

# **Hochschule Darmstadt**

– Fachbereich Maschinenbau und Kunststofftechnik–  
– Studiengang Mechatronik–

## **Datenverarbeitung innerhalb der Produktionstechnik**

**Interaktiver Wissensspeicher mittels eines Bayes Netzes für den  
Einrichtprozess einer Extrusionsblasformmaschine**

Projektbericht der berufspraktischen Phase zur Erlangung des  
akademischen Grades

**Bachelor of Science (B.Sc.)**

vorgelegt von

**Julian Schneider**

Matrikelnummer: 760221

eingereicht am: 6. Dezember 2021

Wintersemester 21/22

Prüfer/ Betreuer : Prof. Dr.-Ing. Karl Kleinmann  
Betriebl. Betreuer : Dipl. Ing. Rebekka Zache  
Betriebl. Betreuer : Dr.-Ing. Uwe Frieß



## SPERRVERMERK

---

Die vorliegende Projektarbeit zur berufspraktischen Phase mit dem Titel *„Datenverarbeitung innerhalb der Produktionstechnik, Interaktiver Wissensspeicher mittels eines Bayes Netzes für den Einrichtprozess einer Extrusionsblasformmaschine“* beinhaltet interne und vertrauliche Informationen des Unternehmens **Fraunhofer-Institut für Werkzeugmaschinen und Umformtechnik IWU**.

Eine Einsicht in diese Projektarbeit ist nicht gestattet. Ausgenommen davon sind der betreuende Dozent Prof.Dr.-Ing. K. Kleinmann, sowie die befugten Mitglieder des Prüfungsausschusses des Fachbereiches Maschinenbau und Kunststofftechnik der Hochschule Darmstadt. Eine Veröffentlichung und Vervielfältigung der Projektarbeit – auch in Auszügen – ist nicht gestattet.

Ausnahmen von dieser Regelung bedürfen einer schriftlichen Genehmigung des Unternehmens **Fraunhofer-Institut für Werkzeugmaschinen und Umformtechnik IWU**.

*Dresden, 6. Dezember 2021*

## EIDESSTATTLICHE ERKLÄRUNG

---

### Eidesstattliche Erklärung

hiermit erkläre ich, **Schneider Julian** geboren am **22.05.1996**, dass ich die vorliegende Arbeit selbstständig erstellt und keine anderen als die angegebenen Hilfsmittel benutzt habe.

Soweit ich auf fremde Materialien, Texte und Gedankengänge zurückgegriffen habe, enthalten meine Ausführungen vollständige und eindeutige Verweise auf die Urheber und Quellen. Alle weiteren Inhalte der vorgelegten Arbeit stammen von mir im urheberrechtlichen Sinn, soweit keine Verweise und Zitate erfolgen.

Ich versichere, dass ich diese Arbeit noch bei keiner Prüfung bzw. Prüfungsbehörde vorgelegt habe.

Mir ist bekannt, dass ein Täuschungsversuch vorliegt, wenn die vorstehende Erklärung sich als unrichtig erweist.

*Dresden, 3. Dezember 2021*



---

Julian Schneider

## VORWORT

---

Die vorliegende Arbeit ist während der im Mechatronik Studium vorgeschriebenen Berufspraktische Phase (BPP) entstanden. Zum Abschluss des Bachelorstudiums ist ein BPP über zwölf Wochen von Seiten der Hochschule vorgesehen. Dieses wird zumeist in Unternehmen abseits der Hochschule durchgeführt. Es sollen somit erste Tätigkeiten eines Ingenieurs ausgeführt und kennen gelernt werden. Auch dient diese Phase der Möglichkeit erste Kontakte in Industriezweige und Branchen zu knüpfen. Folgende BPP wurde am Fraunhofer Institut IWU in Dresden absolviert. Auf Grund meiner Neugierde und dem persönlichen Interesse an der Thematik Datenanalyse, cybertechnische Produktionssysteme und den neuen Künstliche Intelligenz (KI)-Methoden wurde der Schritt in ein neues Themenfeld vollzogen.

Der Projektauftrag lies spannende und interessante Aufgaben, welche zugleich herausfordernd wirkten, vorahnen. Während des Studiums konnte man in den Bereichen Datenanalyse und KI keine Einblicke und Erfahrungen sammeln. So galt es sich in das neue Gebiet einzuarbeiten, um möglichst viel an neuem Wissen zu erlernen. Neben dem persönlichen Interesse an dem Thema, verspricht dieses Gebiet eine steigende Präsenz in der Zukunft.

Aufgrund der aktuellen Covid19-Pandemie wurde die Arbeit vorwiegend abseits des Instituts aus dem Home-Office getätigt, da es untersagt war vor Ort die Büroräume zu nutzen. Trotz der besonderen Umstände möchte ich meiner Betreuerin Dipl. Ing. Rebekka Zache danken. Sie hat stets versucht den Kontakt mit mir zu halten und so gut es ging eine „vor Ort Büroatmosphäre“ herzustellen. Bei Fragen oder Problemen bat sie Ihre Hilfe an und unterstützte stets. Ein weiterer Dank geht an Herr Prof. Dr.-Ing. K. Kleinmann für die immer sehr gute Betreuung der BPP. Er hatte stets ein offenes Ohr und man konnte auf seine Unterstützung vertrauen.

Ich wünsche Ihnen viel Freude beim Lesen dieser Arbeit.

*Dresden, 6. Dezember 2021, Julian Schneider*

## ZUSAMMENFASSUNG

---

Das Projekt trägt das Ziel einen interaktiven Wissensspeicher zu entwickeln, welcher den Bediener beim Einrichtprozess von Extrusionsblasformmaschinen unterstützt. Der Einrichtprozess ist in zwei Phasen zu untergliedern. In Phase eins wird ein sogenannter *Vorformling* erzeugt. In der Projektarbeit wird lediglich diese Phase des Einrichtprozesses betrachtet und bewertet.

In Zusammenarbeit mit dem Industriepartner wurde eine vorläufige Struktur des Bayes-Netzes erstellt. Hier gilt es den vorliegenden Entwurf zu vollenden. Im Anschluss ist ein Algorithmus entwickelt worden, mit dem es möglich ist die wahrscheinlichste Fehlerursache vorherzusagen und dem Bediener somit Empfehlungen zu geben, entsprechend in den Prozess einzugreifen. Im nächsten Schritt ist ein Konzept entwickelt worden, wie Bayes-Netze gelernt und trainiert werden können. Dies ermöglicht den Wissensspeicher individuell für jede Maschine zu trainieren.

Zum Abschluss wurde eine Benutzeroberfläche erstellt, welche zur Interaktion zwischen Assistenzsystem und Benutzer dient. Die Oberfläche stellt Empfehlungen visuell da und dem Nutzer ist es möglich Eingaben zu tätigen. Es wurde eine webbasierte Oberfläche mit dem R kompatiblen Paket *Shiny* entwickelt.

Der vorläufige Entwurf beinhaltet einen Tab zur Übersicht des gesamten Bayes-Netzes, sowie einen Tab zur Anzeige eines einzelnen Fehlers und möglichen Fehlerursachen. Dort kann der Bediener über ein Drop-Down-Menü den für ihn interessanten Fehler auswählen und er erhält ein Bayes-Netz für diesen Fehler. Ein weiterer Tab dient dem Importieren neuer Datensätze sowie dem Trainieren und Lernen des Bayes-Netzes.

Im letzten Tab werden Fehlerursachen berechnet und Empfehlungen zum Eingriff in den Prozess gegeben. Der Einrichter wählt ein aktuell vorliegendes Fehlerbild aus. Anschließend bestätigt er dieses. Er erhält eine graphische Darstellung über das Fehler-Bayes-Netz, welche Fehlerursache vorliegt, sowie eine Textausgabe. Es folgt eine Rückmeldung seitens des Einrichters, ob er wie empfohlen in den Prozess eingreifen konnte und dies den Fehler beseitigte. Ist der Fehler nicht vollständig beseitigt erhält er weitere Empfehlungen.

Es wurden zum jetzigen Zeitpunkt noch keine realen Daten der Maschine gesammelt. Ebenfalls wurden keine Tests bzw. Implementierungen durchgeführt. „Kleine“ Tests für das Lernen/ Trainieren, sowie Vorhersagen der Fehlerursache wurden mittels eigens synthetisch generierten Daten durchgeführt.

# INHALTSVERZEICHNIS

---

## I PROJEKTARBEIT

1	FRAUNHOFER-INSTITUT IWU	2
2	EINLEITUNG	3
2.1	Projektauftrag/ Aufgabenstellung . . . . .	3
3	KONZEPT UND AUFBAU	5
3.1	Konzept . . . . .	5
3.2	Gliederung und Aufbau . . . . .	7
4	UMSETZUNG	8
4.1	R-package <i>bnlearn</i> und Aufbau Versuchsnetz . . . . .	8
4.2	Erstellen Bayes-Netz Strukturen . . . . .	12
4.3	Ermittlung einflussreichster Parameter . . . . .	14
4.4	Datenerhebung und -verarbeitung . . . . .	16
4.5	Benutzeroberfläche mit Shiny . . . . .	19
5	ERGEBNIS/ ZUSTAND ZUM ENDE DES BPP'S	23
6	AUSBLICK UND OFFENE PUNKTE	26

## II ANHANG

A	THEORIE UND VERTIEFUNG	28
A.1	Blasformen ( <i>Extrusionsblasformen</i> ) . . . . .	28
A.1.1	Extrusionsblasformen mit Überquetschen . . . . .	30
A.1.2	Kontinuierliches/ diskontinuierliches Extrusionsblasfor- men . . . . .	31
A.2	Bayes Netze . . . . .	32
A.3	Einstieg in die Programmierung mit R . . . . .	33
A.4	Graphisches Erstellen von Bayes Netzen . . . . .	33
A.4.1	Mittels BayesianFusion GeNIe . . . . .	33
A.4.2	Mittels SamIam . . . . .	35
A.5	Verknüpfung SAMIAM und R . . . . .	36
B	R-CODE	39
B.1	Fibonaccifolge . . . . .	39
B.2	Einlesen und Parameter Lernen Bayes-Netz Beispiel Asia . . . .	39
B.3	Algorithmus zum Zusammenfügen einzelner Sub-Bayes-Netze	40

LITERATUR	42
-----------	----

## ABBILDUNGSVERZEICHNIS

---

Abbildung 2.1	Flussdiagramm <i>aktueller Stand</i> Einrichtprozess Blasformmaschine . . . . .	4
Abbildung 2.2	Fehlerbaum für Einflussfaktoren bei Blasformen Prozessen ( <i>Kunden Dokument</i> ) [7] . . . . .	4
Abbildung 3.1	Allgemeiner Lösungsansatz . . . . .	5
Abbildung 3.2	Flussdiagramm Interaktion Mensch und Assistenzsystem . . . . .	6
Abbildung 4.1	DAG aus Bayes-Netz <i>survey</i> . . . . .	10
Abbildung 4.2	Balkendiagramm Verkehrsmittelnutzung . . . . .	11
Abbildung 4.3	Sub-Bayes-Netz für Fehler „Kratzer“ erstellt mit <i>SamIam</i> . . . . .	12
Abbildung 4.4	Zusammengefügte Sub-Bayes-Netze zu einem Gesamten . . . . .	14
Abbildung 4.5	Flussdiagramm Ermittlung der <i>einflussreichsten Parameter</i> . . . . .	15
Abbildung 4.6	Vorhersage <i>Fehlerursache</i> für Fehler „Kratzer“, absteigende Liste mit Wahrscheinlichkeiten . . . . .	16
Abbildung 4.7	Flussdiagramm der aktuell vorliegenden <i>Datenablage</i> . . . . .	17
Abbildung 4.8	Flussdiagramm der geplanten Datenerfassung und -verarbeitung . . . . .	17
Abbildung 4.9	Ausschnitt der Parameterliste mit Zuordnung zu möglichen Fehlern . . . . .	18
Abbildung 4.10	Architektur und Funktion <i>Shiny</i> . . . . .	19
Abbildung 4.11	Übersicht Benutzeroberfläche, vorhandene Tabs . . . . .	20
Abbildung 4.12	Tab „Bayes-Netz“ mit angezeigtem Fehler <i>Kratzer</i> . . . . .	22
Abbildung 5.1	Tab „Gesamt BayesNetz“ . . . . .	23
Abbildung 5.2	Tab „Inferenz“, Vorhersage Fehlerursache für Fehler <i>Kratzer</i> . . . . .	24
Abbildung 5.3	Tab „Parameter Lernen“ . . . . .	25
Abbildung A.1	Extrusionsblasformen (a) konventionell, (b) konventionell mit Überquetschnung, (c) Manipulationsverfahren, (d) Saugblasverfahren, (e) Einlegeverfahren aus [2] S.32ff. . . . .	28
Abbildung A.2	Prozess Kunststoff-Extrusionsblasformen aus [13] S.16 . . . . .	30
Abbildung A.3	Kontinuierliche und diskontinuierliche Extrusion nach [13] S.40 und S.42 . . . . .	31
Abbildung A.4	Bayes Netz als <i>gerichteter, azyklischer Graph (DAG)</i> . . . . .	32
Abbildung A.5	Bayes Netz Kinobesuch . . . . .	34
Abbildung A.6	Bayes Netz Kinobesuch, Bedingung <i>Thriller, Regen und abends</i> . . . . .	35
Abbildung A.7	<i>SamIam</i> Software zum graphischen Erstellen von Bayes-Netzen . . . . .	35



Abbildung A.8	Bayes-Netz „Asien“ erstellt mit <i>SamIam</i> . . . . .	36
Abbildung A.9	Struktur (DAG) Beispiel <i>Asia</i> . . . . .	37

## TABELLENVERZEICHNIS

---

Tabelle 4.1	Ausschnitt Umfrage <i>survey</i> zur Verkehrsmittelnutzung	9
Tabelle 4.2	Konzeptrahmen <i>Datenformat</i> , vollständiges Datentuple	18
Tabelle A.1	Ausschnitt der Daten <i>Asia</i> . . . . .	37
Tabelle A.2	Wahrscheinlichkeitsverteilung für ein auffälliges Bruströntgen Beispiel <i>Asia</i> . . . . .	38

## LISTINGS

---

Listing 4.1	Erstellen eines <i>DAG</i> für Bayes-Netz <i>survey</i> . . . . .	9
Listing 4.2	Parameter lernen mittels <i>Maximum Likelihood parameter estimation</i> . . . . .	10
Listing 4.3	Inferenz berechnen für $P(S = M, T = car   E = high)$ . .	11
Listing 4.4	Programmrahmen Shiny in <b>R</b> . . . . .	20
Listing 4.5	Programmausschnitt <i>ui.R</i> Tab Bayes-Netz . . . . .	21
Listing 4.6	Programmausschnitt <i>server.R</i> Tab Bayes-Netz . . . . .	22
Listing B.1	Fibonacci-folge programmiert in <b>R</b> . . . . .	39
Listing B.2	Bayes-Netz in R einlesen und bearbeiten, Beispiel <i>Asia</i>	39
Listing B.3	Mehrere Subnetze in R einlesen und zu Einem zusammenfügen . . . . .	40

## ABKÜRZUNGSVERZEICHNIS

---

IWU	Fraunhofer-Institut für Werkzeugmaschinen und Umformtechnik IWU
BPP	Berufspraktische Phase
ZIM	Zentrales Innovationsprogramm Mittelstand
KI	Künstliche Intelligenz
BN	Bayes-Netze
DAG	gerichteter, azyklischer Graph
AP	Arbeits Pakete
CPU	Central Processing Unit
GUI	Graphical User Interface

Teil I

PROJEKTARBEIT

Das Fraunhofer-Institut für Werkzeugmaschinen und Umformtechnik IWU (IWU) in Dresden gehört zur Fraunhofer-Gesellschaft zur Förderung der angewandten Forschung e. V. Das Institut für Werkzeugmaschinen und Umformtechnik gilt als treibende Kraft und ist federführend im Bereich der produktionstechnischen Forschung und Entwicklung. Tätig sind an fünf Standorten (Dresden, Chemnitz, Leipzig, Wolfsburg und Zittau) über 650 Mitarbeiter, welche unterschiedliche Branchen bedienen. Von der klassischen Automobil-Branche, über die Luft- und Raumfahrt, bis hin zur Medizin- und Mikrotechnik. Der Hauptsitz des Institutes befindet sich in Chemnitz.[4]

Das Fraunhofer IWU gilt als Erfindergeist und Wegweiser für ressourceneffiziente Produktion und zugleich intelligente Produktionsanlagen. Ihren Hauptaugenmerk legt das Institut hierbei auf spanende und umformende Prozesstechnologien. Die vollständige Prozesskette wird im Blick behalten. Immer mehr drängen sich hierbei die Themen *Kognitive Produktionssysteme*, die *vollständige Vernetzung von Produktionsanlagen* und *Künstliche Intelligenz* in den Vordergrund.

Ein großes zukunftsorientiertes Projekt ist die Produktion von Brennstoffzellen. Unter der Koordination des IWU findet deutschlandweit der Aufbau einer Forschungsplattform statt. Dort soll das technologische Potential und die kostenoptimierte Produktion weiter vorangetrieben werden. Des Weiteren gilt zu erwähnen die *E<sup>3</sup>-Forschungsfabrik Ressourceneffiziente Produktion*, welche als Ziel eine ressourcenschonende und flexible Produktion mit erneuerbaren Energien verfolgt.

Am Standort Dresden beschäftigt man sich mit den Themen Adaptronik und Akustik, Generative Fertigung sowie Mechanische Fügetechnik und Medizintechnik.[4]

Die Fraunhofer Gesellschaft ist 1949 gegründet worden und weist zur Zeit (Stand 2021) in Deutschland 75 Institute und Forschungseinrichtungen auf. Diese bieten ca. 29.000 Mitarbeiterinnen und Mitarbeitern die Möglichkeit auf einen Arbeitsplatz. Ein Großteil findet sich hiervon in der Natur- und Ingenieurwissenschaft wieder. Europaweit gilt sie als Anführer der angewandten Forschung. Neben den Instituten in Deutschland gibt es noch weitere in europäischen Ländern. Außerdem sind Institute in Nord- und Südamerika, Asien, Afrika und im Nahen Osten zu finden. Kooperationen werden weltweit mit Unternehmen und Forschungspartnern geschlossen. [3]

## EINLEITUNG

---

Die **BPP** wird auf Grundlage einer ausgeschriebenen Stelle für ein aktuell laufendes Industrieprojekt (Zentrales Innovationsprogramm Mittelstand (**ZIM**)) absolviert, für dieses eine studentische Hilfskraft gesucht wird. Das behandelte Projekt wird in Zusammenarbeit mit einem Industriepartner durchgeführt. Dieser ist im Bereich der Blasformtechnik tätig. Neben der Herstellung und Produktion von 3D-Kunststoffhohlkörpern, entwickelt und fertigt er eigene Blasformmaschinen. Diese werden sowohl für die In-House Produktion benötigt, als auch an Kunden verkauft. Er bedient und vertritt hierbei ein breites Kundenspektrum.

Als studentische Hilfskraft bekommt man möglichst viele Einblicke in verschiedene Themengebiete und Bereiche, während man eigenständig und selbstverantwortlich arbeitet. In die übertragenen Aufgaben arbeitet man sich selbstständig ein und erhält bei Bedarf Unterstützung durch seine Betreuerin. Man wird als „vollwertige Arbeitskraft“ eingesetzt und erhält somit die Möglichkeit, wichtige Beiträge zu einem Projekt zuleisten.

Im nächsten Abschnitt wird der Projektauftrag und die Aufgabenstellung detaillierter beschrieben, gefolgt durch die Kapitel Konzept und Umsetzung. Zum Abschluss wird der vorliegende Zustand des Projektes zum Ende der **BPP**, sowie ein Ausblick und offene Punkte beschrieben und aufgezeigt.

### 2.1 PROJEKTAUFTRAG / AUFGABENSTELLUNG

Der Projektauftrag/ Projektbeschreibung sieht vor für den Einrichtprozess einer Blasformmaschine einen „Wissensspeicher/ Assistenzsystem“ zu entwickeln, welcher den Einrichter/ Bediener beim Start einer neuen Produktion/ Einrichtprozess unterstützt und Tipps zur Fehlerbeseitigung vorschlägt. Die sogenannte Parametrierung von Kunststoff-Blasformmaschinen soll auf Grundlage des *wissensbasierten Systems* erfolgen.

Dem Institut wird als Teilprojekt, die Entwicklung eines **KI**-basierten Verfahrens zur Online-Qualitätskontrolle von Kunststoff-Blasformmaschinen, sowie weitere KI-Bestandteile des neuen Optimierungssystems zugeschrieben. Im ersten Schritt wird der Einrichtprozess und in einem zweiten die laufende Produktion betrachtet. Der Wissensspeicher soll auf Basis sogenannter Bayes-Netze (**BN**) aufgebaut und entwickelt werden. Der Industriepartner ist für die Datenerfassung und -Erhebung zuständig, wobei das Institut den Rahmen hierzu definiert und gegebenenfalls unterstützt.

Ziel des Projektes ist es, den Einrichtprozess kosten- und zeiteffizienter zu gestalten und gleichzeitig mehr Objektivität in den Prozess zu bringen. So soll das Wissen der einzelnen Bearbeiter hinsichtlich der Wirkung der Parameter, zusammen mit allgemein bekannten Regeln und Besonderheiten der

Maschine, maschinenlesbar und dauerhaft gespeichert werden. Den Produktionsprozess beeinflussende Eingangsgrößen müssen identifiziert und Möglichkeiten zur automatischen Erfassung entwickelt werden. Es ist hierbei auf eine Sicherstellung der erforderlichen Datenqualität sowie eine intelligente Datenhaltung zu achten. Im Anschluss ist ein Dashboard-Modul gefordert, welches dem Bediener Unterstützung bei der Einrichtung der Blasformmaschine leistet und als Interaktionsmöglichkeit mit dem System dient.[5]

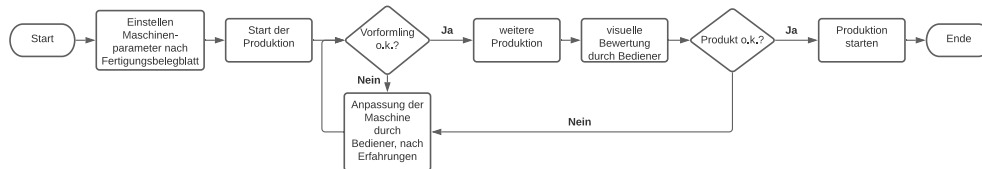


Abbildung 2.1: Flussdiagramm *aktueller Stand* Einrichtprozess Blasformmaschine

Abbildung 2.1 zeigt den aktuellen Einrichtprozess, wie er zur Zeit praktiziert wird. Es ist zu sehen, dass die Tätigkeiten des „Einrichters“ lediglich auf *Erfahrungen und Wissen* basieren. Durch das Projekt wird angestrebt, das vorhandene Wissen in einem System „dauerhaft“ zu speichern und somit nach „Ausscheiden“ eines erfahrenen Mitarbeiters, das Wissen weiterhin zu verwenden. Neben dem eigenen Wissen wird ein Fehlerbaum Abbildung 2.2 verwendet, welcher „über die Jahre hinweg“ entworfen und weiterentwickelt wurde. Zu einem bestehenden Fehler lassen sich somit mögliche Ursachen identifizieren.

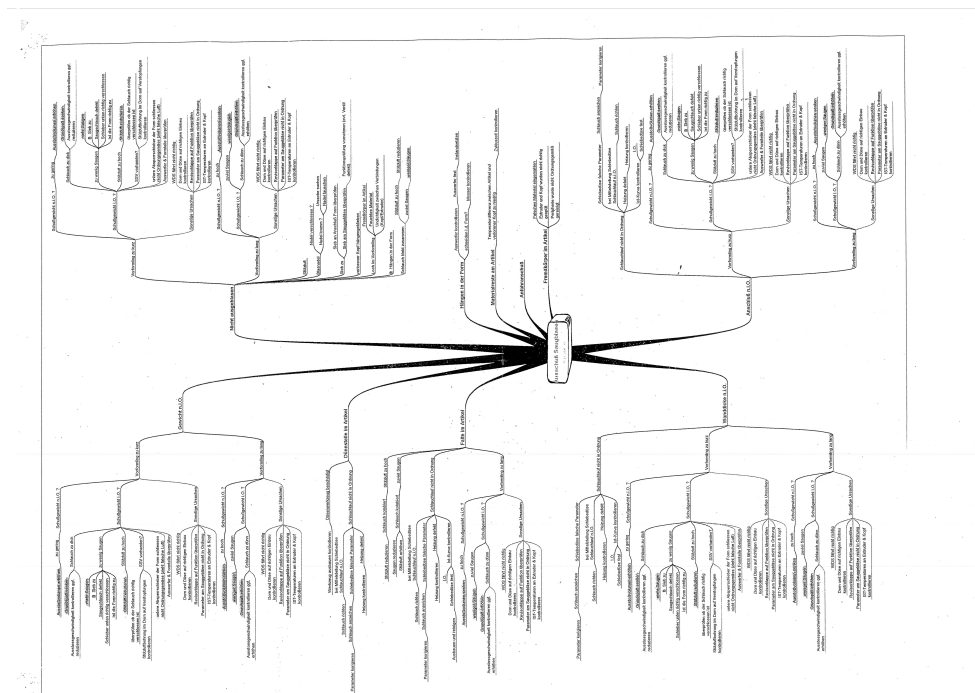


Abbildung 2.2: Fehlerbaum für Einflussfaktoren bei Blasformen Prozessen (Kunden Dokument) [7]

## KONZEPT UND AUFBAU

### 3.1 KONZEPT

In diesem Kapitel geht es darum das Konzept und eine mögliche Umsetzung des Projektes zu beschreiben. Ein klar definierter Rahmen wird durch den Projektauftrag und die -Beschreibung zu Grunde gelegt. Innerhalb diesem wird das Projekt in einzelne Arbeitspakete unterteilt und individuell erläutert. Es gilt daher, kein gesamtes Konzept für das Projekt zu entwerfen, sonder „mehrere kleinere Konzepte für die einzelnen Arbeitspakete“.

Ein allgemeiner Lösungsansatz, welcher den finalen Prozess beschreiben soll, zeigt [Abbildung 3.1](#). Dieser ist aus der Projektbeschreibung heraus entwickelt. Er soll die Unterstützung und Eingriffsmöglichkeit in den Prozess des Einrichtens beschreiben. Ein *Zielvektor* soll verwendet werden, wodurch die Güte eines Produktes beschrieben werden kann und dieser somit Einfluss auf die Berechnungen und Ausgaben des Assistenzsystems besitzt. Es wird in diesem Lösungsansatz lediglich der Einrichtprozess betrachtet. Ein Eingriff während der Produktion ist hier nicht vorgesehen. Es ist mit dem Industriepartner vereinbart, in einem ersten Schritt Einrichtprozess und Produktion getrennt von einander zu betrachten und die Produktion zunächst außen vor zu lassen.

Der Einrichtprozess ermöglicht ebenfalls eine Unterteilung in zwei Phasen, die des Vorformlings und die des fertigen Erzeugnisses. Identische Vorgehensweise ist dort, dass zunächst die Phase des Vorformlings betrachtet wird und in einem Zweiten das fertige Erzeugnis. Sobald die Anforderungen für den Vorformling erfüllt sind (nach Prüfprotokoll), wird in Phase zwei übergegangen und das fertige Produkt beurteilt und gegebenenfalls die Maschine nachgestellt bzw. Parameter angepasst.

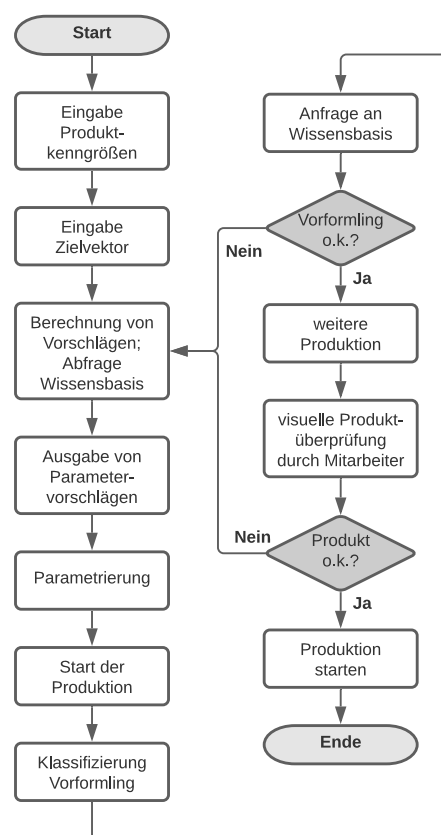


Abbildung 3.1: Allgemeiner Lösungsansatz



Der Aufbau und die Entwicklung des Assistenzsystems werden in zwei Arbeitspakete untergliedert. AP 1 ist die "Datenerfassung und -erhebung" und AP 2 der Aufbau des "Assistenzsystems". Diese Arbeitspakete werden im Folgenden detaillierter beschrieben und erläutert.

#### AP 1: Datenerfassung und -Erhebung

Das Assistenzsystem braucht Daten sowohl zum Trainieren der eigenen Netze, als auch um den aktuellen Zustand der Maschine zu kennen. Die Datenerfassung bildet die *Schnittstelle* zwischen Maschine und Wissensspeicher. Das Institut definiert hierbei den benötigten Rahmen und das Format der Daten. Die finale Umsetzung liegt in den Händen des Industriepartners. Die benötigten Daten lassen sich in zwei bzw. drei Kategorien einteilen. Zum Einen sind es Maschinenparameter, welche in der Steuerung hinterlegt sind und automatisch erfasst werden können und zum Anderen sind es „manuelle Inputs“, hierzu gehören die Fehlerbilder und Parameter, welche nicht automatisch erfasst werden können, sondern die Eingabe des Bedieners erfordern. Wie die Erfassung der Daten implementiert wird, liegt nicht in Händen des Institutes. Das Konzept sieht vor, dass ein Rahmen für das Format der Daten definiert und dem Kunden übergeben wird. Neben diesem Rahmen ist eine weitere Aufgabe zu definieren, welche Daten benötigt werden. Die Daten sollen an einem zentralen Punkt abgelegt sein und dem Assistenzsystem Zugriff bieten. Auf Grundlage der Datenbasis ist es somit möglich das BN und die dazugehörigen Parameter zu lernen und zu trainieren.

#### AP 2: Assistenzsystem

Dieses Arbeitspaket ist das wesentlich größere und aufwendigere, da dort der Wissensspeicher vollständig entworfen und entwickelt wird. In der Projektbeschreibung wird definiert, dass das Assistenzsystem mit Hilfe von Bayes-Netzen arbeiten soll. Hierfür werden für Fehler die Einflussparameter ermittelt und erfasst. Aus der Sammlung dieser einzelnen Bayes-Netze wird ein gesamtes Hauptnetz erstellt. Durch AP 1 verfügt das System über Daten und kann somit das Netz „lernen und trainieren“. Es wird ein Algorithmus entwickelt, mit dem der einflussreichste Parameter zur Behebung des Fehlers ermittelt wird. Zum Abschluss dient ein Dashboard als Schnittstelle und Interaktionsmöglichkeit zum Menschen. Das System hat Zugriff auf die Datenbank und liefert dem Einrichter Hinweise, welche dem Beheben von Fehlern dienen.

Abbildung 3.2 zeigt die Interaktion zwischen Mensch und dem Assistenzsystem und dem geplanten Prozess.

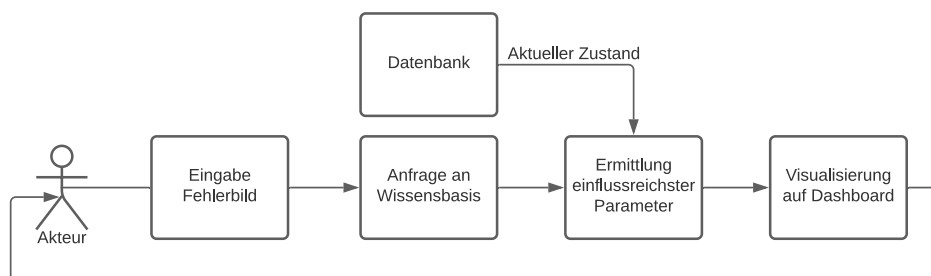


Abbildung 3.2: Flussdiagramm Interaktion Mensch und Assistenzsystem

### 3.2 GLIEDERUNG UND AUFBAU

Die folgende Übersicht soll den Ablauf des Projektes in der Umsetzung bzw. der [BPP](#) veranschaulichen und aufzeigen, welche Tätigkeiten und Themenfelder behandelt werden. Neben der Umsetzung einzelner Arbeitspakete, bedarf es zuvor eine Einarbeitung in bestimmte Themengebiete, da innerhalb des Studiums keine Erfahrungen bzw. Kenntnisse gesammelt worden sind.

#### Themenfelder und Umsetzungen

- Einarbeiten in die Thematik Kunststoff-Extrusionsblasformen und Blasformmaschinen ([Abschnitt A.1](#))
- Einlesen und Auseinandersetzen mit der Thematik Bayes Netze ([Abschnitt A.2](#))
- Einlesen und Einarbeiten in die Programmierung mit R ([Abschnitt A.3](#))
- Einarbeiten in das R-Package „bnlearn“ und Aufbau Versuchsnetz ([Abschnitt 4.1](#))
- Einarbeiten in das graphische Entwerfen eines Bayes-Netzes mittels BayesianFusion „GeNIe“ bzw. „SamIam“ ([Abschnitt A.4](#))
- Verknüpfung „R bnlearn“ und „SamIam“ ([Abschnitt A.5](#))
- Erstellen der Bayes-Netz Struktur, Identifizieren der Einflussparameter auf einen Fehler ([Abschnitt 4.2](#))
- Entwerfen eines Algorithmus zur Ermittlung des einflussreichsten Parameter auf ein Fehlerbild ([Abschnitt 4.3](#))
- Datenerhebung und -verarbeitung, Erstellen der Vorgaben für Industriepartner und Zuarbeit ([Abschnitt 4.4](#))
- Erstellen einer Benutzeroberfläche mittels „Shiny“ zur Interaktion für den Bediener ([Abschnitt 4.5](#))

Im Anschluss an [Kapitel 4](#) Umsetzung wird der Stand der Übergabe zum Ende der [BPP](#) dargestellt. Da es sich um ein Projekt mit zweijähriger Laufzeit handelt, wird dieses innerhalb der zwölf Wochen nicht abgeschlossen. Ebenfalls gilt zu erwähnen, dass der *KickOff* des Projektes vor Beginn meiner berufspraktischen Phase durchgeführt wurde.

## UMSETZUNG

---

In diesem Abschnitt wird die Umsetzung einzelner Arbeits Pakete (AP) und das Gesamtprojekt erläutert. Für die behandelten Themenfelder, eingesetzte Software und Programmiersprache liegen keine Kenntnisse vor. Bevor mit der Umsetzung eines AP begonnen wird, findet eine jeweilige Einarbeitung statt. Im [Anhang A](#) wird detaillierter auf die unterschiedlichen Einarbeitungen eingegangen. In [Abschnitt A.1](#) finden Sie einen Einstieg in das Thema *Blasformen*, den Prozess und das eingesetzte Verfahren *Extrusionsblasformen mit Überquetschen*. [Abschnitt A.2](#) gibt einen kurzen Einblick in das Thema Bayes-Netze, welcher als Grundlage für das vorliegende Projekt dient. In [Abschnitt A.3](#) folgt die eingesetzte Programmiersprache R. Die verwendete Software zum Erstellen der Bayes-Netze ist in [Unterabschnitt A.4.1](#) und [Unterabschnitt A.4.2](#).

Die kommenden Abschnitte beschreiben die Umsetzung einzelner Arbeitspakete und gleichzeitig den zeitlichen Verlauf der Entwicklung des Projektes. Es werden einzelne Abschnitte des Projektes separat betrachtet und umgesetzt. Begonnen wird mit dem R-package *bnlearn* und Aufbau eines Versuchsnetzes.

### 4.1 R-PACKAGE *bnlearn* UND AUFBAU VERSUCHSNETZ

In diesem Abschnitt wird nun das R-package *bnlearn* betrachtet und ein Versuchsnetz für weitere Tests aufgebaut. Im ersten Schritt wird sich in die Dokumentation dieses Paketes eingelesen.<sup>1</sup> Das R-package ist entwickelt worden, um die graphische Struktur eines Bayes-Netzes zu erlernen, Parameter vorherzusagen und Inferenzen zu bestimmen. Herausgebracht wurde es das erste Mal im Jahr 2007 und wird seit dem ständig weiterentwickelt.

Die Dokumentation enthält nützliche Beispiele und auch Beispieldaten, mit denen Parameter erlernt werden können. Auf Grundlage eines Datensatzes aus der Sammlung stammt das später in *Samlam* erstellte Bayes-Netz „Asia“ (s.[Abbildung A.8](#)). Neben der Dokumentation wird das Werk *Bayesian Networks with Examples in R* von „M. Scutari“ und „J.-B. Denis“ verwendet, welches eng mit der Dokumentation zusammenarbeitet.[\[12\]](#)

Im Projekt werden die Daten immer *diskret* vorliegen. Als weitere Möglichkeit gibt es den Fall der *kontinuierlichen* Werte, auch *Gauß Bayes-Netze* genannt. Diesen Fall werden wir jedoch nicht weiter betrachten. Aus dem Buch *Bayesian Networks with Examples in R*[\[12\]](#) wurde das erste Kapitel *The Discrete Case: Multinomial Bayesian Networks* bearbeitet und somit der grundlegende Umgang, die Programmierung und Verwendung von R und *bnlearn* erlernt.

---

<sup>1</sup> Dokumentation des R-package *bnlearn* <https://www.bnlearn.com/>

Als Grundlage für das Beispiel aus dem Kapitel dient eine Umfrage. Man möchte das bevorzugte Verkehrsmittel einer Person vorhersagen können. In der Umfrage werden sowohl Frauen, als auch Männer befragt. Das Alter wird hierbei in drei Kategorien eingeteilt, außerdem unterscheidet man das Geschlecht, die Bildung, die Wohnlage und die Anstellung. In [Tabelle 4.1](#) ist ein Ausschnitt der Daten aus der Umfrage zur Verkehrsmittelnutzung zu sehen. Insgesamt werden 300 Personen unterschiedlicher Kategorien befragt.

AGE	RESIDENCE	EDUCATION	OCCUPATION	SEX	TRAVEL
adult	big	high	emp	F	car
adult	small	uni	emp	M	car
adult	big	uni	emp	F	train
adult	big	high	emp	M	car
adult	big	high	emp	M	car
adult	small	high	emp	F	train
young	small	uni	emp	M	other
old	big	high	self	F	car

Tabelle 4.1: Ausschnitt Umfrage *survey* zur Verkehrsmittelnutzung

Anhand von *Expertenwissen* wird die Struktur des Bayes-Netzes erstellt. Neben der Möglichkeit die Struktur eigenständig zu definieren, besteht die Möglichkeit aus den vorhandenen Daten über Algorithmen die Struktur zu erlernen. Da jedoch im Projekt die Struktur durch „Experten“ definiert und erstellt wird, kann diese Variante vernachlässigt werden. Nachfolgend sind zwei Varianten aufgezeigt, die Struktur in R zu definieren, s.[Listing 4.1](#). Im ersten Fall wird ein leerer Graph mit Knoten erzeugt und anschließend die Abhängigkeiten übergeben. Im zweiten Fall wird die Struktur durch eine formale Gleichung erzeugt. Hierbei werden in eckigen-Klammern die Knoten definiert und hinter einem | die *Eltern*, wenn vorhanden.

```
# Erster Weg ein DAG erstellen zu koennen
dag1 <- empty.graph(nodes = c("A", "S", "E", "O", "R", "T"))
arc.set <- matrix(c("A", "E",
                    "S", "E",
                    "E", "O",
                    "E", "R",
                    "O", "T",
                    "R", "T"),
                 byrow = TRUE, ncol = 2,
                 dimnames = list(NULL, c("from", "to")))
arcs(dag1) <- arc.set
# Zweiter Weg ein DAG zu erstellen, ueber formale Gleichung mit
  Abhaengigkeiten
dag2 <- model2network("[A][S][E|A:S][O|E][R|E][T|O:R]")
```

Listing 4.1: Erstellen eines DAG für Bayes-Netz *survey*

Das Resultat ist in [Abbildung 4.1](#) zu sehen. Die *Wurzelknoten* sind das Alter (*A*) und das Geschlecht (*S*), der *Blattknoten* ist das Verkehrsmittel (*T*). Als *Wurzelknoten* bezeichnet man diejenigen, welche keine Eltern besitzen. Der umgekehrte Fall liegt bei *Blattknoten* vor, diese besitzen keine Kinder. Man könne auch von einem Anfang (*Wurzelknoten*) und Ende (*Blattknoten*) des Graphen sprechen.

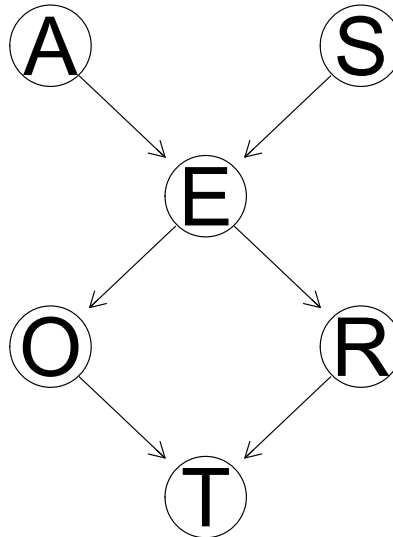


Abbildung 4.1: DAG aus Bayes-Netz *survey*

Das Erstellen der Struktur für den späteren Wissensspeicher wird der erste essentielle Punkt in der Umsetzung des Projektes sein. Der zuvor gezeigte Code dient als Beispiel ein Bayes-Netz direkt mittels *bnlearn* in R umzusetzen. Es besteht zusätzlich die Option, die Struktur mittels einer Software „graphisch“ zu erstellen und diese in R zu importieren. Der Einstieg und die Möglichkeiten der graphischen Umsetzung wird in [Abschnitt A.4](#) beschrieben.

Nachdem die Struktur des Netzes erstellt ist, kann mittels der eingelesenen Daten das *Parameter Lernen* durchgeführt werden. Hierfür bietet R verschiedene Algorithmen, default mäßig wird *Maximum Likelihood parameter estimation* verwendet. Nachfolgend ist die Codezeile ([Listing 4.2](#)) zu sehen mit welcher die Parameter gelernt werden und ein fertiges Bayes-Netz erzeugt wird. Verpflichtend müssen die Struktur und Daten für das Lernen übergeben werden. In diesem Fall wird der default-Algorithmus ebenfalls mit angegeben. Die Zustände, welche ein Knoten annehmen kann, werden aus den Daten heraus extrahiert und müssen nicht vorher definiert werden.

---

```

# learn parameter with data
bn.mle <- bn.fit(dag = dag1, data = survey, method = "mle")

```

---

Listing 4.2: Parameter lernen mittels *Maximum Likelihood parameter estimation*

Mittels den Schritten *Struktur definieren* und *Parameter lernen* ist ein vollständiges und zur weiteren Verwendung verfügbares *Bayes-Netz* erzeugt.

Mit diesem ist es nun möglich Inferenzen für bestimmte Fälle zu berechnen und Vorhersagen zu treffen. Auf der rechten Seite ist das Balkendiagramm für eine berechnete Inferenz dargestellt (s. [Abbildung 4.2](#)). Es werden die Wahrscheinlichkeiten des Verkehrsmittels mit gegebener Wohnlage und Arbeitsverhältnis berechnet. Mathematisch wird dies  $P(T|R,O)$  geschrieben. In der oberen Zeile ist zu sehen, dass das bevorzugte Verkehrsmittel in einer großen Stadt das Auto ist. In der unteren Zeile verändert

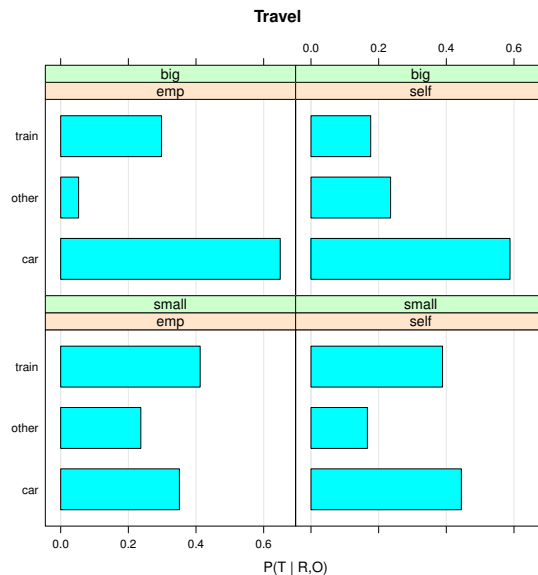


Abbildung 4.2: Balkendiagramm Verkehrsmittel-nutzung

sich das Verhältnis etwas, Zug und Auto gleichen sich aus. Da es sich hierbei um keine realen Umfragedaten handelt, findet keine Wertung statt. Bei einer realen Umfrage würde man vermuten, dass die beiden Zeilen vertauscht würden und in der Großstadt zumeist der Zug verwendet wird, wohingegen sich in einer kleineren Stadt Auto und Zug ausgleichen.

Möchte man die Inferenz für einen bestimmten Fall ermitteln, ist auch dies möglich. In [Listing 4.2](#) wird die Berechnung für den Fall durchgeführt, dass das Geschlecht *männlich* ist und als Verkehrsmittel das *Auto* nutzt, gegeben der Bildung *high*; mathematischer Ausdruck  $P(S = M, T = car | E = high)$ . Der einzutretende Fall wird *event* genannt und gegebene Attribute *evidence*.

---

```
# Approximate Inference
# P(S = M, T = car | E = high)
cpquery(bn, event = (S == 'M') & (T == "car"), evidence = (E == "high"))
```

---

Listing 4.3: Inferenz berechnen für  $P(S = M, T = car | E = high)$

Die Berechnung liefert ein Ergebnis von 0.3487621, welches einer Eintrittswahrscheinlichkeit von 34.876% entspricht. Es lässt sich somit aussagen, dass in 34.876% der Fälle die Person *männlich* ist, als Verkehrsmittel ein *Auto* nutzt, gegeben, dass die Bildung *high* ist.

Nach Abschluss des ersten Kapitels aus *Bayesian Networks with Examples in R* liegen Kenntnisse für das eigene Erstellen und Bearbeiten von Bayes-Netzen vor. Im nächsten Abschnitt wird das Erstellen der Bayes-Netz Struktur beschrieben.

## 4.2 ERSTELLEN BAYES-NETZ STRUKTUREN

Der Prozess des Extrusionsblasformens kann in zwei Phasen unterteilt werden.

- Phase 1: Erzeugen des *Vorformlings*
- Phase 2: Herstellen des *fertigen Erzeugnisses/ Produktes*

Die eventuell auftretenden Fehler werden den beiden Phasen zugeordnet. So gibt es eine Fehlerliste für die Phase *Vorformling* und eine für das *fertige Erzeugnis*. Der Industriepartner stellt diese beiden Listen zur Verfügung. Die Struktur des Bayes-Netzes soll durch Expertenwissen vorgegeben werden. Hierfür findet im wöchentlichen Rhythmus ein 90-minütiger Termin mit dem Industriepartner statt, bei welchem gemeinsam Einflussparameter für einen Fehler und Abhängigkeiten untereinander definiert werden. Die Struktur wird eigenständig festgelegt und aufgebaut. Als Ergebnis wird für jeden Fehler jeweils ein individuelles Subnetz erzeugt, welches am Ende zu einem Gesamten integriert wird.

Die einzelnen Bayes-Netze werden mit der in [Unterabschnitt A.4.2](#) erläuterten Software *SamIam* erzeugt. Das Hauptnetz wird jedoch in *R* gebildet, da dort die weitere Verarbeitung stattfindet.

In [Abbildung 4.3](#) ist das erstellte Bayes-Netz für den Fehler *Kratzer* in der Phase 1 *Vorformling* zu sehen. Als Einflussparameter sind definiert: *Verunreinigung des Rohstoffes*, *Beschädigung Düse*, *verbrannter Rohstoff auf Düsenoberfläche* und *Düsentemperatur*, wobei die Düsentemperatur neben dem Einfluss auf den Fehler *Kratzer* auch Einfluss auf den *verbrannten Rohstoff auf Düsenoberfläche* aufweist. Insgesamt werden für die Phase 1 *Vorformling* 13 Sub-

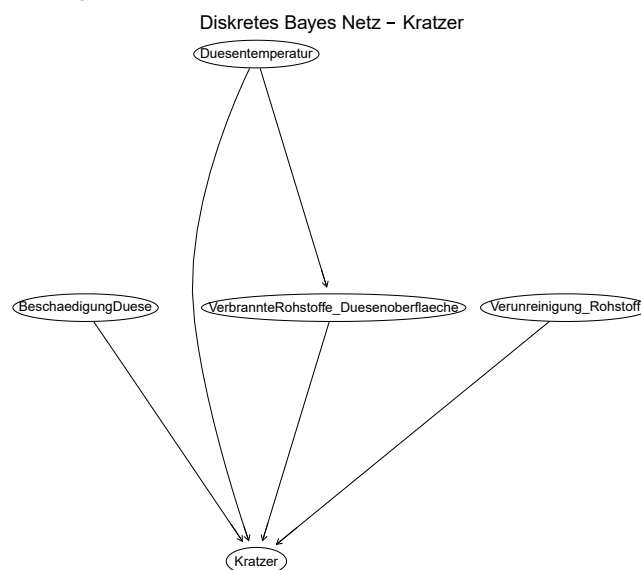


Abbildung 4.3: Sub-Bayes-Netz für Fehler „Kratzer“ erstellt mit *SamIam*

Bayes-Netze erstellt, welche anschließend mittels *bnlearn* zu einem gesamten Netz zusammengefügt werden. Im Nachfolgenden ist eine Liste definierter Fehler zu sehen, welche während des Einrichtprozesses auftreten können.



### Fehlerliste Einrichtprozess Phase „Vorformling“

- |  |                                |
|--|--------------------------------|
| 1. Kratzer   | 7. Durchhängen                 |
| 2. Schmelzbruch                                      | 8. Schlauchkrepeln             |
| 3. Wolkenbildung/ milchig aussehender kalte Schlauch | 9. Schlauchlauf, Geradeauslauf |
| 4. Vorformling glänzend                              | 10. Darmeffekt                 |
| 5. Blasenbildung                                     | 11. Raum/ Qualmbildung         |
| 6. Gardineneffekt                                    | 12. Schlauchlängenschwankungen |

Für das Zusammensetzen in R zu einem Hauptnetz wird ein eigens entworfener Algorithmus (s. [Listing B.3](#)) verwendet. Im ersten Schritt werden die Namen der einzulesenden Sub-Netze angegeben und bei Aufruf der Funktion `readIn_multiNet()` übergeben. Als Rückgabewert erhält man eine Liste mit den einzelnen eingelesenen Sub-Netzen, diese übergibt man im Anschluss der Funktion `buildBNstring()`, welche einen gemeinsamen „Modelstring“ erzeugt. In einem solchen Modelstring werden die Knoten und ihre Abhängigkeiten (Kanten) definiert. Besitzt ein Knoten Eltern, so wird dies formal mit angegeben.

Die formale Gleichung für das Sub-Bayes-Netz „Kratzer“ aus [Abbildung 4.3](#) sieht wie folgt aus:

$$[BeschaedigungDuese][Duesentemperatur][VerbrannteRohstoffe\_Duesenoberflaeche|Duesentemperatur][Verunreinigung\_Rohstoff][Kratzer|BeschaedigungDuese : Duesentemperatur : VerbrannteRohstoffe\_Duesenoberflaeche : Verunreinigung\_Rohstoff]$$

(4.1)

Der „String“ wird anschließend in die für das Erstellen eines *gerichteten azyklischen Graphen* benötigte formale Gleichung umgewandelt. Da dieser möglicherweise Dubletten aufweist, müssen diese entfernt werden. Diese Aufgabe und das Erzeugen des DAG's erfüllt die Funktion `modelstring_to_dag()`. Als Ergebnis erhält man ein großes zusammenhängendes Bayes-Netz, das sogenannte „Hauptnetz“.

[Abbildung 4.4](#) zeigt die ersten sieben Fehler aus der Phase *Vorformling*, welche zu einem Bayes-Netz zusammen gesetzt sind. Die so erstellten Strukturen der Bayes-Netze für Phase 1 *Vorformling* und Phase 2 *fertiges Erzeugnis* (noch nicht umgesetzt) werden abschließend ebenfalls zusammengefügt, somit entsteht eine Struktur über beide Phasen.



### Diskretes Bayes Netz – Gesamtübersicht

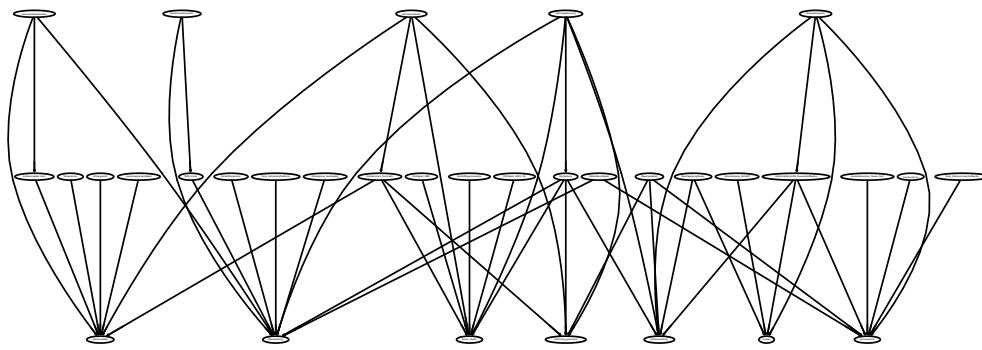


Abbildung 4.4: Zusammengefügte Sub-Bayes-Netze zu einem Gesamten

Ein Bayes-Netz besitzt Vorteile, da es auch mit wenigen Abhängigkeiten „gute“ Vorhersagen treffen kann. Gleichzeitig reduziert die geringe Menge an Abhängigkeiten, die benötigte Datenmenge, die es braucht, dass verwertbare Ergebnisse erzeugt werden können. Als Faustformel gilt, dass man pro Kombination mind. 10 Datensätze benötigt. Beim Entwurf der Struktur gilt es daher die gesamte Anzahl der Abhängigkeiten, sowie die Kantenanzahl auf einen Knoten, auf ein Minimum zu reduzieren.

#### 4.3 ERMITTLUNG EINFLUSSREICHSTER PARAMETER

In diesem Kapitel befassen wir uns mit einer der „Leitfragen“ des Projektes, wie es möglich ist den einflussreichsten Parameter/ die Fehlerursache auf einen anstehenden Fehler zu selektieren und ihn dem Bediener erkenntlich zu machen. Für die Ermittlung wird ein selbst entwickelter Algorithmus verwendet.

Als Voraussetzung gilt, dass der Bediener ein aktuell vorliegendes Fehlerbild auswählt. Im Anschluss werden die für diesen Fehler möglichen zu veränderbaren Einflussparameter selektiert. Im weiteren Verlauf werden innerhalb mehrerer Schleifen unterschiedliche *Inferenzen* für den eingetretenen Fehler und die selektierten Parameter ermittelt. Hierbei werden die Inferenzen zu jedem möglichen Zustand eines Parameters „Y“, mit dem *event*, dass der Fehler eingetreten ist, berechnet;  $P(X = true | Y = lvi)$  mit  $i = 1 \dots n - lvi$ . Die einzelnen berechneten Wahrscheinlichkeiten werden in einer Liste gespeichert. Für jeden Parameter werden Differenzen zwischen zwei möglichen Zuständen berechnet und anschließend in einer weiteren Liste gespeichert. Der Gedanke bei diesem Vorgehen: *Die größte Differenz gibt die beiden Zustände eines Parameters an, wenn man von Zustand „a“ in Zustand „b“ wechselt, dass man die Wahrscheinlichkeit des Auftretens des Fehlers minimiert.*

Abschließend wird die Liste aus Wahrscheinlichkeitsdifferenzen aller selektierten Parameter und ihrer Zustände absteigend sortiert. An oberster Stelle steht somit der *einflussreichste Parameter* mit Zustandswechsel (s. [Ab-](#)

**bildung 4.6** Beispiel Fehler „Kratzer“). Der beschriebene Algorithmus wird in **Abbildung 4.5** durch ein Flussdiagramm dargestellt. Es wird hierbei nicht berücksichtigt, ob dieser Zustandswechsel im vorliegenden Fall möglich ist, da im ersten Schritt die Maschinenanbindung nicht umgesetzt wird. Der Bediener muss dem System eine Rückmeldung geben, ob der Eingriff möglich ist oder nicht. Wenn dieser nicht möglich ist, dann erhält er den nächsten Vorschlag, mit der nächst kleineren Wahrscheinlichkeitsdifferenz. Im Sinne einer KI, ist die Anbindung an die Maschine und somit das Abfragen des aktuellen Zustandes unumgänglich. Bei aktuell vorliegendem Stand arbeitet der Algorithmus auf einer *offline Datenbasis*.

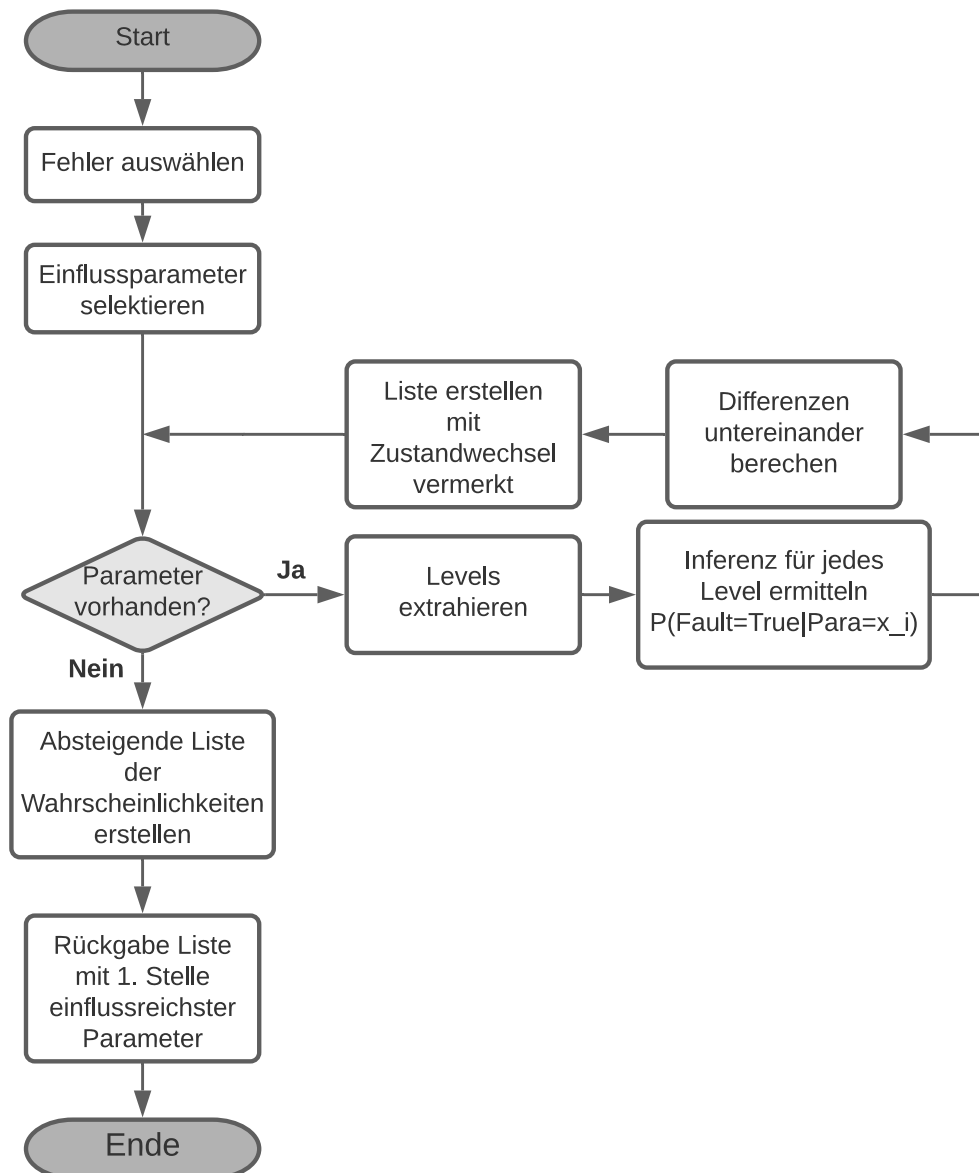


Abbildung 4.5: Flussdiagramm Ermittlung der *einflussreichsten Parameter*

Für den Algorithmus werden mehrere Funktionen definiert und entworfen. Neben der eigentlichen Funktion zur Berechnung der Inferenz, werden

Funktionen definiert, welche die möglichen Einflussparameter aus dem gesamten Bayes-Netz extrahieren, die Zustände eines Parameters ermitteln, Differenzen zwischen den Inferenzen der Zustände eines Parameters berechnen und die Wechselrichtung der Zustände vertauschen, wenn die Differenz einen negativen Wert aufweist. Aufgrund des Umfangs wird auf eine detaillierte Beschreibung der einzelnen Funktionen verzichtet.

Durch diesen Algorithmus ist es möglich die Fehlerursache für ein vorliegendes Fehlerbild vorherzusagen. Im Idealfall soll durch Beseitigen der Fehlerursache, das Fehlerbild vollständig behoben sein, in der Realität werden Fälle auftreten, bei denen mehrere Eingriffe getätigt werden müssen. Daher wurde das Erstellen einer Liste getätigt, um mögliche weitere Fehlerursachen nennen zu können.

**Einflussparameter**

Show  entries Search:

prob	from	to	node
99.30	Ja	Nein	BeschaedigungDuese
28.89	Nein	Ja	VerbrannteRohstoffeDuesenoberflaeche
28.54	200	190	Temp_Duese
0.14	Nein	Ja	VerunreinigungRohstoff

Showing 1 to 4 of 4 entries Previous **1** Next

Abbildung 4.6: Vorhersage Fehlerursache für Fehler „Kratzer“, absteigende Liste mit Wahrscheinlichkeiten

#### 4.4 DATENERHEBUNG UND -VERARBEITUNG

Die Datenerfassung ist ein eminentes Arbeitspaket in diesem Projekt, da ohne Daten, der Wissensspeicher seine Netze nicht lernen kann und die Erfahrungen aus vorhergehenden Einrichtprozessen nicht gespeichert werden. Daten werden benötigt, damit das Bayes-Netz seine Intelligenz erhält.

In [Abschnitt 3.1 Absatz „Datensammlung“](#) wird kurz darauf eingegangen, dass die Daten, welche erhoben werden sollen, in zwei bzw. drei Kategorien zu unterteilen sind. Die Daten, welche in der Steuerung abgelegt sind werden zum aktuellen Zeitpunkt vom Industriepartner archiviert. [Abbildung 4.7](#) zeigt den vorliegenden Archivierungsprozess. Dort werden die Maschinenparameter in einem fünf Minuten Zyklus abgefragt und lokal

gespeichert. Da es sich hierbei um eine Umlaufablage handelt und diese überschrieben wird, werden für eine endgültige und dauerhafte Ablage, die Daten in einem Netzlaufwerk abgelegt.

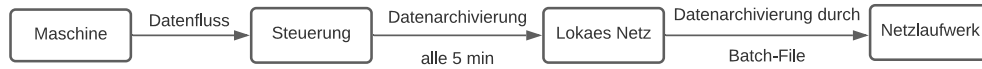


Abbildung 4.7: Flussdiagramm der aktuell vorliegenden *Datenablage*

Die vorgefundene Architektur kann für das Projekt allerdings nicht verwendet werden, da man lediglich Zugriff auf Daten besitzt, welche fünf Minuten „alt“ sind, es jedoch den Zugriff auf „Echtzeitdaten“ benötigt. Somit wird eine direkte Schnittstelle an die Steuerung geplant. Die Datenerfassung sowie -erhebung liegt auf Kundenseite, es ist daher seine Aufgabe, die Schnittstelle und Möglichkeit auf Echtzeitdaten zu erzeugen. [Abbildung 4.8](#) zeigt eine mögliche Umsetzung.

Hierbei wird auf die Central Processing Unit (CPU) zugegriffen. Die abgezogenen Daten werden anschließend in einer Datenbank abgelegt, worauf der Wissenspeicher ebenfalls Zugriffsrechte besitzt. Er kann den aktuellen Stand der Maschine abfragen und seine Netze mit neuen „Erfahrungen“ lernen. Die Hilfestellungen des Assistenzsystems werden auf einem Dashboard dargestellt und der Bediener greift in den Prozess ein. Diese Beschreibung dient dem Kunden als mögliches Konzept bzw. Hilfestellung. Eine Umsetzung von Seiten des Institutes findet aktuell nicht statt.

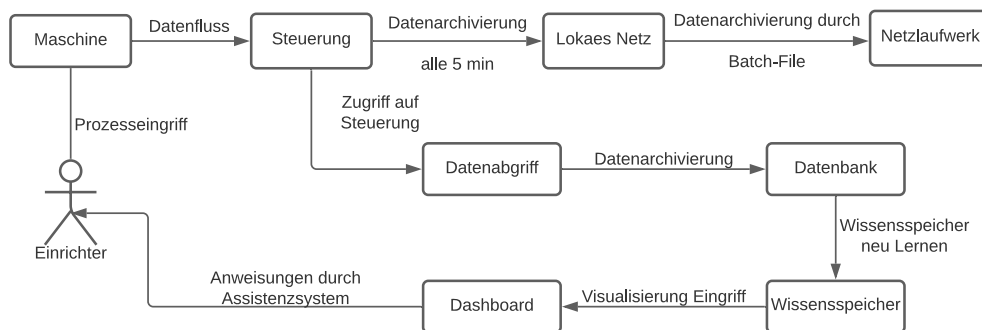


Abbildung 4.8: Flussdiagramm der geplanten Datenerfassung und -verarbeitung

Gemeinsam mit dem Industriepartner werden aus den erstellten Sub-Bayes-Netzen zu möglichen auftretenden Fehlern die benötigten Daten/ Parameter identifiziert und aufgelistet. Dies dient der Definition, welche Daten der Wissenspeicher benötigt. [Abbildung 4.9](#) zeigt einen Ausschnitt der erstellten Liste. Dort ist neben der Zuordnung zum jeweiligen Fehler, auch der Bayes-Netz Name sowie Eingliederung der Parameterart zu finden. In einer weiteren Spalte werden mögliche Zustände eines Knoten definiert. Da mit diskreten Variablen gearbeitet wird, besitzen viele Knoten lediglich zwei Zustände. Dies ist auch im Sinne der Bayes-Netze, da so die Anzahl der möglichen Kombinationen verringert werden kann. Bei Parametern, die einen numerischen Wert überliefern, werden Überlegungen getroffen, wie mit diesen verfahren wird. Eine Möglichkeit wäre eine Einteilung in Intervalle um

somit die Anzahl möglicher Zustände zu begrenzen. Dieser Punkt ist zum aktuellen Zeitpunkt jedoch noch offen und wird weiterhin diskutiert.

Einflussparameter			Vorkonzepte bei Vorformling							
Parameter	Erhebung	Wert	Name in Bayes-Netz	Kratzer	Schmelzdruck	Schleusenbildung	Vorformling/Duesentemp	Blasenbildung	Rauch/Qualität	Gardienstoff
Verbrannter Rohstoff auf Düsenoberfläche	visuell	Ja / Nein	VerbrannterRohstoffDuesenoberfläche	x	x	x				
Verunreinigung des Rohstoffes	visuell	Ja / Nein	VerunreinigungRohstoff	x						
Düsentemperatur	messbar	numerischer Wert	Temp_Duese	x	x	x				
Beschädigung Düse	visuell	Ja / Nein	BeschädigungDuese	x		x				
Materialfeuchtigkeit	visuell	i.O. / zu nass / zu trocken	Materialfeuchtigkeit		x			x		
Innenoberfläche Extrudertemperatur	messbar	numerischer Wert	Temp_Extruderenoberfläche		x					
Regeneratanteil	visuell	numerischer Wert oder i.O. / zu viel / zu wenig	Regeneratanteil		x	x	x			
Ausstossdruck	messbar	numerischer Wert	Ausstossdruck		x					
Außenstemperatur Düse und Kopf	messbar	numerischer Wert	Temp_AussenDuse/Kopf		x					
Kopfheizung	messbar	numerischer Wert	Kopfheizung			x	x	x	x	
Heizungssteuerung	messbar	i.O. / n.i.O.	Heizungssteuerung			x	x	x	x	
Extruderdruck & Durchsatz	messbar	numerischer Wert	Extruderdruck/Durchsatz				x		x	x
Extruderdrehzahl	messbar	numerischer Wert	Extruderdrehzahl				x		x	x
Düsen-Kernkombination	visuell	i.O. / vergrößern / verkleinern	DusenKernkombination					x	x	x
Temperatur Extrudereinflusszone	messbar	numerischer Wert	Temp_Extrudereinflusszone					x		
Rohstofftemperatur	visuell	numerischer Wert oder i.O. / zu warm / zu kalt	Temp_Rohstoff					x		
Einsaugzone auf Beschädigung überprüfen	visuell	Ja / Nein	EinsaugzoneBeschädigt							x
WDS Kurve	visuell	i.O. / n.i.O.	WDSKurve					x		
Düsenanordnung	visuell	i.O. / n.i.O.	DusenAnordnung					x		
Dichtigkeit WSK	visuell	i.O. / n.i.O.	DichtigkeitWSK						x	
Dichtigkeit AKSU	visuell	i.O. / n.i.O.	DichtigkeitAKSU							x
Anströmwinkel des Kerns	visuell	i.O. / zu steil / zu flach	AnstromwinkelKern							x
Ausstossgeschwindigkeit	messbar	numerischer Wert	Ausstossgeschwindigkeit							x
Artikellgewicht	visuell	numerischer Wert oder i.O. / zu leicht / zu schwer	Artikellgewicht							x
Temperaturen	messbar	numerischer Wert	Temperaturen							x
Taktzeit	messbar	numerischer Wert	Taktzeit							
Gegendruck Akku	messbar	numerischer Wert	GegendruckAkku							
WDS Kurve Anfang Ende	visuell	i.O. / n.i.O.	WDSKurveA/E							
Kontamination Kopf	visuell	Ja / Nein	KontaminationKopf							
Zentrierung Düse	visuell	i.O. / n.i.O.	ZentrierungDuese							
Kontamination Düse	visuell	Ja / Nein	KontaminationDuese							
Leck-Stoßluft am Kopf	visuell	Ja / Nein	LeckStoßluftKopf							
Brückenbildung am Trichter	visuell	Ja / Nein	BrueckeTrichter							
Temperatur Extrudereinsaugzone	messbar	numerischer Wert	Temp_Extrudereinsaugzone							
Düsenventile	visuell	i.O. / n.i.O.	Dusenventile							
Regeneratqualität	visuell	i.O. / zu fein / zu grob	Regeneratqualitaet		x	x	x			
Düsenanordnung auf Null stellen	messbar		DusenAnordnungNullstellen							

Abbildung 4.9: Ausschnitt der Parameterliste mit Zuordnung zu möglichen Fehlern

Ein Datensatz/ Datentuple soll wie folgt aussehen (Tabelle 4.2):

Er erhält eine eindeutige und einmalige ID und anschließend werden alle möglichen Fehler sowie Parameter/ Fehlerursache aufgelistet. Ein Datensatz gilt als vollständig und korrekt, wenn für den untersuchten Fehler alle Einflussparameter-Zustände bekannt sind. Im Beispiel Kratzer sind dies die vier Faktoren *BeschädigungDuese*, *Duesentemperatur*, *VerbrannteRohstoffe\_Duesenoberfläche* und *Verunreinigung\_Rohstoff*, deren Zustände bekannt sein müssen. Alle anderen Parameterzustände müssen nicht eingetragen sein, es hat jedoch keine negativen Auswirkungen, wenn sie ebenfalls aufgenommen sind. In einem Datensatz können auch mehrere Fehler gleichzeitig bewertet werden. In diesem Fall müssen die Einflussparameter jedes zu bewertenden Fehlers bekannt sein.

ID	Kratzer	Blasenb.	Fehler...	Duesentemp	Kopftemp	Para...
1	Ja	Nein	-	190	205	...
2	Nein	-	-	200	-	...
3	Nein	Ja	-	200	180	...
4	-	Ja	-	-	200	...
5	Ja	-	-	185	-	...

Tabelle 4.2: Konzeptrahmen *Datenformat*, vollständiges Datentuple

Zum Lernen und Trainieren des Bayes-Netzes und seiner Fehler bedarf es einer exakten Zuordnung zwischen Fehler- und Parameterzustand. Nur so ist es möglich Vorhersagen zu treffen, welcher Einflussparameter die höchste Wahrscheinlichkeit für das Auftreten eines Fehlers besitzt.

## 4.5 BENUTZEROBERFLÄCHE MIT SHINY

Dieser Abschnitt behandelt das Entwerfen und Erstellen einer Benutzeroberfläche. Die Programmiersprache R ist zur Datenverarbeitung und -analyse entwickelt worden. Sie bietet nur bedingt Möglichkeiten ein Graphical User Interface (GUI) für die Interaktion zwischen Assistenzsystem und dem Menschen zu programmieren.

Bei Recherchen zur Erstellung von Benutzeroberflächen ist man auf *Shiny* gestoßen. Das R-package *Shiny* ermöglicht es, webbasierte Oberflächen zu erstellen. Die Webseite bietet Informationen, Beispiele und Tutorials.<sup>2</sup>

In [Abbildung 4.10](#) ist die Architektur von Shiny dargestellt. Ein Benutzer tätigt eine „Eingabe“ über die *Benutzeroberfläche*. Die Eingabedaten werden an den *Server* übermittelt. Dieser besitzt Zugriff auf Daten, Algorithmen und Funktionen. Je nach Input werden Aktionen ausgeführt und die Resultate zurück an die *GUI* gesendet. Diese stellt dem Benutzer die Ausgaben in einer visuellen Form da.

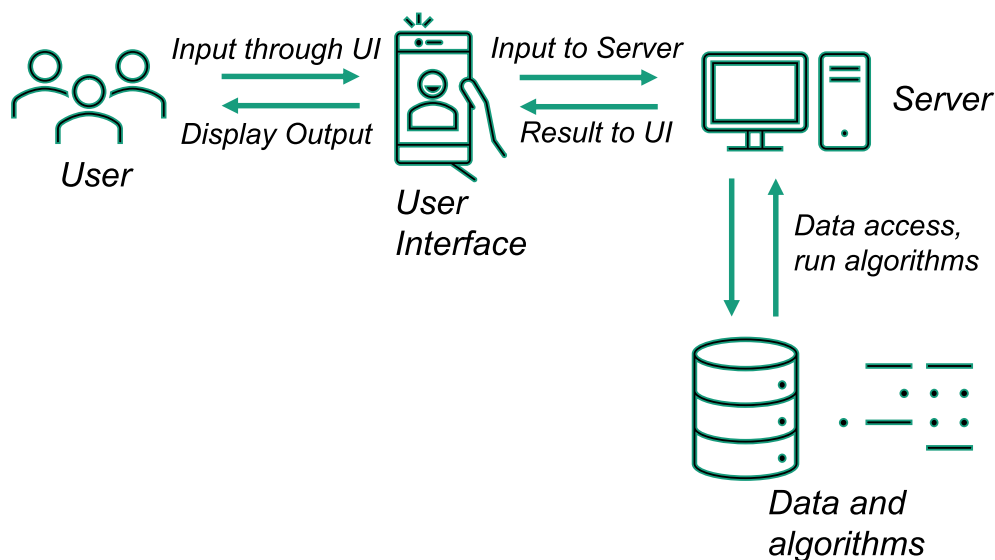


Abbildung 4.10: Architektur und Funktion *Shiny*

Aufgabe ist es zum einen die Benutzeroberfläche und zum anderen die Serverseite zu programmieren. [Listing 4.4](#) zeigt die allgemeine Struktur des Codes einer Shiny Weboberfläche. Im Bereich *ui* wird die eigentliche Benutzeroberfläche programmiert. Man erzeugt das Layout und das Erscheinungsbild. Im Bereich *server* werden Funktionen und Regeln implementiert, die unter gewissen Eingaben ausgeführt werden sollen. Möchte man eine Schaltfläche zum Start einer Berechnung auf der Benutzeroberfläche darstellen, dann wird im Bereich *ui* eine Schaltfläche mit eindeutiger „ID“ definiert und im Bereich *server* für diese „ID“ die Funktion zum Ausführen der Berechnung hinterlegt.

Sowohl *ui* als auch *server* werden an die Funktion *shinyApp* übergeben, um

<sup>2</sup> Webseite des R-package *shiny* <https://shiny.rstudio.com/>

die App zu starten. Sie startet lokal auf dem eigenen Rechner. Allerdings ist es möglich über Angabe der IP-Adresse und des Ports aus einem Browserfenster die Weboberfläche zu öffnen und zu bedienen, z.B. mit der Adresse `http://127.0.0.1:6954`

---

```
library(shiny)
library(shinydashboard)

#####
## Define UI for application ##
#####
ui <- dashboardPage(
  Code ...
)

#####
## Define server logic required ##
#####
server <- function(input, output){
  Code ...
}

#####
## Run the application ##
#####
shinyApp(ui = ui, server = server)
```

---

Listing 4.4: Programmrahmen Shiny in R

Die Oberfläche wird *step-by-step* entworfen und erstellt. Sie wächst und entwickelt sich fortlaufen weiter. Aus diesem Gedanken heraus entstand die Idee für unterschiedliche Bereiche mit Tabs zu arbeiten. [Abbildung 4.11](#) zeigt die zum aktuellen Stand erstellten Tabs. So gibt es einen Tab zur Gesamtübersicht des Bayes-Netzes, einer dient zur Berechnung und Ermittlung der Fehlerursache, ein weiterer dient zum Hochladen neuer Datensätze und dem Trainieren des Netzes.

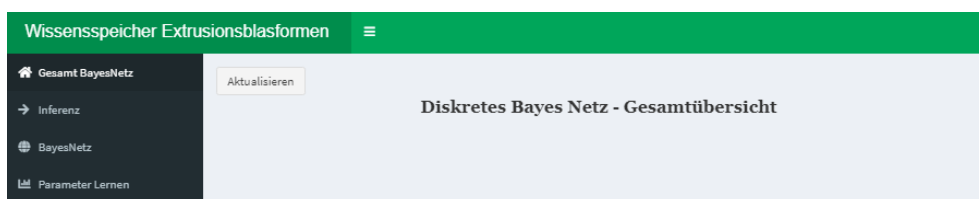


Abbildung 4.11: Übersicht Benutzeroberfläche, vorhandene Tabs

Shiny bietet Bibliotheksfunktionen, mit diesen es ermöglicht wird die graphische Oberfläche zu erstellen und zu gestalten. Es werden unterschiedliche In- und Output Funktionen zur Verfügung gestellt. Im Bereich *ui* wird einem Ein- bzw. Ausgabeelement eine eindeutige ID definiert. Durch diese ist es möglich, dass im Bereich *server* die Werte eines Elementes verändert bzw. abgefragt werden können. Eine Auflistung der Möglichkeiten ist nachfolgend zu sehen.

### Eingabe Möglichkeiten

- |                    |               |                  |                |
|--------------------|---------------|------------------|----------------|
| 1. Buttons         | 4. Date input | 8. Numeric input | 11. Sliders    |
| 2. Single checkbox | 5. Date range | 9. Radio buttons | 12. Text input |
| 3. Checkbox group  | 6. File input | 10. Select box   |                |
|                    | 7. Help text  |                  |                |

### Ausgabe Möglichkeiten

- |              |          |          |
|--------------|----------|----------|
| 1. DataTable | 3. Image | 5. Table |
| 2. Raw HTML  | 4. Plot  | 6. Text  |

In [Listing 4.5](#) und [Listing 4.6](#) ist der Programmcode für den Tab „Bayes-Netz“ bzw. „Fehlerübersicht“ dargestellt. Wie im oberen Abschnitt beschrieben, bedarf es der Programmierung sowohl von *ui* als auch von *server*.

---

```
## Tab BAYESNET
tabItem(tabName = "bayesnetz",
  fluidPage(
    column(width = 12,
      h1("Anzeige der einzelnen Bayes Netze"),
      ## chose bayesian net
      box(width = NULL,
        selectInput(inputId = "bayesnet",
          label = "Wählen Sie ein Bayes
            Netz aus",
          choices = c(get_available_nets()
            )),
        helpText("Ausgabe Bayes Netz"),
        actionButton(inputId = "idi_ab_bayes_
          refresh",
            label = "Ausgabe",
            icon = icon("sync"), width =
              100)
        ),
      ## plot bayesian net
      box(width = NULL,
        plotOutput(outputId = "ido_bayesnet_solo_
          _plot",
            height = 1000)
        )))
),# end Tab Bayes net
```

---

Listing 4.5: Programmausschnitt *ui.R* Tab Bayes-Netz

Die Benutzeroberfläche wird wie folgt aufgebaut. An oberster Stelle wird eine Überschrift mit dem Befehl `h1(<Überschrift>)` eingefügt. Anschließend werden zwei „Boxen“ erstellt, welche der räumlichen Abtrennung dienen. In



der oberen Box wird ein Drop-Down-Menü eingefügt, über das man einen einzelnen Fehler auswählen kann und über eine Schaltfläche „Ausgabe“ das zum Fehler gehörige Bayes-Netz plottet. Das Darstellen des Bayes-Netzes wird im Bereich *server* umgesetzt. Dort wird *reactive* auf das Event der Schaltflächenbetätigung zum Plotten des Graphen reagiert. Es wird eine passende Funktion zum Einlesen des Sub-Netzes sowie zum Plotten des Graphen aufgerufen.

```
##### TAB BAYESIAN NETWORKS #####
## plot bayesian network
output$ido_bayesnet_solo_plot <- renderPlot({
  input$idi_ab_bayes_refresh
  isolate(bayesiannet <- read.net(paste0(input$bayesnet, ".net",
    sep = "")))
  isolate(renderGraph(highlight_fault(draw_BayesNet(bayesnet =
    bayesiannet),
    input$bayesnet)))
})
```

Listing 4.6: Programmausschnitt *server.R* Tab Bayes-Netz

In [Abbildung 4.12](#) ist die Oberfläche dargestellt, wie sie ein Bediener vorfindet.

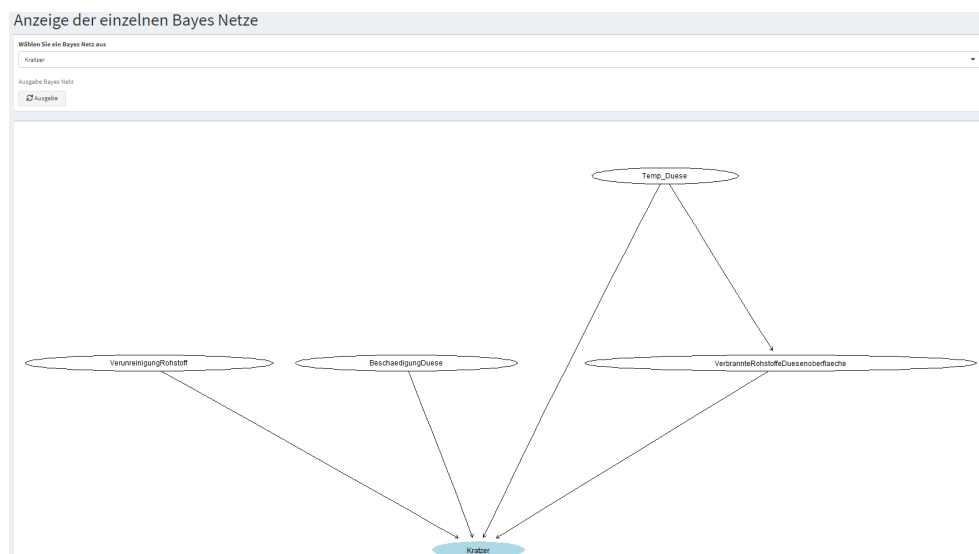


Abbildung 4.12: Tab „Bayes-Netz“ mit angezeigtem Fehler *Kratzer*

Die vollständige Oberfläche ist im [Kapitel 5](#) Ergebnis/ Zustand zum Ende des BPP's abgebildet.

## ERGEBNIS/ ZUSTAND ZUM ENDE DES BPP'S

Das Projekt zur Entwicklung eines interaktiven Wissensspeichers besitzt, wie zu Beginn beschrieben, eine Laufzeit von 2 Jahren. Der Stand zum Ende der berufspraktischen Phase stellt dahingehend einen Zwischenstand und nicht den Endzustand des Gesamtprojektes dar.

Die Bayes-Netz Struktur befindet sich in einem vorläufig vollendeten Zustand. Mögliche Fehler während der Phase *Vorformling* sind identifiziert und Fehlerursachen mit Abhängigkeiten definiert. Die einzelnen Strukturen gilt es zu überprüfen und zu validieren, dieser Schritt ist zum aktuellen Zeitpunkt offen. Für das Zusammenfügen der einzelnen Netze zu einem Großen/ Gesamten ist ein eigens entwickelter Prozess vorhanden. Desweiteren ist ein Algorithmus entwickelt worden, mit diesem es möglich ist die wahrscheinlichste Fehlerursache für ein aktuell vorliegendes Fehlerbild vorherzusagen.

Im nächsten Schritt ist ein Konzept entworfen worden, dass das Lernen und Trainieren des Bayes-Netzes umsetzt. Neben Algorithmen und Funktionen ist eine vorläufige Benutzeroberfläche erstellt worden. Die nachfolgenden Abbildungen zeigen die zum aktuellen Zeitpunkt entworfene Benutzeroberfläche

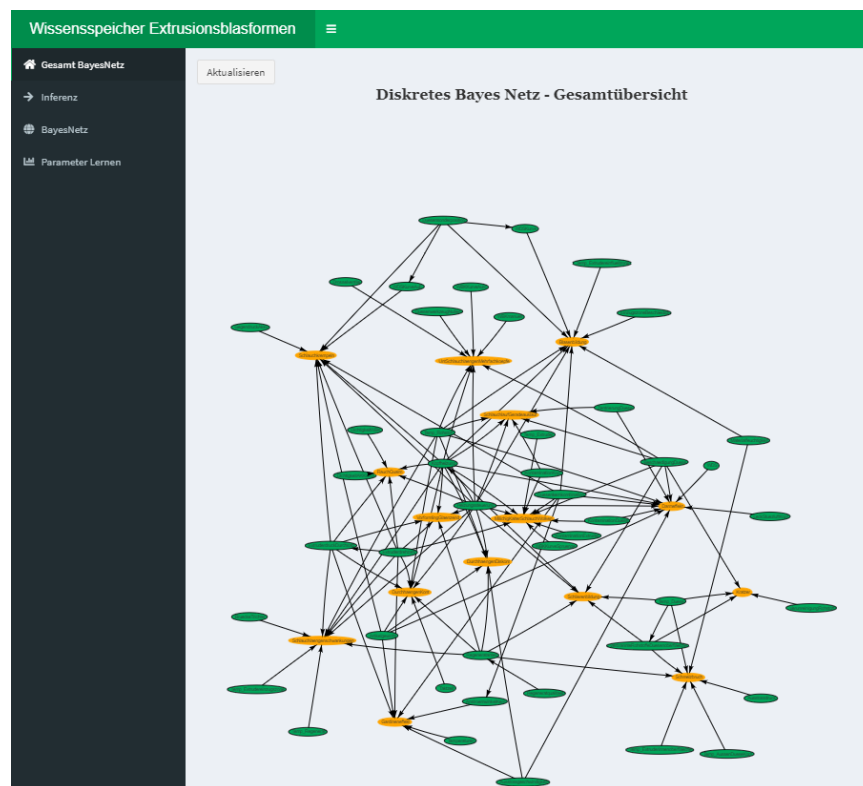


Abbildung 5.1: Tab „Gesamt BayesNetz“

Abbildung 5.1 zeigt die Übersicht des erstellten Hauptnetzes. Der Bediener besitzt die Möglichkeit den Graphen interaktiv zu bedienen. Neben heranzoomen und verschieben des Netzes, ist es möglich einzelne Knoten zu „greifen“ und somit zu verschieben. Möchte man Abhängigkeiten zu einem Fehler hervorheben, gelingt dies durch Klicken auf den jeweiligen Knoten. Diejenigen Knoten, welche eine Kante zum ausgewählten aufweisen bleiben in ihrer Farbe unverändert, andere werden ausgegraut und somit in den Hintergrund verlagert.

Zur Vorhersage der Fehlerursache dient der Tab „Inferenz“ (s. Abbildung 5.2)

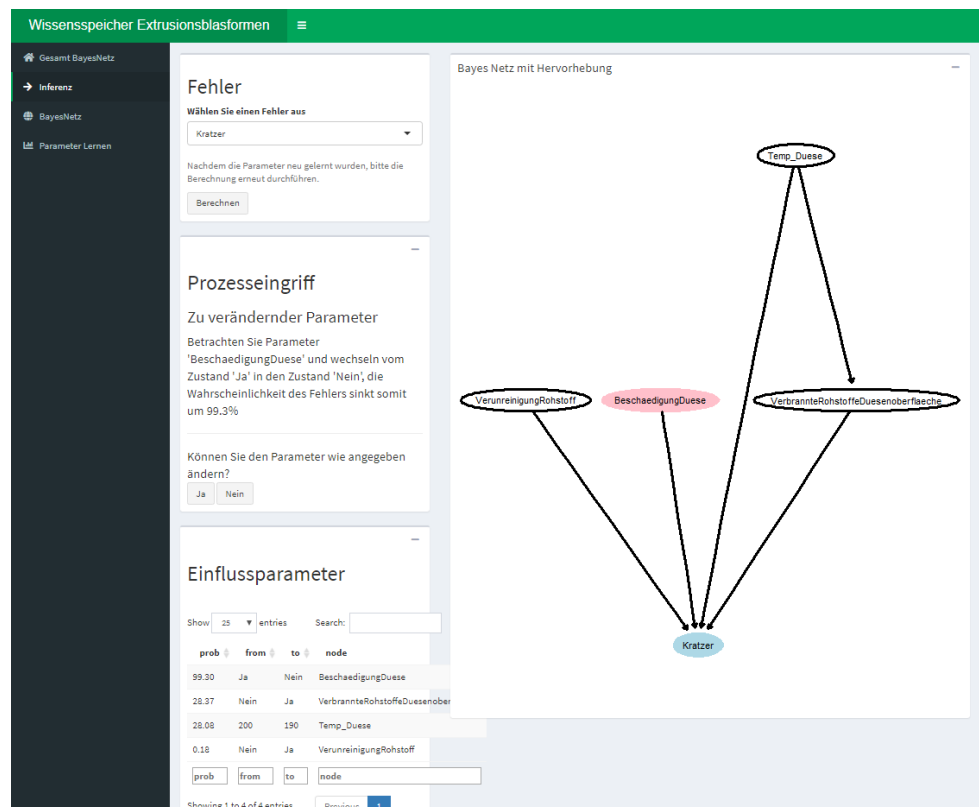


Abbildung 5.2: Tab „Inferenz“, Vorhersage Fehlerursache für Fehler *Kratzer*

Der Bediener wählt im oberen Bereich das anstehende Fehlerbild aus und betätigt anschließend die Schaltfläche *Berechnen*. Im rechten Fensterabschnitt wird das Bayes-Netz des ausgewählten Fehlers dargestellt. Mit einem Rotton wird der Knoten hervorgehoben, welcher als wahrscheinlichste Fehlerursache vorhergesagt wurde. Auf der linken Seite wird der zu verändernde Knoten und seine Wertänderung, die vorgenommen werden soll, als Text ausgegeben. Unterhalb gibt es die Möglichkeit dem Assistenzsystem mitzuteilen, ob der Eingriff möglich ist oder nicht. Da die Vorhersage des Wissensspeichers ohne Anbindung an die Maschine arbeitet, gibt es die Möglichkeit, dass ein Wertewechsel vorgeschlagen wird, der in Realität nicht umzusetzen ist, da sich die Maschine bereits in diesem Zustand befindet. Ist es möglich den Knoten wie empfohlen zu verändern, muss anschließend Rückmeldung gegeben werden, ob der Fehler beseitigt wurde oder ein weiterer Vor-

schlag ausgegeben werden muss. Im unteren Bereich erhält der Einrichter eine Übersicht über die berechneten Vorhersagen mit Wahrscheinlichkeiten und Wertwechsel.

Tab „Parameter Lernen“ ist in [Abbildung 5.3](#) zu sehen. Dort erhält der Bediener die Möglichkeit neue Datensätze einzulesen und anschließend das Netz weiter zu trainieren. Sind neue Daten vorhanden, welche dem Wissenspeicher zugeführt werden sollen, ist wie folgt vorzugehen. Der Datensatz muss die vorgegebene Struktur aufweisen und als csv-Datei vorliegen. Über ein Explorer-Fenster besteht die Möglichkeit eine Datei auszuwählen und hochzuladen. Ist das Hochladen abgeschlossen, erscheint eine Tabelle mit den neuen Daten. Diese gilt es in einem nächsten Schritt zu speichern und anschließend das Bayes-Netz mit dem neuen Datensatz zu trainieren. Das Trainieren muss aktiv ausgeführt werden und wird nicht automatisch gestartet. Nachdem das Netz trainiert wurde, erhält der Bediener den Hinweis, dass sich die Parameter verändert haben und die Vorhersagen für einen aufgetretenen Fehler neu berechnet werden müssen.

**Wissenspeicher Extrusionsblasformen**

**Parameter Bayes Netz lernen**

1. Laden Sie die neuen Daten als .csv Datei hoch
2. Speichern Sie die Daten mittels 'Save'
3. Lernen Sie die Parameter mittels 'Parameter Lernen'

**Lade Parameterliste**

Durchsuchen... kratzer\_test\_daten\_3.csv

Upload complete

Save Parameter lernen

**Liste eingelesener Daten**

Show 25 entries

Kratzer	BeschädigungDuese	Temp_Duese	VerbrannteRohstoffeDuesenoberflaeche	VerunreinigungRohstoff
Ja	Ja	200	Nein	Nein
Ja	Ja	200	Nein	Nein
Ja	Ja	200	Nein	Nein
Ja	Ja	200	Nein	Nein
Ja	Ja	200	Nein	Nein
Ja	Ja	200	Nein	Nein
Ja	Ja	200	Nein	Nein
Ja	Ja	200	Nein	Nein

Abbildung 5.3: Tab „Parameter Lernen“

Zum aktuellen Zeitpunkt liegen keine realen Produktionsdaten vor. Für das Sub-Netz *Kratzer* sind synthetische Daten generiert worden, welche allerdings nicht realitätsnah sind. Sie wurden benötigt um das Trainieren und Vorhersagen der Fehlerursache zu testen. Diese Tests wurden anhand des Fehlers *Kratzer* durchgeführt. Implementierungen vor Ort bzw. Tests mit realen Daten haben zum aktuellen Zeitpunkt nicht stattgefunden. Der Kunde/ Industriepartner beginnt in einem naheliegenden Zeitraum mit der Datenerhebung und -erfassung.

## AUSBLICK UND OFFENE PUNKTE

---

In diesem Kapitel werden offene Punkte sowie ein Ausblick thematisiert. In [Kapitel 5](#) wurde beschrieben, dass lediglich Fehler für die Phase *Vorformling* bearbeitet und Strukturen erstellt wurden. Hier gilt es die Struktur weiter zu vollenden, sowie die Phase *fertiges Produkt* und dort auftretende Fehler zu bearbeiten.

In einem nächsten Schritt muss die Datenerfassung betrachtet werden, da diese noch nicht erfasst wurden. Man benötigt jedoch zum Lernen und Trainieren des Bayes-Netzes eine gewisse Anzahl an Datensätzen. Über die Möglichkeiten der Umsetzung an beide Kategorien von Daten zu gelangen, bedarf es seitens des Instituts einer Unterstützung. Die Maßnahmen hierfür muss der Industriepartner durchführen.

Aufgrund der fehlenden Daten konnten mit dem aktuell vorliegenden Stand des Assistenzsystems keine Testläufe durchgeführt werden. Diese gilt es schnellst möglich durchzuführen, sobald ausreichend Daten erfasst wurden. Ebenfalls sollten in diesem Zuge eine vor Ort Implementierung, sowie Tests während des realen Einrichtprozesses durchgeführt werden.

Der Wissensspeicher berechnet die Vorhersagen und Empfehlungen auf Basis von „offline-Daten“. Der aktuelle Maschinenzustand ist unbekannt und kann lediglich durch den Bediener festgestellt werden. Aufgrund dessen können Fälle auftreten, bei denen das Assistenzsystem Empfehlungen ausgibt, welche mit dem aktuellen Maschinenzustand nicht kompatibel sind. Hier gilt es eine „Live-Anbindung“ an die Maschine herzustellen und somit die Vorhersagen und Empfehlungen *Zustands abhängig* zu tätigen. Neben dem Abfragen des aktuellen Zustands ist es möglich die Datenerfassung in den Prozess zu integrieren. Da es einige Fehlerursachen gibt, welche der Bediener zum aktuellen Zeitpunkt für die Datenerfassung manuell eintragen muss, gilt es zu diskutieren, ob zusätzliche Sensorik den Automatisierungsgrad der Datenerfassung erhöhen kann. Neben der Steigerung des Automatisierungsgrades bei der Datenerhebung wird gleichzeitig die Fehleranfälligkeit reduziert.

Als letzten offenen Punkt gilt es die Benutzeroberfläche zu erwähnen, welche in ihrer Funktion und Aussehen vollumfänglich programmiert werden muss. Hier sind aktuell die zuvor gezeigten Entwürfe als mögliche Lösung erstellt worden.

Teil II

ANHANG

### A.1 BLASFORMEN (*extrusionsblasformen*)

In der Projektbeschreibung wird ein Wissensspeicher für eine Extrusionsblasformmaschine gefordert. Aus dem Studium liegen keine Kenntnisse im Bereich Blasformen vor. Mit Hilfe einer *betriebsinternen* Präsentation und einer Bachelorthesis über das Extrusionsblasformen (Verfahren und Maschinen), sowie öffentlicher Literatur, werden erste Einblicke und Knowhow in dem Themengebiet gewonnen.

Blasformen im Allgemeinen beschreibt eine Verfahrensfamilie zum Herstellen von Kunststoffhohlkörpern. In diesem Bericht wird sich auf das Extrusionsblasformen beschränkt und es wird lediglich ein kleiner Überblick über dieses Themengebiet aufgezeigt. In den letzten Jahren ist neben dem Extrusionsblasformen das Streckblasformen in den Vordergrund gerückt. Gemeinsam haben diese beiden Verfahren, dass ein thermoplastisch verformbarer Vorformling in einer Negativform aufgeblasen wird, durch Kühlung der Kunststoff erstarrt und ein fester Hohlkörper entnommen werden kann. [13]

Das **Extrusionsblasformen** selbst wird nochmal in unterschiedliche Unterverfahren kategorisiert.

- Konventionelles Extrusionsblasformen
- Konventionelles Extrusionsblasformen mit Überquetschen
- Manipulationsverfahren
- Saugblasverfahren
- Einlegeverfahren

Nachfolgende [Abbildung A.1](#) zeigt die verschiedenen Extrusionsblasformverfahren. Es ist jeweils der Vorformling und das Werkzeug und die unterschiedliche Komplexität der möglichen Hohlkörperformen zu sehen.

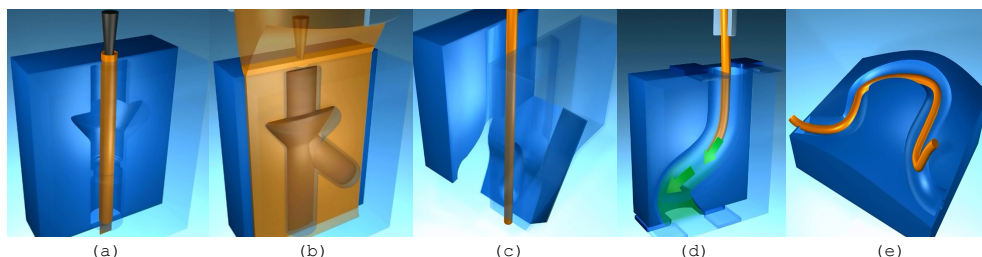


Abbildung A.1: Extrusionsblasformen (a) konventionell, (b) konventionell mit Überquetschung, (c) Manipulationsverfahren, (d) Saugblasverfahren, (e) Einlegeverfahren aus [2] S.32ff.

Im Projekt wird eine Maschine für das Extrusionsblasformen mit Überquetschen betrachtet, daher wird im Folgenden nochmal einzeln auf dieses Verfahren eingegangen. Jedoch ist im Allgemeinen der Prozess für alle Verfahren identisch, Unterschiede sind lediglich in einzelnen Punkten enthalten. Neben Unterschieden im Prozess, weisen die Verfahren signifikante Abweichungen im Ressourcenverbrauch auf und auch in der Möglichkeit der Komplexität der 3D-Hohlkörper.

### Allgemeiner Prozess des Extrusionsblasformens

1. Kunststoff in einem Extruder erhitzen und plastifizieren.
2. Schmelze von einer waagerechten in eine senkrechte Bewegung umlenken und erzeugen eines schlauchförmigen Vorformlings.
3. Werkzeug (zwei Halbschalen) um Vorformling herum schließen und oberes und unteres Ende quetschen.
4. Blasdorn/ Blasnadel in Form einfahren.
5. Vorformling aufblasen und an gekühlten Wänden des Werkzeuges abkühlen und erstarren.
6. Entlüften und Öffnen der Werkzeughälften.
7. Entnehmen des Hohlkörpers und Entfernen von Überquetschungen und Butzen.

[13]

Nachfolgende Abbildung (A.2) zeigt in Grafiken die einzelnen Schritte des Extrusionsblasformens. Nachdem die Schmelze erhitzt und plastifiziert ist, wird sie durch den Extruder über eine Schnecke in den Kopf befördert. Aus einer waagerechten Bewegung entsteht im Folgenden eine Senkrechte. Durch eine Düse wird der Thermoplast aus dem Kopf gedrückt und zu einem Vorformling geformt. Besitzt dieser die richtige Länge, werden die Werkzeughälften zu gefahren und das obere und untere Ende eingequetscht. Nachdem der Blasdorn eingefahren ist, wird der Vorformling aufgeblasen und gegen die Innenwand des Werkzeuges gedrückt, bis der Thermoplast erstarrt. Zum Abschluss wird das Werkzeug auf gefahren, der Kunststoffhohlkörper aus der Form entnommen und entstandene Butzen entfernt. In einem letzten Schritt ist es z.B. möglich eine Dichtigkeitsprüfung durch zu führen.



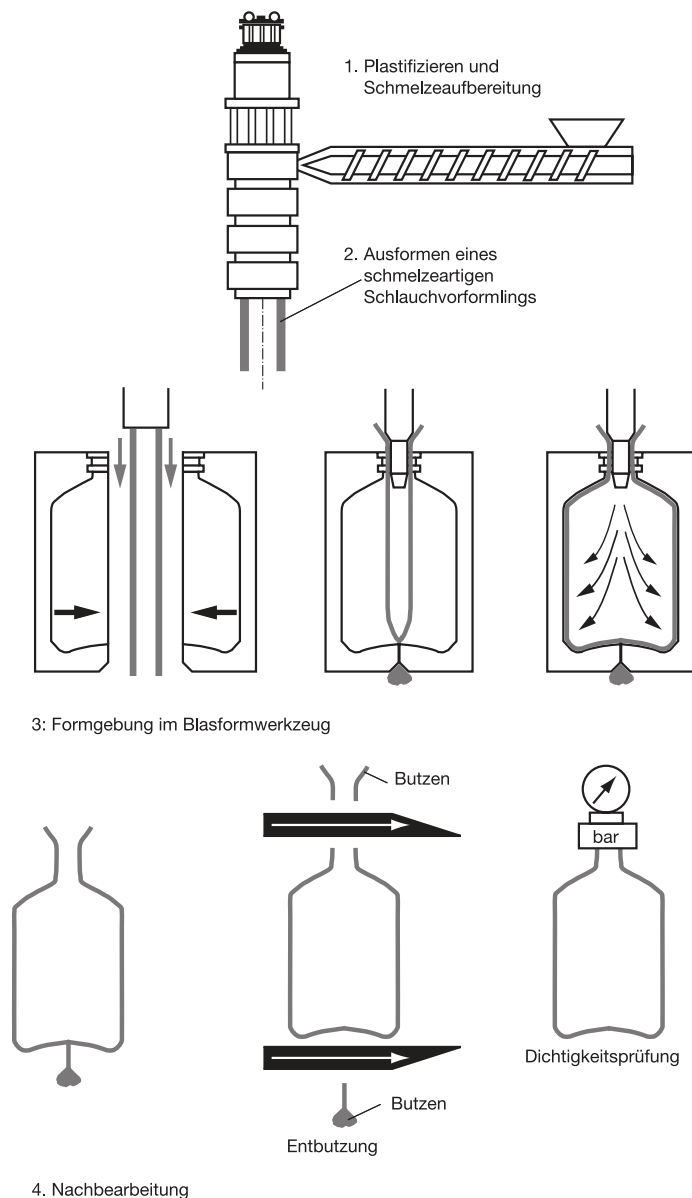


Abbildung A.2: Prozess Kunststoff-Extrusionsblasformen aus [13] S.16

#### A.1.1 Extrusionsblasformen mit Überquetschen

Beim konventionellen Extrusionsblasformen mit Überquetschen handelt es sich um das einfachste Fertigungsverfahren. Hierbei findet der Materialeinsatz nicht Ressourcen effizient statt. Wie in [Abbildung A.1\(b\)](#) zu sehen, wird ein Vorformling benötigt, welcher die Breite des späteren Hohlkörpers aufweist. Für das Überquetschen werden neben dem großen Materialverbrauch (großer Extruder nötig) hohe Schließkräfte benötigt. An den Stellen der Überquetschung weist die Wanddicke ihr Maximum auf und wird zur Formtie-

fe hin minimaler. Nach dem Ausformen des Hohlkörpers besitzt dieser eine vollständig umlaufende Quetschnaht, welche neben einer Schwachstelle auch Nacharbeit erfordert. Als Vorteile lassen sich hier günstige Formkosten und kurze Rüst- und Einstellprozesse erwähnen.[9]

#### A.1.2 *Kontinuierliches/ diskontinuierliches Extrusionsblasformen*

Ein nicht zu vernachlässigender Unterschied innerhalb eines Verfahrens ist die Art und Weise, wie die Schmelze für den Vorformling ausgestoßen wird. Hierbei unterscheidet man *kontinuierliches* und *diskontinuierliches* Ausstoßen der Schmelze. Bei kontinuierlichem Ausstoßen fließt ununterbrochen thermoplastische Schmelze aus dem Kopf heraus und formt so den schlauchförmigen Vorformling. Die Ausstoßgeschwindigkeit ist hierbei stark von der Extruderdrehzahl abhängig. Diese Art und Weise wird zumeist bei kurzen und zähen Thermoplasten eingesetzt, welche zugleich eine geringe Masse aufweisen, somit kann eine unkontrollierte Schlauchlängung bzw. Wanddickenreduzierung verhindert werden. Bei diskontinuierlichem Ausstoßen wird die gesamte Schmelze für einen Vorformling in einem Speicher gesammelt und anschließend stoßartig nach Draußen befördert. Dieses Verfahren wird angewandt, wenn ein langer bzw. schwerer Vorformling benötigt wird oder es sich um einen Thermoplast handelt, welcher nur eine geringe Zähigkeit aufweist.[2]

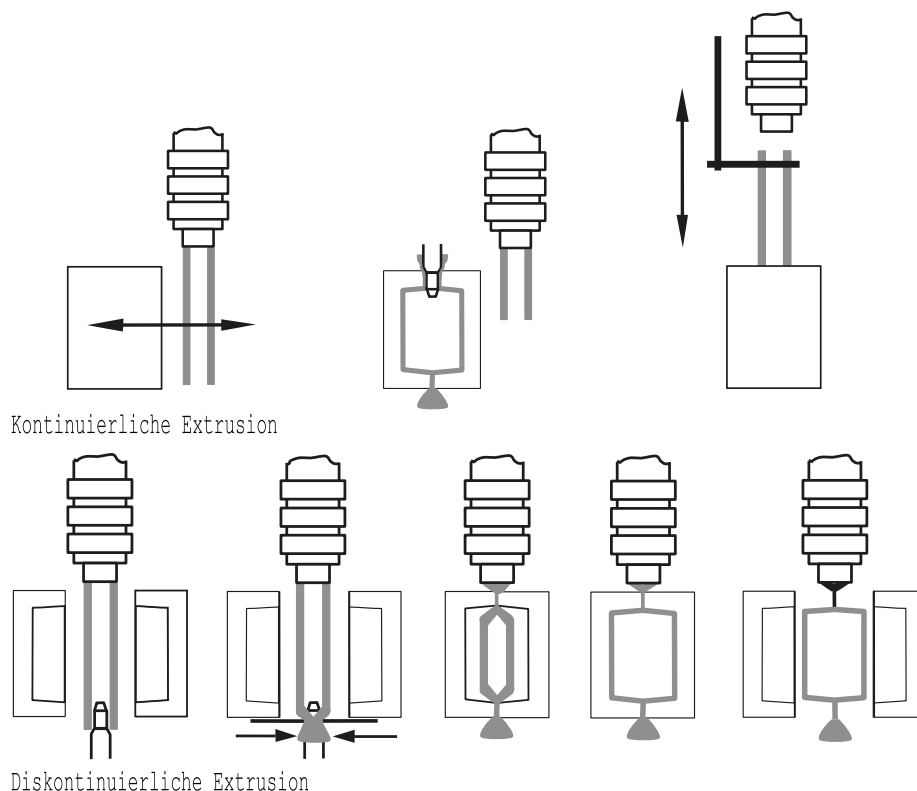


Abbildung A.3: Kontinuierliche und diskontinuierliche Extrusion nach [13] S.40 und S.42

## A.2 BAYES NETZE

Bayes-Netze dienen der Datenanalyse und dem Erstellen von Wahrscheinlichkeitsmodellen. Oft liegt eine Vielzahl von Daten vor und man möchte Vorhersagen treffen. Daten werden zumeist in Tabellenform abgelegt und gespeichert. Hierbei beschreibt eine Zeile eine Beobachtung und eine Spalte die Werte eines spezifischen Attributes. Eine Zeile besitzt für jedes Attribut einen Wert. Bayes-Netze bieten die Möglichkeit Attribute zu verknüpfen, Abhängigkeiten herzustellen und das Auftreten von Ereignissen vorherzusagen.[8]

Die graphische Darstellung eines Bayes-Netzes verwendet die Methode des *gerichteter, azyklischer Graph (DAG)*. Hierbei wird eine endliche Menge *Knoten* durch gerichtete *Kanten* (Pfeile) miteinander verbunden, daher der Name *gerichtet*. Neben den gerichteten Kanten ist es wichtig, dass keine Schleifen/ Zyklen entstehen dürfen, daher *azyklisch* s. [Abbildung A.4](#).<sup>[11]</sup>

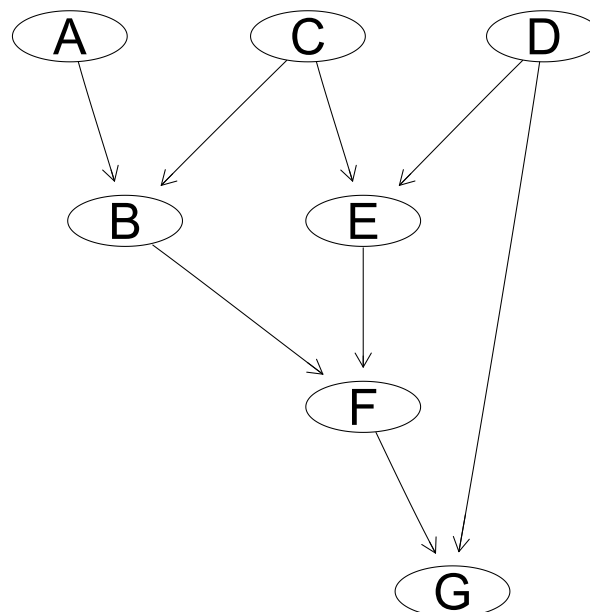


Abbildung A.4: Bayes Netz als *gerichteter, azyklischer Graph (DAG)*

Jeder Knoten besitzt mindestens zwei Zustände/ Levels. Einem Zustand wird eine Verteilung zugeordnet. Die Summe aller Zustände eines Knotens ergibt 1. Besitzt ein Knoten  $X$  nachfolgende Knoten, dann werden diese Kinder von  $X$  genannt, z.B.  $Kinder(A)=B$ . Die Vorgänger eines Knoten  $X$  werden Eltern genannt, z.B.  $Eltern(B)=A,C$ . Mittels *Inferenz* kann man Wahrscheinlichkeiten eines Attributes unter *gegebenen* Bedingungen bestimmen. Angewendet wird hierbei unter anderem der *Satz von Bayes*  $P(B|A) = \frac{P(A|B) \cdot P(B)}{P(A)}$  [6]

In [Abbildung A.4](#) bestehe die Möglichkeit mittels Inferenz die Wahrscheinlichkeit für Knoten  $F$  unter gegebenen  $A$  und  $C$  zu bestimmen. Mathematisch drückt man dies wie folgt aus:  $P(F|A, C)$ . Die Erläuterung der angewandten Mathematik und die Berechnungen der Bayes-Netze wird in diesem Bericht

nicht weitergehend beschrieben. Es wird bei Bedarf auf die angegebene Literatur verwiesen.

### A.3 EINSTIEG IN DIE PROGRAMMIERUNG MIT R

Im Mechatronik Studium werden Inhalte in der Programmierung mit C und C++ vermittelt. Später wird in der Vertiefung Robotik die Programmiersprache Python eingeführt. Kenntnisse über die Programmierung mit R werden jedoch nicht behandelt.

Bevor mit der späteren Erstellung eines Programmcode begonnen wird, wird mit der Literatur von U. Ligges "Programmieren mit R" [10] und D. Obszeka "Statistisches Programmieren mit R" [1] ein grundlegendes Wissen erlernt. Beide Werke bieten Beispiele zum Nach- bzw. Selbst-Programmieren an. Die Programmiersprache R ist eine "Open Source" Software und optimiert für die Datenanalyse und Grafiken. Durch das Hinzufügen von Zusatzpaketen (*packages*) können Methoden und Technologien ergänzt werden. [10]

Für die Programmierung wird R-Studio verwendet. Dieses Programm besitzt neben einem *Quellcodefenster*, eine *Konsole*, eine *Variablenübersicht* und ein *Navigationsbereich*, dort können auch Plots ausgegeben werden. R-Studio unterstützt beim Programmieren und gibt dem Programmierer Vorschläge zur Vervollständigung seiner Codezeile.

Die Werke werden verwendet um die Syntax und das grundlegende Umgehen und Verwenden der Sprache R zu erlernen. Neben kleinen mathematischen Beispielen (Berechnung des *BodyMaßIndex* oder das *Lösen einer quadratischen Gleichung*) werden Funktionen definiert und angewendet.

Der folgende Code Listing B.1 zeigt zwei Wege auf die Fibonaccifolge zu berechnen. Wird im ersten Schritt für jedes "n" die Gleichung zur Berechnung vollständig ausgeschrieben, so wird im zweiten Schritt eine Funktion definiert, welche lediglich den Parameter "n" übergeben bekommt und die Ergebnisse zurück gibt. Im letzten Abschnitt wird ein Vergleich durchgeführt, welcher als Ergebnis ein *TRUE* liefert. Auf weitere Beispiele wird auf Grund des Umfangs verzichtet.

### A.4 GRAPHISCHES ERSTELLEN VON BAYES NETZEN

#### A.4.1 Mittels *BayesianFusion* GeNIe

Das R-package *bnlearn* bietet die Möglichkeit mit "reinem" Code ein Bayes-Netz aufzubauen. Schneller, intuitiver und leichter gelingt dies durch eine graphische Oberfläche. *BayesFusion, LLC*<sup>1</sup> bietet die Möglichkeit mit ihrer Software *GeNIe* ein Bayes Netz graphisch aufzubauen. Die Software ist für

<sup>1</sup> Homepage *BayesianFusion*

[https://www.bayesfusion.com/?gclid=CjwKCAjwzt6LBhBeEiwAbPG0gXh\\_nTLNoyEVWk9IC707\\_c\\_mFh4tSVxXYlqVAX8Fq\\_vn73Ya56EfhoC32MQAvD\\_BwE](https://www.bayesfusion.com/?gclid=CjwKCAjwzt6LBhBeEiwAbPG0gXh_nTLNoyEVWk9IC707_c_mFh4tSVxXYlqVAX8Fq_vn73Ya56EfhoC32MQAvD_BwE)

die allgemein üblichen Betriebssysteme verfügbar und benötigt keine besonderen Ressourcen.

Nach Start der Software und Öffnen eines neuen Netzwerkes, kann man per "Drag-and-Drop" die einzelnen Komponenten für das spätere Bayes-Netz in den Arbeitsbereich ziehen.

Nachdem die benötigten Knoten eingefügt und Abhängigkeiten durch Pfeile/ Kanten erzeugt worden sind, werden die Eigenschaften der einzelnen Knoten angepasst. Es werden Zustände und Wahrscheinlichkeitsverteilungen eingetragen. Zum Abschluss wird das Bayes-Netz berechnet und die Verteilungen ausgegeben. Durch eine Layoutumstellung werden die Verteilungen in einem Balkendiagramm an jedem Knoten angezeigt, so ist es möglich einen schnellen Gesamtüberblick zu erhalten. [Abbildung A.5](#) zeigt ein fertig erstelltes Bayes-Netz mittels GeNIe. Die eingetragenen Verteilungen beruhen auf *fiktiven und ausgedachten* Werten. Das Beispiel wurde für den Industriepartner erstellt, um Ihm das Thema Bayes-Netze und seine Funktion zu erläutern.

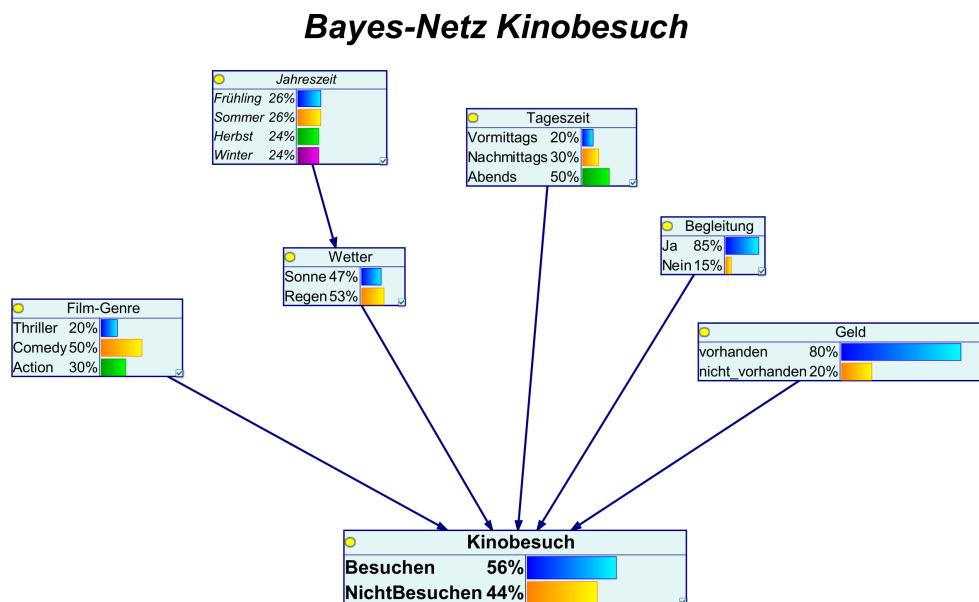


Abbildung A.5: Bayes Netz Kinobesuch

Im Beispielnetz ([Abbildung A.5](#)) wird untersucht, welche Parameter auf die Entscheidung, ob man das Kino besucht oder nicht, Einfluss haben. Als Einflussparameter werden das *Film-Genre*, das *Wetter*, die *Tageszeit*, *Begleitung* und *Geld* aufgelistet, wobei das *Wetter* abhängig von der *Jahreszeit* ist. Untereinander werden keine Abhängigkeiten erzeugt, somit bleibt das Bayes-Netz übersichtlich. Grundsätzlich gilt beim Erstellen eines Bayes-Netzes, dass man nur Abhängigkeiten einfügt, welche relevant und wichtig sind. Abhängigkeiten die nicht auftreten können werden entfernt. Mit den angegebenen Wahrscheinlichkeiten ergibt sich, dass ein Kinobesuch zu 56% stattfindet.

*GeNIe* bietet die Möglichkeit Zustände eines Knotens vorzugeben und somit für gegebene Bedingungen einen Kinobesuch zu ermitteln. Gibt man vor, dass ein *Thriller* läuft, es *regnet* und der Film *abends* spielt, so steigt die Wahrscheinlichkeit für einen Kinobesuch auf 84%. (Abbildung A.6)

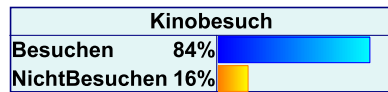


Abbildung A.6: Bayes Netz Kinobesuch, Bedingung *Thriller*, *Regen* und *abends*

Neben dem Erstellen eines Bayes-Netzes bietet die Software auch die Möglichkeit Daten einzulesen und Parameter zu ermitteln. Somit könnte aus einer Umfrage die Verteilungen für die Knoten anhand der Daten ermittelt werden. Man spricht hierbei von "Parameter Lernen". Speichert man das erstellte Bayes-Netz im Format "Hugin File .net" ab, so kann man dies mit *bnlearn* in *R* laden und weiter verwenden.

Da es sich bei dieser Software allerdings um ein kostenpflichtiges Produkt handelt, wurde nach einer Alternative gesucht. Die Alternative wird im nachfolgenden Abschnitt erläutert. Weitere Vertiefungen in die Software *GeNIe* finden nicht statt.

#### A.4.2 Mittels *SamIam*

*SamIam*<sup>2</sup> steht für *Sensitivity Analysis, Modeling, Inference and More*. Es handelt sich hierbei um eine "Open Source" Software der *Automated Reasoning Group, University of California*.

Da es sich um ein kostenloses Produkt handelt und auch die Möglichkeit bietet graphisch Bayes-Netze zu erstellen, wird der Fokus im Folgenden auf *SamIam* gelegt. [Abbildung A.7](#) zeigt die gestartete Software. Die Bedienung ist vergleichbar mit der Software *GeNIe*, ebenfalls werden per "Drag-and-Drop" einzelne Knoten und Pfeile/ Kanten für Abhängigkeiten erstellt. In den Eigenschaften eines Knotens erstellt man die Zustände mit ihren jeweiligen Verteilungen.

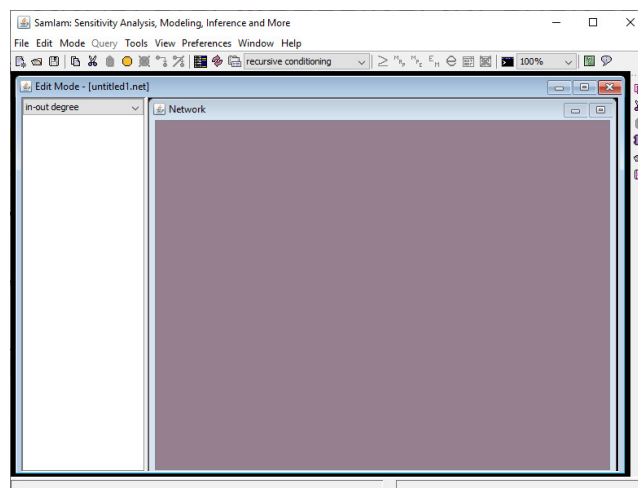


Abbildung A.7: *SamIam* Software zum graphischen Erstellen von Bayes-Netzen

<sup>2</sup> <http://reasoning.cs.ucla.edu/samiam/>

Abbildung A.8 zeigt ein weiteres Beispiel Bayes-Netz, in diesem Fall mit *SamIam* erstellt. Es handelt sich hierbei um ein Beispiel von *bnlearn*. Im späteren Verlauf, wird dieses Beispiel verwendet, um die Verknüpfung zwischen *SamIam* und R zu testen. Das Beispiel untersucht die Einflussparameter auf ein auffälliges Bruströntgen und Atemnot. Direkt beeinflussbare/ veränderbare Knoten sind der "Besuch in Asien" und der Knoten "Rauchen". *Tuberkulose*, *Bronchitis* und *Lungenkrebs* sind Folgen aus den beiden vorherigen Knoten. Die Blattknoten sind ein auffälliges Bruströntgen (chest\_X\_ray) und Atemnot (dyspnoea).

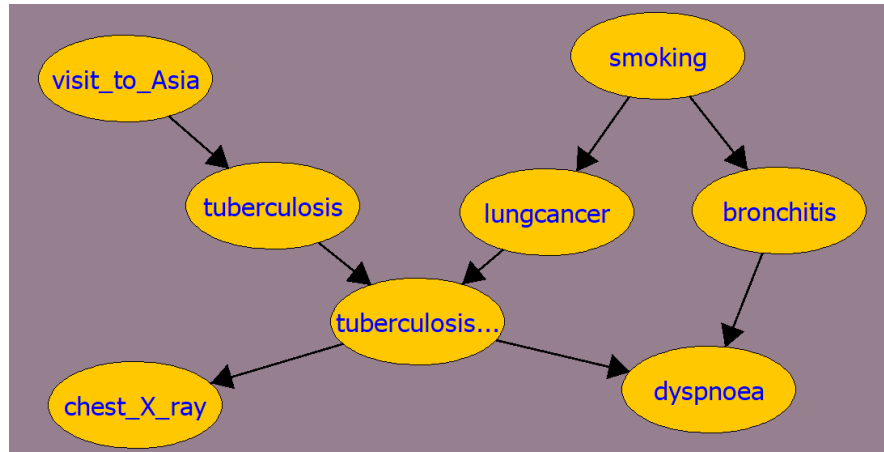


Abbildung A.8: Bayes-Netz „Asien“ erstellt mit *SamIam*

Ebenfalls besteht die Möglichkeit sich für vorliegende Bedingungen die Wahrscheinlichkeiten eines auffälligen Bruströntgens anzeigen zu lassen. Die visuelle Darstellung ist im Vergleich zu *GeNIe* etwas pragmatischer und einfacher. Für das Projekt jedoch sind die Funktionen und Darstellungen ausreichend.

Nachdem der Umgang mit *SamIam* im Bezug auf das Erstellen einer Struktur eines Bayes-Netzes erlernt ist, wird im nächsten Schritt der Verknüpfung zwischen *SamIam* und *bnlearn* betrachtet und getestet.

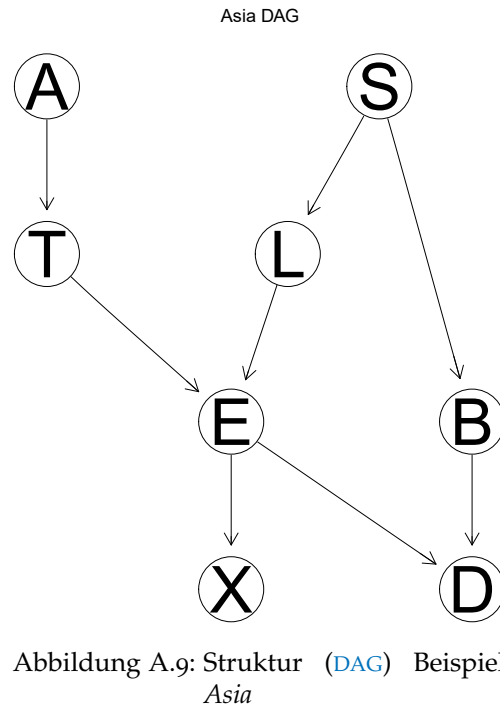
## A.5 VERKNÜPFUNG SAMIAM UND R

Der Gedanke für das Projekt ist, dass die Struktur des Bayes-Netzes bzw. mehrerer Bayes-Netze graphisch mit *SamIam* erstellt und als *.net* (*Hugin Net File*) abgespeichert werden. Das verwendete Format macht es möglich, ein Bayes-Netz mittels *bnlearn* einzulesen und weiter zu verarbeiten. In diesem Abschnitt wird ein Beispiel aus der *bnlearn* Dokumentation verwendet, da für dieses Daten vorliegen und somit ein realer Fall nachgebildet werden kann.

Zu Beginn wird das Bayes-Netz eingelesen und als Variable abgelegt. Zum Überprüfen der Struktur wird im nächsten Schritt das Bayes-Netz graphisch dargestellt s. [Abbildung A.9](#). Für eine erleichterte Übersicht und verringerte

Schreibarbeit werden die Knoten mit Kürzeln dargestellt. In [Abbildung A.8](#) ist die Struktur in Klarnamen zu sehen.

Zum Lernen der Parameter benötigt man ein noch nicht vollständiges Bayes-Netz (Struktur ausreichend). [Listing B.2](#) zeigt den Programmcode für die Verarbeitung. Im ersten Schritt wird aus dem eingelesenen Bayes-Netz die formale Gleichung der Struktur extrahiert. Im nächsten Schritt wird diese verwendet, um eine reine Struktur (dag) zu erstellen. Das Paket *bnlearn* stellt die Daten für das Beispiel *Asia* zur Verfügung. Diese werden eingelesen und in zwei Teile aufgeteilt. Der erste Teil dient hierbei dem Lernen der Parameter, wohingegen der zweite Teil dem späteren Testen dient. [Tabelle A.1](#) zeigt einen Ausschnitt der Daten. Insgesamt enthält der Datensatz 5000 Befragungen.



A	S	T	L	B	E	X	D
no	yes	no	no	yes	no	no	yes
no	yes	no	no	no	no	no	no
no	no	yes	no	no	yes	yes	yes
no	no	no	no	yes	no	no	yes
no	no	no	no	no	no	no	yes
no	yes	no	no	no	no	no	yes
no	no	no	no	no	no	no	no
no	yes	no	no	yes	no	no	yes
no	yes	no	no	yes	no	no	yes
no	yes	no	no	yes	no	no	yes

Tabelle A.1: Ausschnitt der Daten *Asia*

Mittels den Trainingsdaten werden die Parameter für das Bayes-Netz erlernt. In diesem Schritt erhält jeder Knoten seine möglichen Zustände mit Wahrscheinlichkeitsverteilung. Die Klasse der entstehenden Variable ist nun "bn.fit". Der Befehl `coef(bn.asia)` gibt die Verteilungen eines jeden Knoten aus. Es ist auch möglich die Wahrscheinlichkeiten für einen Knoten zu betrachten, hierfür übergibt man als Parameter nicht das gesamte Netz sondern nur



den gewünschten Knoten. [Tabelle A.2](#) zeigt die Verteilung für ein auffälliges *Bruströntgen* ( $X$ ) gegeben seiner direkten Abhängigkeit *Tuberkulose* bzw. *Lungenkrebs oder Bronchitis* ( $E$ ).

X \ E	E	
	Yes	No
Yes	0.994	0.046
No	0.006	0.954

Tabelle A.2: Wahrscheinlichkeitsverteilung für ein auffälliges Bruströntgen Beispiel *Asia*

Nachdem die Parameter erlernt sind, ist das Bayes-Netz vollständig und für den "Einsatz" bereit. Neben Berechnungen von Inferenzen für bestimmte Fälle ist es auch möglich ein Knoten in seinem Wert vorherzusagen. Die eingelesenen Daten sind in zwei Teile aufgeteilt. Die Testdaten werden im Folgenden verwendet, um vorher zu sagen, ob das Bruströntgen Auffälligkeiten aufweist oder nicht. Bei einem Vergleich mit den Originaldaten kann man die Sicherheit einer korrekten Vorhersage bestimmen. Es werden für 2500 weitere Beobachtungen durch das Netz vorhergesagt, ob das Bruströntgen *positiv* oder *negativ* ausfällt. Der Vergleich mit den Originaldaten ergibt, dass lediglich in 94 Fällen eine falsche Vorhersage getroffen wird. In 96.24% kann das Bayes-Netz das korrekte Ergebnis vorhersagen. Verbessern lässt sich der Wert durch eine größere Anzahl an Daten für das Lernen. Analog zur Einführung in *bnlearn* können Inferenzen berechnet werden.

## R-CODE

---

### B.1 FIBONACCIFOLGE

---

```
# Fibonaccifolge (an)n>=1
# allgemeine Formel
fibonaccifolge <- function(n){
  a <- (1 / sqrt(5)) * (((1+sqrt(5))/2)^n - ((1-sqrt(5))/
    2)^n)

  return(a)
}

# unterschiedliche Folgen n=8,9,10
n <- 8
a8 <- (1 / sqrt(5)) * (((1+sqrt(5))/2)^n - ((1-sqrt(5))/2)^n)
n <- 9
a9 <- (1 / sqrt(5)) * (((1+sqrt(5))/2)^n - ((1-sqrt(5))/2)^n)
n <- 10
a10 <- (1 / sqrt(5)) * (((1+sqrt(5))/2)^n - ((1-sqrt(5))/2)^n)

# Ausgabe der Folge
a_man <- c(a8,a9,a10)
a_man

# Aufruf ueber Funktion
n = c(8,9,10)
a_aut <- fibonaccifolge(n)
a_aut

# Vergleich
all.equal(a_man, a_aut)
```

---

Listing B.1: Fibonaccifolge programmiert in R

### B.2 EINLESEN UND PARAMETER LERNEN BAYES-NETZ BEISPIEL ASIA

---

```
# load net, create with SamIam
net <- read.net("asia_dag.net")

# plot net
graphviz.plot(net, main = "Asia Bayes Network")

# extract modelstring
asia.modelstring <- modelstring(net)
```

```

# generate new dag
dag <- model2network(asia.modelstring)

# load data
data(asia)

# learn parameters
traing.data <- asia[1:2500,]
bn.asia <- bn.fit(dag, data = traing.data, method = "mle")

# show parameters
coef(bn.asia)

# predict values node X
test.data <- asia[2501:nrow(asia),]
predict.X <- predict(bn.asia, node = "X", data = test.data, method = "
    bayes-lw")
all.equal(test.data[,7], predict.X)

```

---

Listing B.2: Bayes-Netz in R einlesen und bearbeiten, Beispiel *Asia*

### B.3 ALGORITHMUS ZUM ZUSAMMENFÜGEN EINZELNER SUB-BAYES-NETZE

---

```

## read in multiple bayes nets from working directory
# return list with multiple bayes nets
readIn_multiNet <- function(net_names){
  bnet <- list(NULL)
  for(i in 1:length(net_names)){
    # merge strings
    cmd = paste("read.net('", net_names[i], ".net'")", sep = "")
    # execute
    bnet[[i]] <- eval(parse(text = cmd))
  }

  return(bnet)
}

## build a modelstring from multiple bayes nets
# return string
buildBNstring <- function(BN){
  stringBN <- modelstring(BN[[1]])
  for (i in 2:length(BN)) {
    stringBN <- paste0(stringBN, modelstring(BN[[i]]), sep = "")
  }

  return(stringBN)
}

```

```
## separate each brackets and put together an array
# return separated string
collaps_modelstring <- function(modelstring){
  string <- strsplit(modelstring, split = '']')
  col_string <- ""
  for(i in 1:length(string[[1]])){
    temp <- string[[1]][i]
    temp <- paste0(temp, "]", collapse = "")
    col_string[i] <- temp
  }
  return(col_string)
}

## remove duplicates from array and build a DAG
# return DAG
modelstring_to_dag <- function(modelstring){
  mstring <- unique(modelstring)
  mstring <- paste0(mstring, collapse = "")

  dag <- model2network(mstring)

  return(dag)
}

## Read in multiple Bayes-nets and build one DAG
bn_names <- c("Kratzer", "Schmelzbruch", "Schlierenbildung", "Rauch_Qualm",
  "Blasenbildung", "Gardineneffekt")
bayesnets <- readIn_multiNet(bn_names)
bigstring <- buildBNstring(bayesnets)
modelstring <- collaps_modelstring(bigstring)
dag <- modelstring_to_dag(modelstring)
```

---

Listing B.3: Mehrere Subnetze in R einlesen und zu Einem zusammenfügen

## LITERATUR

---

- [1] Daniel Obszelka und Andreas Baierl. *Statistisches Programmieren mit R. Eine ausführliche, übersichtliche, spannende und praxiserprobte Einführung*. Hrsg. von Sybille Thelen. 1. Aufl. Wiesbaden, Germany: Springer Fachmedien Wiesbaden GmbH, 2020.
- [2] M. Erfert. *Grundlagen der Fertigungsverfahren mit Systemanalyse der Maschine und des Prozesses*. Präsentation, intern und vertrauenswürdig. 2017.
- [3] Fraunhofer. *Fraunhofer-Gesellschaft zur Förderung der angewandten Forschung e.V.* 2021. URL: <https://www.fraunhofer.de/>.
- [4] Fraunhofer IWU. *Fraunhofer-Institut für Werkzeugmaschinen und Umformtechnik IWU*. 2021. URL: <https://www.iwu.fraunhofer.de/>.
- [5] Fraunhofer Institut IWU und Industriepartner. *Entwicklung wissensbasierten System zur Parametrierung von Kunststoff-Blasformmaschinen*. Projektbeschreibung, internes Dokument, vertrauenswürdig. 2021.
- [6] Uffe B. Kjærulff und Anders L. Madsen. *Bayesian Networks and Influence Diagrams: A Guide to Construction and Analysis*. 2. Aufl. New York: Springer Science+Business Media New York, 2013.
- [7] MBK Maschinenbau Koetke. *Einflussfaktoren bei Blasformprozessen*.
- [8] Rudolf Kruse, Christian Borgelt, Christian Braune, Frank Klawonn, Christian Moewes und Matthias Steinbrecher. *Computational Intelligence Eine methodische Einführung in Künstliche Neuronale Netze, Evolutionäre Algorithmen, Fuzzy-Systeme und Bayes-Netze*. Hrsg. von Prof. Dr. Wolfgang Bibel, Prof. Dr. Rudolf Kruse und Prof. Dr. Bernhard Nebel. 2. Aufl. Wiesbaden, Germany: Springer Fachmedien Wiesbaden GmbH, 2015.
- [9] D. Kötke. *Technischer Leitfaden für Extrusionsblasformmaschinen*. Bachelorarbeit, internes Dokument, vertrauenswürdig. 2015.
- [10] Professor Dr. Uwe Ligges. *Programmieren mit R. Statistiken und ihre Auswertung*. 3. Aufl. Berlin und Heidelberg, Germany: Springer-Verlag Berlin Heidelberg, 2008.
- [11] Marco Scutari. *Understanding Bayesian Networks with Examples in R*. Department of Statistics University of Oxford.
- [12] Marco Scutari und Jean-Baptiste Denis. *Bayesian Networks With Examples in R*. 2. Aufl. New York: CRC Press Taylor & Francis Group, 2021.
- [13] Dr.-Ing. Michael Thielen, Prof. Dr.-Ing. Peter Gust und Dr.-Ing. Klaus Hartwig. *Blasformen von Kunststoffhohlkörpern*. Hrsg. von Jörg Strohbach. 2. Aufl. München, Germany: Carl Hanser Verlag GmbH & Co. KG, 2020.