
	100h < x < 300h / 100h < x < 300h	25,67%
	>300h / 100h < x < 300h	3,42%
	< 100h / > 300h	11,83%
	100h < x < 300h / > 300h	8,46%
> 300h / > 300h	13,04%	

Abbildung 69: Bedingte Wahrscheinlichkeiten für Knoten VTT in Zeitpunkt t+1

5.5.1.2 Generierung der erweiterten Wahrscheinlichkeitstabellen für ESC_RFH

Um den Zustand des Moduls im t+1 Zeitpunkt feststellen zu können, wird wie folgt vorgegangen: Es wird ermittelt mit welcher Wahrscheinlichkeit das Modul ESC im Zustand t+1 eine bestimmte Anzahl an Einsatzstunden erreicht hat, unter der Voraussetzung im Zustand t einem definierten Zustand unterlegen zu sein.

Beispiel: Mit welcher Wahrscheinlichkeit betragen die Einsatzstunden in Zeitpunkt t+1 <1000h unter der Voraussetzung, dass in Zeitpunkt t ebenfalls der Zustand <1000h festzustellen war (Zustand <1000h / <1000h)?

Die Ausweitung der möglichen Zustände für den Zeitpunkt t+1 wird in Abbildung 70 gezeigt.

		Zustand	P
t	< 1000h		33,33%
	1000h < x < 2500h		33,33%
	>2500h		33,33%

→

		Zustand	P
t + 1	< 1000h / < 1000h		XX,XX%
	1000h < x < 2500h / < 1000h		XX,XX%
	>2500h / < 1000h		XX,XX%
	1000h / 1000h < x < 2500h		XX,XX%
	1000h < x < 2500h / 1000h < x < 2500h		XX,XX%
	>2500h / 1000h < x < 2500h		XX,XX%
	1000h / >2500h		XX,XX%
	1000h < x < 2500h / > 2500h		XX,XX%
	> 2500h / > 2500h		XX,XX%

Abbildung 70: Erweiterung der Wahrscheinlichkeitstabelle für ESC_RFH

Zur Füllung dieser erweiterten Tabelle mit Wahrscheinlichkeiten werden die Reinigungsdaten herangezogen. Aus diesen können Zeitpunkte und zugehörige Einsatzstunden entnommen werden.

Zunächst wurde, wie bereits bei dem Knoten VTT eruiert, in welchen Intervallen das besagte Modul getauscht wird (siehe Abbildung 71).

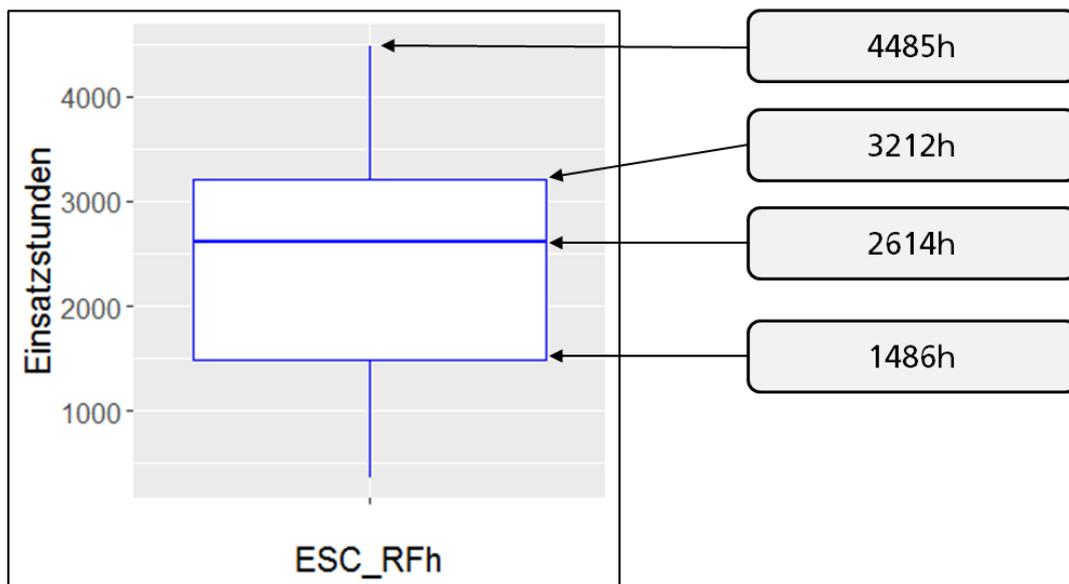


Abbildung 71: Boxplot Einsatzstunden ESC_RFH bis zum Austausch

Auch in diesem Fall wird der Zustand t+1 mit dem 14.12.2017 festgelegt, welcher 720h in der Zukunft liegt. Um die RFh des Moduls zum Zeitpunkt t+1 feststellen zu können, werden, wie auch schon anhand des Knotens VTT gezeigt, drei Entscheidungsbäume für die 3 zugehörigen Zustandsbereiche (<1000h, 1000h < x < 2500h und >2500h) entwickelt. Zur Komplexitätsreduktion werden die folgenden Annahmen getroffen:

- Mit 25% Wahrscheinlichkeit erfolgt die Reinigung nach 1486h RFh (Unteres Quantil)
- Mit 50% Wahrscheinlichkeit erfolgt die Reinigung nach 2614h RFh (Median)
- Mit 25% erfolgt die Reinigung nach 4485 RFh (max. Wert)

5.5.1.2.1 Entscheidungsbaum für ESC_RFH < 1000h in Zeitpunkt t:

Mit Hilfe dieses Entscheidungsbaumes werden die folgenden Zustände ermittelt:

- < 1000h / < 1000h
- 1000h < x < 2500h / 1000h
- >2500h / 1000h

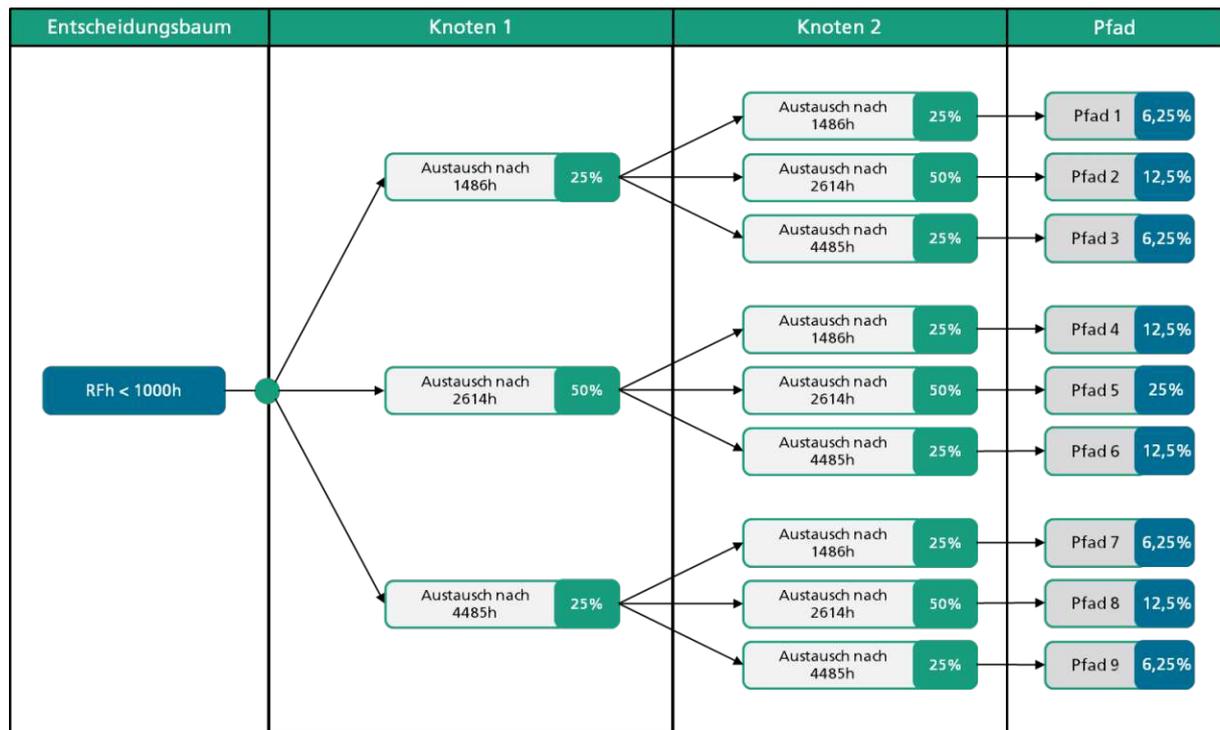


Abbildung 72: Entscheidungsbaum für Zustand t < 1000h

Da im Zustand t alle Zustände zwischen 0 und 999h möglich sind, muss dieser Baum 1000-mal erstellt und durchgerechnet werden um den Status zu Zeitpunkt t+1 zu ermitteln.

Auf die detaillierte Aufschlüsselung der Inhalte des Entscheidungsbaums und die zugehörige Darstellung der in R abgebildeten Tabelle wird verzichtet (ident zu bereits in Kapitel 5.5.1.1 Tabelle 30 gezeigtem Vorgehen). Die Ergebnisse können Abbildung 73 entnommen werden.

	Zustand	P
t + 1	< 1000h / < 1000h	33,93%
	1000h < x < 2500h / < 1000h	65,98%
	>2500h / < 1000h	0,09%

Abbildung 73: Bereich der Einsatzstunden (ESC) der Maschine in Zeitpunkt t+1 unter der Voraussetzung, dass Einsatzstunden in Zeitpunkt t < 1000h

5.5.1.2.2 Entscheidungsbaum für ESC_RFH $1000h < x < 2500h$ in Zeitpunkt t:

Mit Hilfe dieses Entscheidungsbaumes werden die folgenden Zustände ermittelt:

- $< 1000h / 1000h < x < 2500h$
- $1000h < x < 2500h / 1000h < x < 2500h$
- $>2500h / 1000h < x < 2500h$

Auch dieser Entscheidungsbaum muss nach der bereits gezeigten Methode geteilt, gewichtet und wieder zusammengefügt werden. Auch hier sollen nur die erzielten Ergebnisse gezeigt werden (siehe Abbildung 74).

	Zustand	P
t + 1	$< 1000h / 1000h < x < 2500h$	38,35%
	$1000h < x < 2500h / 1000h < x < 2500h$	43,9%
	$>2500h / 1000h < x < 2500h$	17,75%

Abbildung 74: Bereich der Einsatzstunden (ESC) der Maschine in Zeitpunkt t+1 unter der Voraussetzung, dass Einsatzstunden in Zeitpunkt t $1000h < x < 2500h$

5.5.1.2.3 Entscheidungsbaum für ESC_RFH $>2500h$ in Zeitpunkt t:

Mit Hilfe dieses Entscheidungsbaumes werden die folgenden Zustände ermittelt:

- $< 1000h / >2500h$
- $1000h < x < 2500h / >2500h$
- $>2500h / >2500h$

Aufgrund des analogen Vorgehens werden wie bereits in vorherigen Kapiteln nur die Ergebnisse gezeigt (siehe Abbildung 75):

	Zustand	P
t + 1	$< 1000h / >2500h$	40,56%
	$1000h < x < 2500h / >2500h$	0,00%
	$>2500h / >2500h$	59,44%

Abbildung 75: Bereich der Einsatzstunden (ESC) der Maschine in Zeitpunkt t+1 unter der Voraussetzung, dass Einsatzstunden in Zeitpunkt t $>2500h$

5.5.1.2.4 Erweiterte Wahrscheinlichkeitstabelle für ESC_RFH

Für die Ermittlung der gesamten erweiterten Wahrscheinlichkeitstabelle müssen die Ergebnisse der drei zusammengeführten Bäume wieder mit 33,33% (Auftrittswahrscheinlichkeiten der Zustände in Zeitpunkt t) multipliziert werden. Die zugehörigen Ergebnisse werden in Abbildung 76 visualisiert.

Zustand	P
< 1000h	33,33%
1000h < x < 2500h	33,33%
>2500h	33,33%

→

Zustand	P
< 1000h / < 1000h	11,31%
1000h < x < 2500h / < 1000h	21,99%
>2500h / < 1000h	0,03%
1000h / 1000h < x < 2500h	12,78%
1000h < x < 2500h / 1000h < x < 2500h	14,63%
>2500h / 1000h < x < 2500h	5,91%
1000h / >2500h	13,51%
1000h < x < 2500h / > 2500h	0,00%
> 2500h / > 2500h	19,81%

Abbildung 76: Bedingte Wahrscheinlichkeiten für Knoten ESC_RFH in Zeitpunkt t+1

5.6 Validierung des Modells

Um zu überprüfen ob das präsentierte Modell ordnungsgemäß funktioniert, muss es validiert werden. Die Validierung soll sowohl am statischen als auch dem dynamischen Modell erfolgen. Überprüft werden die erzielten Ergebnisse mittels des Validierungsdatensets.

5.6.1 Ergebnisse des statischen Netzwerkes

Um die Wahrscheinlichkeit zu den definierten Ausfallbereichen zu ermitteln, müssen die in Abbildung 77 dargestellten Anfragen gestellt werden. Dabei wird ersichtlich, dass das Modul bzw. das Submodul mit einer Wahrscheinlichkeit von 61% erst in über 36h ausfallen wird.

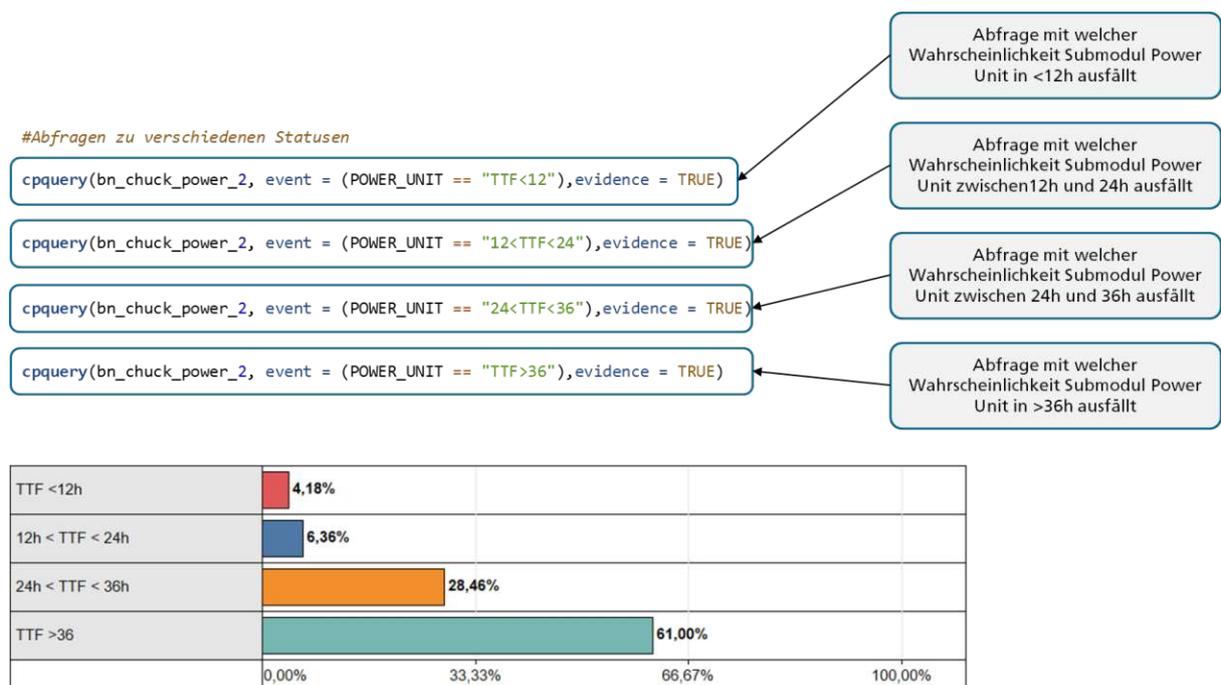


Abbildung 77: Abfragen zur Ergebnisfindung und Ergebnisse des statischen Netz

Da der Zeitpunkt, zu dem die Prognose durchgeführt wurde, der 14.11.2017 ist, wurde zur Kontrolle ein Abgleich mit diesen Daten durchgeführt. Hierbei wurde eruiert, dass der nächste Breakdown der Maschine am 17.11.2017 erfolgte, wodurch die Richtigkeit des prognostizierten Zeitraums bestätigt wird.

5.6.2 Ergebnisse des dynamischen Netzwerkes

In diesem Unterkapitel soll zunächst das Ergebnis des Use-Cases präsentiert werden. Danach erfolgt eine Vorgehensempfehlung unter erweiterten Annahmen.

▪ Ergebnisse des Use Cases:

Bei der Validierung des dynamischen Netzes wurden die erweiterten Wahrscheinlichkeitstabellen in das Programm eingefügt und die Berechnung erneut nach dem bereits in Kapitel 5.6.1 gezeigten Schema durchgeführt. Dabei ergab sich folgendes Bild (siehe Abbildung 78):

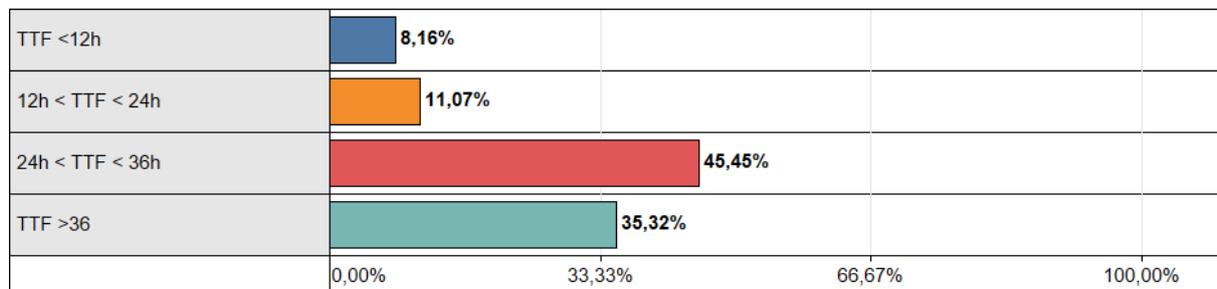


Abbildung 78: Ergebnisse des dynamischen Netzes

Das Ergebnis zeigt, dass eine Prognose, die für den 14.12.2017 durchgeführt wird ergibt, dass die Maschine mit 45,45% Wahrscheinlichkeit zwischen den nächsten 24 und 36h ungeplant ausfallen wird. Unter dem Hinzuziehen des Validierungsdatensets wurde bestätigt, dass der Ausfall am 15.12.2017 gegen 08:00 stattfand. Somit war der Zeitpunkt mit 32h innerhalb des zuvor prognostizierten Zeitintervalls. Um sicherzugehen, ob es sich bei dem Ausfall auch um einen E-Chuck bezogenen handelte, wurde ein Abgleich mit dem Instandhaltungsmaßnahmenbuch vollzogen, der den gezeigten Anwendungsfall bestätigt.

▪ Ergebnisse unter erweiterten Annahmen:

Im Zuge dieser Arbeit erfolgte die vollständige Erstellung eines Bayes'schen Netzwerkes zur Ausfallprognose nur für das Submodul des Moduls E-Chuck. Das folgende Gedankenexperiment soll zeigen, wie im Falle einer vollständigen Erstellung und Parametrisierung aller Submodule, die Ausfallsprognose durchgeführt werden würde.

Annahme:

Tabelle 31: Ausfallwahrscheinlichkeiten aller Submodule (Annahme)

Submodul	TTF < 12h	12h < TTF < 24h	24h < TTF < 36h	TTF > 36h
RF Power	3,5%	15,5%	60%	21%
Gas Delivery	9,5%	34,5%	30%	26%
Coolant	12%	18%	20%	50%

Da das Modul Gas Delivery mit 34,5% durch das Zeitintervall $12h < TTF < 24h$ am kritischsten scheint, würde die Instandhaltungsplanung auf diesen Zeitraum angepasst werden. Zusätzlich wäre es natürlich empfehlenswert, im Zuge der Wartung auch den Bereich RF Power zu überprüfen, da dieses den nächsten kritischen Zustand darstellt.

6 Ergebnisdiskussion

6.1 Ergebnisse durchgeführter Analysen

Im Zuge dieser Arbeit wurde ein Modell vorgestellt, das im Stande ist, den Ausfall von spezifischem Equipment, das zur Halbleiterfertigung benötigt wird, zu prognostizieren. Hierfür wurden Daten analysiert, Zusammenhänge zwischen Ausfall beeinflussender Parameter gebildet und ein Vorschlag zur Modellierung eines Bayes'schen Netzwerks zur Ausfallsprognose präsentiert.

- **Erstellung:**

Bei der Erstellung des Netzwerkes mussten mehrere Versuche gestartet werden. Zunächst sollte die Modellierung auf Basis von Modulen und zugehörigen Komponenten geschehen. Aus dem Verschleiß- und Ausfallverhalten besagter Bauteile wurde versucht, eine Verbindung zum Ausfallverhalten der Maschine herzustellen. Dieser erste, genauere und meist auch in der Literatur empfohlene Ansatz musste im Zuge der Datenanalyse aufgrund der mangelnden und unvollständigen Dokumentation von historischen Instandhaltungsmaßnahmen, verworfen werden. In weiterer Folge wurde ein zweites Netz entworfen, welches aufgrund von auf Prozesskennzahlen basierenden Limitverletzungen und Alarmen, zukünftige Ausfälle prädiktieren soll. Dieses Modell lag zunächst in statischer Form vor und wurde mittels der Einführung von Entscheidungsbäumen zu einem dynamischen Netz erweitert.

- **Ergebnisse:**

Wie in Kapitel 5.6 beschrieben, kann das vorgeschlagene Modell als Unterstützung agieren, Ausfälle zu prognostizieren. Hierbei ist jedoch anzumerken, dass diese Prognose sich nur auf ein im Vorfeld definiertes Zeitintervall bezieht. Dies liegt vor allem an der Natur des Bayes'schen Netzwerkes, die eine Diskretisierung der Zustände der einzelnen Knoten verlangt. Vor allem das Ergebnis des statischen Netzwerkes hebt diese Schwäche hervor. Es konnte richtigerweise bestimmt werden, dass die Maschine erst in über 36h ausfallen wird. Über den darauffolgenden Bereich kann jedoch keine Auskunft gegeben werden.

6.2 Ergebnisse in Bezug auf die Instandhaltungsplanung

Durch die Schaffung eines „Forecasts“, der Aussagen bezüglich des Zeitintervalls, in dem ein Maschinenausfall voraussichtlich erfolgen wird, ermöglicht, können Vorteile für die Instandhaltungsplanung gewonnen werden. Im Kapitel 5.6 wird beispielsweise prognostiziert, dass das betrachtete Submodul RF Power des E-Chuck, in einem Zeitintervall zwischen 24 und 36 Stunden ausfallen wird. Demzufolge kann die

Schichtplanung angepasst werden, sodass Instandhaltungspersonal vor Ort ist, welches die benötigten Kompetenzen besitzt, diesen Bereich der Maschine zu reparieren bzw. zu warten. Zusätzlich kann, sofern eine Gesamtreinigung der Kammer in einem ähnlichen Zeitraum anstehen würde, eine Kombination dieser beiden Maßnahmen in Betracht gezogen werden. Durch die Bündelung der prognostizierten und geplanten Instandhaltungsmaßnahme kann eine zusätzliche Produktivitätssteigerung der Anlage erzielt werden.

6.3 Vergleich der erzielten Ergebnisse mit anderen Vorgehensmethodiken

6.3.1 Vergleich mit Ergebnissen aus Literatur

In der Literatur gibt es bereits einige Beispiele zur Diagnose von Maschinen in der Halbleiterfertigung mittels Bayes'schen Netzen (siehe Kapitel 4.3). Diese fokussieren jedoch hauptsächlich auf die Prognose des „Equipment Health Factors“ (EHF)¹⁶⁵ oder auf die Fehlerdiagnose zur Unterstützung bei der Ausfallbehebung¹⁶⁶. Modelle zur Prognose des Ausfallsverhaltens in definierten Zeitintervallen wurden im Zuge der Literaturrecherche nicht gefunden.

6.3.2 Vergleich mit Ergebnissen eines Forschungsprojektes mit selbigem Use-Case

Im Zuge eines parallel zu dieser Arbeit durchgeführten Forschungsprojektes, welches ebenfalls die Prognostizierung von Ausfällen zum Ziel hatte, konnten im Vergleich zum hier vorgestellten Modell genauere Ergebnisse erzielt werden. Dies liegt daran, dass bei der Erstellung des Modells unterschiedlichste Machine-Learning Verfahren angewendet und teilweise auch kombiniert wurden. Bei der Erstellung von Bayes'schen Netzwerken muss vor allem immer auf die Simplifizierung des Netzes geachtet werden. Folglich wurden viele, eventuell wichtige Einflussparameter vernachlässigt oder blieben sogar unberücksichtigt, wodurch die Prognosegenauigkeit des Modells beeinträchtigt wird. Jedoch ist zu berücksichtigen, dass auch bei der Anwendung anderer Machine-Learning Techniken die Erstellung von Modellen mit hoher Prognosegenauigkeit und Präzision eine große Herausforderung darstellt. So wurde zum Beispiel ein Ansatz zur Lösung der Problemstellung mittels eines speziellen Regressions-Algorithmus aufgrund schlechter Ergebnisse verworfen.

¹⁶⁵ Bouaziz and Zamaï, 2012

¹⁶⁶ Nawaz et al., 2014

6.4 SWOT Analyse

Im nachfolgenden wird eine SWOT Analyse durchgeführt, um den entwickelten Ansatz zur Instandhaltung bewerten zu können.

Tabelle 32: SWOT Analyse¹⁶⁷

SWOT Analyse		Interne Analyse	
		Stärken (Strengths)	Schwächen (Weaknesses)
Externe Analyse	Chancen (Opportunities)	Die Modellierung ermöglicht die vereinfachte Darstellung von Abhängigkeiten innerhalb komplexer Systeme.	Die Erhöhung der Datenqualität und die damit verbundene Bestimmung komponentenspezifischer Einflüsse auf Ausfälle können verwendet werden, um das Modell anzupassen und um die Fehlerdiagnose zu ermöglichen.
		Die Methodik Bayes'scher Netze ist für den Bereich der Instandhaltung geeignet.	Das erstellte Modell ist nur für ein Submodul der Maschine erstellt worden. Eine Erweiterung auf alle Module würde folglich noch einiges an Zeit in Anspruch nehmen
			Eine dynamische Erweiterung des Modells ist unbedingt erforderlich, um auf sich ändernde Einflüsse reagieren zu können.
	Gefahren (Threats)	Ungenauigkeiten können durch eine Erhöhung der Datenqualität verringert werden.	Annahmen und Vereinfachungen können das Prognoseergebnis beeinflussen.
		Durch die Methodik werden Regeln definiert, die gleichzeitig als Beschränkungen gesehen werden können.	Subjektive Einflüsse bei der manuellen Erstellung des Netzwerkes können zur Verfälschung führen.
			Komponentenspezifische Einflüsse (z.B. Verschleiß) konnten nicht ausreichend berücksichtigt werden.

¹⁶⁷ vgl.Homburg, 2017

6.5 Ergebnisse in Bezug auf die Forschungsfragen

Forschungsfrage: Können mit Hilfe eines geeignet parametrisierten und erprobten Bayes'schen Netzwerkes, ungeplante Ausfälle, des für die Fertigung von Halbleiterprodukten benötigten Equipments prognostiziert werden, sodass ungeplanten Stillständen vorgebeugt werden kann und folglich eine funktionierende Wertschöpfung gewährleistet wird?

Im Zuge dieser Arbeit konnte festgestellt, dass es möglich ist, ein Bayes'sches Netzwerk zu entwerfen, das zur Ausfallprognose verwendet werden kann. Folglich können Bayes'sche Netze im Zuge von prädiktiver Instandhaltung eingesetzt werden, bergen aber aufgrund des großen Aufwands bei der Erstellung, der häufig notwendigen Vereinfachung und der subjektiven Einflüsse einige und teilweise gravierende Nachteile zu anderen Vorgehensmethoden.

Forschungsfrage: Wie kann die Effizienz des entwickelten Ansatzes im Vergleich zu anderen Methoden zur Ausfallprognose beurteilt werden?

Im Zuge dieser Diplomarbeit wurde sehr viel Zeit und Aufwand in die Entwicklung des Modells gesteckt. Oftmals mussten auch bereits entwickelte Ansätze aufgrund von fehlenden/unvollständigen Daten oder auch fehlendem Expertenwissen bezüglich des Produktionsprozesses wieder verworfen werden. Folglich wurde das Modell aufgrund der bereits angesprochenen Vereinfachungen und der Notwendigkeit zeitweise Annahmen treffen zu müssen, stark simplifiziert. Demnach muss die kritische Frage gestellt werden, ob dieser Aufwand, in Bezug auf die Genauigkeit des Ergebnisses (Prognose von Zeitintervallen), gerechtfertigt ist. Im Vergleich zu dem im Forschungsprojekt erstellten Ansatz, der eine viel größere Menge an Parametern berücksichtigt, ist dieser Ansatz folglich von einer geringeren Effizienz bezüglich Input und Output begleitet. Es gilt jedoch zu berücksichtigen, dass besagtes Forschungsprojekt in Kooperation mehrerer Forschungseinrichtungen und dem Einsatz wesentlich größerer personaler Ressourcen erfolgte.

Forschungsfrage: Lassen sich allgemeine Aussagen über Eignung/Nicht-Eignung der verglichenen Methoden für die Halbleiterindustrie ableiten?

Wie bereits erwähnt leidet die Prognosegenauigkeit des Modells unter den getroffenen Annahmen und Vereinfachungen, die im Zuge der Erstellung von Bayes'schen Netzen notwendig werden (Simplifizierung des Systems, definierte Zustände je Knoten). Somit kann die Behauptung aufgestellt werden, dass sich die Anwendung von Bayes'schen Netzwerken in der Instandhaltung, wie in der Literatur empfohlen, besonders im Bereich der Fehlerdiagnose eignet. Prognostizierung von Zeiten, innerhalb derer die Maschine ausfällt, wäre, wie bereits in den theoretischen Grundlagen dieser Arbeit erläutert, dem Bereich der prädiktiven Instandhaltung zuzuordnen. Dies wäre im Fall des präsentierten Use-Cases die Prognose des Zeitintervalls. Präskriptive

Instandhaltung sieht darüber hinaus auch eine Maßnahmenableitung beziehungsweise -spezifizierung vor, also die Festlegung, welche Bauteile fehlerhaft sind, um das Instandhaltungspersonal bei der Behebung der Störung zu unterstützen. Daraus resultiert, dass ein effizienter Einsatz von Bayes'schen Netzwerken eher als Unterstützung, im Bereich der präskriptiven Instandhaltung vermutet werden kann als in der prädiktiven.

6.6 Ergebnisse in Bezug auf die Problemstellung

Die Problemstellung adressiert die Schwierigkeit für Unternehmen, die für ihre Anwendungsfälle passende Methode zur Datenanalyse und Instandhaltungsstrategie zu finden. Vor allem das historisch bedingte Wachstum der Unternehmen und das häufig ungenügende Anpassen der Datensysteme an die neuen Rahmenbedingungen verschärfen dieses Dilemma zusätzlich. Da die Datengrundlage in jedem Unternehmen aufgrund von Einflüssen wie zum Beispiel der Buchungsqualität/disziplin, unterschiedlich ist, kann keine allgemeine wissenschaftlich fundierte Empfehlung bezüglich der Anwendung spezifischer Datenanalyse- und Modellierungsalgorithmen abgegeben werden. Weitere Forschungsaktivitäten in Richtung eines Datenreifegradmodells, das bewertet in welcher Form Daten vorliegen und basierend auf dieser Bewertung definieren, welche Methoden angewendet werden können, sind notwendig, um dieser Problematik zukünftig gegenüberzutreten. Ein im Jahr 2018 von Bernerstätter präsentierter Ansatz adressiert diese Problematik. Dieser sieht einen Vorschlag zur Bewertung des Reifegrads von Daten innerhalb definierter Kategorien vor. Darüber hinaus sollen zusätzlich zur Bewertung der Daten auch die Software- und Hardwaresysteme evaluiert werden, um somit eine ganzheitliche Bewertung der relevanten IT-Umgebung zu ermöglichen.¹⁶⁸

6.7 Conclusio

Der Bereich der Halbleiterfertigung, mit besonderem Fokus auf die Disziplin des Halbleiterätzens wird von einem hohen Grad an Komplexität begleitet. Dies liegt daran, dass in diesem Gebiet sämtliche Bereiche der Chemie und Physik aufeinandertreffen. Demzufolge ist es äußerst aufwändig, ein möglichst allumfassendes Wissen über die gesamte Prozesskette zu erlangen. Die Erstellung und Erprobung eines Bayes'schen Netzes für den gezeigten Use-Case gestaltet sich als sehr aufwändig. Zunächst muss sehr viel Domänenwissen bezüglich der Fertigungsprozesse aufgebaut werden, um Zusammenhänge und Input in Form von Expertenwissen verstehen zu können. Da das Netzwerk nicht automatisch mittels diverser mathematischer Modelle (z.B. Hill-Climbing Algorithmen) erstellt werden konnte, musste es manuell modelliert werden. Bei der manuellen Erstellung solcher Netze muss somit auf Basis von Expertenwissen

¹⁶⁸ Bernerstätter, 2018, S.31ff

und von gefundenen Zusammenhängen aus den Daten erfolgen. Damit erfolgt die Erstellung aufgrund von subjektiven Eindrücken der erstellenden Personen. Dies birgt die Gefahr, dass Zusammenhänge und Abhängigkeiten unberücksichtigt bleiben. Zusätzlich wurde das Unterfangen durch unvollständige Daten bzw. dem fehlenden Informationsgehalt der Daten erschwert. Die Prognose des Zustandes und der damit verbundenen Zeitintervalle bezieht sich vor allem auf die Wahrscheinlichkeit, dass ein Alarm oder eine Limitverletzung auftritt. Diese sind in erster Linie von den Einsatzstunden des Moduls und der Kammer abhängig. Ein Bezug zu weiteren Einflüssen auf Alarme und Limitverletzungen, zum Beispiel der von kritischen Komponenten, konnte aufgrund der ungenügenden Datenqualität des Instandhaltungsmaßnahmenbuches nicht hergestellt werden.

7 Ausblick

Da der Validierungszeitraum sehr kurz bemessen war, konnte im Zuge der Validierung die Prognosefähigkeit des Modells nur anhand von zwei Beispielen gezeigt werden. Folglich müsste für eine holistische Erprobung des Modells ein größeres Datenset bereitgestellt werden, welches während der Durchführung dieser Diplomarbeit nicht vorhanden war.

Um weitere Forschung im Bereich der Bayes'schen Netze zur prädiktiven Instandhaltung in der Halbleiterfertigung zu betreiben, müssen des Weiteren Maßnahmen gesetzt werden, um die Problematik der Ungenauigkeit adressieren zu können. Diese werden im nachfolgenden erörtert.

- **Erhöhung der Datenqualität**

Um zukünftig in der Lage zu sein, das Modell auf Bauteil- und Komponenten-Ebene zu erstellen (siehe Kapitel 5.3) ist eine Erhöhung des Informationsgehaltes und der Qualität der Daten unerlässlich. Da vor allem das Instandhaltungsmaßnahmenbuch diesbezüglich Schwächen aufzeigt, sollte bei diesem gestartet werden. Vor allem Informationen bezüglich ausgetauschter oder gewarteter Bauteile müssen in Zukunft in diesem System hinterlegt werden. Eine Möglichkeit zur Implementierung wäre zum Beispiel die Erweiterung der Buchungsmaske mit Sub-Menüs, in dem die Instandhalter die kontrollierten oder auch getauschten Bauteile mittels „Checkboxen“ zuordnen können. Auch die Einführung von standardisierten Textbausteinen bei der Ausfallbehebung könnte in Betracht gezogen werden, um zukünftige Analysen mittels Text-Mining zu ermöglichen.

- **Erforschung weiterer Zusammenhänge im Ausfallverhalten**

Im Zuge dieser Arbeit wurde das Auftreten bestimmter Events mit den bereits von der Maschine geleisteten Einsatzstunden verknüpft. Weitere Zusammenhänge konnten nicht ermittelt werden. Durch die bereits angesprochene Erhöhung des Dateninformationsgrades könnten erweiterte Analysetechniken und Machine-Learning Verfahren, wie zum Beispiel Text-Mining oder Neuronale Netze, zum Einsatz kommen.

8 Anhang

8.1 R-Skripten

8.1.1 Aufbereitung Limitverletzungen und TTF

```

# Load Libraries
library(ggplot2)
library(dplyr)
library(zoo)
library(lubridate)
library(stringr)
# -----
# Load Breakdown Status Information
# -----
#rm(List=ls())
dFchamber='L2303-C'
LVchamber='L2303_PM3'
foldername='PM3'

setwd("C:/Users/FAB/Desktop/RohdatenInfineonSD4M/Extrahierte_Daten")

#dF ist das File RTC enhanced
f <- file('RTCenhanced_L2303.csv', open="r", encoding="UTF-16LE")
dF <- read.csv(f, header = TRUE, sep = ";", fill = TRUE)

setwd("C:/Users/FAB/Desktop/RohdatenInfineonSD4M/APC_Limit_Violations")
LVdata <- read.table('Limit_Violations_L2300.tab', header = T, sep = "\t", fill =
TRUE,nrows=400000)

subsetdF = subset(dF, dF$EQUIPMENT==dFchamber)
subsetLV = subset(LVdata, LVdata$Equipment==LVchamber)
subsetdF$DATE= as.POSIXct(subsetdF$TIME_STAMP, format = "%Y-%m-%d %H:%M:%S")
#subsetLV=subset(subsetLV, grepl( "_F$", subsetLV$KEYNUMBER))
subsetLV$DATE=as.POSIXct(subsetLV$RUNSTART, format = "%Y-%m-%dT%H:%M:%S")

LVFeatures=data.frame(subsetLV$DATE,subsetLV$KEYNUMBER, subsetLV$LL,subsetLV$TESTV
ALUE, subsetLV$UL, subsetLV$Recipe, subsetLV$Basic.type, subsetLV$RECOMMENDATION,
subsetLV$COMMENTS)
names(LVFeatures)=c('DATE','KEYNUMBER','LL','TESTVALUE','UL','Recipe','Basic.Type'
,'RECOMMENDATION','Comments')
dFFeatures=data.frame(subsetdF$DATE, subsetdF$EVENT_BIG_CATEGORY)
names(dFFeatures)=c('DATE','Status')

lv_fullldata=full_join(LVFeatures,dFFeatures)
## Joining, by = "DATE"

lv_fullldata=lv_fullldata[order(lv_fullldata$DATE),c(1:length(lv_fullldata))]
rownames(lv_fullldata) <- 1:nrow(lv_fullldata)

#um Datensatz beginnt mit 30.6.2017 und endet mit 31.12.2017
lv_fullldata <- lv_fullldata %>%
  filter(DATE > "2017-06-30", DATE <= "2017-12-31" )

#lv_fullldata=lv_fullldata[c(2110:10000),]

```

```

#neue spalte einfügen
lv_fulldata$StatusStop=NA
#wenn in status limitviolation oder breakdown angezeigt wird dann wird die neue spalte mit STOP markiert (für Berechnung)
lv_fulldata$StatusStop[which(lv_fulldata$Status=='LimitViolation')]='STOP'
lv_fulldata$StatusStop[which(lv_fulldata$Status=='BREAKDOWN')]='STOP'
LVinds=which(lv_fulldata$StatusStop=='STOP')
lv_fulldata$TimeDiff=0
lv_fulldata$TimeDiff[1:LVinds[1]]=lv_fulldata$DATE[LVinds[1]]-lv_fulldata$DATE[1:LVinds[1]]

for (jj in 2:length(LVinds)){
  lv_fulldata$TimeDiff[(LVinds[jj-1]+1):LVinds[jj]]=lv_fulldata$DATE[LVinds[jj]]-lv_fulldata$DATE[(LVinds[jj-1]+1):LVinds[jj]]
}
#lv_fulldata$TimeDiff=as.factor(lv_fulldata$TimeDiff)
lv_fulldata$TimeDiff=lv_fulldata$TimeDiff/(60*60) #Display in Hours

```

8.1.2 Aufbereitung Alarme und TTF

```

library(readxl)
library(ggplot2)
library(corrplot)
library(dplyr)
library(zoo)
library(gdata)
library(lubridate)

###
# Run Code Add_AlarmCats_ALLDates.R to obtain AlarmSubset!!
###
#keep(Alarmsubset, sure=TRUE)
#Laden der extrahierten ALarm_rohdaten
setwd("C:/Users/FAB/Desktop/RohdatenInfineonSD4M/Extrahierte_Daten")

APC_Alarme <- read.csv("APC_Alarmlogs_Rohdaten.csv", header = TRUE, sep = ",", dec = ".")
names(APC_Alarme)[names(APC_Alarme) == 'Absolute_Time'] <- "DATE"
APC_Alarme$DATE <- ymd_hms(APC_Alarme$DATE)

setwd("C:/Users/FAB/Desktop/RohdatenInfineonSD4M/Extrahierte_Daten")
f <- file('RTCenhanced_L2303.csv', open="r", encoding="UTF-16LE")
dF <- read.csv(f, header = TRUE, sep = ";", fill = TRUE)
dF$DATE= as.POSIXct(dF$TIME_STAMP, format = "%Y-%m-%d %H:%M:%S")
dFA = subset(dF, dF$EQUIPMENT=='L2303-A')
dFB = subset(dF, dF$EQUIPMENT=='L2303-B')
dFC = subset(dF, dF$EQUIPMENT=='L2303-C')
rm(dF)

AlarmFeatures=data.frame(APC_Alarme$DATE,APC_Alarme$AlarmId, APC_Alarme$AlarmName,
APC_Alarme$AlarmCat, APC_Alarme$AlarmText)
names(AlarmFeatures)=c('DATE', 'ID', 'NAME', 'CAT', 'Text')
dFA_Feat=data.frame(dFA$DATE, dFA$EVENT_BIG_CATEGORY)
names(dFA_Feat)=c('DATE', 'StatusA')
dFB_Feat=data.frame(dFB$DATE, dFB$EVENT_BIG_CATEGORY)
names(dFB_Feat)=c('DATE', 'StatusB')
dFC_Feat=data.frame(dFC$DATE, dFC$EVENT_BIG_CATEGORY)
names(dFC_Feat)=c('DATE', 'StatusC')

alarme_fulldata=full_join(AlarmFeatures, dFA_Feat)

```

```

alarme_fullldata=alarme_fullldata[order(alarme_fullldata$DATE),c(1:length(alarme_fullldata))]
rownames(alarme_fullldata) <- 1:nrow(alarme_fullldata)
#alarme_fullldata=alarme_fullldata[c(3582:109300),]

alarme_fullldata$AstatusStop=NA
alarme_fullldata$AstatusStop[which(alarme_fullldata$StatusA=='LimitViolation')]='STOP'
alarme_fullldata$AstatusStop[which(alarme_fullldata$StatusA=='BREAKDOWN')]='STOP'

alarme_fullldata$BstatusStop=NA
alarme_fullldata$BstatusStop[which(alarme_fullldata$StatusB=='LimitViolation')]='STOP'
alarme_fullldata$BstatusStop[which(alarme_fullldata$StatusB=='BREAKDOWN')]='STOP'

alarme_fullldata$CstatusStop=NA
alarme_fullldata$CstatusStop[which(alarme_fullldata$StatusC=='LimitViolation')]='STOP'
alarme_fullldata$CstatusStop[which(alarme_fullldata$StatusC=='BREAKDOWN')]='STOP'

Ainds=which(alarme_fullldata$AstatusStop=='STOP')
Binds=which(alarme_fullldata$BstatusStop=='STOP')
Cinds=which(alarme_fullldata$CstatusStop=='STOP')

alarme_fullldata$ATimeDiff=0
alarme_fullldata$BTimeDiff=0
alarme_fullldata$CTimeDiff=0

alarme_fullldata$ATimeDiff[1:Ainds[1]]=alarme_fullldata$DATE[Ainds[1]]-alarme_fullldata$DATE[1:Ainds[1]]
alarme_fullldata$BTimeDiff[1:Binds[1]]=alarme_fullldata$DATE[Binds[1]]-alarme_fullldata$DATE[1:Binds[1]]
alarme_fullldata$CTimeDiff[1:Cinds[1]]=alarme_fullldata$DATE[Cinds[1]]-alarme_fullldata$DATE[1:Cinds[1]]

for (jj in 2:length(Ainds)){
  alarme_fullldata$ATimeDiff[(Ainds[jj-1]+1):Ainds[jj]]=alarme_fullldata$DATE[Ainds[jj]]-alarme_fullldata$DATE[(Ainds[jj-1]+1):Ainds[jj]]
}
alarme_fullldata$ATimeDiff=alarme_fullldata$ATimeDiff/(60*60) #Display in Hours

for (jj in 2:length(Binds)){
  alarme_fullldata$BTimeDiff[(Binds[jj-1]+1):Binds[jj]]=alarme_fullldata$DATE[Binds[jj]]-alarme_fullldata$DATE[(Binds[jj-1]+1):Binds[jj]]
}
alarme_fullldata$BTimeDiff=alarme_fullldata$BTimeDiff/(60*60) #Display in Hours

for (jj in 2:length(Cinds)){
  alarme_fullldata$CTimeDiff[(Cinds[jj-1]+1):Cinds[jj]]=alarme_fullldata$DATE[Cinds[jj]]-alarme_fullldata$DATE[(Cinds[jj-1]+1):Cinds[jj]]
}
alarme_fullldata$CTimeDiff=alarme_fullldata$CTimeDiff/(60*60) #Display in Hours

#Definieren eines zusätzlichen Dataframe der nur mehr Informationen über PM3 beinhaltet
#Subset sodass nur Einträge von PM3 aufscheinen
library(stringr)
liste_PM3_filter_1 <- c(which(str_detect(alarme_fullldata$NAME, "PM3")==TRUE))
liste_all_alarms_2 <- c(which(str_detect(alarme_fullldata$NAME, "<AllAlarms>")==TRUE))
liste_all_alarms_3 <- c(which(alarme_fullldata$CstatusStop=="STOP"))

```

```

liste_PM3_gesamt <- c(liste_PM3_filter_1, liste_all_alarms_2, liste_all_alarms_3)

#Löschen der Hilfsvariablen damit nicht so viele Variablen belegt sind
rm(liste_PM3_filter_1)
rm(liste_all_alarms_2)
rm(liste_all_alarms_3)

alarme_fullldata_PM3 <- alarme_fullldata[liste_PM3_gesamt,] %>%
  arrange(DATE)%>%
  filter(DATE > "2017-06-30", DATE <= "2017-12-31" ) %>% #
  um Datensatz beginnt mit 30.6.2017 und endet mit 31.12.2017
  mutate(ID_test = ID - 30000000)

beibehalten <- which(alarme_fullldata_PM3$ID_test > 0 | is.na(alarme_fullldata_PM3$I
D_test)==TRUE )
alarme_fullldata_PM3 <- alarme_fullldata_PM3[beibehalten,]

alarme_fullldata_PM3$StatusA <- NULL
alarme_fullldata_PM3$StatusB <- NULL
alarme_fullldata_PM3$AStatusStop <- NULL
alarme_fullldata_PM3$BStatusStop <- NULL
alarme_fullldata_PM3$ATimeDiff <- NULL
alarme_fullldata_PM3$BTimeDiff <- NULL
alarme_fullldata_PM3$ID_test <- NULL

```

8.1.3 Einlesen der prozessspezifischen Kennzahlen

```

library(readxl)
library(ggplot2)
library(corrplot)
library(dplyr)
library(zoo)
library(gdata)
library(lubridate)
#Einlesen APC Daten
setwd("C:/Users/FAB/Desktop/RohdatenInfineonSD4M/Extrahierte_Daten")
APC_kennzahlen <- read.csv("APC_Kennzahlen.csv", header = TRUE, sep = ",", dec =
".")
APC_kennzahlen$RUNEND <- ymd_hms(APC_kennzahlen$RUNEND)
APC_kennzahlen$RUNSTART <- ymd_hms(APC_kennzahlen$RUNSTART)
APC_kennzahlen$BATCHSTART <- ymd_hms(APC_kennzahlen$BATCHSTART)
APC_kennzahlen$BATCHEND <- ymd_hms(APC_kennzahlen$BATCHEND)
APC_kennzahlen$BAT_ID <- as.factor(APC_kennzahlen$BAT_ID)
names(APC_kennzahlen)[names(APC_kennzahlen) == 'RUNSTART'] <- "DATE"
APC_kennzahlen_PM3 <- APC_kennzahlen %>%
  filter(Chamber == "PM3")

```

8.1.4 Bestimmung TTF für Biasvoltage, Clampvoltage und Generator Power (LV und Alarme)

```

#Berechnung der TTFs für Limitviolations

#####
#####
#BIASVOLTAGE TTF LV
#####
#####
#in welchen zeilen befindet sich biasvoltage
bias_rows <- which((str_detect(lv_fulldata$KEYNUMBER, "BIAS"))==TRUE)

#erstellen eines neuen dataframes der nur biasvoltage Limitviolations enthält
bias_frame <- lv_fulldata[bias_rows,]
rownames(bias_frame) <- 1:nrow(bias_frame)

#erstellen eines trainings und eines validierungsdatensatzes
bias_frame_training <- filter(bias_frame, DATE <= "2017-11-14 22:50:29 CET")
bias_frame_validierung <- filter(bias_frame, DATE > "2017-11-14 22:50:29 CET")

#berechnung der TTF von Biasvoltage related Limitviolations
filter_vektor <- rep(1, nrow(bias_frame_training))

bias_frame_training_filter <- cbind(bias_frame_training, filter_vektor)

bias_frame_training_filter <- spread(bias_frame_training_filter, RECOMMENDATION, f
ilter_vektor)

#leerer Vektor für ttf zeiten
ttf_vektor_biasvoltage <- c()

#schleife um leeren vektor mit ttf zeiten zu füllen
for (i in 1:length(bias_frame_training_filter$DATE)) {
  if(is.na(bias_frame_training_filter$STOPEQUIPMENTPROCESSINGCHA[i]) == TRUE | is.
na(bias_frame_training_filter$INHIBITCHA[i])){
    ttf_vektor_biasvoltage[i] <- bias_frame_training_filter$TimeDiff[i]
  } else {
    i <- i+1
  }
}

#na s dropen
drop <- which(is.na(ttf_vektor_biasvoltage)==TRUE)
ttf_vektor_biasvoltage <- ttf_vektor_biasvoltage[-drop]

#ttf Biasvoltage
lv_ttf_bias <- mean(ttf_vektor_biasvoltage)
quantile(ttf_vektor_biasvoltage)

#####
#####
#CLAMPVOLTAGE TTF LV
#####
#####
#in welchen zeilen befindet sich clampvoltage
clamp_rows <- which((str_detect(lv_fulldata$KEYNUMBER, "CLAMP"))==TRUE)

```

```

#erstellen eines neuen dataframes der nur biasvoltage Limitviolations enthält
clamp_frame <- lv_fullldata[clamp_rows,]
rownames(clamp_frame) <- 1:nrow(clamp_frame)

#erstellen eines trainings und eines validierungsdatensatzes
clamp_frame_training <- filter(clamp_frame, DATE <= "2017-11-14 22:50:29 CET")
clamp_frame_validierung <- filter(clamp_frame, DATE > "2017-11-14 22:50:29 CET")

# erstellen eines vektors um zu überprüfen ob HoldLot und inhibitcha vorkommt

clamp_recommendation <- unique(clamp_frame_training$RECOMMENDATION)
bias_recommendation <- unique(bias_frame_training$RECOMMENDATION)

#berechnung der TTF von CLAMPvoltage related Limitviolations

filter_vektor <- rep(1, nrow(clamp_frame_training))

clamp_frame_training_filter <- cbind(clamp_frame_training, filter_vektor)

clamp_frame_training_filter <- spread(clamp_frame_training_filter, RECOMMENDATION,
filter_vektor)

#Leerer Vektor für ttf zeiten
ttf_vektor_clamp <- c()

#schleife um leeren vektor mit ttf zeiten zu füllen
#zuerst muss überprüft werden ob zustand STOPEquipment und INHIBITCHA vorkommen (w
enn nicht würde sonst eine Fehlermeldung ausgespuckt werden)

if(sum(str_count(clamp_recommendation, "STOPE")) < 1 | sum(str_count(clamp_recommen
dation, "INHIBIT")) < 1){
  ttf_vektor_clamp <- clamp_frame_training_filter$TimeDiff
} else { for (i in 1:length(clamp_frame_training_filter$DATE)) {

  if(is.na(clamp_frame_training_filter$STOPEQUIPMENTPROCESSINGCHA[i]) ==
TRUE | is.na(clamp_frame_training_filter$INHIBITCHA[i])){
    ttf_vektor_clamp[i] <- clamp_frame_training_filter$TimeDiff[i]
  } else {
    i <- i+1
  }
}
}

#na s dropen
#if Schleife um zu überprüfen ob drop Vektor Leer ist
#wenn ja darf drop nicht durchgeführt werden weil sich sonst inhalt des ttf vektor
s löscht

drop <- which(is.na(ttf_vektor_clamp)==TRUE)

if (sum(drop)==0){
  ttf_vektor_clamp <- ttf_vektor_clamp
} else {
  ttf_vektor_clamp <- ttf_vektor_clamp[-drop]
}

#ttf Clampvoltage
lv_ttf_clamp <- mean(ttf_vektor_clamp)
quantile(ttf_vektor_clamp)
#####
#####
#PWR MHZ Generator TTF LV

```

```
#####
#####

pwrmh_rows <- which((str_detect(lv_fulldata$KEYNUMBER, "MHZ"))==TRUE)

#erstellen eines neuen dataframes der nur biasvoltage Limitviolations enthält
pwrmh_frame <- lv_fulldata[pwrmh_rows,]
rownames(pwrmh_frame) <- 1:nrow(pwrmh_frame)

#erstellen eines trainings und eines validierungsdatensatzes
pwrmh_frame_training <- filter(pwrmh_frame, DATE <= "2017-11-14 22:50:29 CET")
pwrmh_frame_validierung <- filter(pwrmh_frame, DATE > "2017-11-14 22:50:29 CET")

# erstellen eines vektors um zu überprüfen ob Holdlot und inhibitcha vorkommt

pwrmh_recommendation <- unique(pwrmh_frame_training$RECOMMENDATION)

#berechnung der TTF von Biasvoltage related Limitviolations

filter_vektor <- rep(1, nrow(pwrmh_frame_training))

pwrmh_frame_training_filter <- cbind(pwrmh_frame_training, filter_vektor)

pwrmh_frame_training_filter <- spread(pwrmh_frame_training_filter, RECOMMENDATION, filter_vektor)

#Leerer Vektor für ttf zeiten
ttf_vektor_pwrmh <- c()

#schleife um leeren vektor mit ttf zeiten zu füllen
#zuerst muss überprüft werden ob zustand STOPEquipment und INHIBITCHA vorkommen (wenn nicht würde sonst eine Fehlermeldung ausgespuckt werden)

if(sum(str_count(pwrmh_recommendation, "STOPE")) < 1 | sum(str_count(pwrmh_recommendation, "INHIBIT")) < 1){
  ttf_vektor_pwrmh <- pwrmh_frame_training_filter$TimeDiff
} else { for (i in 1:length(pwrmh_frame_training_filter$DATE)) {

  if(is.na(pwrmh_frame_training_filter$STOPEQUIPMENTPROCESSINGCHA[i]) == TRUE | is.na(pwrmh_frame_training_filter$INHIBITCHA[i])){
    ttf_vektor_pwrmh[i] <- pwrmh_frame_training_filter$TimeDiff[i]
  }else{
    i <- i+1
  }
}
}

#na s dropen
#if Schleife um zu überprüfen ob drop Vektor Leer ist
#wenn ja darf drop nicht durchgeführt werden weil sich sonst inhalt des ttf vektor s löscht

drop_pwrmh <- which(is.na(ttf_vektor_pwrmh)==TRUE)

if (sum(drop_pwrmh)==0){
  ttf_vektor_pwrmh <- ttf_vektor_pwrmh
} else {
  ttf_vektor_pwrmh <- ttf_vektor_pwrmh[-drop]
}

#ttf PWR MHZ
```

```

lv_ttf_pwrMHz <- mean(ttf_vektor_pwrMHz)
quantile(ttf_vektor_pwrMHz)

#ploten der 3 erzeugten Vektoren

boxplot(ttf_vektor_biasvoltage, ttf_vektor_clamp, ttf_vektor_pwrMHz)

TTF_LIMITVIOLATIONS_PWRMHZ <- data.frame(lv_ttf_bias, lv_ttf_clamp, lv_ttf_pwrMHz)

#####
#####
#Berechnung der TTFS für ALARME
#####
#####

#subset dataframe der nur noch alarme enthält die Chuck related sind

chuck_alarm <- which(str_detect(alarme_fulldata_PM3$NAME, "Chuck"))
ESC_alarm <- which(str_detect(alarme_fulldata_PM3$NAME, "ESC"))
all_alarms <- which(str_detect(alarme_fulldata_PM3$NAME, "<AllAlarms"))
stops <- which(is.na(alarme_fulldata_PM3$NAME))

echuck_liste <- c(chuck_alarm, ESC_alarm, all_alarms, stops)

subset_alarme_fulldata_PM3 <- alarme_fulldata_PM3[echuck_liste,] %>%
  arrange(DATE)
#####
#####
#Biasvoltage TTF Alarme
#####
#####
#Filtern um Alarme die Biasvoltage related sind herauszufinden
biasvoltage_alarms_zeilen1 <- which(str_detect(subset_alarme_fulldata_PM3$NAME, "B
ia"))
biasvoltage_alarms_zeilen2 <- which(str_detect(subset_alarme_fulldata_PM3$Text, "B
ia"))
biasvoltage_alarms_zeilen3 <- c(biasvoltage_alarms_zeilen1, biasvoltage_alarms_zei
len2)
#variablen Löschen
rm(biasvoltage_alarms_zeilen1)
rm(biasvoltage_alarms_zeilen2)
#erstellen des subsets mit nur biasvoltage related alarms
biasvoltage_frame_alarm <- subset_alarme_fulldata_PM3[biasvoltage_alarms_zeilen3,]
#erstellen des trainingsdatasets
biasvoltage_frame_alarm_training <- filter(biasvoltage_frame_alarm, DATE <= "2017-
11-14 22:50:29 CET")
#erstellen des validierungsdatasets
biasvoltage_frame_alarm_validierung <- filter(biasvoltage_frame_alarm, DATE > "201
7-11-14 22:50:29 CET")

##!!!!
alarm_ttf_bias <-27.82 #aus Kammer 2 --> keine Werte in Kammer 3

#####
#####
# CLAMP Alarme
#####
#####

```

```

#Filtern um Alarme die PWRMHZ related sind herauszufinden

clamp_alarms_zeilen1 <- which(str_detect(subset_alarme_fulldata_PM3$NAME, "clamp"))
)
clamp_alarms_zeilen2 <- which(str_detect(subset_alarme_fulldata_PM3$Text, "clamp"))
)
clamp_alarms_zeilen3 <- c(clamp_alarms_zeilen1, clamp_alarms_zeilen2)

#variablen Löschen
rm(clamp_alarms_zeilen1)
rm(clamp_alarms_zeilen2)

#erstellen des subsets mit nur biasvoltage related alarms
clamp_frame_alarm <- subset_alarme_fulldata_PM3[clamp_alarms_zeilen3,]
#erstellen des trainingsdatasets
clamp_frame_alarm_training <- filter(clamp_frame_alarm, DATE <= "2017-11-14 22:50:
29 CET")
#erstellen des validierungsdatasets
clamp_frame_alarm_validierung <- filter(clamp_frame_alarm, DATE > "2017-11-14 22:5
0:29 CET")

alarm_ttf_clamp <- mean(clamp_frame_alarm_training$CTimeDiff)

quantile(clamp_frame_alarm_training$CTimeDiff)
#####
#####
# PWRMHZ Alarme
#####
#####
#Filtern um Alarme die PWRMHZ related sind herauszufinden

pwrmh_alarms_zeilen1 <- which(str_detect(subset_alarme_fulldata_PM3$NAME, "MHz"))
pwrmh_alarms_zeilen2 <- which(str_detect(subset_alarme_fulldata_PM3$Text, "MHz"))
pwrmh_alarms_zeilen3 <- c(pwrmh_alarms_zeilen1, pwrmh_alarms_zeilen2)

#variablen Löschen
rm(pwrmh_alarms_zeilen1)
rm(pwrmh_alarms_zeilen2)

#erstellen des subsets mit nur biasvoltage related alarms
pwrmh_frame_alarm <- subset_alarme_fulldata_PM3[pwrmh_alarms_zeilen3,]
#erstellen des trainingsdatasets
pwrmh_frame_alarm_training <- filter(pwrmh_frame_alarm, DATE <= "2017-11-14 22:5
0:29 CET")
#erstellen des validierungsdatasets
pwrmh_frame_alarm_validierung <- filter(pwrmh_frame_alarm, DATE > "2017-11-14 22
:50:29 CET")

alarm_ttf_pwrmh <- mean(pwrmh_frame_alarm_training$CTimeDiff)
quantile(pwrmh_frame_alarm_training$CTimeDiff)

TTF_ALARMS_PWRMHZ <- data.frame(alarm_ttf_bias, alarm_ttf_clamp, alarm_ttf_pwrmh)

```

8.1.5 Berechnung der Auftretenswahrscheinlichkeiten für Biasvoltage, Clampvoltage und Generator Power

```

#####
#####
# PWRMHZ Alarme

```

```
#####
#####
#Filtern um Alarme die PWRMHZ related sind herauszufinden

pwrmhz_alarms_zeilen1 <- which(str_detect(subset_alarme_fulldata_PM3$NAME, "MHZ"))
pwrmhz_alarms_zeilen2 <- which(str_detect(subset_alarme_fulldata_PM3$Text, "MHZ"))
pwrmhz_alarms_zeilen3 <- c(pwrmhz_alarms_zeilen1, pwrmhz_alarms_zeilen2)

#variablen Löschen
rm(pwrmhz_alarms_zeilen1)
rm(pwrmhz_alarms_zeilen2)

#erstellen des subsets mit nur pwrmhz related alarms
pwrmhz_frame_alarm <- subset_alarme_fulldata_PM3[pwrmhz_alarms_zeilen3,]
#erstellen des trainingsdatasets
pwrmhz_frame_alarm_training <- filter(pwrmhz_frame_alarm, DATE <= "2017-11-14 22:5
0:29 CET")
#erstellen des validierungsdatasets
pwrmhz_frame_alarm_validierung <- filter(pwrmhz_frame_alarm, DATE > "2017-11-14 22
:50:29 CET")

alarm_ttf_pwrmhz <- mean(pwrmhz_frame_alarm_training$CTimeDiff)
quantile(pwrmhz_frame_alarm_training$CTimeDiff)

TTF_ALARMS_PWRMHZ <- data.frame(alarm_ttf_bias, alarm_ttf_clamp, alarm_ttf_pwrmhz)

#Auftreten PWRMHZ

#APC_Kennzahlen in trainingsdatenset einteilen
#Datenset von LV und Alarme geht von 30.6.2017 bis 31.12.2017

apc_kennzahlen_PM3_training <- APC_kennzahlen_PM3 %>%
  filter(DATE <= "2017-11-14 22:50:29 CET") %>%
  arrange(DATE)

#ausfiltern von WAC Step

apc_kennzahlen_PM3_training <- apc_kennzahlen_PM3_training[which(apc_kennzahlen_PM
3_training$Recipe.Type!="CLEAN"),]

anzahl_pwrmhz_lv <- length(which((str_detect(lv_fulldata$KEYNUMBER, "MHZ"))==TRUE)
)
cpt_lv_pwrmhz_true <- anzahl_pwrmhz_lv/length(apc_kennzahlen_PM3_training$DATE)
cpt_lv_pwrmhz_false <- 1-cpt_lv_pwrmhz_true

#####
#####
#Berechnung mit fulldata Datensatz als Grundlage
apc_kennzahlen_PM3_training <- lv_fulldata %>%
  filter(DATE <= "2017-11-14 22:50:29 CET") %>%
  arrange(DATE)

anzahl_pwrmhz_lv <- length(which((str_detect(lv_fulldata$KEYNUMBER, "MHZ"))==TRUE)
)
cpt_lv_pwrmhz_true <- anzahl_pwrmhz_lv/length(apc_kennzahlen_PM3_training$DATE)
cpt_lv_pwrmhz_false <- 1-cpt_lv_pwrmhz_true

#####
#####
#Trainingsdatenset für die Alarme
```

```

alarme_fullldata_PM3_training <- alarme_fullldata_PM3 %>%
  filter(DATE <= "2017-11-14 22:50:29 CET") %>%
  arrange(DATE)

#Subset Datenset wo nur ECHUCK PWRMHZ relevante alarme aufscheinen
subset_alarme_fullldata_PM3_training <- subset_alarme_fullldata_PM3 %>%
  filter(DATE <= "2017-11-14 22:50:29 CET"
) %>%
  arrange(DATE)

#zeilen finden die MHz enthalten (für PWRMHZ)
pwrmhz_alarms_zeilen1 <- which(str_detect(subset_alarme_fullldata_PM3_training$NAME
, "MHz"))
pwrmhz_alarms_zeilen2 <- which(str_detect(subset_alarme_fullldata_PM3_training$Text
, "MHz"))
pwrmhz_alarms_zeilen3 <- c(pwrmhz_alarms_zeilen1, pwrmhz_alarms_zeilen2)
rm(pwrmhz_alarms_zeilen1)
rm(pwrmhz_alarms_zeilen2)

anzahl_pwrmhz_alarme <- length(pwrmhz_alarms_zeilen3)
cpt_alarm_pwrmhz_true <- anzahl_pwrmhz_alarme/length(alarme_fullldata_PM3_training$
DATE)
cpt_alarm_pwrmhz_false <- 1-cpt_alarm_pwrmhz_true

pwrmhz_cpt <- data.frame(cpt_alarm_pwrmhz_true, cpt_alarm_pwrmhz_false, cpt_lv_pwr
mhz_true, cpt_lv_pwrmhz_false)

apc_kennzahlen_PM3_training <- lv_fullldata %>%
  filter(DATE <= "2017-11-14 22:50:29 CET") %>%
  arrange(DATE)

#ausfiltern von WAC Step

#apc_kennzahlen_PM3_training <- apc_kennzahlen_PM3_training[which(apc_kennzahlen_P
M3_training$Recipe.Type!="CLEAN"),]

anzahl_clamp_lv <- length(which((str_detect(lv_fullldata$KEYNUMBER, "CLAMP"))==TRUE
))
cpt_lv_clamp_true <- anzahl_clamp_lv/length(apc_kennzahlen_PM3_training$DATE)
cpt_lv_clamp_false <- 1-cpt_lv_clamp_true

#####
#####
#Berechnung mit fullldata Datensatz als Grundlage
apc_kennzahlen_PM3_training <- lv_fullldata %>%
  filter(DATE <= "2017-11-14 22:50:29 CET") %>%
  arrange(DATE)

anzahl_clamp_lv <- length(which((str_detect(lv_fullldata$KEYNUMBER, "CLAMP"))==TRUE
))
cpt_lv_clamp_true <- anzahl_pwrmhz_lv/length(apc_kennzahlen_PM3_training$DATE)
cpt_lv_clamp_false <- 1-cpt_lv_clamp_true

#####
#####

#Trainingsdatenset für die Alarme
alarme_fullldata_PM3_training <- alarme_fullldata_PM3 %>%
  filter(DATE <= "2017-11-14 22:50:29 CET") %>%
  arrange(DATE)

```

```

#Subset Datenset wo nur ECHUCK PWRMHZ relevante alarme aufscheinen
subset_alarme_fullldata_PM3_training <- subset_alarme_fullldata_PM3 %>%
  filter(
    DATE <= "2017-11-14 22:50:29 CET"
  ) %>%
  arrange(
    DATE
  )

#zeilen finden die MHz enthalten (für PWRMHZ)
clamp_alarms_zeilen1 <- which(str_detect(subset_alarme_fullldata_PM3_training$NAME,
"clamp"))
clamp_alarms_zeilen2 <- which(str_detect(subset_alarme_fullldata_PM3_training$Text,
"clamp"))

clamp_alarms_zeilen4 <- which(str_detect(subset_alarme_fullldata_PM3_training$NAME,
"Dech"))
clamp_alarms_zeilen5 <- which(str_detect(subset_alarme_fullldata_PM3_training$Text,
"Dech"))

clamp_alarms_zeilen3 <- c(clamp_alarms_zeilen1, clamp_alarms_zeilen2, clamp_alarms_
_zeilen4, clamp_alarms_zeilen5)
rm(clamp_alarms_zeilen1)
rm(clamp_alarms_zeilen2)

anzahl_clamp_alarme <- length(clamp_alarms_zeilen3)
cpt_alarm_clamp_true <- anzahl_clamp_alarme/length(alarme_fullldata_PM3_training$DA
TE)
#cpt_alarm_clamp_true <- 0.1
cpt_alarm_clamp_false <- 1-cpt_alarm_clamp_true

clamp_cpt <- data.frame(cpt_alarm_clamp_true, cpt_alarm_clamp_false, cpt_lv_clamp_
true, cpt_lv_clamp_false)

#Auftreten Biasvoltage
#APC_Kennzahlen in trainingsdatenset einteilen
#Datenset von LV und Alarme geht von 30.6.2017 bis 31.12.2017

apc_kennzahlen_PM3_training <- lv_fullldata %>%
  filter(
    DATE <= "2017-11-14 22:50:29 CET"
  ) %>%
  arrange(
    DATE
  )

#ausfiltern von WAC Step
#apc_kennzahlen_PM3_training <- apc_kennzahlen_PM3_training[which(apc_kennzahlen_P
M3_training$Recipe.Type!="CLEAN"),]
#T_PROC LV auch dazugenommen . Diese sind allgemeine Prozessfehler und können auch
von APC Kennzahlen ausgelöst werden
#T_PROC ist die Ätzdauer. Diese hängt von einigen Parametern ab. Höherer Bias kann
zB zu einer höheren Ätzrate führen wodurch die Ätztiefe schneller
#erreicht wird als gewünscht
#T_PROC ohne BAT_ID
anzahl_bias_lv <- length(which((str_detect(lv_fullldata$KEYNUMBER, "BIAS"))==TRUE))
+ length(which((str_detect(lv_fullldata$KEYNUMBER, "T_PROC"))==TRUE))
cpt_lv_bias_true <- anzahl_bias_lv/length(apc_kennzahlen_PM3_training$DATE)
cpt_lv_bias_false <- 1-cpt_lv_bias_true

#####
#####

alarme_fullldata_PM3_training <- alarme_fullldata_PM3 %>%
  filter(
    DATE <= "2017-11-14 22:50:29 CET"
  ) %>%
  arrange(
    DATE
  )

#Subset Datenset wo nur ECHUCK PWRMHZ relevante alarme aufscheinen

```

```

subset_alarme_fullldata_PM3_training <- subset_alarme_fullldata_PM3 %>%
  filter(DATE <= "2017-11-14 22:50:29 CET") %>%
  arrange(DATE)
#zeilen finden die MHz enthalten (für PWRMHZ)
bias_alarms_zeilen1 <- which(str_detect(subset_alarme_fullldata_PM3_training$NAME,
"BIAS"))
bias_alarms_zeilen2 <- which(str_detect(subset_alarme_fullldata_PM3_training$Text,
"BIAS"))
bias_alarms_zeilen3 <- c(bias_alarms_zeilen1, bias_alarms_zeilen2)
rm(bias_alarms_zeilen1)
rm(bias_alarms_zeilen2)

anzahl_bias_alarme <- length(bias_alarms_zeilen3)
cpt_alarm_bias_true <- anzahl_bias_alarme/length(alarme_fullldata_PM3_training$DATE
)

#!!!!
cpt_alarm_bias_true <- cpt_lv_bias_true
cpt_alarm_bias_false <- 1-cpt_alarm_bias_true

bias_cpt <- data.frame(cpt_alarm_bias_true, cpt_alarm_bias_false, cpt_lv_bias_true
, cpt_lv_bias_false)

```

8.1.6 Aufbereitung Reinigungsdaten

```

#Einlesen der isShare Daten
library(readxl)
library(dplyr)
library(ggplot2)
library(zoo)
library(xts)

setwd("C:/Users/FAB/Desktop/RohdatenInfineonSD4M/Extrahierte_Daten/ebook und
ishare")
ishare <- read_xlsx("iShare.xlsx")
ishare$Tool <- as.factor(ishare$Tool)
ishare_Pm3 <- ishare[c(which(ishare$Tool == "L2303-PM3"),which(ishare$Tool ==
"L2303-C")),]
ishare_Pm3 <- ishare_Pm3 %>%
  arrange(StartZeit)
ishare_training <- ishare_Pm3 %>%
  filter(StartZeit <= "2017-11-14 22:50:29 CET") %>%
  arrange(StartZeit)
ishare_Pm3_red <- ishare_Pm3
ishare_Pm3_red$X_1 <- NULL
ishare_Pm3_red$Tool <- NULL
#ishare_Pm3_red$Reinigungsgrund <- NULL
ishare_Pm3_red$Quarz_Window_RFh <- NULL
ishare_Pm3_red$Nozzle_RFh <- NULL
ishare_Pm3_red$Pendulum_RFh <- NULL
ishare_Pm3_red$OES_Sichtfenster_RFh <- NULL
ishare_Pm3_red$`XML-filename` <- NULL
names(ishare_Pm3_red)[names(ishare_Pm3_red)=="StartZeit"] <- "TIME_STAMP"

```

```

ggplot(ishare_Pm3, aes(x = StartZeit, y = ESC_RFh)) + geom_point()
ggplot(ishare_Pm3, aes(x = StartZeit, y = VTT_RFh)) + geom_point()
boxplot(ishare_Pm3$VTT_RFh)
median(ishare_Pm3$VTT_RFh)
mean(ishare_Pm3$ESC_RFh)
summary(ishare_Pm3$VTT_RFh)
#####
#####
#Einlesen ebook
ebook <- read_excel("L2303_ebook_PM3_180823_FAB.xls", sheet = "L2303_Flex_8")
#####
#####
#plots für Limitviolations
#####
#####
#für alle 3 parameter gültig
ebook_lv <- ebook %>%
  filter(DOWN_EVENT == "LV")
ishare_plus_ebook_lv <- full_join(ishare_Pm3_red, ebook_lv, by = "TIME_STAMP")
ishare_plus_ebook_lv <- arrange(ishare_plus_ebook_lv, TIME_STAMP)
ishare_plus_ebook_lv$DOWN_EVENT <- as.factor(ishare_plus_ebook_lv$DOWN_EVENT)
ishare_plus_ebook_lv$`Komponente 1` <- as.factor(ishare_plus_ebook_lv$`Komponente
1`)

```

8.1.7 Erstellung Entscheidungsbaum VTT RFH <100h

```

#erstellen eines dataframes um alle möglichkeiten für VTT <100h abbilden zu können
spalte1 <- c(rep(rep(0,9), 100))
spalte2 <- c(rep(c(rep(283,3), rep(368,3), rep(380,3)), 100))
spalte3 <- c(rep(c(rep(0.25,3), rep(0.5,3), rep(0.25,3)),100))
spalte4 <- c(rep(rep(c(283, 368, 380), 3),100))
spalte5 <- c(rep(rep(c(0.25, 0.5, 0.25), 3), 100))

bis_knoten <- c(rep(0,100))
decision_tree_100_test <- data.frame(spalte1,bis_knoten, spalte2)

j <- 1
k <- 10
for(i in 2:100){
  decision_tree_100_test$spalte1[k] <- j
  decision_tree_100_test$spalte1[k+1] <- j
  decision_tree_100_test$spalte1[k+2] <- j
  decision_tree_100_test$spalte1[k+3] <- j
  decision_tree_100_test$spalte1[k+4] <- j
  decision_tree_100_test$spalte1[k+5] <- j
  decision_tree_100_test$spalte1[k+6] <- j
  decision_tree_100_test$spalte1[k+7] <- j
  decision_tree_100_test$spalte1[k+8] <- j
  j <- j+1
}

```

```

k <- k+9
}

for(i in 1:length(decision_tree_100_test$bis_knoten)){
  decision_tree_100_test$bis_knoten[i] <- decision_tree_100_test$spalte2[i]-decision_tree_100_test$spalte1[i]
}

decision_tree_100_test <- data.frame(decision_tree_100_test, spalte3, spalte4, spalte5)

decision_tree_100_test <- decision_tree_100_test %>%
  mutate(Wahrscheinlichkeit = spalte3*spalte5) %>%
  mutate(Summe = bis_knoten+spalte4) %>%
  mutate(zeitpunkt_t1=720) %>%
  mutate(Diff = zeitpunkt_t1 - Summe) %>%
  mutate(Zustand = 0)

#schleife die die VTT Stunden zum Zeitpunkt 720h (14.12.17) ausrechnet
for(i in 1:length(decision_tree_100_test$Diff)){
  if(decision_tree_100_test$Diff[i] < 0){
    decision_tree_100_test$Zustand[i] <- decision_tree_100_test$zeitpunkt_t1[i] - decision_tree_100_test$bis_knoten[i]
  } else {
    decision_tree_100_test$Zustand[i] <- decision_tree_100_test$Diff[i]
  }
}

#für den fall das sich rfh nicht mit 2 knoten ausgegangen wären
for(i in 1:length(decision_tree_100_test$Zustand)){
  if(decision_tree_100_test$Zustand[i] > decision_tree_100_test$spalte4[i]){
    decision_tree_100_test$Zustand[i] <- decision_tree_100_test$Zustand[i]-decision_tree_100_test$spalte4[i]
  }
}

#schleife die werte zu bereich zuordnet

for(i in 1:length(decision_tree_100_test$Zustand)){
  if(decision_tree_100_test$Zustand[i]<100){
    decision_tree_100_test$Bereich[i] <- "<100h"
  } else if (decision_tree_100_test$Zustand[i]>100 & decision_tree_100_test$Zustand[i]<300 ) {
    decision_tree_100_test$Bereich[i] <- "100< x < 300"
  } else {
    decision_tree_100_test$Bereich[i] <- ">300"
  }
}

#hinterlegen der prozentuellen gewichtung(jeder Pfad mit prozentzahl mal 1%(100 Pfade))

decision_tree_100_test <- decision_tree_100_test %>%
  mutate(Anteil = Wahrscheinlichkeit*0.01)

#wahrscheinlichkeiten am 14.12 wenn mit zustand <100 begonnen wird
t_1_100_t_100 <- sum(decision_tree_100_test$Anteil[which(decision_tree_100_test$Bereich == "<100h")])
t_1_100x300_t_100 <- sum(decision_tree_100_test$Anteil[which(decision_tree_100_test$Bereich == "100< x < 300")])

```

```
t$Bereich == "100< x < 300"]])
t_1_300_t_100 <- sum(decision_tree_100_test$Anteil[which(decision_tree_100_test$Be
reich == ">300"]])
```

8.1.8 Erstellung Entscheidungsbaum VTT RFH 100h < x < 300h

#erstellen eines dataframes um alle möglichkeiten für VTT 100<x<300 abbilden zu können

#zwei Bäume erforderlich -->erster Baum bis 283h

```
spalte1 <- c(rep(rep(100,27), 183))
knoten1_stunden <- c(rep(c(rep(283,9), rep(368,9), rep(380,9)),183))
knoten1_prob <- c(rep(c(rep(0.25,9), rep(0.5,9), rep(0.25,9)),183))
knoten2_stunden <- c(rep(c(rep(c(rep(283,3), rep(368,3), rep(380,3)),3)),183))
knoten2_prob <- c(rep(c(rep(c(rep(0.25,3), rep(0.5,3), rep(0.25,3)),3)),183))
knoten3_stunden <- c(rep(rep(c(283, 368, 380),9),183))
knoten3_prob <- c(rep(rep(c(0.25, 0.5, 0.25),9),183))
```

```
bis_knoten <- c(rep(0,4941))
decision_tree_100x300_1 <- data.frame(spalte1,bis_knoten, knoten1_stunden)
```

```
j <- 101
```

```
k <- 28
```

```
for(i in 2:183){
```

```
  decision_tree_100x300_1$spalte1[k] <- j
  decision_tree_100x300_1$spalte1[k+1] <- j
  decision_tree_100x300_1$spalte1[k+2] <- j
  decision_tree_100x300_1$spalte1[k+3] <- j
  decision_tree_100x300_1$spalte1[k+4] <- j
  decision_tree_100x300_1$spalte1[k+5] <- j
  decision_tree_100x300_1$spalte1[k+6] <- j
  decision_tree_100x300_1$spalte1[k+7] <- j
  decision_tree_100x300_1$spalte1[k+8] <- j
  decision_tree_100x300_1$spalte1[k+9] <- j
  decision_tree_100x300_1$spalte1[k+10] <- j
  decision_tree_100x300_1$spalte1[k+11] <- j
  decision_tree_100x300_1$spalte1[k+12] <- j
  decision_tree_100x300_1$spalte1[k+13] <- j
  decision_tree_100x300_1$spalte1[k+14] <- j
  decision_tree_100x300_1$spalte1[k+15] <- j
  decision_tree_100x300_1$spalte1[k+16] <- j
  decision_tree_100x300_1$spalte1[k+17] <- j
  decision_tree_100x300_1$spalte1[k+18] <- j
  decision_tree_100x300_1$spalte1[k+19] <- j
  decision_tree_100x300_1$spalte1[k+20] <- j
  decision_tree_100x300_1$spalte1[k+21] <- j
  decision_tree_100x300_1$spalte1[k+22] <- j
  decision_tree_100x300_1$spalte1[k+23] <- j
  decision_tree_100x300_1$spalte1[k+24] <- j
  decision_tree_100x300_1$spalte1[k+25] <- j
  decision_tree_100x300_1$spalte1[k+26] <- j
```

```
  j <- j+1
  k <- k+27
```

```
}
```

```
for(i in 1:length(decision_tree_100x300_1$bis_knoten)){
  decision_tree_100x300_1$bis_knoten[i] <- decision_tree_100x300_1$knoten1_stunden
  [i] - decision_tree_100x300_1$spalte1[i]
```

```

}

decision_tree_100x300_1 <- data.frame(decision_tree_100x300_1,knoten1_prob, knoten
2_stunden, knoten2_prob, knoten3_stunden, knoten3_prob )

decision_tree_100x300_1 <- decision_tree_100x300_1 %>%
  mutate(Wahrscheinlichkeit = knoten1_prob*knoten2_prob*knoten3_prob) %>%
  mutate(Summe = bis_knoten+knoten2_stunden+knoten3_stunden) %>%
  mutate(zeitpunkt_t1=720) %>%
  mutate(Diff = zeitpunkt_t1 - Summe) %>%
  mutate(Zustand = 0)

#schleife die die VTT Stunden zum Zeitpunkt 720h (14.12.17) ausrechnet
for(i in 1:length(decision_tree_100x300_1$Diff)){
  if(decision_tree_100x300_1$Diff[i] < 0){
    decision_tree_100x300_1$Zustand[i] <- decision_tree_100x300_1$zeitpunkt_t1[i]
- decision_tree_100x300_1$bis_knoten[i]- decision_tree_100x300_1$kn
oten2_stunden[i]
  } else {
    decision_tree_100x300_1$Zustand[i] <- decision_tree_100x300_1$Diff[i]
  }
}

#schleife die werte zu bereich zuordnet

for(i in 1:length(decision_tree_100x300_1$Zustand)){
  if(decision_tree_100x300_1$Zustand[i]<100){
    decision_tree_100x300_1$Bereich[i] <- "<100h"
  } else if (decision_tree_100x300_1$Zustand[i]>100 & decision_tree_100x300_1$Zust
and[i]<300 ) {
    decision_tree_100x300_1$Bereich[i] <- "100< x < 300"
  } else {
    decision_tree_100x300_1$Bereich[i] <- ">300"
  }
}

#hinterlegen der prozentuellen gewichtung(jeder Pfad mit prozentzahl mal 0,5%(200
Pfade))
decision_tree_100x300_1 <- decision_tree_100x300_1 %>%
  mutate(Anteil = Wahrscheinlichkeit*(1/183))

#wahrscheinlichkeiten am 14.12 wenn mit zustand 100<x<300 begonnen wird
t_1_100_t_100x300_1 <- sum(decision_tree_100x300_1$Anteil[which(decision_tree_100x
300_1$Bereich == "<100h")])
t_1_100x300_t_100x300_1 <- sum(decision_tree_100x300_1$Anteil[which(decision_tree_
100x300_1$Bereich == "100< x < 300")])
t_1_300_t_100x300_1 <- sum(decision_tree_100x300_1$Anteil[which(decision_tree_100x
300_1$Bereich == ">300")])

#####
#zweiter Baum start 283 bis 300

spalte1 <- c(rep(rep(283,18), 17))
knoten1_stunden <- rep(c(rep(368,9), rep(380,9)), 17)
knoten1_prob <- rep(c(rep(0.75,9), rep(0.25,9)), 17)
knoten2_stunden <- rep(c(rep(c(rep(283, 3), rep(368, 3), rep(380, 3)),2)),17)
knoten2_prob <- rep(c(rep(c(rep(0.25, 3), rep(0.5, 3), rep(0.25, 3)),2)),17)
knoten3_stunden <- rep(c(rep(c(283, 368, 380),6)),17)
knoten3_prob <- rep(c(rep(c(0.25, 0.5, 0.25),6)),17)

```

```

bis_knoten <- c(rep(0,306))
decision_tree_100x300_2 <- data.frame(spalte1,bis_knoten, knoten1_stunden)

j <- 284
k <- 19
for(i in 2:17){
  decision_tree_100x300_2$spalte1[k] <- j
  decision_tree_100x300_2$spalte1[k+1] <- j
  decision_tree_100x300_2$spalte1[k+2] <- j
  decision_tree_100x300_2$spalte1[k+3] <- j
  decision_tree_100x300_2$spalte1[k+4] <- j
  decision_tree_100x300_2$spalte1[k+5] <- j
  decision_tree_100x300_2$spalte1[k+6] <- j
  decision_tree_100x300_2$spalte1[k+7] <- j
  decision_tree_100x300_2$spalte1[k+8] <- j
  decision_tree_100x300_2$spalte1[k+9] <- j
  decision_tree_100x300_2$spalte1[k+10] <- j
  decision_tree_100x300_2$spalte1[k+11] <- j
  decision_tree_100x300_2$spalte1[k+12] <- j
  decision_tree_100x300_2$spalte1[k+13] <- j
  decision_tree_100x300_2$spalte1[k+14] <- j
  decision_tree_100x300_2$spalte1[k+15] <- j
  decision_tree_100x300_2$spalte1[k+16] <- j
  decision_tree_100x300_2$spalte1[k+17] <- j

  j <- j+1
  k <- k+18
}

for(i in 1:length(decision_tree_100x300_2$bis_knoten)){
  decision_tree_100x300_2$bis_knoten[i] <- decision_tree_100x300_2$knnoten1_stunden
[i]- decision_tree_100x300_2$spalte1[i]
}

decision_tree_100x300_2 <- data.frame(decision_tree_100x300_2, knoten1_prob, knoten
2_stunden, knoten2_prob, knoten3_stunden, knoten3_prob )

decision_tree_100x300_2 <- decision_tree_100x300_2 %>%
  mutate(Wahrscheinlichkeit = knoten1_prob*knoten2_prob*knoten3_prob) %>%
  mutate(Summe = bis_knoten+knnoten2_stunden+knnoten3_stunden) %>%
  mutate(zeitpunkt_t1=720) %>%
  mutate(Diff = zeitpunkt_t1 - Summe) %>%
  mutate(Zustand = 0)

#schleife die die VTT Stunden zum Zeitpunkt 720h (14.12.17) ausrechnet
for(i in 1:length(decision_tree_100x300_2$Diff)){
  if(decision_tree_100x300_2$Diff[i] < 0){
    decision_tree_100x300_2$Zustand[i] <- decision_tree_100x300_2$zeitpunkt_t1[i]
- decision_tree_100x300_2$bis_knoten[i]- decision_tree_100x300_2$knnoten2_stunden[i]
  } else {
    decision_tree_100x300_2$Zustand[i] <- decision_tree_100x300_2$Diff[i]
  }
}

#schleife die werte zu bereich zuordnet

for(i in 1:length(decision_tree_100x300_2$Zustand)){
  if(decision_tree_100x300_2$Zustand[i]<100){

```

```

    decision_tree_100x300_2$Bereich[i] <- "<100h"
  } else if (decision_tree_100x300_2$Zustand[i]>100 & decision_tree_100x300_2$Zust
and[i]<300 ) {
    decision_tree_100x300_2$Bereich[i] <- "100< x < 300"
  } else {
    decision_tree_100x300_2$Bereich[i] <- ">300"
  }
}

#hinterlegen der prozentuellen gewichtung(jeder Pfad mit prozentzahl mal 0,5%(200
Pfade))

decision_tree_100x300_2 <- decision_tree_100x300_2 %>%
  mutate(Anteil = Wahrscheinlichkeit*(1/17))

#wahrscheinlichkeiten am 14.12 wenn mit zustand 100<x<300 begonnen wird
t_1_100_t_100x300_2 <- sum(decision_tree_100x300_2$Anteil[which(decision_tree_100x
300_2$Bereich == "<100h")])
t_1_100x300_t_100x300_2 <- sum(decision_tree_100x300_2$Anteil[which(decision_tree_
100x300_2$Bereich == "100< x < 300")])
t_1_300_t_100x300_2 <- sum(decision_tree_100x300_2$Anteil[which(decision_tree_100x
300_2$Bereich == ">300")])

###
#für <100x<300 wird nun gewichteter mittelwert errechnet (da zwei bäume)

#bis 283h mit 91,5% (183/215) gewichtet danach mit 8,5%

t_1_100_t_100x300_1 <- sum(decision_tree_100x300_1$Anteil[which(decision_tree_100x
300_1$Bereich == "<100h")])*0.915
t_1_100x300_t_100x300_1 <- sum(decision_tree_100x300_1$Anteil[which(decision_tree_
100x300_1$Bereich == "100< x < 300")])*0.915
t_1_300_t_100x300_1 <- sum(decision_tree_100x300_1$Anteil[which(decision_tree_100x
300_1$Bereich == ">300")])*0.915

t_1_100_t_100x300_2 <- sum(decision_tree_100x300_2$Anteil[which(decision_tree_100x
300_2$Bereich == "<100h")])*0.085
t_1_100x300_t_100x300_2 <- sum(decision_tree_100x300_2$Anteil[which(decision_tree_
100x300_2$Bereich == "100< x < 300")])*0.085
t_1_300_t_100x300_2 <- sum(decision_tree_100x300_2$Anteil[which(decision_tree_100x
300_2$Bereich == ">300")])*0.085

t_1_100_t_100x300 <- sum(t_1_100_t_100x300_1, t_1_100_t_100x300_2)
t_1_100x300_t_100x300 <- sum(t_1_100x300_t_100x300_1, t_1_100x300_t_100x300_2)
t_1_300_t_100x300 <- sum(t_1_300_t_100x300_1, t_1_300_t_100x300_2)

```

8.1.9 Erstellung Entscheidungsbaum VTT RFH > 300h

```

#im zeitpunkt 1 können Logischerweise nur 368h und 380h auftreten
#nur möglich bis zeitpunkt 368h

```

```

spalte1 <- c(rep(rep(300,18), 68))
knoten1_stunden <- c(rep(c(rep(368,9), rep(380,9)),68))
knoten1_prob <- c(rep(c(rep(0.75,9), rep(0.25,9)),68))
knoten2_stunden <- c(rep(c(rep(c(rep(283,3), rep(368,3), rep(380,3)),2)),68))
knoten2_prob <- c(rep(c(rep(c(rep(0.25,3), rep(0.5,3), rep(0.25,3)),2)),68))
knoten3_stunden <- c(rep(rep(c(283, 368, 380),6),68))
knoten3_prob <- c(rep(rep(c(0.25, 0.5, 0.25),6),68))

```

```

bis_knoten <- c(rep(0,1224))
decision_tree_300 <- data.frame(spalte1,bis_knoten, knoten1_stunden)

j <- 301
k <- 19
for(i in 2:68){
  decision_tree_300$spalte1[k] <- j
  decision_tree_300$spalte1[k+1] <- j
  decision_tree_300$spalte1[k+2] <- j
  decision_tree_300$spalte1[k+3] <- j
  decision_tree_300$spalte1[k+4] <- j
  decision_tree_300$spalte1[k+5] <- j
  decision_tree_300$spalte1[k+6] <- j
  decision_tree_300$spalte1[k+7] <- j
  decision_tree_300$spalte1[k+8] <- j
  decision_tree_300$spalte1[k+9] <- j
  decision_tree_300$spalte1[k+10] <- j
  decision_tree_300$spalte1[k+11] <- j
  decision_tree_300$spalte1[k+12] <- j
  decision_tree_300$spalte1[k+13] <- j
  decision_tree_300$spalte1[k+14] <- j
  decision_tree_300$spalte1[k+15] <- j
  decision_tree_300$spalte1[k+16] <- j
  decision_tree_300$spalte1[k+17] <- j

  j <- j+1
  k <- k+18
}

for(i in 1:length(decision_tree_300$bis_knoten)){
  decision_tree_300$bis_knoten[i] <- decision_tree_300$knnoten1_stunden[i]- decisio
n_tree_300$spalte1[i]
}

decision_tree_300 <- data.frame(decision_tree_300,knoten1_prob, knoten2_stunden, k
noten2_prob, knoten3_stunden, knoten3_prob )

decision_tree_300 <- decision_tree_300 %>%
  mutate(Wahrscheinlichkeit = knoten1_prob*knoten2_prob*knoten3_prob) %>%
  mutate(Summe = bis_knoten+knnoten2_stunden+knnoten3_stunden) %>%
  mutate(zeitpunkt_t1=720) %>%
  mutate(Diff = zeitpunkt_t1 - Summe) %>%
  mutate(Zustand = 0)

#schleife die die VTT Stunden zum Zeitouunkt 720h (14.12.17) ausrechnet
for(i in 1:length(decision_tree_300$Diff)){
  if(decision_tree_300$Diff[i] < 0){
    decision_tree_300$Zustand[i] <- decision_tree_300$zeitpunkt_t1[i] - decision_t
ree_300$bis_knoten[i] - decision_tree_300$knnoten2_stunden[i]
  } else {
    decision_tree_300$Zustand[i] <- decision_tree_300$Diff[i]
  }
}
#schleife die werte zu bereich zuordnet

for(i in 1:length(decision_tree_300$Zustand)){
  if(decision_tree_300$Zustand[i]<100){
    decision_tree_300$Bereich[i] <- "<100h"
  } else if (decision_tree_300$Zustand[i]>100 & decision_tree_300$Zustand[i]<300 )
{
    decision_tree_300$Bereich[i] <- "100< x < 300"
  }
}

```

```

} else {
  decision_tree_300$Bereich[i] <- ">300"
}
}
#andere gewichtung weil nur 68h
decision_tree_300 <- decision_tree_300 %>%
  mutate(Anteil = Wahrscheinlichkeit*0.01470588)
#wahrscheinlichkeiten am 14.12 wenn mit zustand >300 begonnen wird
sum(decision_tree_300$Anteil[which(decision_tree_300$Bereich == "<100h")])

## [1] 0.3547794

sum(decision_tree_300$Anteil[which(decision_tree_300$Bereich == "100< x < 300")])

## [1] 0.254136

sum(decision_tree_300$Anteil[which(decision_tree_300$Bereich == ">300")])

## [1] 0.3910845

#zweiter baum für >368h
spalte1 <- c(rep(rep(368,9), 12))
knoten1_stunden <- c(rep(rep(380,9),12))
knoten1_prob <- c(rep(rep(1,9),12))
knoten2_stunden <- rep(c(rep(c(rep(283,3), rep(368,3), rep(380,3))), 12)
knoten2_prob <- rep(c(rep(c(rep(0.25,3), rep(0.5,3), rep(0.25,3))), 12)
knoten3_stunden <- rep(c(rep(c(283, 368, 380),3)),12)
knoten3_prob <- rep(c(rep(c(0.25, 0.50, 0.25),3)),12)

bis_knoten <- c(rep(0,108))
decision_tree_300_zwei <- data.frame(spalte1,bis_knoten, knoten1_stunden)

j <- 369
k <- 10
for(i in 2:12){
  decision_tree_300_zwei$spalte1[k] <- j
  decision_tree_300_zwei$spalte1[k+1] <- j
  decision_tree_300_zwei$spalte1[k+2] <- j
  decision_tree_300_zwei$spalte1[k+3] <- j
  decision_tree_300_zwei$spalte1[k+4] <- j
  decision_tree_300_zwei$spalte1[k+5] <- j
  decision_tree_300_zwei$spalte1[k+6] <- j
  decision_tree_300_zwei$spalte1[k+7] <- j
  decision_tree_300_zwei$spalte1[k+8] <- j

  j <- j+1
  k <- k+9
}

for(i in 1:length(decision_tree_300_zwei$bis_knoten)){
  decision_tree_300_zwei$bis_knoten[i] <- decision_tree_300_zwei$knoten1_stunden[i]
  decision_tree_300_zwei$spalte1[i]
}

decision_tree_300_zwei <- data.frame(decision_tree_300_zwei,knoten1_prob, knoten2_stunden, knoten2_prob, knoten3_stunden, knoten3_prob )

decision_tree_300_zwei <- decision_tree_300_zwei %>%
  mutate(Wahrscheinlichkeit = knoten1_prob*knoten2_prob*knoten3_prob) %>%
  mutate(Summe = bis_knoten+knoten2_stunden+knoten3_stunden) %>%
  mutate(zeitpunkt_t1=720) %>%

```

```

mutate(Diff = zeitpunkt_t1 - Summe) %>%
mutate(Zustand = 0)

#schleife die die VTT Stunden zum Zeitpunkt 720h (14.12.17) ausrechnet
for(i in 1:length(decision_tree_300_zwei$Diff)){
  if(decision_tree_300_zwei$Diff[i] < 0){
    decision_tree_300_zwei$Zustand[i] <- decision_tree_300_zwei$zeitpunkt_t1[i] -
decision_tree_300_zwei$bis_knoten[i] - decision_tree_300_zwei$knoten2_stunden[i]
  } else {
    decision_tree_300_zwei$Zustand[i] <- decision_tree_300_zwei$Diff[i]
  }
}
#schleife die werte zu bereich zuordnet
for(i in 1:length(decision_tree_300_zwei$Zustand)){
  if(decision_tree_300_zwei$Zustand[i]<100){
    decision_tree_300_zwei$Bereich[i] <- "<100h"
  } else if (decision_tree_300_zwei$Zustand[i]>100 & decision_tree_300_zwei$Zustand[i]<300 ) {
    decision_tree_300_zwei$Bereich[i] <- "100< x < 300"
  } else {
    decision_tree_300_zwei$Bereich[i] <- ">300"
  }
}
#andere gewichtung da nur 12 steps
decision_tree_300_zwei <- decision_tree_300_zwei %>%
  mutate(Anteil = Wahrscheinlichkeit*0.08333333)
#wahrscheinlichkeiten am 14.12 wenn mit zustand >300 begonnen wird
sum(decision_tree_300_zwei$Anteil[which(decision_tree_300_zwei$Bereich == "<100h")
])
## [1] 0.375

sum(decision_tree_300_zwei$Anteil[which(decision_tree_300_zwei$Bereich == "100< x
< 300")])
## [1] 0.0625

sum(decision_tree_300_zwei$Anteil[which(decision_tree_300_zwei$Bereich == ">300")
])
## [1] 0.5625

###
#für >300 wird nun gewichteter mittelwert errechnet (da zwei bäume)
#bis 368h mit 85% gewichtet danach mit 15%
t_1_100_t_300_1 <- sum(decision_tree_300$Anteil[which(decision_tree_300$Bereich ==
"<100h")])*0.85
t_1_100x300_t_300_1 <- sum(decision_tree_300$Anteil[which(decision_tree_300$Bereic
h == "100< x < 300")])*0.85
t_1_300_t_300_1 <- sum(decision_tree_300$Anteil[which(decision_tree_300$Bereich ==
">300")])*0.85
t_1_100_t_300_1
## [1] 0.3015625

t_1_100x300_t_300_1
## [1] 0.2160156

t_1_300_t_300_1
## [1] 0.3324218

```

```

t_1_100_t_300_2 <- sum(decision_tree_300$Anteil[which(decision_tree_300$Bereich ==
"<100h")]) * 0.15
t_1_100x300_t_300_2 <- sum(decision_tree_300$Anteil[which(decision_tree_300$Bereic
h == "100< x < 300")]) * 0.15
t_1_300_t_300_2 <- sum(decision_tree_300$Anteil[which(decision_tree_300$Bereich ==
">300")]) * 0.15

t_1_100_t_300 <- sum(c(t_1_100_t_300_1, t_1_100_t_300_2))
t_1_100x300_t_300 <- sum(c(t_1_100x300_t_300_1, t_1_100x300_t_300_2))
t_1_300_t_300 <- sum(c(t_1_300_t_300_1, t_1_300_t_300_2))
t_1_100_t_300

## [1] 0.3547794

t_1_100x300_t_300

## [1] 0.254136

t_1_300_t_300

## [1] 0.3910845

```

8.1.10 Erstellung Entscheidungsbaum ESC_RFH < 1000h

#baum für <1000h

```

spalte1_esc <- rep(rep(0,9),1000)
knoten1_stunden_esc <- rep(c(rep(1486,3), rep(2614,3), rep(4485,3)),1000)
knoten1_prob_esc <- rep(c(rep(0.25,3), rep(0.5,3), rep(0.25,3)),1000)
knoten2_stunden_esc <- rep(c(rep(c(1486, 2614, 4485),3)),1000)
knoten2_prob_esc <- rep(c(rep(c(0.25, 0.50, 0.25),3)),1000)

bis_knoten_esc <- c(rep(0,9000))
decision_tree_esc_1000 <- data.frame(spalte1_esc,bis_knoten_esc, knoten1_stunden_e
sc)
j <- 1
k <- 10
for(i in 2:1000){
  decision_tree_esc_1000$spalte1_esc[k] <- j
  decision_tree_esc_1000$spalte1_esc[k+1] <- j
  decision_tree_esc_1000$spalte1_esc[k+2] <- j
  decision_tree_esc_1000$spalte1_esc[k+3] <- j
  decision_tree_esc_1000$spalte1_esc[k+4] <- j
  decision_tree_esc_1000$spalte1_esc[k+5] <- j
  decision_tree_esc_1000$spalte1_esc[k+6] <- j
  decision_tree_esc_1000$spalte1_esc[k+7] <- j
  decision_tree_esc_1000$spalte1_esc[k+8] <- j
  j <- j+1
  k <- k+9
}
for(i in 1:length(decision_tree_esc_1000$bis_knoten_esc)){
  decision_tree_esc_1000$bis_knoten_esc[i] <- decision_tree_esc_1000$knoten1_stund
en_esc[i]-decision_tree_esc_1000$spalte1_esc[i]
}

decision_tree_esc_1000 <- data.frame(decision_tree_esc_1000, knoten1_prob_esc, kno
ten2_stunden_esc, knoten2_prob_esc)

decision_tree_esc_1000 <- decision_tree_esc_1000 %>%
  mutate(Wahrscheinlichkeit = knoten1_prob_esc*knoten2_prob_esc) %>%

```

```

mutate(Summe = bis_knoten_esc+knoten2_stunden_esc) %>%
mutate(zeitpunkt_t1=720) %>%
mutate(Diff = zeitpunkt_t1 - Summe) %>%
mutate(Zustand = 0)

#schleife die die ESC Stunden zum Zeitpunkt 720h (14.12.17) ausrechnet
for(i in 1:length(decision_tree_esc_1000$Diff)){
  if((decision_tree_esc_1000$spalte1_esc[i]+decision_tree_esc_1000$zeitpunkt_t1[i]
) < 1486){
    decision_tree_esc_1000$Zustand[i] <- decision_tree_esc_1000$spalte1_esc[i]+720
  } else if((decision_tree_esc_1000$spalte1_esc[i]+decision_tree_esc_1000$zeitpunkt_t1[i]) > 1486){
    if(decision_tree_esc_1000$knoten1_stunden_esc[i]==1486){
      decision_tree_esc_1000$Zustand[i] <- decision_tree_esc_1000$spalte1_esc[i]
+decision_tree_esc_1000$zeitpunkt_t1[i]-1486
    } else {
      decision_tree_esc_1000$Zustand[i] <- decision_tree_esc_1000$spalte1_esc[i]
+720
    }
  }
}

#schleife die werte zu bereich zuordnet

for(i in 1:length(decision_tree_esc_1000$Zustand)){
  if(decision_tree_esc_1000$Zustand[i]<1000){
    decision_tree_esc_1000$Bereich[i] <- "<1000h"
  } else if (decision_tree_esc_1000$Zustand[i]>1000 & decision_tree_esc_1000$Zustand[i]<2500 ) {
    decision_tree_esc_1000$Bereich[i] <- "1000< x < 2500"
  } else {
    decision_tree_esc_1000$Bereich[i] <- ">2500"
  }
}

#hinterlegen der prozentuellen gewichtung(jeder Pfad mit prozentzahl mal 1%(100 Pfade))

decision_tree_esc_1000 <- decision_tree_esc_1000 %>%
  mutate(Anteil = Wahrscheinlichkeit*0.001)

#wahrscheinlichkeiten am 14.12 wenn mit zustand <100 begonnen wird
t_1_1000_t_1000_esc <- sum(decision_tree_esc_1000$Anteil[which(decision_tree_esc_1000$Bereich == "<1000h")])
t_1_1000x2500_t_1000_esc <- sum(decision_tree_esc_1000$Anteil[which(decision_tree_esc_1000$Bereich == "1000< x < 2500")])
t_1_2500_t_1000_esc <- sum(decision_tree_esc_1000$Anteil[which(decision_tree_esc_1000$Bereich == ">2500")])

```

8.1.11 Erstellung Entscheidungsbaum ESC_RFH 1000h < x < 2500h

```

#baum für 1000h<x<2500h nur bis 1486 möglich danach nur noch zustand 2614 oder 2500 möglich
spalte1_esc <- rep(rep(1000,9),486)
knoten1_stunden_esc <- rep(c(rep(1486,3), rep(2614,3), rep(4485,3)),486)
knoten1_prob_esc <- rep(c(rep(0.25,3), rep(0.5,3), rep(0.25,3)),486)
knoten2_stunden_esc <- rep(c(rep(c(1486, 2614, 4485),3)),486)

```

```

knoten2_prob_esc <- rep(c(rep(c(0.25, 0.50, 0.25),3)),486)

bis_knoten_esc <- c(rep(0,4374))
decision_tree_esc_1000x2500_1 <- data.frame(spalte1_esc,bis_knoten_esc, knoten1_st
unden_esc)

j <- 1001
k <- 10
for(i in 2:486){
  decision_tree_esc_1000x2500_1$spalte1_esc[k] <- j
  decision_tree_esc_1000x2500_1$spalte1_esc[k+1] <- j
  decision_tree_esc_1000x2500_1$spalte1_esc[k+2] <- j
  decision_tree_esc_1000x2500_1$spalte1_esc[k+3] <- j
  decision_tree_esc_1000x2500_1$spalte1_esc[k+4] <- j
  decision_tree_esc_1000x2500_1$spalte1_esc[k+5] <- j
  decision_tree_esc_1000x2500_1$spalte1_esc[k+6] <- j
  decision_tree_esc_1000x2500_1$spalte1_esc[k+7] <- j
  decision_tree_esc_1000x2500_1$spalte1_esc[k+8] <- j

  j <- j+1
  k <- k+9
}

for(i in 1:length(decision_tree_esc_1000x2500_1$bis_knoten_esc)){
  decision_tree_esc_1000x2500_1$bis_knoten_esc[i] <- decision_tree_esc_1000x2500_1
$knoten1_stunden_esc[i]-decision_tree_esc_1000x2500_1$spalte1_esc[i]
}

decision_tree_esc_1000x2500_1 <- data.frame(decision_tree_esc_1000x2500_1, knoten1
_prob_esc, knoten2_stunden_esc, knoten2_prob_esc)

decision_tree_esc_1000x2500_1 <- decision_tree_esc_1000x2500_1 %>%
  mutate(Wahrscheinlichkeit = knoten1_prob_esc*knoten2_prob_esc) %>%
  mutate(Summe = bis_knoten_esc+knoten2_stunden_esc) %>%
  mutate(zeitpunkt_t1=720) %>%
  mutate(Diff = zeitpunkt_t1 - Summe) %>%
  mutate(Zustand = 0)

#schleife die die ESC Stunden zum Zeitouunkt 720h (14.12.17) ausrechnet
for(i in 1:length(decision_tree_esc_1000x2500_1$Diff)){
  if((decision_tree_esc_1000x2500_1$spalte1_esc[i]+decision_tree_esc_1000x2500_1$z
eitpunkt_t1[i]) < 1486){
    decision_tree_esc_1000x2500_1$Zustand[i] <- decision_tree_esc_1000x2500_1$spal
te1_esc[i]+720
  } else if((decision_tree_esc_1000x2500_1$spalte1_esc[i]+decision_tree_esc_1000x2
500_1$zeitpunkt_t1[i]) > 1486){
    if(decision_tree_esc_1000x2500_1$knoten1_stunden_esc[i]==1486){
      decision_tree_esc_1000x2500_1$Zustand[i] <- decision_tree_esc_1000x2500_1$sp
alte1_esc[i]+decision_tree_esc_1000x2500_1$zeitpunkt_t1[i]-1486
    } else {
      decision_tree_esc_1000x2500_1$Zustand[i] <- decision_tree_esc_1000x2500_1$sp
alte1_esc[i]+720
    }
  }
}

#schleife die werte zu bereich zuordnet
for(i in 1:length(decision_tree_esc_1000x2500_1$Zustand)){
  if(decision_tree_esc_1000x2500_1$Zustand[i]<1000){

```

```

    decision_tree_esc_1000x2500_1$Bereich[i] <- "<1000h"
  } else if (decision_tree_esc_1000x2500_1$Zustand[i]>1000 & decision_tree_esc_1000x2500_1$Zustand[i]<2500 ) {
    decision_tree_esc_1000x2500_1$Bereich[i] <- "1000< x < 2500"
  } else {
    decision_tree_esc_1000x2500_1$Bereich[i] <- ">2500"
  }
}

#hinterlegen der prozentuellen gewichtung(jeder Pfad mit prozentzahl mal 1%(100 Pfade))
decision_tree_esc_1000x2500_1 <- decision_tree_esc_1000x2500_1 %>%
  mutate(Anteil = Wahrscheinlichkeit*(1/486))

#wahrscheinlichkeiten am 14.12 wenn mit zustand 1000<x<2500 begonnen wird (erster Teil des Baumes zweiter folgt unten)
t_1_1000_t_1000x2500_esc_1 <- sum(decision_tree_esc_1000x2500_1$Anteil[which(decision_tree_esc_1000x2500_1$Bereich == "<1000h")])
t_1_1000x2500_t_1000x2500_esc_1 <- sum(decision_tree_esc_1000x2500_1$Anteil[which(decision_tree_esc_1000x2500_1$Bereich == "1000< x < 2500")])
t_1_2500_t_1000x2500_esc_1 <- sum(decision_tree_esc_1000x2500_1$Anteil[which(decision_tree_esc_1000x2500_1$Bereich == ">2500")])
#####
#zweiter Baum für 1000<x<2500 ab 1486
spalte1_esc <- rep(rep(1487,6),1014)
knoten1_stunden_esc <- rep(c(rep(2614,3), rep(4485,3)),1014)
knoten1_prob_esc <- rep(c(rep(0.75,3), rep(0.25,3)),1014)
knoten2_stunden_esc <- rep(c(rep(c(1486, 2614, 4485),2)),1014)
knoten2_prob_esc <- rep(c(rep(c(0.25, 0.50, 0.25),2)),1014)
bis_knoten_esc <- c(rep(0,6084))
decision_tree_esc_1000x2500_2 <- data.frame(spalte1_esc,bis_knoten_esc, knoten1_stunden_esc)

j <- 1487
k <- 7
for(i in 2:1014){
  decision_tree_esc_1000x2500_2$spalte1_esc[k] <- j
  decision_tree_esc_1000x2500_2$spalte1_esc[k+1] <- j
  decision_tree_esc_1000x2500_2$spalte1_esc[k+2] <- j
  decision_tree_esc_1000x2500_2$spalte1_esc[k+3] <- j
  decision_tree_esc_1000x2500_2$spalte1_esc[k+4] <- j
  decision_tree_esc_1000x2500_2$spalte1_esc[k+5] <- j

  j <- j+1
  k <- k+6
}

for(i in 1:length(decision_tree_esc_1000x2500_2$bis_knoten_esc)){
  decision_tree_esc_1000x2500_2$bis_knoten_esc[i] <- decision_tree_esc_1000x2500_2$knoten1_stunden_esc[i]-decision_tree_esc_1000x2500_2$spalte1_esc[i]
}

decision_tree_esc_1000x2500_2 <- data.frame(decision_tree_esc_1000x2500_2, knoten1_prob_esc, knoten2_stunden_esc, knoten2_prob_esc)

decision_tree_esc_1000x2500_2 <- decision_tree_esc_1000x2500_2 %>%
  mutate(Wahrscheinlichkeit = knoten1_prob_esc*knoten2_prob_esc) %>%
  mutate(Summe = bis_knoten_esc+knoten2_stunden_esc) %>%
  mutate(zeitpunkt_t1=720) %>%
  mutate(Diff = zeitpunkt_t1 - Summe) %>%
  mutate(Zustand = 0)

```

```

#schleife die die ESC Stunden zum Zeitpunkt 720h (14.12.17) ausrechnet
for(i in 1:length(decision_tree_esc_1000x2500_2$Zustand)){
  if(decision_tree_esc_1000x2500_2$bis_knoten_esc[i]<720){
    decision_tree_esc_1000x2500_2$Zustand[i] <- 720-decision_tree_esc_1000x2500_2$
bis_knoten_esc[i]
  }else{
    decision_tree_esc_1000x2500_2$Zustand[i] <- decision_tree_esc_1000x2500_2$spal
te1_esc[i]+720
  }
}
#schleife die werte zu bereich zuordnet
for(i in 1:length(decision_tree_esc_1000x2500_2$Zustand)){
  if(decision_tree_esc_1000x2500_2$Zustand[i]<1000){
    decision_tree_esc_1000x2500_2$Bereich[i] <- "<1000h"
  } else if (decision_tree_esc_1000x2500_2$Zustand[i]>1000 & decision_tree_esc_100
0x2500_2$Zustand[i]<2500 ) {
    decision_tree_esc_1000x2500_2$Bereich[i] <- "1000< x < 2500"
  } else {
    decision_tree_esc_1000x2500_2$Bereich[i] <- ">2500"
  }
}
#hinterlegen der prozentuellen gewichtung(jeder Pfad mit prozentzahl mal 1%(100 Pf
ade))
decision_tree_esc_1000x2500_2 <- decision_tree_esc_1000x2500_2 %>%
  mutate(Anteil = Wahrscheinlichkeit*(1/1014))
#wahrscheinlichkeiten am 14.12 wenn mit zustand 1000<x<2500 begonnen wird (erster
Teil des Baumes zweiter folgt unten)
t_1_1000_t_1000x2500_esc_2 <- sum(decision_tree_esc_1000x2500_2$Anteil[which(decis
ion_tree_esc_1000x2500_2$Bereich == "<1000h")])
t_1_1000x2500_t_1000x2500_esc_2 <- sum(decision_tree_esc_1000x2500_2$Anteil[which(
decision_tree_esc_1000x2500_2$Bereich == "1000< x < 2500")])
t_1_2500_t_1000x2500_esc_2 <- sum(decision_tree_esc_1000x2500_2$Anteil[which(decis
ion_tree_esc_1000x2500_2$Bereich == ">2500")])
#Zusammenführen der beiden Bäume (erster mit 32,4% gewichtet zweiter mit 67,6%)
t_1_1000_t_1000x2500_esc_1 <- sum(decision_tree_esc_1000x2500_1$Anteil[which(decis
ion_tree_esc_1000x2500_1$Bereich == "<1000h")])*0.324
t_1_1000x2500_t_1000x2500_esc_1 <- sum(decision_tree_esc_1000x2500_1$Anteil[which(
decision_tree_esc_1000x2500_1$Bereich == "1000< x < 2500")])*0.324
t_1_2500_t_1000x2500_esc_1 <- sum(decision_tree_esc_1000x2500_1$Anteil[which(decis
ion_tree_esc_1000x2500_1$Bereich == ">2500")])*0.324

t_1_1000_t_1000x2500_esc_2 <- sum(decision_tree_esc_1000x2500_2$Anteil[which(decis
ion_tree_esc_1000x2500_2$Bereich == "<1000h")])*0.676
t_1_1000x2500_t_1000x2500_esc_2 <- sum(decision_tree_esc_1000x2500_2$Anteil[which(
decision_tree_esc_1000x2500_2$Bereich == "1000< x < 2500")])*0.676
t_1_2500_t_1000x2500_esc_2 <- sum(decision_tree_esc_1000x2500_2$Anteil[which(decis
ion_tree_esc_1000x2500_2$Bereich == ">2500")])*0.676

t_1_1000_t_1000x2500_esc <- sum(t_1_1000_t_1000x2500_esc_1, t_1_1000_t_1000x2500_e
sc_2)
t_1_1000x2500_t_1000x2500_esc <- sum(t_1_1000x2500_t_1000x2500_esc_1, t_1_1000x250
0_t_1000x2500_esc_2)
t_1_2500_t_1000x2500_esc <- sum(t_1_2500_t_1000x2500_esc_1, t_1_2500_t_1000x2500_e
sc_2)

```

8.1.12 Erstellung Entscheidungsbaum ESC_RFH > 2500h

#baum für >2500 wieder abbruch nach 2614 (danach im ersten knoten wieder nur 4485 wartung möglich)

```

spalte1_esc <- rep(rep(2500,6),114)
knoten1_stunden_esc <- rep(c(rep(2614,3), rep(4485,3)),114)
knoten1_prob_esc <- rep(c(rep(0.75,3), rep(0.25,3)),114)
knoten2_stunden_esc <- rep(c(rep(c(1486, 2614, 4485),2)),114)
knoten2_prob_esc <- rep(c(rep(c(0.25, 0.50, 0.25),2)),114)
bis_knoten_esc <- c(rep(0,684))
decision_tree_esc_2500_1 <- data.frame(spalte1_esc,bis_knoten_esc,
knoten1_stunden_esc)
j <- 2501
k <- 7
for(i in 2:114){
  decision_tree_esc_2500_1$spalte1_esc[k] <- j
  decision_tree_esc_2500_1$spalte1_esc[k+1] <- j
  decision_tree_esc_2500_1$spalte1_esc[k+2] <- j
  decision_tree_esc_2500_1$spalte1_esc[k+3] <- j
  decision_tree_esc_2500_1$spalte1_esc[k+4] <- j
  decision_tree_esc_2500_1$spalte1_esc[k+5] <- j

  j <- j+1
  k <- k+6
}
for(i in 1:length(decision_tree_esc_2500_1$bis_knoten_esc)){
  decision_tree_esc_2500_1$bis_knoten_esc[i] <-
decision_tree_esc_2500_1$knoten1_stunden_esc[i]-
decision_tree_esc_2500_1$spalte1_esc[i]
}

decision_tree_esc_2500_1 <- data.frame(decision_tree_esc_2500_1, knoten1_prob_esc,
knoten2_stunden_esc, knoten2_prob_esc)
decision_tree_esc_2500_1 <- decision_tree_esc_2500_1 %>%
  mutate(Wahrscheinlichkeit = knoten1_prob_esc*knoten2_prob_esc) %>%
  mutate(Summe = bis_knoten_esc+knoten2_stunden_esc) %>%
  mutate(zeitpunkt_t1=720) %>%
  mutate(Diff = zeitpunkt_t1 - Summe) %>%
  mutate(Zustand = 0)

#schleife die die ESC Stunden zum Zeitpunkt 720h (14.12.17) ausrechnet
for(i in 1:length(decision_tree_esc_2500_1$Zustand)){
  if(decision_tree_esc_2500_1$bis_knoten_esc[i]<720){
    decision_tree_esc_2500_1$Zustand[i] <- 720-
decision_tree_esc_2500_1$bis_knoten_esc[i]
  }else{
    decision_tree_esc_2500_1$Zustand[i] <-
decision_tree_esc_2500_1$spalte1_esc[i]+720
  }
}

```

```

}

#schleife die werte zu bereich zuordnet

for(i in 1:length(decision_tree_esc_2500_1$Zustand)){
  if(decision_tree_esc_2500_1$Zustand[i]<1000){
    decision_tree_esc_2500_1$Bereich[i] <- "<1000h"
  } else if (decision_tree_esc_2500_1$Zustand[i]>1000 &
decision_tree_esc_2500_1$Zustand[i]<2500 ) {
    decision_tree_esc_2500_1$Bereich[i] <- "1000< x < 2500"
  } else {
    decision_tree_esc_2500_1$Bereich[i] <- ">2500"
  }
}

#hinterlegen der prozentuellen gewichtung(jeder Pfad mit prozentzahl mal 1%(100
Pfade))
decision_tree_esc_2500_1 <- decision_tree_esc_2500_1 %>%
  mutate(Anteil = Wahrscheinlichkeit*(1/114))
#wahrscheinlichkeiten am 14.12 wenn mit zustand 1000<x<2500 begonnen wird (erster
Teil des Baumes zweiter folgt unten)
t_1_1000_t_2500_esc_1 <-
sum(decision_tree_esc_2500_1$Anteil[which(decision_tree_esc_2500_1$Bereich ==
"<1000h")])
t_1_1000x2500_t_2500_esc_1 <-
sum(decision_tree_esc_2500_1$Anteil[which(decision_tree_esc_2500_1$Bereich ==
"1000< x < 2500")])
t_1_2500_t_2500_esc_1 <-
sum(decision_tree_esc_2500_1$Anteil[which(decision_tree_esc_2500_1$Bereich ==
">2500")])
#####
#zweiter Baum für >2500 ab 2614
spalte1_esc <- rep(rep(2614,3),1872)
knoten1_stunden_esc <- rep(rep(4485,3),1872)
knoten1_prob_esc <- rep(rep(1,3),1872)
knoten2_stunden_esc <- rep(c(1486, 2614, 4485),1872)
knoten2_prob_esc <- rep(c(0.25, 0.50, 0.25),1872)

bis_knoten_esc <- c(rep(0,5616))
decision_tree_esc_2500_2 <- data.frame(spalte1_esc,bis_knoten_esc,
knoten1_stunden_esc)

j <- 2615
k <- 4
for(i in 2:1872){
  decision_tree_esc_2500_2$spalte1_esc[k] <- j
  decision_tree_esc_2500_2$spalte1_esc[k+1] <- j
  decision_tree_esc_2500_2$spalte1_esc[k+2] <- j

  j <- j+1
}

```

```

k <- k+3
}

for(i in 1:length(decision_tree_esc_2500_2$bis_knoten_esc)){
  decision_tree_esc_2500_2$bis_knoten_esc[i] <-
decision_tree_esc_2500_2$knoten1_stunden_esc[i]-
decision_tree_esc_2500_2$spalte1_esc[i]
}

decision_tree_esc_2500_2 <- data.frame(decision_tree_esc_2500_2, knoten1_prob_esc,
knoten2_stunden_esc, knoten2_prob_esc)

decision_tree_esc_2500_2 <- decision_tree_esc_2500_2 %>%
  mutate(Wahrscheinlichkeit = knoten1_prob_esc*knoten2_prob_esc) %>%
  mutate(Summe = bis_knoten_esc+knoten2_stunden_esc) %>%
  mutate(zeitpunkt_t1=720) %>%
  mutate(Diff = zeitpunkt_t1 - Summe) %>%
  mutate(Zustand = 0)

#schleife die die ESC Stunden zum Zeitpunkt 720h (14.12.17) ausrechnet

for(i in 1:length(decision_tree_esc_2500_2$Zustand)){
  if(decision_tree_esc_2500_2$bis_knoten_esc[i]<720){
    decision_tree_esc_2500_2$Zustand[i] <- 720-
decision_tree_esc_2500_2$bis_knoten_esc[i]
  }else{
    decision_tree_esc_2500_2$Zustand[i] <-
decision_tree_esc_2500_2$spalte1_esc[i]+720
  }
}

#schleife die werte zu bereich zuordnet
for(i in 1:length(decision_tree_esc_2500_2$Zustand)){
  if(decision_tree_esc_2500_2$Zustand[i]<1000){
    decision_tree_esc_2500_2$Bereich[i] <- "<1000h"
  } else if (decision_tree_esc_2500_2$Zustand[i]>1000 &
decision_tree_esc_2500_2$Zustand[i]<2500 ) {
    decision_tree_esc_2500_2$Bereich[i] <- "1000< x < 2500"
  } else {
    decision_tree_esc_2500_2$Bereich[i] <- ">2500"
  }
}

#hinterlegen der prozentuellen gewichtung(jeder Pfad mit prozentzahl mal 1%(100
Pfade))

decision_tree_esc_2500_2 <- decision_tree_esc_2500_2 %>%
  mutate(Anteil = Wahrscheinlichkeit*(1/1872))
#wahrscheinlichkeiten am 14.12 wenn mit zustand 1000<x<2500 begonnen wird (erster

```

Teil des Baumes zweiter folgt unten)

```
t_1_1000_t_2500_esc_2 <-
sum(decision_tree_esc_2500_2$Anteil[which(decision_tree_esc_2500_2$Bereich ==
"<1000h")])
t_1_1000x2500_t_2500_esc_2 <-
sum(decision_tree_esc_2500_2$Anteil[which(decision_tree_esc_2500_2$Bereich ==
"1000< x < 2500")])
t_1_2500_t_2500_esc_2 <-
sum(decision_tree_esc_2500_2$Anteil[which(decision_tree_esc_2500_2$Bereich ==
">2500")])
#Zusammenführen der beiden Bäume (erster mit 5,74% gewichtet zweiter mit 94,26%)
t_1_1000_t_2500_esc_1 <-
sum(decision_tree_esc_2500_1$Anteil[which(decision_tree_esc_2500_1$Bereich ==
"<1000h")])*0.05743073
t_1_1000x2500_t_2500_esc_1 <-
sum(decision_tree_esc_2500_1$Anteil[which(decision_tree_esc_2500_1$Bereich ==
"1000< x < 2500")])*0.05743073
t_1_2500_t_2500_esc_1 <-
sum(decision_tree_esc_2500_1$Anteil[which(decision_tree_esc_2500_1$Bereich ==
">2500")])*0.05743073
t_1_1000_t_2500_esc_2 <-
sum(decision_tree_esc_2500_2$Anteil[which(decision_tree_esc_2500_2$Bereich ==
"<1000h")])*0.9425693
t_1_1000x2500_t_2500_esc_2 <-
sum(decision_tree_esc_2500_2$Anteil[which(decision_tree_esc_2500_2$Bereich ==
"1000< x < 2500")])*0.9425693
t_1_2500_t_2500_esc_2 <-
sum(decision_tree_esc_2500_2$Anteil[which(decision_tree_esc_2500_2$Bereich ==
">2500")])*0.9425693
t_1_1000_t_2500_esc <- sum(t_1_1000_t_2500_esc_1, t_1_1000_t_2500_esc_2)
t_1_1000x2500_t_2500_esc <- sum(t_1_1000x2500_t_2500_esc_1,
t_1_1000x2500_t_2500_esc_2)
t_1_2500_t_2500_esc <- sum(t_1_2500_t_2500_esc_1, t_1_2500_t_2500_esc_2)
```

8.1.13 Erstellung und fitten des Bayes'Netz

```
library(bnlearn)
library(readxl)
chuck_power_unit_2 <-
model2network("[RFH][VTT][Biasvoltage_Alarm|RFH:VTT][Biasvoltage_LV|RFH:VTT][Clamp
voltage_LV|RFH:VTT][Clampvoltage_Alarm|RFH:VTT][RF_POWER_LV|RFH:VTT][RF_POWER_Alar
m|RFH:VTT][POWER_UNIT|Biasvoltage_LV:Biasvoltage_Alarm:Clampvoltage_LV:Clampvoltag
e_Alarm:RF_POWER_LV:RF_POWER_Alarm]")

graphviz.plot(chuck_power_unit_2, layout = "dot")
formatierung_modul <- list(nodes = "POWER_UNIT", col = "black", fill = "grey")
hl <- list(arcs = vstructs(chuck_power_unit_2, arcs = TRUE), lwd = 1, col =
"black", lty = "solid")
pp <- graphviz.plot(chuck_power_unit_2, highlight = formatierung_modul, shape =
```

```

"circle")
vtt=c("<100h", "100< x < 200", ">300")
abnutzung_esc=c("<1000h", "1000 < x < 2500h ", ">2500h")
auftreten = c("TRUE", "FALSE")
TTF_RANGE= c("TTF<12", "12<TTF<24", "24<TTF<36", "TTF>36")
#-----
-----
VTT.cpt <- array(c(1/3,1/3,1/3), dim = 3, dimnames = list(VTT = vtt))
esc_rfh.cpt <- array(c(1/3,1/3,1/3), dim = 3, dimnames = list(esc_rfh =
abnutzung_esc))
#-----
-----
#2 EBENE
aufschlag_vtt <-2
aufschlag_esc_rfh <-3
Biasvoltage_Alarm.cpt <- array(c(cpt_alarm_bias_true, cpt_alarm_bias_false,
#ESC_RFH<1000h VTT<100h
cpt_alarm_bias_true*(aufschlag_vtt*1+aufschlag_esc_rfh*0), 1-
cpt_alarm_bias_true*(aufschlag_vtt*1+aufschlag_esc_rfh*0), #ESC_RFH<1000h VTT
100<x<300h
cpt_alarm_bias_true*(aufschlag_vtt*2+aufschlag_esc_rfh*0), 1-
cpt_alarm_bias_true*(aufschlag_vtt*2+aufschlag_esc_rfh*0), #ESC RFH <1000 VTT
>300h
cpt_alarm_bias_true*(aufschlag_vtt*0+aufschlag_esc_rfh*1), 1-
cpt_alarm_bias_true*(aufschlag_vtt*0+aufschlag_esc_rfh*1), #ESC_RFH 1000h<x<2500
VTT<100h
cpt_alarm_bias_true*(aufschlag_vtt*1+aufschlag_esc_rfh*1), 1-
cpt_alarm_bias_true*(aufschlag_vtt*1+aufschlag_esc_rfh*1), #ESC_RFH 1000h<x<2500
VTT 100<x<300h
cpt_alarm_bias_true*(aufschlag_vtt*2+aufschlag_esc_rfh*1), 1-
cpt_alarm_bias_true*(aufschlag_vtt*2+aufschlag_esc_rfh*1), #ESC_RFH 1000h<x<2500
VTT >300h
cpt_alarm_bias_true*(aufschlag_vtt*0+aufschlag_esc_rfh*2), 1-
cpt_alarm_bias_true*(aufschlag_vtt*0+aufschlag_esc_rfh*2), #ESC_RFH >3000h
VTT<100h
cpt_alarm_bias_true*(aufschlag_vtt*1+aufschlag_esc_rfh*2), 1-
cpt_alarm_bias_true*(aufschlag_vtt*1+aufschlag_esc_rfh*2), #ESC_RFH >3000h VTT
100<x<300h
cpt_alarm_bias_true*(aufschlag_vtt*2+aufschlag_esc_rfh*1.5), 1-
cpt_alarm_bias_true*(aufschlag_vtt*2+aufschlag_esc_rfh*1.5)), #ESC_RFH >3000h VTT
>300h
                                     dim = c(2, 3, 3), dimnames = list(Biasvoltage_Alarm
= auftreten, VTT = vtt, RFH = abnutzung_esc))

Biasvoltage_LV.cpt <- array(c(cpt_lv_bias_true, cpt_lv_bias_false, #ESC_RFH<1000h
VTT<100h
cpt_lv_bias_true*(aufschlag_vtt*1+aufschlag_esc_rfh*0), 1-
cpt_lv_bias_true*(aufschlag_vtt*1+aufschlag_esc_rfh*0), #ESC_RFH<1000h VTT
100<x<300h

```

```

cpt_lv_bias_true*(aufschlag_vtt*2+aufschlag_esc_rfh*0), 1-
cpt_lv_bias_true*(aufschlag_vtt*2+aufschlag_esc_rfh*0), #ESC_RFH <1000 VTT >300h
cpt_lv_bias_true*(aufschlag_vtt*0+aufschlag_esc_rfh*1), 1-
cpt_lv_bias_true*(aufschlag_vtt*0+aufschlag_esc_rfh*1), #ESC_RFH 1000h<x<2500
VTT<100h
cpt_lv_bias_true*(aufschlag_vtt*1+aufschlag_esc_rfh*1), 1-
cpt_lv_bias_true*(aufschlag_vtt*1+aufschlag_esc_rfh*1), #ESC_RFH 1000h<x<2500 VTT
100<x<300h
cpt_lv_bias_true*(aufschlag_vtt*2+aufschlag_esc_rfh*1), 1-
cpt_lv_bias_true*(aufschlag_vtt*2+aufschlag_esc_rfh*1), #ESC_RFH 1000h<x<2500 VTT
>300h
cpt_lv_bias_true*(aufschlag_vtt*0+aufschlag_esc_rfh*2), 1-
cpt_lv_bias_true*(aufschlag_vtt*0+aufschlag_esc_rfh*2), #ESC_RFH >3000h VTT<100h
cpt_lv_bias_true*(aufschlag_vtt*1+aufschlag_esc_rfh*2), 1-
cpt_lv_bias_true*(aufschlag_vtt*1+aufschlag_esc_rfh*2), #ESC_RFH >3000h VTT
100<x<300h

cpt_lv_bias_true*(aufschlag_vtt*2+aufschlag_esc_rfh*1.5), 1-
cpt_lv_bias_true*(aufschlag_vtt*2+aufschlag_esc_rfh*1.5), #ESC_RFH >3000h VTT
>300h

dim = c(2, 3, 3), dimnames = list(Biasvoltage_LV =
auftreten, VTT = vtt, RFH = abnutzung_esc)
Clampvoltage_Alarm.cpt <- array(c(cpt_alarm_clamp_true, cpt_alarm_clamp_false,
#ESC_RFH<1000h VTT<100h
cpt_alarm_clamp_true*(aufschlag_vtt*2+aufschlag_esc_rfh*0), 1-
cpt_alarm_clamp_true*(aufschlag_vtt*2+aufschlag_esc_rfh*0), #ESC_RFH<1000h VTT
100<x<300h
cpt_alarm_clamp_true*(aufschlag_vtt*3+aufschlag_esc_rfh*0), 1-
cpt_alarm_clamp_true*(aufschlag_vtt*3+aufschlag_esc_rfh*0), #ESC_RFH <1000 VTT
>300h
cpt_alarm_clamp_true*(aufschlag_vtt*0+aufschlag_esc_rfh*2), 1-
cpt_alarm_clamp_true*(aufschlag_vtt*0+aufschlag_esc_rfh*2), #ESC_RFH 1000h<x<2500
VTT<100h
cpt_alarm_clamp_true*(aufschlag_vtt*2+aufschlag_esc_rfh*2), 1-
cpt_alarm_clamp_true*(aufschlag_vtt*2+aufschlag_esc_rfh*2), #ESC_RFH 1000h<x<2500
VTT 100<x<300h
cpt_alarm_clamp_true*(aufschlag_vtt*3+aufschlag_esc_rfh*2), 1-
cpt_alarm_clamp_true*(aufschlag_vtt*3+aufschlag_esc_rfh*2), #ESC_RFH 1000h<x<2500
VTT >300h
cpt_alarm_clamp_true*(aufschlag_vtt*0+aufschlag_esc_rfh*3), 1-
cpt_alarm_clamp_true*(aufschlag_vtt*0+aufschlag_esc_rfh*3), #ESC_RFH >3000h
VTT<100h
cpt_alarm_clamp_true*(aufschlag_vtt*2+aufschlag_esc_rfh*3), 1-
cpt_alarm_clamp_true*(aufschlag_vtt*2+aufschlag_esc_rfh*3), #ESC_RFH >3000h VTT
100<x<300h
cpt_alarm_clamp_true*(aufschlag_vtt*3+aufschlag_esc_rfh*3), 1-
cpt_alarm_clamp_true*(aufschlag_vtt*3+aufschlag_esc_rfh*3), #ESC_RFH >3000h VTT
>300h

dim = c(2, 3, 3), dimnames =

```

```

list(Clampvoltage_Alarm = auftreten, VTT = vtt, RFH = abnutzung_esc))
Clampvoltage_LV.cpt <- array(c(cpt_lv_clamp_true, cpt_lv_clamp_false,
#ESC_RFH<1000h VTT<100h
cpt_lv_clamp_true*(aufschlag_vtt*2+aufschlag_esc_rfh*0), 1-
cpt_lv_clamp_true*(aufschlag_vtt*2+aufschlag_esc_rfh*0), #ESC_RFH<1000h VTT
100<x<300h
cpt_lv_clamp_true*(aufschlag_vtt*3+aufschlag_esc_rfh*0), 1-
cpt_lv_clamp_true*(aufschlag_vtt*3+aufschlag_esc_rfh*0), #ESC RFH <1000 VTT >300h

cpt_lv_clamp_true*(aufschlag_vtt*0+aufschlag_esc_rfh*2), 1-
cpt_lv_clamp_true*(aufschlag_vtt*0+aufschlag_esc_rfh*2), #ESC_RFH 1000h<x<2500
VTT<100h

cpt_lv_clamp_true*(aufschlag_vtt*2+aufschlag_esc_rfh*2), 1-
cpt_lv_clamp_true*(aufschlag_vtt*2+aufschlag_esc_rfh*2), #ESC_RFH 1000h<x<2500 VTT
100<x<300h

cpt_lv_clamp_true*(aufschlag_vtt*3+aufschlag_esc_rfh*2), 1-
cpt_lv_clamp_true*(aufschlag_vtt*3+aufschlag_esc_rfh*2), #ESC_RFH 1000h<x<2500 VTT
>300h
cpt_lv_clamp_true*(aufschlag_vtt*0+aufschlag_esc_rfh*3), 1-
cpt_lv_clamp_true*(aufschlag_vtt*0+aufschlag_esc_rfh*3), #ESC_RFH >3000h VTT<100h
cpt_lv_clamp_true*(aufschlag_vtt*2+aufschlag_esc_rfh*3), 1-
cpt_lv_clamp_true*(aufschlag_vtt*2+aufschlag_esc_rfh*3), #ESC_RFH >3000h VTT
100<x<300h
cpt_lv_clamp_true*(aufschlag_vtt*3+aufschlag_esc_rfh*3), 1-
cpt_lv_clamp_true*(aufschlag_vtt*3+aufschlag_esc_rfh*3), #ESC_RFH >3000h VTT
>300h

                                dim = c(2, 3, 3), dimnames = list(Clampvoltage_LV =
auftreten, VTT = vtt, RFH = abnutzung_esc))
RF_POWER_Alarm.cpt <- array(c(cpt_alarm_pwrhz_true, cpt_alarm_pwrhz_false,
#ESC_RFH<1000h VTT<100h
cpt_alarm_pwrhz_true*(aufschlag_vtt*1+aufschlag_esc_rfh*0), 1-
cpt_alarm_pwrhz_true*(aufschlag_vtt*1+aufschlag_esc_rfh*0), #ESC_RFH<1000h VTT
100<x<300h
cpt_alarm_pwrhz_true*(aufschlag_vtt*2+aufschlag_esc_rfh*0), 1-
cpt_alarm_pwrhz_true*(aufschlag_vtt*2+aufschlag_esc_rfh*0), #ESC RFH <1000 VTT
>300h
cpt_alarm_pwrhz_true*(aufschlag_vtt*0+aufschlag_esc_rfh*1), 1-
cpt_alarm_pwrhz_true*(aufschlag_vtt*0+aufschlag_esc_rfh*1), #ESC_RFH 1000h<x<2500
VTT<100h
cpt_alarm_pwrhz_true*(aufschlag_vtt*1+aufschlag_esc_rfh*1), 1-
cpt_alarm_pwrhz_true*(aufschlag_vtt*1+aufschlag_esc_rfh*1), #ESC_RFH 1000h<x<2500
VTT 100<x<300h
cpt_alarm_pwrhz_true*(aufschlag_vtt*2+aufschlag_esc_rfh*1), 1-
cpt_alarm_pwrhz_true*(aufschlag_vtt*2+aufschlag_esc_rfh*1), #ESC_RFH 1000h<x<2500
VTT >300h
cpt_alarm_pwrhz_true*(aufschlag_vtt*0+aufschlag_esc_rfh*2), 1-
cpt_alarm_pwrhz_true*(aufschlag_vtt*0+aufschlag_esc_rfh*2), #ESC_RFH >3000h

```

VTT<100h

```
cpt_alarm_pwrhz_true*(aufschlag_vtt*1+aufschlag_esc_rfh*2), 1-
cpt_alarm_pwrhz_true*(aufschlag_vtt*1+aufschlag_esc_rfh*2), #ESC_RFH >3000h VTT
```

100<x<300h

```
cpt_alarm_pwrhz_true*(aufschlag_vtt*2+aufschlag_esc_rfh*2), 1-
cpt_alarm_pwrhz_true*(aufschlag_vtt*2+aufschlag_esc_rfh*2), #ESC_RFH >3000h VTT
>300h
```

```
dim = c(2, 3, 3), dimnames = list(RF_POWER_Alarm =
auftreten, VTT = vtt, RFH = abnutzung_esc))
RF_POWER_LV.cpt <- array(c(cpt_lv_pwrhz_true, cpt_lv_pwrhz_false, #ESC_RFH<1000h
VTT<100h
```

```
cpt_lv_pwrhz_true*(aufschlag_vtt*1+aufschlag_esc_rfh*0), 1-
cpt_lv_pwrhz_true*(aufschlag_vtt*1+aufschlag_esc_rfh*0), #ESC_RFH<1000h VTT
100<x<300h
```

```
cpt_lv_pwrhz_true*(aufschlag_vtt*2+aufschlag_esc_rfh*0), 1-
cpt_lv_pwrhz_true*(aufschlag_vtt*2+aufschlag_esc_rfh*0), #ESC RFH <1000 VTT >300h
```

```
cpt_lv_pwrhz_true*(aufschlag_vtt*0+aufschlag_esc_rfh*1), 1-
cpt_lv_pwrhz_true*(aufschlag_vtt*0+aufschlag_esc_rfh*1), #ESC_RFH 1000h<x<2500
VTT<100h
```

```
cpt_lv_pwrhz_true*(aufschlag_vtt*1+aufschlag_esc_rfh*1), 1-
cpt_lv_pwrhz_true*(aufschlag_vtt*1+aufschlag_esc_rfh*1), #ESC_RFH 1000h<x<2500
VTT 100<x<300h
```

```
cpt_lv_pwrhz_true*(aufschlag_vtt*2+aufschlag_esc_rfh*1), 1-
cpt_lv_pwrhz_true*(aufschlag_vtt*2+aufschlag_esc_rfh*1), #ESC_RFH 1000h<x<2500
VTT >300h
```

```
cpt_lv_pwrhz_true*(aufschlag_vtt*0+aufschlag_esc_rfh*2), 1-
cpt_lv_pwrhz_true*(aufschlag_vtt*0+aufschlag_esc_rfh*2), #ESC_RFH >3000h VTT<100h
cpt_lv_pwrhz_true*(aufschlag_vtt*1+aufschlag_esc_rfh*2), 1-
cpt_lv_pwrhz_true*(aufschlag_vtt*1+aufschlag_esc_rfh*2), #ESC_RFH >3000h VTT
100<x<300h
```

```
cpt_lv_pwrhz_true*(aufschlag_vtt*2+aufschlag_esc_rfh*2), 1-
cpt_lv_pwrhz_true*(aufschlag_vtt*2+aufschlag_esc_rfh*2), #ESC_RFH >3000h VTT
>300h
```

```
dim = c(2, 3, 3), dimnames = list(RF_POWER_LV =
auftreten, VTT = vtt, RFH = abnutzung_esc))
```

```
#-----
-----
```

#3 EBENE

#einlesen der daten aus excel

```
setwd("S:/8_Austausch/TAN-FAB/DA_FAB/Diplomarbeit/Erstellung BN/Excel
Auswertungen")
```

```
x <- read_excel("Versuch_cpt Liste_181007.xlsx", sheet = "R_import_neu")
```

```
x <- as.data.frame(x)
```

#schleife die w befüllt damit werte in netz verwendet werden können

```
w <- c()
```

```
j <- 1
```

```
for(i in 1:length(x$`TTF<12`)){
```

```

w[j] <- x[i,1]
w[j+1] <- x[i,2]
w[j+2] <- x[i,3]
w[j+3] <- x[i,4]
j <- j+4
}
Power_UNIT_cpt_liste <-w
POWER_UNIT.cpt <- array( c(Power_UNIT_cpt_liste), dim = c(4,2,2,2,2,2,2),
                        dimnames = list( POWER_UNIT = TTF_RANGE,
                                         Biasvoltage_Alarm = auftreten,
                                         Biasvoltage_LV = auftreten,
                                         Clampvoltage_Alarm = auftreten,
                                         Clampvoltage_LV = auftreten,
                                         RF_POWER_Alarm = auftreten,
                                         RF_POWER_LV = auftreten))
##-----
-----
#Erstellen des Netzes (Fitten des Netzes)
cpt <- list(VTT = VTT.cpt,
           RFH = esc_rfh.cpt,
           Biasvoltage_Alarm = Biasvoltage_Alarm.cpt,
           Biasvoltage_LV = Biasvoltage_LV.cpt,
           Clampvoltage_Alarm = Clampvoltage_Alarm.cpt,
           Clampvoltage_LV = Clampvoltage_LV.cpt,
           RF_POWER_Alarm = RF_POWER_Alarm.cpt,
           RF_POWER_LV = RF_POWER_LV.cpt,
           POWER_UNIT = POWER_UNIT.cpt)
bn_chuck_power_2 <- custom.fit(chuck_power_unit_2, dist = cpt)
#Abfragen zu verschiedenen Statusen
cpquery(bn_chuck_power_2, event = (POWER_UNIT == "TTF<12"),evidence = TRUE)
cpquery(bn_chuck_power_2, event = (POWER_UNIT == "12<TTF<24"),evidence = TRUE)
cpquery(bn_chuck_power_2, event = (POWER_UNIT == "24<TTF<36"),evidence = TRUE)
cpquery(bn_chuck_power_2, event = (POWER_UNIT == "TTF>36"),evidence = TRUE)
#cpquery(bn_chuck_power_2, event = (Biasvoltage_Alarm == "TRUE"), evidence =
POWER_UNIT == "TTF<12")
#cpquery(bn_chuck_power_2, event = (VTT == ">300"), evidence = POWER_UNIT ==
"24<TTF<36")

```

9 Literaturverzeichnis

Abu-Samah, A., Shahzad, M.K., Zamai, E., Said, A.B.: Failure Prediction Methodology for Improved Proactive Maintenance using Bayesian Approach, in: IFAC-PapersOnLine 48, S.844–851, 2015 <https://doi.org/10.1016/j.ifacol.2015.09.632>

Acatech: Smart Maintenance für Smart Factories: Mit intelligenter Instandhaltung Industrie 4.0 vorantreiben, in: acatech POSITION (Oktober 2015)

Aghezzaf, E.-H., Khatab, A., Tam, P.L.: Optimizing production and imperfect preventive maintenance planning's integration in failure-prone manufacturing systems, in: Reliability Engineering & System Safety 145, S.190–198, 2016 <https://doi.org/10.1016/j.ress.2015.09.017>

Ansari, F., Glawar, R., Sihn, W.: Prescriptive Maintenance of CPPS by Integrating Multi- modal Data with Dynamic Bayesian Networks , 2017

Asano, K., Hatakeyama, F., Yatsuzuka, K.: Fundamental Study of an Electrostatic Chuck for Silicon Wafer Handling, in: IEEE TRANSACTIONS ON INDUSTRY APPLICATIONS 38, S.840-845, 2002

Batini, C., Cappiello, C., Francalanci, C., Maurino, A.: Methodologies for data quality assessment and improvement, in: ACM Computing Surveys 41, S.1–52, 2009 <https://doi.org/10.1145/1541880.1541883>

Bauer, A., Günzel, H.: Data-Warehouse-Systeme: Architektur, Entwicklung, Anwendung, 4. überarbeitete und erweiterte Auflage, dpunkt.verlag Heidelberg, 2013, ISBN 978-3-89864-785-4

Bayes, T.: An Essay towards solving a Problem in the Doctrine of Chances. By the late Rev. Mr. Bayes, communicated by Mr. Price, in a letter to John Canton, M. A. and F. R. S., 1763

Beierle, C., Kern-Isberner, G.: Methoden wissensbasierter Systeme: Grundlagen, Algorithmen, Anwendungen, 4. verb. Aufl. ed, Studium. Vieweg + Teubner, Wiesbaden, 2008, ISBN 978-3-8348-0504-1

Bernerstätter, R.: Data Maturity for Smart Factory Applications - An Assessment Model, in: ACTA TECHNICA CORVIENSIS – Bulletin of Engineering TOME XI, Fascicule 1 [January-March], 2018

Biedermann, H.: Ersatzteilmanagement: effiziente Ersatzteillogistik für Industrieunternehmen, 2. erw. und aktualisierte Auflage, Springer Berlin, 2008, ISBN 978-3-540-00850-7

Biedermann, H.: Anlagenmanagement: Managementwerkzeuge zur Rationalisierung, In: Schriftenreihe Erfolgspotentiale für Unternehmer und Führungskräfte, Verl. TÜV Rheinland, Köln, 1990

Bouaziz, M.-F., Zamaï, E.: Equipment Health Factor prediction for complex semiconductor manufacturing facility, in: IFAC Proceedings Volumes 45, S. 1005–1010, 2012, <https://doi.org/10.3182/20120523-3-RO-2023.00377>

Bouaziz, M.F., Zamaï, E., Duvivier, F.: Towards Bayesian network methodology for predicting the equipment health factor of complex semiconductor systems, in: International Journal of Production Research 51, S.4597–4617, 2013, <https://doi.org/10.1080/00207543.2013.775525>

Chapman, P., Khabaza, T., Shearer, C.: CRISP-DM 1.0: Step-by-step data mining guide, 1999

Dagum, P., Galper, A., Horvitz, E.: Dynamic Network Models for Forecasting, in: Uncertainty in Artificial Intelligence. Elsevier, S. 41–48., 1992, <https://doi.org/10.1016/B978-1-4832-8287-9.50010-4>

Deutsches Institut für Normung: DIN 31051:2012-09 Grundlagen der Instandhaltung, 2012

Deutsches Institut für Normung: DIN EN ISO 9001 Qualitätsmanagementsysteme-Anforderungen (ISO 9001:2015); Deutsch und Englische Fassung EN ISO 9001:2015, 2015

Deutsches Institut für Normung: DIN EN 13306 Instandhaltung-Begriffe der Instandhaltung Dreisprachige Fassung EN 13306:2017, 2018

Eppler, M.J.: Managing information quality: increasing the value of information in knowledge-intensive products and processes, 2nd ed. ed. Springer, Berlin New York, 2006, ISBN 978-3-540-31408-0

Fischer, L.: BPM everywhere: Internet of Things, Process of Everything, Future Strategies, S.L., 2015, ISBN 978-0-9863214-1-2

Frawley, W.J.: Knowledge Discovery in Databases: An Overview, MIT PR, 1992, ISBN 978-0262660709

Gabriel, R., Gluchowski, P., Pastwa, A.: Data warehouse & data mining, 1. Nachdr. Oktober 2011. ed, Informatik. W3L-Verl, Herdecke Witten, 2011, ISBN 978-3-937137-66-7

Glawar, R., Karner, M., Nemeth, T., Matyas, K., Sihh, W.: An Approach for the Integration of Anticipative Maintenance Strategies within a Production Planning and Control Model, in: *Procedia CIRP* 67, S. 46–51, 2018, <https://doi.org/10.1016/j.procir.2017.12.174>

Hadidi, L.A., Turki, U.M.A., Rahim, A.: Integrated models in production planning and scheduling, maintenance and quality: a review, in: *International Journal of Industrial and Systems Engineering* Vol.10, S.21-50, 2012, <https://doi.org/10.1504/IJISE.2012.044042>

Hatakeyama, F., Asano, K.: Fundamental Characteristics of Electrostatic Wafer Chuck with Insulating Sealant, in: *IEEE TRansactions on Industry Applications* Vol.36, S.510-516, 2000, DOI: 10.1109/28.833768

Heckerman, D., Geiger, D., Chickering, D.M.: Learning Bayesian networks: The combination of knowledge and statistical data, in: *Machine Learning* Vol.20, S.197-243, 1995, <https://doi.org/10.1007/BF00994016>

Helmis, S., Hollmann, R.: *Webbasierte Datenintegration: Ansätze zur Messung und Sicherung der Informationsqualität in heterogenen Datenbeständen unter Verwendung eines vollständig webbasierten Werkzeuges*, 1. Aufl. ed Vieweg + Teubner Research Ausgezeichnete Arbeiten zur Informationsqualität. Vieweg + Teubner, Wiesbaden, 2009, ISBN 978-3-8348-0723-6

Hinrichs, H.: Datenqualitätsmanagement in Data Warehouse-Umgebungen, in: Heuer, A., Leymann, F., Priebe, D. (Eds.): *Datenbanksysteme in Büro, Technik Und Wissenschaft*. Springer Berlin Heidelberg, Berlin, Heidelberg, S. 187–206, 2001, https://doi.org/10.1007/978-3-642-56687-5_15

Homburg, C.: *Grundlagen des Marketingmanagements: Einführung in Strategie, Instrumente, Umsetzung und Unternehmensführung*, 5. überarbeitete und erweiterte Auflage, Springer Gabler, Wiesbaden, 2017, ISBN 978-3-658-13653-6

Hoppenstedt, B., Pryss, R., Treß, A., Biechele, B., Reichert, M.: *Datengetriebene Module für Predictive Maintenance* 3, S. 21-23, 2017

Hölbfer, S.: *Modell zur Auswahl von Instandhaltungsstrategien in anlageintensiven Industriebetrieben*, Leoben, Montanuniversität Leoben, Univ., Diss., 2014

Jensen, F.V., Nielsen, T.D.: *Bayesian networks and decision graphs*, 2.Auflage, Information science and statistics. Springer-Verlag, New York, 2007, ISBN 978-0-387-68281-5

Jun, H.-B., Kim, D.: A Bayesian network-based approach for fault analysis, in: Expert Systems with Applications 81, S.332–348, 2017, <https://doi.org/10.1016/j.eswa.2017.03.056>

Karim, R., Westerberg, J., Galar, D., Kumar, U.: Maintenance Analytics – The New Know in Maintenance, in: IFAC-PapersOnLine 49, S. 214–219, 2016, <https://doi.org/10.1016/j.ifacol.2016.11.037>

Khoshafian, D.S., Rostetter, C.: Digital Prescriptive Maintenance: Disrupting Manufacturing Value Streams through Internet of Things, Big Data, and Dynamic Case Management, 2015

Kraaijeveld, P., Druzdzel, M., Onisko, A., Wasyluk, H: GeNIeRate: An Interactive Generator of Diagnostic Bayesian Network Models, 2018

Kroninger, F., Streckfuss, N., Frey, L., Falter, T., Ryzlewicz, C., Pfitzner, L., Ryssel, H.: Application of advanced contamination analysis for qualification of wafer handling systems and chucks, in: Applied Surface Science 63, S. 93-98, 1993

Kruse, R., Borgelt, C., Braune, C., Klawonn, F., Moewes, C., Steinbrecher, M.: Computational intelligence: eine methodische Einführung in künstliche neuronale Netze, evolutionäre Algorithmen, Fuzzy-Systeme und Bayes-Netze, 2. überarbeitete und erweiterte Auflage. ed, Computational intelligence. Springer Vieweg, Wiesbaden, 2015, ISBN 978-3-658-10903-5

Lazar, M., Vang, H., Brosselard, P., Raynaud, C., Cremillieu, P., Leclercq, J.-L., Descamps, A., Scharholz, S., Planson, D.: Deep SiC etching with RIE, in: Superlattices and Microstructures 40, S. 388–392, 2006, <https://doi.org/10.1016/j.spmi.2006.06.015>

Lee, Y.W. (Ed.): Journey to data quality. MIT Press, Cambridge, Mass, 2006, ISBN 978-0-262-12287-0

Liggins, M.E., Hall, D.L., Llinas, J.: Handbook of multisensor data fusion: theory and practice, 2nd ed. The electrical engineering and applied signal processing series. CRC Press, Boca Raton Florida., 2009, ISBN 9781420053098

Matyas, K.: Instandhaltungslogistik: Qualität und Produktivität steigern, 6., überarbeitete Auflage. ed, Praxisreihe Qualitätswissen. Hanser, München, 2016, ISBN 978-3-446-44614-4

Matyas, K., 2002. Ganzheitliche Optimierung durch individuelle Instandhaltungsstrategien, in: Industrie Management 18, S. 13-16, 2002

Matyas, K., Nemeth, T., Kovacs, K., Glawar, R.: A procedural approach for realizing prescriptive maintenance planning in manufacturing industries, in: CIRP Annals 66, S.461–464, 2016 <https://doi.org/10.1016/j.cirp.2017.04.007>

May, G.S., Spanos, C.J.: Fundamentals of semiconductor manufacturing and process control, IEEE Wiley-Interscience, Hoboken, N.J., 2006, ISBN 978-0-471-78406-7

Mobley, R.K.: An introduction to predictive maintenance, 2nd edition, Butterworth-Heinemann, Amsterdam New York, 2002, ISBN 978-0-7506-7531-4

Moubray, J., Kugler, W.: RCM - die hohe Schule der Zuverlässigkeit von Produkten und Systemen, Verl. Moderne Industrie, Landsberg, 1996, ISBN 978-3478916301

Nagarajan, R., Scutari, M., Lèbre, S.: Bayesian networks in R: with applications in systems biology, Use R! Springer, New York, 2013, ISBN 978-1461464457

Naïm, P.: Réseaux bayésiens. Eyrolles, Paris, 2008, ISBN 978-2-212-11972-5

Naumann, F.: Datenqualität, in: Informatik-Spektrum 30, S.27–31, 2007, <https://doi.org/10.1007/s00287-006-0125-5>

Nawaz, J.M., Arshad, M.Z., Hong, S.J.: Fault Diagnosis in Semiconductor Etch Equipment Using Bayesian Networks, in: JSTS:Journal of Semiconductor Technology and Science 14, S.252–261, 2014, <https://doi.org/10.5573/JSTS.2014.14.2.252>

Nemeth, T., Ansari, F., Sihm, W., Haslhofer, B., Schindler, A.: PriMa-X: A reference model for realizing prescriptive maintenance and assessing its maturity enhanced by machine learning, in: Procedia CIRP 72, S.1039–1044, 2018, <https://doi.org/10.1016/j.procir.2018.03.280>

Nemeth, T., Bernerstätter, R., Glawar, R., Matyas, K., Sihm, W.: Instandhaltung 4.0: Sicherstellung von Produktqualität und Anlagenverfügbarkeit durch einen echtzeitbasierten Instandhaltungsleitstand, in: ZWF Zeitschrift für wirtschaftlichen Fabrikbetrieb 110, S. 569–573, 2015 <https://doi.org/10.3139/104.111373>

Novotn, Z.: Reactive ion etching of dielectric and semiconductor layers in CHF₃ and C₂F₆, in: Czechoslovak Journal of Physics Vol.38, S.689-695, 1988

Palm, W.: Die Instandhaltung von Maschinen und maschinellen Anlagen im Industriebetrieb, R.G. Fischer, Frankfurt, 1981, ISBN 978-3883232621

Pawellek, G.: Integrierte Instandhaltung und Ersatzteillogistik: Vorgehensweisen, Methoden, Tools, Springer Vieweg, Berlin, 2013, ISBN 978-3662486665

Pearl, J.: Probabilistic reasoning in intelligent systems: networks of plausible inference, Rev. 2. print., The Morgan Kaufmann series in representation and reasoning, Kaufmann, San Francisco Calif., 1988, ISBN 0-934613-73-7

Pearl, J.: Fusion, propagation, and structuring in belief networks, in: *Artificial Intelligence* 29, S. 241–288, 1986, [https://doi.org/10.1016/0004-3702\(86\)90072-X](https://doi.org/10.1016/0004-3702(86)90072-X)

Prüß, H., Nebl, T.: Instandhaltungspolitik im Rahmen einer ganzheitlichen Anlagenwirtschaft, in: *ZWF Zeitschrift für wirtschaftlichen Fabrikbetrieb* 101, S. 728–734, 2006, <https://doi.org/10.3139/104.101091>

Qin, S., McTeer, A.: Wafer dependence of Johnsen–Rahbek type electrostatic chuck for semiconductor processes, in: *Journal of Applied Physics* 102, 2007, <https://doi.org/10.1063/1.2778633>

Rahm, E., Do, H.H.: Data Cleaning: Problems and Current Approaches, in: *Bulletin of the technical committee on Data Engineering Vol.23 No.4*, IEEE Computer Society, 2002

Reichel, J., Müller, G., Haeffs, J.: *Betriebliche Instandhaltung*, 2. Auflage. ed, VDI-Buch Springer Vieweg, Berlin, 2018, ISBN 978-3-662-53134-1

Rohweder, D.J.P., Kasten, G., Malzahn, D., Piro, A., Schmid, J.: Informationsqualität – Definitionen, Dimensionen und Begriffe, in: *Daten- und Informationsqualität: Auf dem Weg zur Information Excellence*, S. 23-46, 4.Auflage, Springer Vieweg, 2008 ISBN 978-3-658-21993-2

SAID, A.B., SHAHZAD, M.K., ZAMAI, E., HUBAC, S., TOLLENAERE, M.: A Bayesian network based approach to improve the effectiveness of maintenance actions in Semiconductor Industry, in: *2nd European conference of the prognostics and health management society Vol.5*, 2014

Schenk, M.: *Instandhaltung technischer Systeme: Methoden und Werkzeuge zur Gewährleistung eines sicheren und wirtschaftlichen Anlagenbetriebs*. Springer, Berlin, 2010, ISBN 978-3642039485

Schucan, C.P.: *Effektivitätssteigerung mittels konzeptionellem Informationsmanagement*, Zürich, ETH Zürich, Univ., Diss., 1999, <https://doi.org/10.3929/ethz-a-003829265>

Schumacher, A., Erol, S., Sihm, W.: A Maturity Model for Assessing Industry 4.0 Readiness and Maturity of Manufacturing Enterprises, in: *Procedia CIRP* 52, S.161–166, 2016, <https://doi.org/10.1016/j.procir.2016.07.040>

Shearn, M., Sun, X., Henry, M.D., Yariv, A., Scherer, A.: *Advanced Plasma Processing: Etching, Deposition, and Wafer Bonding Techniques for Semiconductor Applications*, in: *Semiconductor Technologies* 27, S.81-105, 2010

Silberschatz, A., Tuzhilin, A.: What makes patterns interesting in knowledge discovery systems, in: IEEE Transactions on Knowledge and Data Engineering 8, S.970–974, 1996, <https://doi.org/10.1109/69.553165>

Slaby, D., Krasselt, R.: Industriebetriebslehre. 1: Anlagenwirtschaft. Oldenbourg, München, 1998, ISBN 978-3486244861

Spickenheuer, W.: Optimale vorbeugende Instandhaltung, in: Brockhoff, K., Dinkelbach, W., Kall, P., Pressmar, D.B., Spicher, K. (Eds.), Vorträge Der Jahrestagung 1977 / Papers of the Annual Meeting 1977 DGOR. Physica-Verlag HD, Heidelberg, S. 337–346, 1978, https://doi.org/10.1007/978-3-662-00409-8_49

Sturm, A., Förster, R.: Maschinen- und Anlagendiagnostik für die zustandsbezogene Instandhaltung, B.G. Teubner, Stuttgart, 1990, ISBN 9783322998149

Sun, Y., Cheng, J., Lu, Y., Hou, Y., Ji, L.: Design space of electrostatic chuck in etching chamber, in: Journal of Semiconductors 36, 2015, <https://doi.org/10.1088/1674-4926/36/8/084004>

Tiddens, W.W., Braaksma, A.J.J., Tinga, T.: The Adoption of Prognostic Technologies in Maintenance Decision Making: A Multiple Case Study, in: Procedia CIRP Vol. 38, S.171–176, 2015, <https://doi.org/10.1016/j.procir.2015.08.028>

Vogel-Heuser, B., Bauernhansl, T., Ten Hompel, M.: Handbuch Industrie 4.0. Bd. 1: Produktion, 2. erweiterte und bearbeitete Auflage. Springer Reference Technik. Springer Vieweg, Berlin, 2017, ISBN 978-3662452783

Wand, Y., Wang, R.Y.: Anchoring data quality dimensions in ontological foundations, in: Communications of the ACM Vol.39 No. 1, S. 86–95, 1996, <https://doi.org/10.1145/240455.240479>

Wang, R.Y., Strong, D.M.: Beyond Accuracy: What Data Quality Means to Data Consumers, in: Journal of Management Information Systems 12, S.5–33, 1996

Warnecke, H.-J., Bauer, C.-O.: Instandhaltungsmanagement, 2., völlig überarb. Aufl. ed, Handbuch Instandhaltung. Verl. TÜV Rheinland, Köln, 1992, ISBN 978-3-88585-822-5

Weber, P., Medina-Oliva, G., Simon, C., lung, B.: Overview on Bayesian networks applications for dependability, risk analysis and maintenance areas, in: Engineering Applications of Artificial Intelligence Vol. 25, S.671–682, 2012 <https://doi.org/10.1016/j.engappai.2010.06.002>

Wittig, F.: Maschinelles Lernen Bayes'scher Netze für benutzeradaptive Systeme, Saarbrücken, Universität des Saarlandes, Univ., Diss., 2002

Winkler, W.E.: Methods for evaluating and creating data quality, in: Information Systems 29, S. 531–550, 2004, <https://doi.org/10.1016/j.is.2003.12.003>

Wirth, R., Hipp, J.: CRISP-DM: Towards a Standard Process Model for Data Mining, 2000

Würthele, V.G.: Datenqualitätsmetrik für Informationsprozesse: Datenqualitätsmanagement mittels ganzheitlicher Messung der Datenqualität, Zürich, ETH Zürich, Univ., Diss., 2003, <https://doi.org/10.3929/ethz-a-004650156>

Yatsuzuka, K., Toukairin, J., Asano, K., Aonuma, S.: Electrostatic chuck with a thin ceramic insulation layer for wafer holding, in: Conference Record of the 2001 IEEE Industry Applications Conference, S. 399–403, 2001 <https://doi.org/10.1109/IAS.2001.955451>

Zack, M.H.: Managing Codified Knowledge, in: Sloan Management Review S.45-58, 1999

Zülch, G., Jagdev, H.S., Stock, P.: Integrating human aspects in production management: IFIP TC5/WG5.7 proceedings of the International Conference on Human Aspects in Production Management, Karlsruhe, Germany, IFIP Springer New York, 2005, ISBN 978-0-387-23065-8

10 Abbildungsverzeichnis

Abbildung 1: Unterteilung der Instandhaltung nach DIN 31051	5
Abbildung 2: Trade-Off zwischen Instandhaltungskosten und –intensität.....	6
Abbildung 3: Unterscheidung von Instandhaltungsstrategien	8
Abbildung 4: PF-Kurve	9
Abbildung 5: Empfohlene Auswahl von Instandhaltungsmaßnahmen für die vorausschauende Instandhaltung	10
Abbildung 6: Regelkreis der Instandhaltung	11
Abbildung 7: Entwicklung der Instandhaltungstechniken	12
Abbildung 8: Entwicklung der Wissensbasierten Instandhaltung	13
Abbildung 9: Problematik einer abteilungsübergreifenden Instandhaltungsplanung..	14
Abbildung 10: Architektur eines zukünftigen Instandhaltungsplanungssystems	15
Abbildung 11: Verknüpfung von Daten in der „Instandhaltung 4.0“	16
Abbildung 12: PriMa Modell für präskriptive Instandhaltung	18
Abbildung 13: PriMa-X Portfolio Matrix	19
Abbildung 14: Iterativer Zyklus des PriMa-X Referenzmodells	19
Abbildung 15: Instandhaltungs-Informations-Prozess	21
Abbildung 16: Erweitertes Begriffsschema nach Schucan	21
Abbildung 17: Datenqualitätsmerkmale nach Wang and Strong.....	22
Abbildung 18: Klassifikation von Problemen der Datenqualität in Datenquellen.....	23
Abbildung 19: Datenqualitätsradar nach Würthele	24
Abbildung 20: Einteilung der 15 IQ Dimensionen in 4 IQ Kategorien	26
Abbildung 21: 4 Ebenen der CRISP-DM Methodik für Data-Mining.....	28
Abbildung 22: Phasen des CRISP-DM-Prozessmodells.....	29
Abbildung 23: CRISP-DM Referenzmodell Phasen mit zugeordneten Aufgaben	31
Abbildung 24: ASUM-DM Zyklus von IBM	33
Abbildung 25: Beispiel für ein Bayes'sches Netz	34
Abbildung 26: Vorgangsweise bei der Erstellung von Bayes'schen Netzwerken	37
Abbildung 27: DBN mit Zeitscheiben und unterschiedlichen Knotentypen	38
Abbildung 28: Bayes'sches Netz zur Fehlerdiagnose in der Halbleiterätzung	39
Abbildung 29: Ergebnisse der Fehlerdiagnose	40
Abbildung 30: Ermittlung des Equipment Health Factor	41
Abbildung 31: Prozesskammern der Referenzmaschine M3 und zugehöriger Prozessablauf.....	44
Abbildung 32: Aufbau einer Prozesskammer.....	45
Abbildung 33: Aufbau des ESC Moduls.....	47
Abbildung 34: Erstmodellierung des Bayes'schen Netz für die gesamte Ätzkammer	55
Abbildung 35: Netzwerk E-Chuck	55
Abbildung 36: Zuordnung der Kennzahl zu Modulen und Klassifizierung anhand des E-Chucks	56

Abbildung 37: Fusionierung Statusmeldungen und Limitverletzungen	58
Abbildung 38: Berechnung der TTF Zeiten.....	59
Abbildung 39: Auszug einiger Limitverletzungen und zugehörige TTF	59
Abbildung 40: Auszug einiger Alarmer und zugehörige TTF	60
Abbildung 41: Bayes'Netz zur Ausfallprognose für das Modul E-Chuck.....	61
Abbildung 42: Submodell des E-Chucks	62
Abbildung 43: Bedingte Wahrscheinlichkeiten für Knoten ESC_RFH.....	63
Abbildung 44: Bedingte Wahrscheinlichkeiten für Knoten VTT_RFH	64
Abbildung 45: Einfluss der Einsatzzeit des E-Chuck Moduls auf das Auftreten von Biasvoltage Limitverletzungen	65
Abbildung 46: CPT Tabelle Biasvoltage LV	67
Abbildung 47: CPT Tabelle Biasvoltage Alarm	69
Abbildung 48: CPT Tabelle Clampvoltage LV.....	70
Abbildung 49: CPT Tabelle Clampvoltage Alarm.....	72
Abbildung 50: CPT Tabelle Generator LV	73
Abbildung 51: CPT Tabelle Generator Alarm	75
Abbildung 52: Ermittlung der TTF Mittelwerte.....	80
Abbildung 53: Gewichtung der TTF Zeiten	80
Abbildung 54: Wahrscheinlichkeitstabelle für Knoten RF Power	81
Abbildung 55: Erstellung des statischen Netzes in R	82
Abbildung 56: Erklärung der Code Fragmente bei der Erstellung des Netzwerkes ...	82
Abbildung 57: Definieren der Knotenzustände	83
Abbildung 58: Erstellen der Wahrscheinlichkeitstabelle für den Knoten Power Unit..	83
Abbildung 59: Fitten des Netzes.....	84
Abbildung 60: Dynamisches Bayes'sches Netz zur Ausfallprognose	85
Abbildung 61: Erweiterung der Wahrscheinlichkeitstabelle für VTT_RFH	86
Abbildung 62: Boxplot Einsatzstunden VTT_RFH bis zur nächsten Reinigung	86
Abbildung 63: Entscheidungsbaum für Zustand $t < 100h$	87
Abbildung 64: Bereich der Einsatzstunden (VTT) der Maschine in Zeitpunkt $t+1$ unter der Voraussetzung, dass Einsatzstunden in Zeitpunkt $t < 100h$	90
Abbildung 65: Zweiter Entscheidungsbaum für $100h < x < 300h$	91
Abbildung 66: Ausschnitt des R Scripts zur Berechnung des zweiten Entscheidungsbaums	92
Abbildung 67: Bereich der Einsatzstunden (VTT) der Maschine in Zeitpunkt $t+1$ unter der Voraussetzung, dass Einsatzstunden in Zeitpunkt $t > 100h < x < 300h$	92
Abbildung 68: Bereich der Einsatzstunden (VTT) der Maschine in Zeitpunkt $t+1$ unter der Voraussetzung, dass Einsatzstunden in Zeitpunkt $t > 300h$	93
Abbildung 69: Bedingte Wahrscheinlichkeiten für Knoten VTT in Zeitpunkt $t+1$	93
Abbildung 70: Erweiterung der Wahrscheinlichkeitstabelle für ESC_RFH	94
Abbildung 71: Boxplot Einsatzstunden ESC_RFH bis zum Austausch.....	94
Abbildung 72: Entscheidungsbaum für Zustand $t < 1000h$	95

Abbildung 73: Bereich der Einsatzstunden (ESC) der Maschine in Zeitpunkt $t+1$ unter der Voraussetzung, dass Einsatzstunden in Zeitpunkt $t < 1000h$	95
Abbildung 74: Bereich der Einsatzstunden (ESC) der Maschine in Zeitpunkt $t+1$ unter der Voraussetzung, dass Einsatzstunden in Zeitpunkt $t \ 1000h < x < 2500h$	96
Abbildung 75: Bereich der Einsatzstunden (ESC) der Maschine in Zeitpunkt $t+1$ unter der Voraussetzung, dass Einsatzstunden in Zeitpunkt $t > 2500h$	96
Abbildung 76: Bedingte Wahrscheinlichkeiten für Knoten ESC_RFH in Zeitpunkt $t+1$	97
Abbildung 77: Abfragen zur Ergebnisfindung und Ergebnisse des statischen Netz ..	97
Abbildung 78: Ergebnisse des dynamischen Netzes	98

11 Formelverzeichnis

Formel 1: Gemeinsame Verteilungsfunktion.....	35
Formel 2: Gesamte Verteilungsfunktion anhand des in Abbildung 25 präsentierten Beispiels	35
Formel 3: Satz von Bayes.....	36
Formel 4: Kalkulation des globalen EHF	41
Formel 5: Spaltenkombination der einzelnen Knoten	62
Formel 6: Berechnung der Auftretenswahrscheinlichkeit anhand des Knotens Biasvoltage Limitverletzung	66
Formel 7: Spaltenzahl des Knotens Biasvoltage LV	67
Formel 8: Spaltenzahl des Knoten Biasvoltage Alarm	68
Formel 9: Spaltenzahl des Knoten Clampvoltage LV	70
Formel 10: Spaltenzahl des Knoten Clampvoltage Alarm	71
Formel 11: Spaltenzahl des Knoten Generator Power LV	73
Formel 12: Spaltenzahl des Knoten Generator Power Alarm	74
Formel 13: Spaltenzahl des Knoten RF Power.....	75

12 Tabellenverzeichnis

Tabelle 1: Methodik des Instandhaltungsleitstands	16
Tabelle 2: 15 Dimensionen der Informationsqualität.....	26
Tabelle 3: Prognoseergebnisse	41
Tabelle 4: Struktur der APC_Alarmlogs	50
Tabelle 5: Ausschnitt Limitverletzungen	50
Tabelle 6: Exemplarischer Auszug aus dem Instandhaltungsmaßnahmenbuchs.....	52
Tabelle 7: Auszug der Reinigungsdaten.....	53
Tabelle 8: Diskretisierung der kontinuierlichen Variable ESC_RFH	63
Tabelle 9: Diskretisierung der kontinuierlichen Variable VTT_RFH	64
Tabelle 10: Zustände Biasvoltage LV	66
Tabelle 11: Wahrscheinlichkeitstabelle Biasvoltage LV	67
Tabelle 12: Zustände Biasvoltage Alarm	68
Tabelle 13: Wahrscheinlichkeitstabelle Biasvoltage Alarm	68
Tabelle 14: Zustände Clampvoltage LV	69
Tabelle 15: Wahrscheinlichkeitstabelle Clampvoltage LV	70
Tabelle 16: Zustände Clampvoltage Alarm.....	71
Tabelle 17: Tabelle Clampvoltage Alarm.....	71
Tabelle 18: Zustände Generator Power LV	72
Tabelle 19: Wahrscheinlichkeitstabelle Generator Power LV	73
Tabelle 20: Zustände Generator Power Alarm	74
Tabelle 21: Generator Power Alarm	74
Tabelle 22: Zustände Knoten RF Power.....	75
Tabelle 23: Mögliche Zustandskombinationen der Elternknoten	76
Tabelle 24: TTF Biasvoltage Alarm	77
Tabelle 25: TTF Biasvoltage LV	77
Tabelle 26: TTF Clampvoltage Alarm	78
Tabelle 27: Clampvoltage LV.....	78
Tabelle 28: TTF Generator Power Alarm.....	79
Tabelle 29: TTF Generator Power LV.....	79
Tabelle 30: Auszug Tabelle Entscheidungsbaum für VTT_RFH Zeitpunkt $t < 100h$...	88
Tabelle 31: Ausfallwahrscheinlichkeiten aller Submodule (Annahme).....	99
Tabelle 32: SWOT Analyse	102

13 Abkürzungsverzeichnis

BN	Bayes'sches Netzwerk
CPPS	Cyber Physical Production Systems
DBN	Dynamische Bayes'sche Netze
DIN	Deutsche Industrienorm
DM	Data Mining
DQ	Datenqualität
ESC	Electro-Static-Chuck
FMEA	Fehlermöglichkeits- und Einflussanalyse
IQ	Informationsqualität
KBM	Wissensbasierte Instandhaltung
LV	Limitverletzungen
OTG	Obere Toleranzgrenze
RIE	Reactive Ion Etching
RF	Radio-Frequency
RfH	Einsatzstunden
TTF	Time to Failure
UTG	Untere Toleranzgrenze
VTT	Komplettreinigung der Kammer
VW	Vergleichswert