h_da

HOCHSCHULE DARMSTADT
UNIVERSITY OF APPLIED SCIENCES

Darmstadt University

- Department of Mechanical Engineering and Plastics Technology-
- Course of studies Mechatronik-

Data processing within production engineering

Interactive knowledge repository using a Bayesian network for the setup process of an extrusion blow molding machine

Project report of the professional practical phase for obtaining the academic degree

Bachelor of Science (B.Sc.)

submitted by

Julian Schneider

Matriculation number: 760221

Filed on: 6 December 2021

Winter semester 21/22

Examiner/ Supervisor : Prof. Dr.-Ing. Karl Kleinmann Operat. Supervisor : Dipl. Ing. Rebekka Zache Company supervisor Supervisor : Dr.-Ing. Uwe Frieß

Julian Schneider: *Datenverarbeitung innerhalb der Produktionstechnik*, Interaktiver Wissensspeicher mittels eines Bayes Netzes für den Einrichtungsprozess einer Extrusionsblasformmaschine , © 6. Dezember 2021

S P E R V E R M E R K

Dresden, 6 December 2021

AFFIDAVIT

affidavit
I, **Schneider Julian born on 22.05.1996**, hereby declare that I have written this thesis on my own and that I have not used any other aids than those indicated.

Insofar as I have d r a w n on external materials, texts and thought processes, my explanations contain complete and unambiguous references to the authors and sources. All other contents of the presented work originate from me in the copyright sense, as far as no references and quotations are made.

I assure that I have not yet submitted this work to any examination or examination authority.

I understand that an attempt to deceive has been made if the above declaration proves to be incorrect.

Dresden, 3 December 2021

J. Schneider

Julian Schneider

V O RW O RT

This thesis was written during the practical phase (BPP) of the mechatronics degree program. At the end of the bachelor's degree, the university requires a BPP lasting twelve weeks. This is usually carried out in companies outside the university. Thus, the first activities of an engineer are to be carried out and become acquainted with. This phase also serves as an opportunity to establish initial contacts in industrial sectors and branches. The following BPP was completed at the Fraunhofer Institute IWU in Dresden. Due to my curiosity and personal interest in the topic of data analysis, cyber-technical production systems and the new artificial intelligence (AI) methods, the step into a new subject area was taken.

The project assignment foreshadowed exciting and interesting tasks, which seemed challenging at the same time. During my studies, I was not able to gain any insights and experience in the areas of data analysis and AI. Thus, it was necessary to familiarize oneself with the new field in order to learn as much new knowledge as possible. Besides the personal interest in the topic, this field promises an increasing presence in the future. Due to the current Covid19 pandemic, the work was mainly done away from the institute from the home office, as it was forbidden to use the office space on site. Despite the special circumstances I would like to thank my supervisor Dipl. Ing. Rebekka Zache. She always tried to keep in contact with me and to create an "on-site office atmosphere" as good as possible. In case of questions or problems she offered her help and always supported me. Further thanks go to Prof. Dr.-Ing. K. Kleinmann for the always very good support of the BPP. He always had an open mind.

ear and you could trust his support.

I hope you enjoy reading this work.

Dresden, 6 December 2021, Julian Schneider

COMMUNICATION

The aim of the project is to develop an interactive knowledge repository that supports the operator during the setup process of extrusion blow molding machines. The setup process can be divided into two phases. In phase one a so-called *preform is* produced. In the project work, only this phase of the setup process is considered and evaluated.

In cooperation with the industry partner, a preliminary structure of the Bayes network was created. Here it is necessary to complete the present design. Subsequently, an algorithm has been developed with which it is possible to predict the most probable cause of the error and thus to give the operator recommendations to intervene in the process accordingly. In the next step a concept has been developed how Bayes nets can be learned and trained. This makes it possible to train the knowledge store individually for each machine.

Finally, a user interface was created for the interaction between the assistance system and the user. The interface presents recommendations visually and the user is able to make inputs. A web-based interface was developed with the R-compatible package *Shiny.*

The preliminary design contains a tab for the overview of the entire Bayes net, as well as a tab for the display of an individual error and possible error causes. There, the operator can select the error he is interested in via a drop-down menu and he receives a Bayes net for this error. Another tab is used to import new data sets as well as to train and learn the Bayes net.

In the last tab, error causes are calculated and recommendations for intervention in the process are given. The setter selects a current error pattern. He then confirms this. He receives a graphical representation via the error Bayes network, which error cause is present, as well as a text output. The operator then provides feedback as to whether he was able to intervene in the process as recommended and whether this eliminated the error. If the error is not completely eliminated, he receives further recommendations.

At the moment, no real data of the machine has been collected. Likewise, no tests or implementations have been carried out. "Small" tests for learning/training, as well as predictions of the cause of the error, were carried out using synthetically generated data.

INTERNALVERITY

A P P L I C A T I O N P R O V I D E S

TABELLEVERATIONS

LISTINGS

IWU	Fraunhofer Institute for Machine Tools and Forming Technology IWU
BPP	Practical phase
ZIM	Central Innovation Programme for SMEs
AI	Artificial Intelligence
BN	Bayes nets
DAG	directed acyclic graph
AP	Work packages
CPU	Central Processing Unit
GUI	Graphical User Interface

Part I

PROJECTION

1 F R A U N H O F E R - I N S T I T U T I W U

The Fraunhofer Institute for Machine Tools and Forming Technology IWU (IWU) in Dresden is part of the Fraunhofer-Gesellschaft zur Förderung der angewandte Forschung e. V. The Institute for Machine Tools and Forming Technology is considered a driving force and is a leader in the field of production engineering research and development. More than 650 employees are active at five locations (Dresden, Chemnitz, Leipzig, Wolfsburg and Zittau), serving different industries. From the classic automotive industry, to aerospace, to medical and microtechnology. The headquarters of the institute is located in Chemnitz.[4]

The Fraunhofer IWU is regarded as an inventive genius and pioneer for resource-efficient production and, at the same time, intelligent production plants.
The main focus of the institute is on machining and forming process technologies. The complete process chain is kept in view. The topics of *cognitive production systems*, the *complete networking of production plants* and *artificial intelligence are increasingly coming to the* fore.

A major future-oriented project is the production of fuel cells. Under the coordination of IWU, a research platform is being set up throughout Germany. There, the technological potential and cost-optimized production are to be further advanced. Another project worth mentioning is the *E3 research factory for resource-efficient production*, which aims to achieve resource-efficient and flexible production using renewable energies.
The Dresden site is involved in adaptronics and acoustics, generative manufacturing, and mechanical joining and medi- cal engineering.[4]

The Fraunhofer Gesellschaft was founded in 1949 and currently (as of 2021) has 75 institutes and research facilities in Germany. These offer approximately 29,000 employees the opportunity of a job. The majority of these are in the natural and engineering sciences. Europe-wide, it is considered the leader in applied research. In addition to the institutes in Germany, there are others in other European countries. Furthermore, institutes can be found in North and South America, Asia, Africa and the Middle East. Cooperations are established with companies and research partners worldwide. [3]

2 ANALLING

The BPP is completed on the basis of an advertised position for a currently running industrial project (Central Innovation Programme for SMEs (ZIM)), for which a student assistant is being sought. The project in question is being carried out in cooperation with an industrial partner. This partner is active in the field of blow moulding technology. In addition to the manufacture and production of 3D plastic hollow bodies, he develops and manufactures blow molding machines. These are used for in-house production as well as sold to customers. He serves and represents a wide range of customers.

As a student assistant, you get as much insight as possible into various topics and areas, while working independently and on your own responsibility. You work independently on the assigned tasks and receive support from your supervisor if necessary. You are deployed as a "full-fledged worker" and thus have the opportunity to make important contributions to a project.

In the next section, the project mandate and the task are described in more detail, followed by the chapters Concept and Implementation. Finally, the current state of the project at the end of the BPP, as well as an outlook and open points are described and shown.

2.1 project assignment/task definition

The project order/project description provides for the development of a "knowledge memory/assistance system" for the setup process of a blow molding machine, which supports the setter/operator when starting a new production/setup process and suggests tips for troubleshooting. The so-called parameterization of plastic blow molding machines is to be carried out on the basis of the *knowledge-based system.*

The development of an AI-based process for online quality control of plastic blow molding machines, as well as other AI components of the new optimization system, is attributed to the institute as a subproject. In the first step, the set-up process and in a second step the running production will be considered. The knowledge repository is to be built and developed on the basis of so-called Bayes nets (BN). The industrial partner is responsible for the data acquisition and collection, whereby the institute defines the framework for this and supports it if necessary.

The aim of the project is to make the set-up process more cost- and time-efficient and at the same time to bring more objectivity into the process. Thus, the knowledge of the individual processors regarding the effect of the parameters, together with generally known rules and peculiarities of the machine, machine-readable and permanently stored. Input variables influencing the production process must be identified and possibilities for automatic recording must be developed. Attention must be paid to ensuring the required data quality and intelligent data management. Subsequently, a dashboard module is required which supports the operator in setting up the blow molding machine and serves as an interaction option with the system[5].

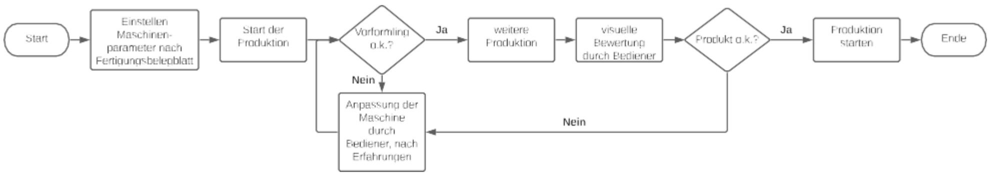

Start
Einstellen Maschinen-parameter nach Fertigungsbelegblatt
Start der Produktion
Vorformling o.k.?
Ja
weitere Produktion
visuelle Bewertung durch Bediener
Produkt o.k.?
Ja
Produktion starten
Ende
Nein
Anpassung der Maschine durch Bediener, nach Erfahrungen
Nein

Figure 2.1: Flow chart *current status of the* blow molding machine setup process

Figure 2.1 shows the current setup process as it is currently practiced. It can be seen that the activities of the "setter" are only based on *experience and knowledge.* The project aims to store the existing knowledge in a system "permanently" and thus to continue to use the knowledge after the "retirement" of an experienced employee. In addition to the own knowledge, a *fault tree* Figure 2.2 is used, which has been designed and developed "over the years". Possible causes of an existing error can thus be identified.

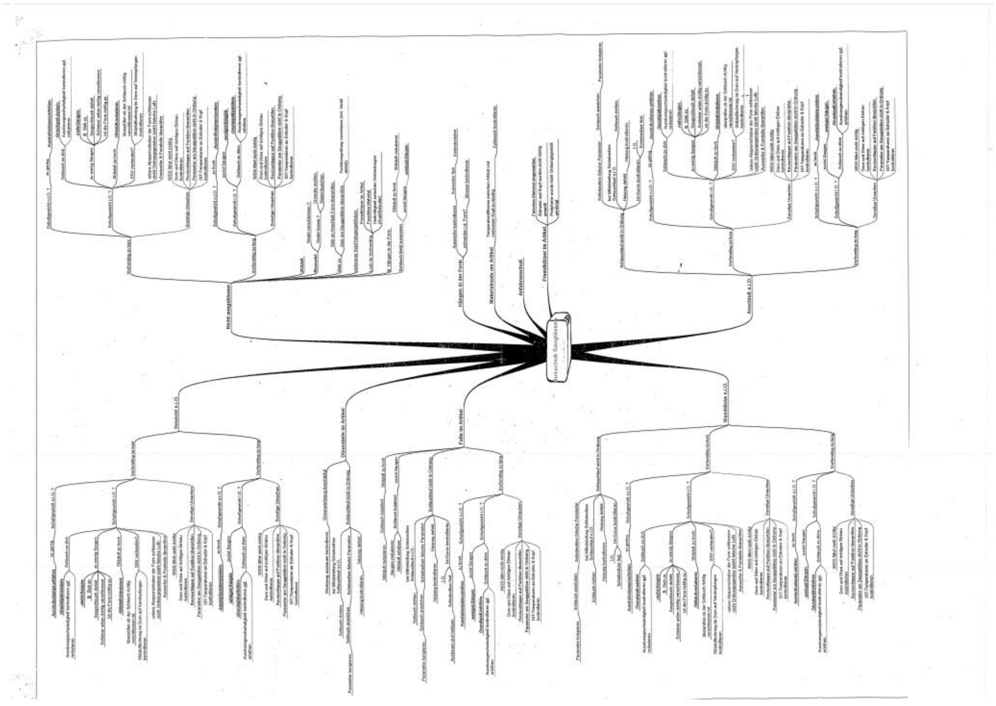


Figure 2.2: Fault tree for influencing factors in blow molding processes (*customer document*) [7].

3 C O N C E P T A N D A U D I T I O N

3.1 concept

This chapter describes the concept and a possible implementation of the project. A clearly defined framework is based on the project assignment and description. Within this, the project is divided into individual work packages and explained individually. Therefore, it is not necessary to design a complete concept for the project, but "several smaller concepts for the individual work packages".

A general approach to describe the final process is shown in Figure 3.1, which is developed from the project description. It is intended to describe the support and the possibility of intervention in the process of setting up. A *target vector* is to be used, whereby the quality of a product can be described and this therefore has an influence on the calculations and outputs of the assistance system. In this approach only the setup process is considered. An intervention during the production is not intended here. It has been agreed with the industry partner that, in a first step, the setup process and production will be considered separately from each other and that production will initially be left out of the equation.

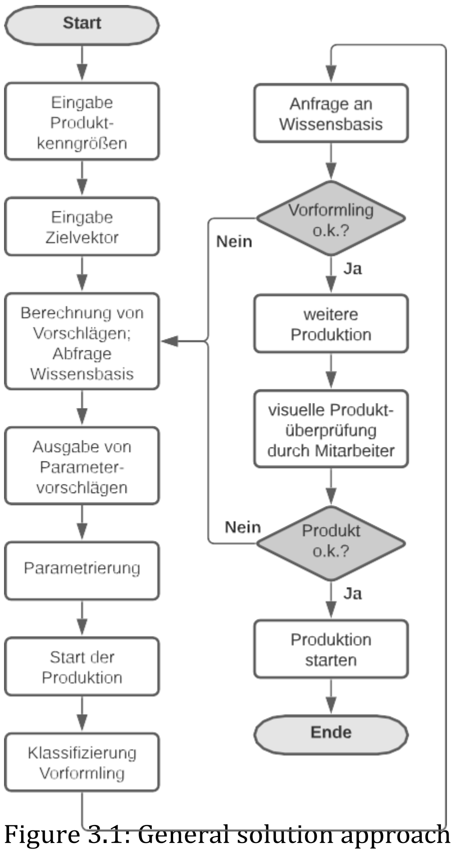

Start
Eingabe Produkt-kenngrößen
Eingabe Zielvektor
Berechnung von Vorschlägen; Abfrage Wissensbasis
Ausgabe von Parameter-vorschlägen
Parametrierung
Start der Produktion
Klassifizierung Vorformling
Anfrage an Wissensbasis
Vorformling o.k.?
Nein
Ja
weitere Produktion
visuelle Produkt-überprüfung durch Mitarbeiter
Produkt o.k.?
Nein
Ja
Produktion starten
Ende

Figure 3.1: General solution approach

The setup process also allows- if there is a subdivision into two phases, of the preform and that of the finished product. The identical procedure is that first the phase of the preform is of the preform is considered and in a second phase the finished product. As soon as the requirements for the preform are met (according to the test protocol), the second phase is carried out and the finished product is assessed and, if necessary, the machine is readjusted or parameters are adjusted.

The construction and development of the assistance system is divided into two work packages. WP 1 is the "data acquisition and collection" and WP 2 is the construction of the "assistance system". These work packages are described and explained in more detail below.

WP 1: Data collection and compilation

The assistance system needs data both to train its own networks and to know the current state of the machine. Data acquisition forms the *interface* between the machine and the knowledge store. The institute defines the required framework and the format of the data. The final implementation is in the hands of the industry partner.

The required data can be divided into two or three categories. On the one hand, there are machine parameters that are stored in the control system and can be recorded automatically, and on the other hand, there are "manual inputs", which include the error patterns and parameters that cannot be recorded automatically but require the input of the operator. How the data acquisition is implemented is not in the hands of the institute. The concept provides that a framework for the format of the data is defined and handed over to the customer. In addition to this framework, another task is to define which data is required. The data should be stored at a central point and provide access to the assistance system. On the basis of the database it is thus possible to learn and train the BN and the associated parameters.

AP 2: Assistance system

This work package is the much larger and more elaborate one, since the knowledge repository is completely designed and developed there. The project description defines that the assistance system should work with the help of Bayesian networks. For this purpose, the influence parameters for errors are determined and recorded. From the collection of these single Bayes nets a complete main net is created. Through AP 1 the system has data and can therefore "learn and train" the net. An algo- rithm is developed to determine the most influential parameter to fix the fault. Finally, a dashboard serves as an interface and interaction facility with the human. The system has access to the database and provides the setter with information that can be used to correct errors.

Figure 3.2 shows the interaction between the human and the assistance system and the planned process.

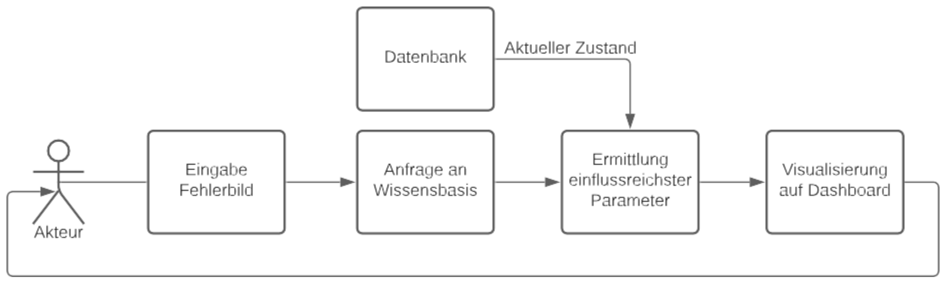


Figure 3.2: Flowchart interaction human and assistance system

3.2 structure and layout

The following overview is intended to illustrate the course of the project in the implementation or the BPP and to show which activities and topics are dealt with. In addition to the implementation of individual work packages, a familiarization with certain topics is required beforehand, since no experience or knowledge has been gained during the studies.

Thematic areas and implementation

- Familiarization with the subject of plastic extrusion blow moulding and blow moulding machines (Section A.1)
- Introduction to and discussion of Bayesian networks (section A.2)
- Learning and familiarization with programming with R (Section A.3)
- Familiarisation with the R package "bnlearn" and construction of the experimental network (section 4.1)
- Familiarization with the graphical design of a Bayesian network using BayesianFusion "GeNIe" or "SamIam" (Section A.4)
- Linking "R bnlearn" and "SamIam" (section A.5)
- Creating the Bayes net structure, identifying the parameters influencing a fault (section 4.2)
- Designing an algorithm to determine the most influential parameter on a fault pattern (Section 4.3)
- Data collection and processing, preparation of the specifications for industry partners and subcontracting (section 4.4)
- Creating a user interface using "Shiny" for operator interaction (Section 4.5)

Following Chapter 4 Implementation, the status of the handover at the end of the BPP is presented. As this is a two-year project, it will not be completed within the twelve weeks. It is also important to mention that the *kick-off of* the project was carried out before the beginning of my professional practice phase.

4 CONSUMPTION

This section explains the implementation of individual work packages (AP) and the overall project. No knowledge is available for the topics covered, software used and programming language. Before the implementation of an AP is started, a respective familiarization takes place. Appendix A describes the different familiarization processes in more detail. Section A.1 gives an introduction to *blow molding*, the process and the used method *extrusion blow molding with squeezing over*. Section A.2 gives a brief insight into the topic of Bayes nets, which serves as the basis for this project. Section A.3 describes the programming language **R**. The software used to create the Bayes nets is described in subsection A.4.1 and subsection A.4.2.

The following sections describe the implementation of individual work packages and at the same time the chronological development of the project. Individual sections of the project are considered and implemented separately. We will start with the **R-package** *bnlearn* and the construction of an experimental network.

4.1 r-package *bnlearn* and set-up experimental network

In this section, the **R-package** *bnlearn* will be examined and a test network for further tests will be built. The first step is to read the documentation of this package. [1] The **R-package was** developed to learn the graphical structure of a Bayes net, to predict parameters and to determine inferences. It was first released in 2007 and has been under continuous development ever since.

The documentation contains useful examples and also sample data with which parameters can be learned. Based on a dataset from the collection, the Bayesian network created later in *SamIam is* derived
"Asia" (see Figure A.8). In addition to the documentation, the work *Bayesi- an Networks with Examples in R* by "M. Scutari" and "J.-B. Denis" is used, which works closely with the documentation.[12]
In the project the data will always be *discrete.* Another possibility is the case of *continuous* values, also called *Gauss Bayes networks.* However, we will not consider this case further. From the book *Bayesian Networks with Examples in R* [12] the first chapter *The Discrete Case: Multinomial Bayesian Networks* was worked on and thus the basic handling, programming and use of **R** and *bnlearn* was learned.

1 Documentation of the **R-package** *bnlearn* `https://www.bnlearn.com/`

A survey serves as the basis for the example from the chapter. One would like to be able to predict the preferred means of transport of a person. In the survey, both women and men are asked. The age is divided into three categories, and a distinction is also made between gender, education, residential location and employment. Table 4.1 shows an extract of the data from the survey on transport use. A total of 300 people of different categories are interviewed.

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

Table 4.1: Excerpt from *survey* on transport use

Expert knowledge is used to create the structure of the Bayesian network. In addition to the possibility of defining the structure independently, there is also the possibility of learning the structure from the existing data via algorithms. However, since in the project the structure is defined and created by "experts", this variant can be neglected. In the following, two variants are shown to define the structure in **R**, see Listing 4.1. In the first case, an empty graph with nodes is created and then the dependencies are passed. In the second case, the structure is generated by a for- male equation. Here, the nodes are defined in square brackets and the *parents,* if any, are defined behind a |.

```
# First way to create a DAG
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
# Second way to create a DAG, via formal equation with
    dependencies
dag2 <- model2network(" [A][ S] [E|A: S] [O|E] [R|E] [T|O: R] ")
```

Listing 4.1: Creating a *DAG* for Bayes net *survey*

The result can be seen in Figure 4.1. The *root nodes* are age (A) and sex (S), and the *leaf node* is the mode of transport (T). Root nodes are those which have no parents. The reverse is true for leaf nodes, which have no children. One could also speak of a beginning (*root node*) and end (*leaf node*) of the graph.

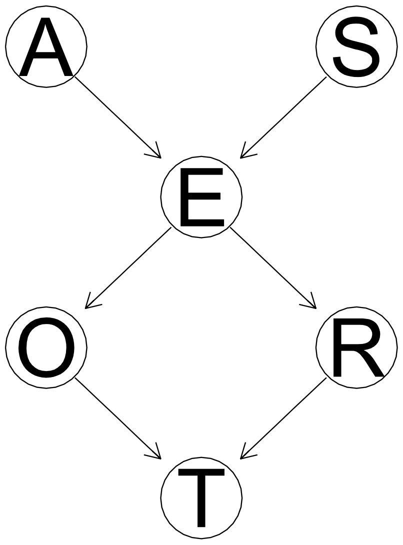


Figure 4.1: DAG from Bayes net *survey*

The creation of the structure for the later knowledge store will be the first essential point in the implementation of the project. The code shown above serves as an example to i m p l e m e n t a Bayes net directly in **R** using *bnlearn*. In addition, there is the option to implement the structure using a software "graphically" and to import these into **R.** The start and the possibilities of the graphical implementation are described in section A.4.

After the structure of the network is created, the *parameter learning* can be performed using the read-in data. For this purpose **R** offers different algorithms, by default *Maximum Likelihood parameter estimation* is used. In the following you can see the code line (Listing 4.2) with which the parameters are learned and a finished Bayes net is generated. Mandatorily, the structure and data for learning must be passed. In this case the default algorithm is also specified. The states that a node can assume are extracted from the data and do not have to be defined beforehand.

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

Listing 4.2: Parameter learning using *maximum likelihood parameter estimation*

By means of the steps *Define structure* and *Learn parameters* a complete *Bayes net* is generated which is available for further use.

With this it is now possible to calculate inferences for certain cases and to make predictions. The bar chart for a calculated inference is shown on the right-hand side (see Figure 4.2). The probabilities of the means of transport with a given place of residence and employment are calculated. Mathematically, this is written P $(T/R, O)$.

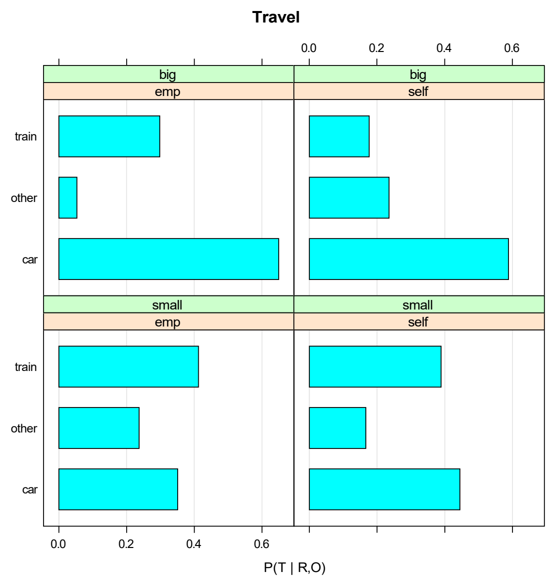


Figure 4.2: Bar chart Means of transport-use

The top line shows that the preferred mode of transport in a large city is the car. In the bottom row the the ratio evens out somewhat, train and car. Since this is not real survey data, there is no scoring. In a real survey, one would assume that the two lines would be reversed and that the train is mostly used in the big city, whereas in a smaller city car and train balance each other out.

If one wishes to determine the inference for a particular case, this is also possible. In Listing 4.2, the calculation is performed for the case where the gender is *male* and the mode of transport is *car*, given the formation *high*; mathematical expression $P(S = M, T = car/E = high)$. The case to occur is called *event* and given attributes *evidence*.

```
# Approximate Inference
# P(S = M,T = car | E = high)
cpquery(bn, event = (S == "M") & (T == " car"), evidence = (E == "high"))
```

Listing 4.3: Calculate inference for $P(S = M, T = car/E = high)$

The calculation yields a result of 0.3487621, which corresponds to a probability of occurrence of 34.876%. It can be stated that in 34.876% of the cases the person is *male*, uses a *car* as means of transport, given that the education is *high.*

After completing the first chapter of *Bayesian Networks with Examples in R, you will be able to* create and edit your own Bayesian networks. The next section describes how to create the Bayesian network structure.

4.2 create bayes-net structures

The extrusion blow molding process can be divided into two phases.

- Phase 1: Creation of the *preform*
- Phase 2: Manufacture of the *finished article/product*

Any defects that occur are assigned to the two phases. Thus, there is a defect list for the *preform* phase and one for the *finished product*. The industry partner provides these two lists. The structure of the Bayes net is to be determined by expert knowledge. For this purpose, a 90-minute meeting is held with the industry partner every week, during which influence parameters for a defect and interdependencies are jointly defined. The structure is defined and set up independently. As a result, an individual subnet is created for each fault, which is then integrated into a complete network.

The individual Bayes nets are generated using the *SamIam* software explained in subsection A.4.2. However, the main net is formed in **R**, since further processing takes place there.

Figure 4.3 shows the created Bayesian network for the defect *scratch in* the phase 1 *preform. The* influence parameters are defined as: *Impurity of the raw material, damage to the nozzle, burnt raw material on the nozzle surface* and *nozzle temperature*, whereby the nozzle temperature has an influence on the *burnt raw material on the nozzle surface in* addition to the influence on the defect *scratch.* All in all, for the phase 1 *preform* 13 sub-

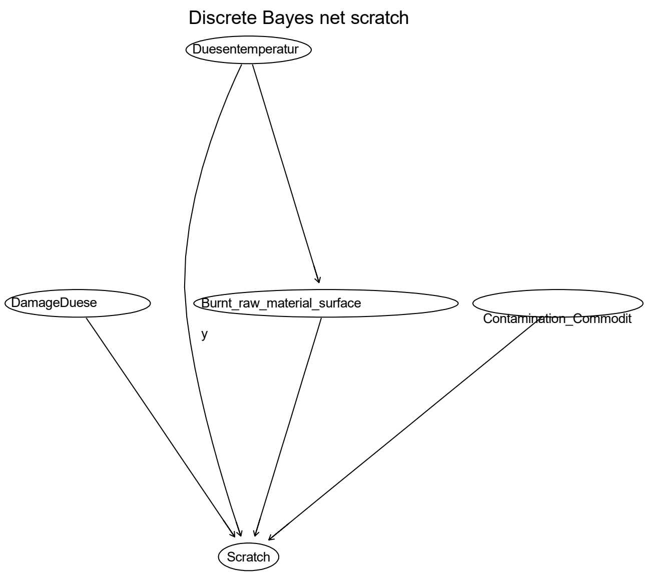


Figure 4.3: Sub-Bayes network for error "Scratch" created with *SamIam*

Bayes nets are created, which are then merged into a complete net using *bnlearn.* Below is a list of defined errors that can occur during the setup process.

Error list setup process phase "preform

1. Scratch
2. Melt fracture
3. Cloud formation/ milky looking cold hose
4. Preform glossy
5. Blistering
6. Curtain effect
7. Sag
8. Tube rolling
9. Hose run, straight run
10. Intestinal Effect
11. Room/ smoke formation
12. Hose length variations

A specially developed algorithm (see Listing B.3) is used for the assembly in **R to** form a main net. In the first step the names of the subnets to be read are specified and passed to the function *readIn_multiNet().* The function *buildBNstring() then* returns a list of all subnets to be read. This list is then passed to the function buildBNstring*()* which creates a common model string. In such a model string the nodes and their dependencies (edges) are defined. If a node has parents, this is formally specified.
The formal equation for the sub-Bayes network "scraper" from Figure 4.3 looks like this:

$$[DamageDuese][Duesentemperatur][VerbrannteRohsto\,f\,e\_Duesenober\,f\,Fläche|Duesentemperatur][Verunreinigung\_Rohsto\,f\,f][Kratzer|Baedigu\,ngDuese : Duesentemperatur : VerbrannteRohsto\,f\;\;e\_Duesenober\,f\,Fläche : Verunreinigung\_Rohsto\,f\;\;f\;\;] \tag{4.1}$$

The "string" is then converted into the formal equation needed to create a *directed acyclic graph.* Since this may contain duplicates, these must be removed. This task and the creation of the DAG is done by the function *modelstring_to_dag()*. The result is a large connected Bayes net, the so-called "main net".

Figure 4.4 shows the first seven defects from the *preform* phase, which are combined to form a Bayesian network. The structures of the Bayesian networks created in this way for phase 1 *preform* and phase 2 *finished product* (not yet implemented) are then also combined to create a structure for both phases.

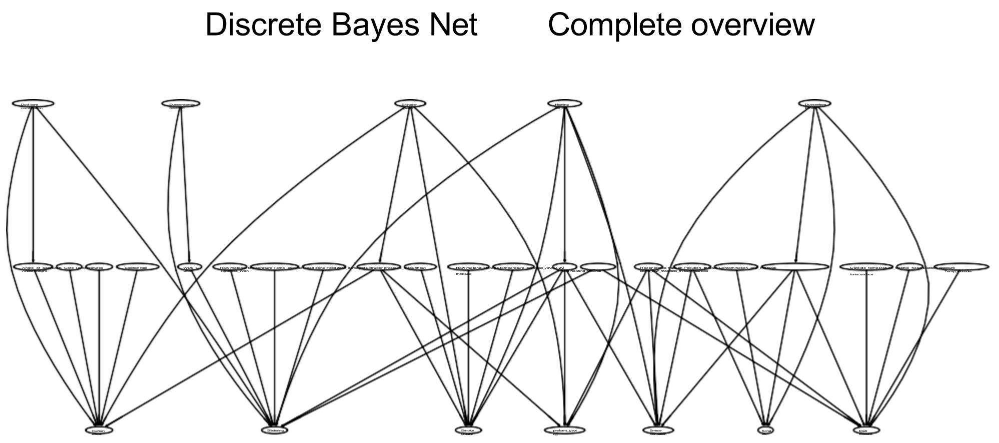


Figure 4.4: Sub-Bayes networks merged to form a whole

A Bayesian network has advantages because it can make "good" predictions even with few dependencies. At the same time, the small number of dependencies reduces the amount of data needed to generate usable results. As a rule of thumb, you need at least 10 data sets per combination. When designing the structure, it is therefore important to reduce the total number of dependencies and the number of edges on a node to a minimum.

4.3 determination of most influential parameters

In this chapter we deal with one of the "leading questions" of the project, how it is possible to select the most influential parameter/cause of an error and to make it recognizable to the operator. A self-developed algorithm is used for the determination.

The prerequisite is that the operator selects a current error image. Subsequently, the influence parameters that can be changed for this error are selected. In the further course of several loops, different *inferences* are determined for the error that has occurred and the selected parameters. Here the inferences for each possible state of a parameter "Y", with the *event that* the error has occurred, are calculated; $P(X = true/Y = lvl_i)$ with $i = 1 \ldots. n - lvl$. The individual calculated probabilities are stored in a list. For each parameter, differences between two possible states are calculated and then stored in another list. The idea behind this procedure: *The largest difference gives the two states of a parameter when changing from state "a" to state "b" that one minimizes the probability of the error occurring*.

Finally, the list of probability differences of all selected parameters and their states is sorted in descending order. The *most influential parameter* with a change of state is at the top of the list.

Figure 4.6 Example error "Scratch"). The described algorithm is represented by a flowchart in Figure 4.5. It is not considered here whether this change of state is possible in the present case, since the machine connection is not implemented in the first step. The operator must provide the system with feedback as to whether the intervention is possible or not. If it is not possible, then he receives the next suggestion, with the next smaller probability difference. In the sense of an AI, the connection to the machine and thus the query of the current state is inevitable. With the current state, the algorithm works on an *offline database*.

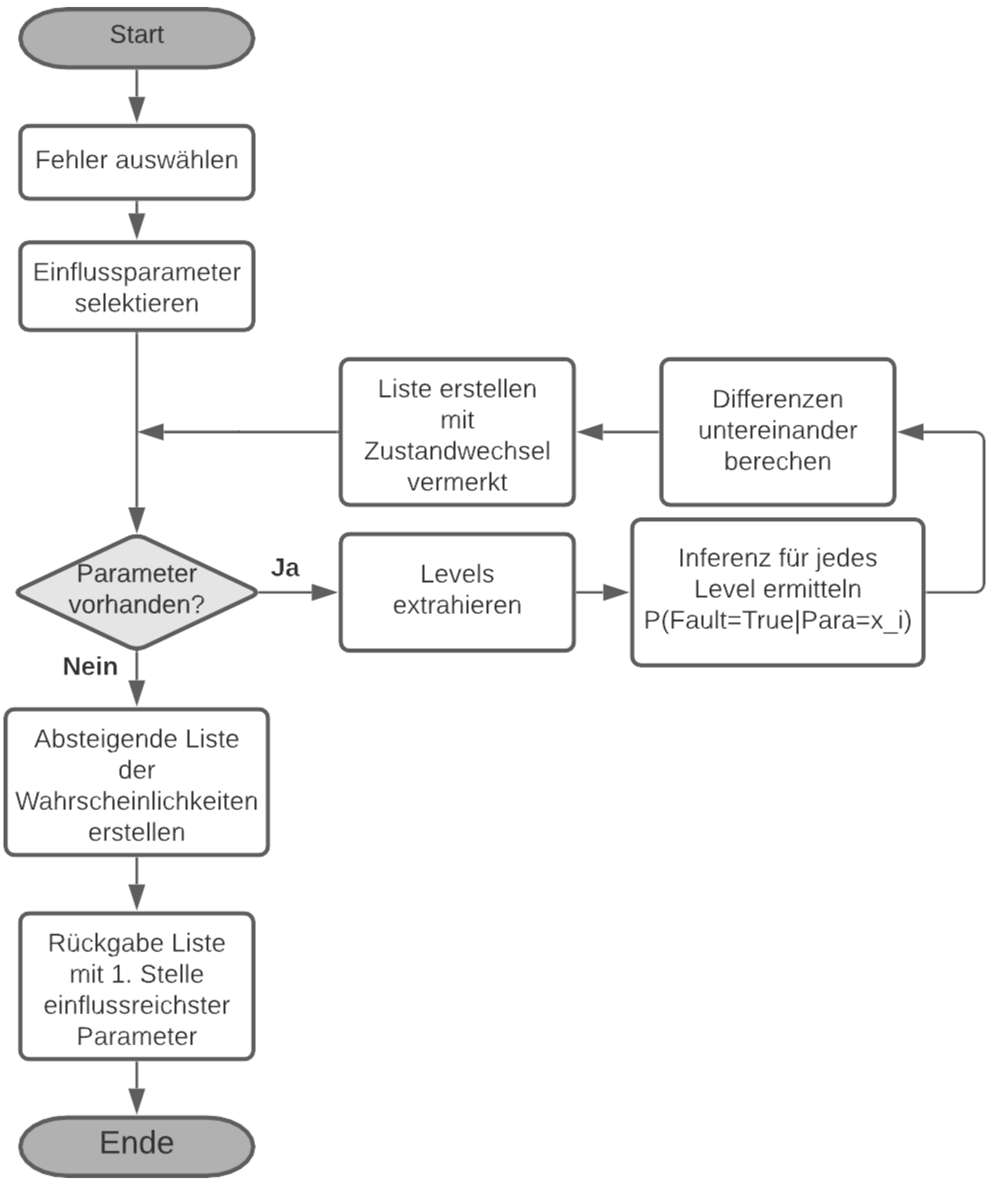
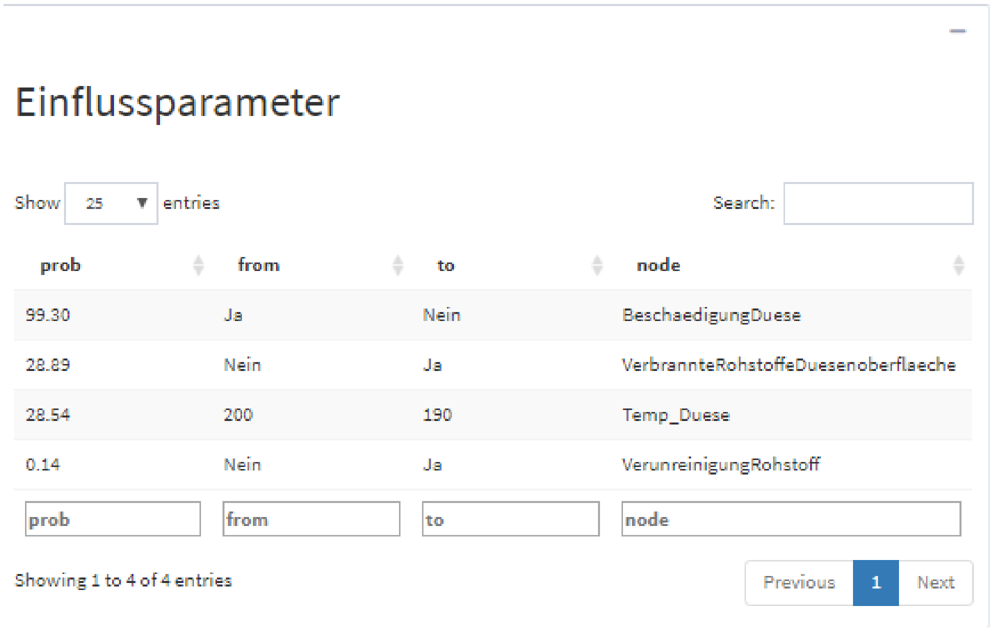

Start
Fehler auswählen
Einflussparameter selektieren
Liste erstellen mit Zustandwechsel vermerkt
Differenzen untereinander berechen
Parameter vorhanden?
Ja
Levels extrahieren
Inferenz für jedes Level ermitteln P(Fault=True|Para=x_i)
Nein
Absteigende Liste der Wahrscheinlichkeiten erstellen
Rückgabe Liste mit 1. Stelle einflussreichster Parameter
Ende

Figure 4.5: Flowchart determination of the *most influential parameters*

Several functions are defined and developed for the algorithm. In addition to the actual function for calculating the inference, the following functions are defined

Functions are defined which extract the possible influencing parameters from the entire Bayes net, determine the states of a parameter, calculate differences between the inferences of the states of a parameter and swap the alternating direction of the states if the difference has a negative value. Due to the scope, a de- tailed description of the individual functions is omitted.

This algorithm makes it possible to predict the cause of a fault for a given fault pattern. Ideally, the error pattern should be completely eliminated by eliminating the cause of the error, but in reality cases will occur in which several interventions must be made. Therefore, the creation of a list was made, in order to be able to name possible further error causes.



Einflussparameter

Show 25 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
prob	from	to	node

Showing 1 to 4 of 4 entries

Figure 4.6: Prediction *error cause for error "Scratch"*, descending list with probabilities

4.4 data collection and processing

Data acquisition is an eminent work package in this project because without data, the knowledge store cannot learn its networks and the experiences from previous setup processes are not stored. Data is needed to give the Bayesian network its intelligence.
Section 3.1, paragraph "Data collection" briefly explains that the data to be collected must be divided into two or three categories. The data stored in the controller are currently archived by the industry partner. Figure 4.7 shows the archiving process. Here, the machine parameters are queried in a five-minute cycle and stored locally.

stored. Since this is a circular file and is overwritten, the data is stored in a network drive for final and permanent storage.

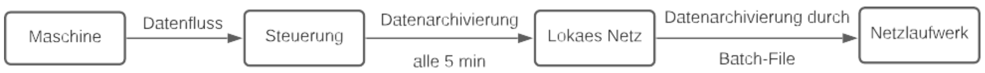


Figure 4.7: Flowchart of the currently available *data store*

However, the architecture found cannot be used for the project, as one only has access to data that is five minutes old.
are "old", but it needs access to "real-time data". Thus, a direct interface to the controller is planned. The data acquisition and
-collection is on the customer side, it is therefore his task to generate the interface and possibility on real-time data. Figure 4.8 shows a possible implementation.
Here, the Central Processing Unit (CPU) is accessed. The retrieved data is then stored in a database, to which the knowledge store also has access rights. It can query the current status of the machine and learn its networks with new "experiences". The assistance provided by the assistance system is displayed on a dashboard and the operator intervenes in the process. This description serves the customer as a possible concept or assistance. The institute is not currently implementing this concept.

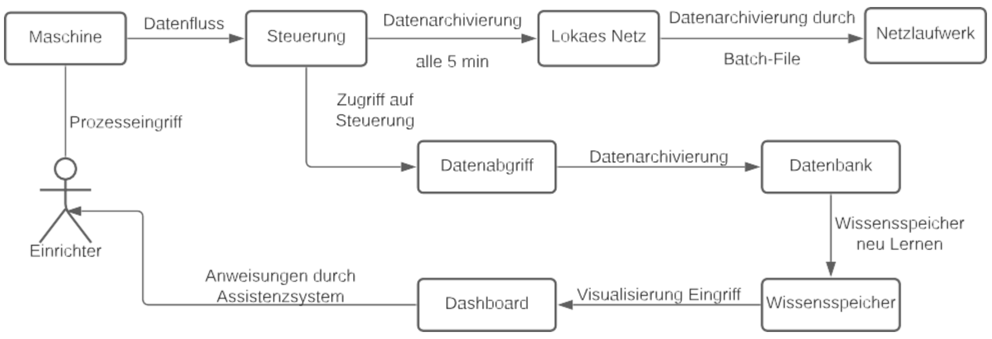
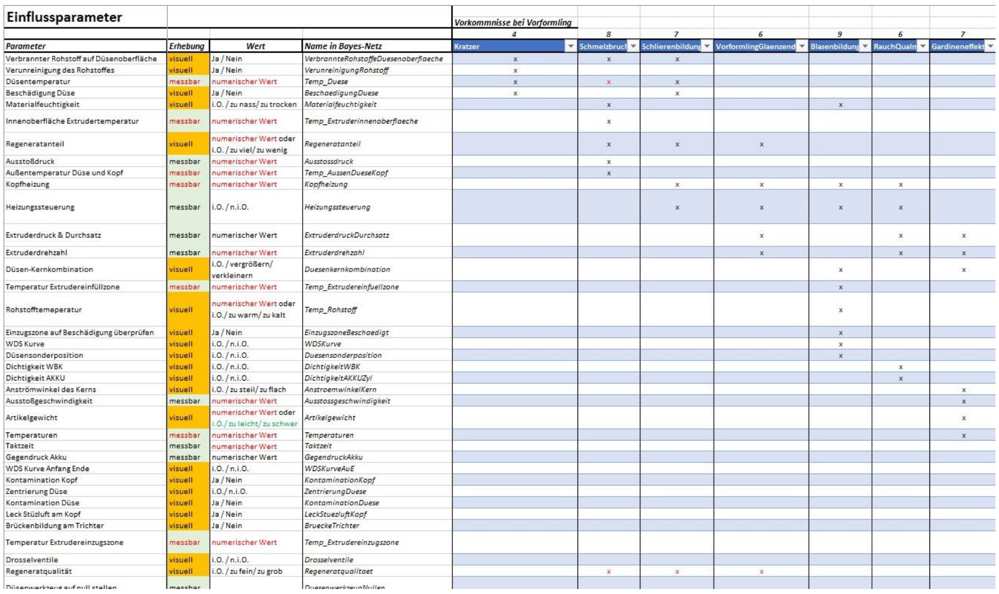


Figure 4.8: Flow chart of the planned data acquisition and processing

Together with the industry partner, the required data/parameters are identified and listed from the created sub-bayes networks for possible errors. This serves to define which data the knowledge store requires. Figure 4.9 shows an excerpt of the created list. In addition to the assignment to the respective error, the Bayes net name as well as the inclusion of the parameter type can be found there. In another column possible states of a node are defined. Since we work with discrete variables, many nodes have only two states. This is also in the sense of Bayes nets, because the number of possible combinations can be reduced. In the case of parameters that transmit a numerical value, considerations are made as to how to deal with them. One possibility would be a division into intervals around thus limit the number of possible states. However, this point is still open at the current time and is still being discussed.



Einflussparameter				Vorkommnisse bei Vorformling						
				4	8	7	6	9	6	7
Parameter	*Erhebung*	*Wert*	*Name in Bayes-Netz*	Kratzer	Schmelzbruc	Schlierenbildun	VorformlingGlaenzend	Blasenbildun	RauchQualn	Gardineneffekt
Verbrannter Rohstoff auf Düsenoberfläche	visuell	Ja / Nein	*VerbrannteRohstoffeDuesenoberflaeche*	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	*BeschaedigungDuese*	x		x				
Materialfeuchtigkeit	visuell	i.O. / zu nass/ zu trocken	*Materialfeuchtigkeit*		x			x		
Innenoberfläche Extrudertemperatur	messbar	numerischer Wert	*Temp_Extruderinnenoberflaeche*		x					
Regeneratanteil	visuell	numerischer Wert oder i.O. / zu viel/ zu wenig	*Regeneratanteil*		x	x	x			
Ausstoßdruck	messbar	numerischer Wert	*Ausstossdruck*		x					
Außentemperatur Düse und Kopf	messbar	numerischer Wert	*Temp_AussenDueseKopf*		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	*ExtruderdruckDurchsatz*				x		x	x
Extruderdrehzahl	messbar	numerischer Wert	*Extruderdrehzahl*				x		x	x
Düsen-Kernkombination	visuell	i.O. / vergrößern/ verkleinern	*Duesenkernkombination*					x		x
Temperatur Extrudereinfüllzone	messbar	numerischer Wert	*Temp_Extrudereinfuellzone*					x		
Rohstofftemeperatur	visuell	numerischer Wert oder i.O./ zu warm/ zu kalt	*Temp_Rohstoff*					x		
Einzugszone auf Beschädigung überprüfen	visuell	Ja / Nein	*EinzugszoneBeschaedigt*					x		
WDS Kurve	visuell	i.O. / n.i.O.	*WDSKurve*					x		
Düsensonderposition	visuell	i.O. / n.i.O.	*Duesensonderposition*					x		
Dichtigkeit WBK	visuell	i.O. / n.i.O.	*DichtigkeitWBK*						x	
Dichtigkeit AKKU	visuell	i.O. / n.i.O.	*DichtigkeitAKKUZyl*						x	
Anströmwinkel des Kerns	visuell	i.O. / zu steil/ zu flach	*AnstroemwinkelKern*							x
Ausstoßgeschwindigkeit	messbar	numerischer Wert	*Ausstossgeschwindigkeit*							x
Artikelgewicht	visuell	numerischer Wert oder i.O./ zu leicht/ zu schwer	*Artikelgewicht*							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.	*WDSKurveAuE*							
Kontamination Kopf	visuell	Ja / Nein	*KontaminationKopf*							
Zentrierung Düse	visuell	i.O./ n.i.O.	*ZentrierungDuese*							
Kontamination Düse	visuell	Ja / Nein	*KontaminationDuese*							
Leck Stüzluft am Kopf	visuell	Ja / Nein	*LeckStuezluftKopf*							
Brückenbildung am Trichter	visuell	Ja / Nein	*BrueckeTrichter*							
Temperatur Extrudereinzugszone	messbar	numerischer Wert	*Temp_Extrudereinzugszone*							
Drosselventile	visuell	i.O. / n.i.O.	*Drosselventile*							
Regeneratqualität	visuell	i.O. / zu fein/ zu grob	*Regeneratqualitaet*		x	x	x			
Düsenwerkzeug auf null stellen	messbar		*DuesenwerkzeugNullen*							

Figure 4.9: Section of the parameter list with assignment to possible errors

A record/data tuple should look like the following (Table 4.2):
It receives a unique *ID* and then all possible errors and parameters/cause of error are listed. A data record is considered to be complete and correct if all influencing parameter states are known for the examined fault. In the example of the scraper these are the four factors *Damage_throat, Thoroughfare_temperature, Burnt_raw_material_through_surface and Contamination_raw_material*, whose states must be known. All other parameter states do not have to be entered, but it has no negative effects if they are also included. Several defects can also be evaluated simultaneously in one data record. In this case, the influence parameters of each defect to be evaluated must be known.

ID	Scratch	bladder b.	Error. . . .	Duesentemp	Head temp	Para. . . .
1	Yes	No	-	190	205	. . .
2	No	-	-	200	-	. . .
3	No	Yes	-	200	180	. . .
4	-	Yes	-	-	200	. . .
5	Yes	-	-	185	-	. . .

Table 4.2: Concept frame *data format*, complete data tuple

Learning and training the Bayesian network and its errors requires an exact assignment between error and parameter state. Only in this way is it possible to predict which influencing parameter has the highest probability for the occurrence of an error.

4.5 user interface with shiny

This section covers the design and creation of a user interface. The programming language **R** has been developed for data processing and analysis. It offers only limited possibilities for programming a graphical user interface (GUI) for interaction between the assistance system and the human. It offers only limited possibilities to program a Graphical User Interface (GUI) for the interaction between the assistance system and the human.
While researching the creation of user interfaces, one came across *Shiny.* The **R-package** *Shiny* makes it possible to create web-based user interfaces. The website offers information, examples and tutorials. [2]

Figure 4.10 shows the architecture of Shiny. A user makes an "input" via the *user interface*. The input data is transmitted to the *server.* The server has access to data, algorithms and functions. Depending on the input, actions are executed and the results are sent back to the *GUI*. This presents the output to the user in a visual form.

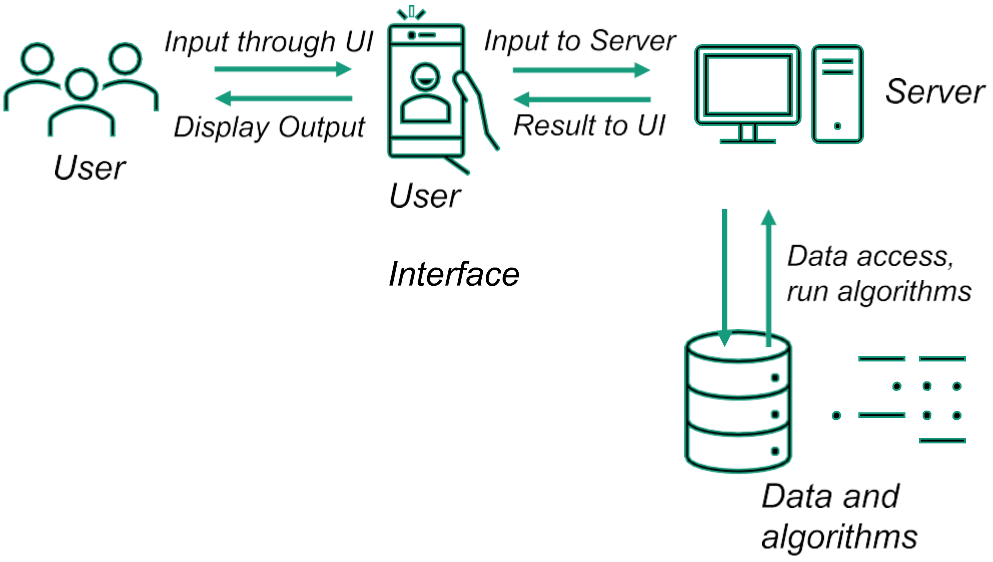


Figure 4.10: Architecture and Function *Shiny*

The task is to program the user interface on the one hand and the server side on the other. Listing 4.4 shows the general structure of the code of a Shiny web interface. The actual user interface is programmed in the *ui* area. You create the layout and the appearance. In the *server area,* functions and rules are implemented that are to be executed under certain inputs. If you want to display a button for starting a calculation on the user interface, then a button with a unique "ID" is defined in the *ui area* and the function for executing the calculation is stored in the *server area* for this "ID".
Both *ui* and *server* are passed to the *shinyApp* function in order to

2 **R-package** *shiny* website `https://shiny.rstudio.com/`

to start the app. It starts locally on the user's own computer. However, it is possible to open and operate the web interface from a browser window by specifying the IP address and the port, e.g. with the address *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: Program frame Shiny in **R**

The interface is designed and created *step-by-step.* It grows and develops continuously. From this thought the idea arose to work with tabs for different areas. Figure 4.11 shows the tabs created for the current state. There is one tab for the overall view of the Bayes net, one for the calculation and determination of the cause of the error, and another for uploading new data sets and training the net.

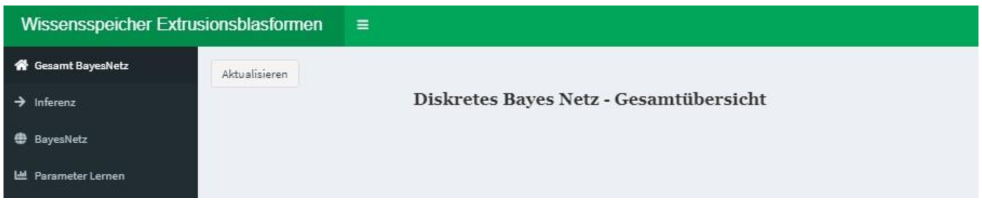

Wissensspeicher Extrusionsblasformen
≡
Gesamt BayesNetz
Inferenz
BayesNetz
Parameter Lernen
Aktualisieren
Diskretes Bayes Netz - Gesamtübersicht

Figure 4.11: Overview user interface, existing tabs

Shiny offers library functions, with which it is possible to create and design the graphical user interface. Different in- and output functions are provided. In the area *ui* a unique ID is defined for an input or output element. This ID makes it possible to change or query the values of an element in the *server area.* A list of the possibilities is shown below.

Input possibilities

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

Output possibilities

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

Listing 4.5 and Listing 4.6 show the program code for the tab "Bayes net" and "Error overview", respectively. As described in the above section, both *ui* and *server* need to be programmed.

```
## Tab BAYESNET
tabItem(tabName = " bayesnetz",
        fluidPage(
                column(width = 12,
                        h1("Display of individual Bayes nets"), ##
                        chose bayesian net
                        box(width = NULL,
                                selectInput(inputId = " bayesnet",
                                        label = "Choose a Bayes
                                            Net out."
                                        choices = c(get_available_nets()
                                            )),
                                helpText("Output Bayes net"),
                                actionButton(inputId = " idi_ab_bayes_
                                    refresh",
                                        label = "output",
                                        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: Program section *ui.R* Tab Bayes net

The user interface is constructed as follows. In the topmost position, a heading is inserted with the command *h1(<heading>)*. Then two "boxes" are created, which serve as spatial separation. In

In the upper box, a drop-down menu is inserted, via which a single error can be selected and the Bayes net associated with the error can be plotted via an "Output" button. The display of the Bayes net is implemented in the *server* area. There it *reacts reactive to the* event of the button operation for plotting the graph. A suitable function is called to read in the subnet and to plot the graph.

```
##### 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: Program excerpt *server.R* tab Bayes net

Figure 4.12 shows the interface as an operator would find it.

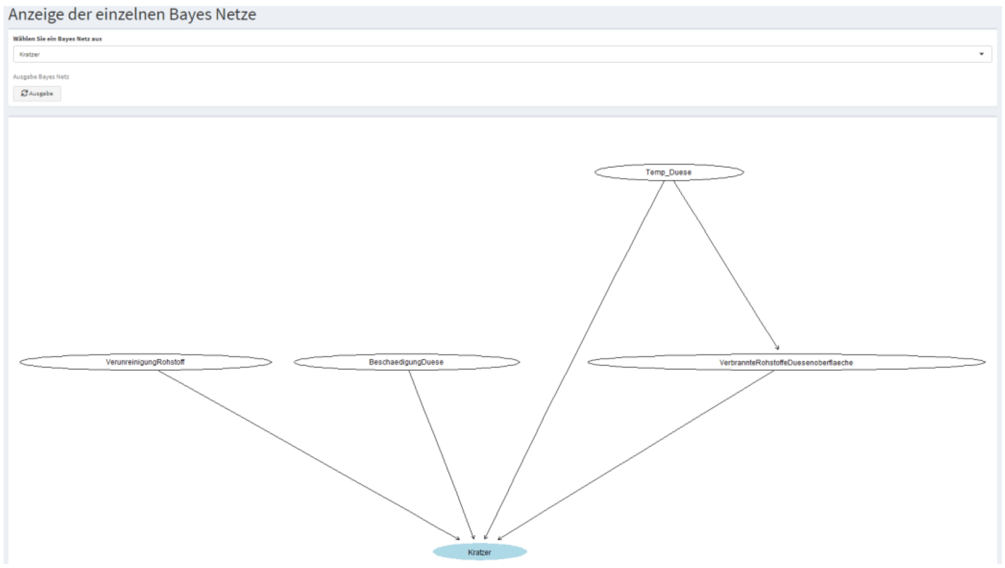

Anzeige der einzelnen Bayes Netze
Wählen Sie ein Bayes Netz aus
Kratzer
Ausgabe Bayes Netz
Ausgabe
Temp_Duese
VerunreinigungRohstoff
BeschaedigungDuese
VerbrannteRohstoffeDuesenoberflaeche
Kratzer

Figure 4.12: Tab "Bayes net" with displayed error *Scratcher*

The complete surface is shown in chapter 5 Result/ State at the end of the BPP.

5 S E R V I C E / F O R M A T I O N O F T H E B P P ' S

The project for the development of an interactive knowledge repository has a duration of 2 years, as described at the beginning. The status at the end of the practical phase therefore represents an interim status and not the final status of the overall project.

The Bayes net structure is in a provisionally completed state. Possible errors during the *preform* phase have been identified and error causes with dependencies defined. The individual structures must be checked and validated; this step is currently open. A specially developed process is available for combining the individual meshes into a large/overall mesh. Furthermore, an algorithm has been developed with which it is possible to predict the most probable error cause for a current error pattern.

In the next step, a concept has been designed that implements the learning and training of the Bayes net. Besides algorithms and functions a preliminary user interface has been created. The following figures show the currently designed user interface

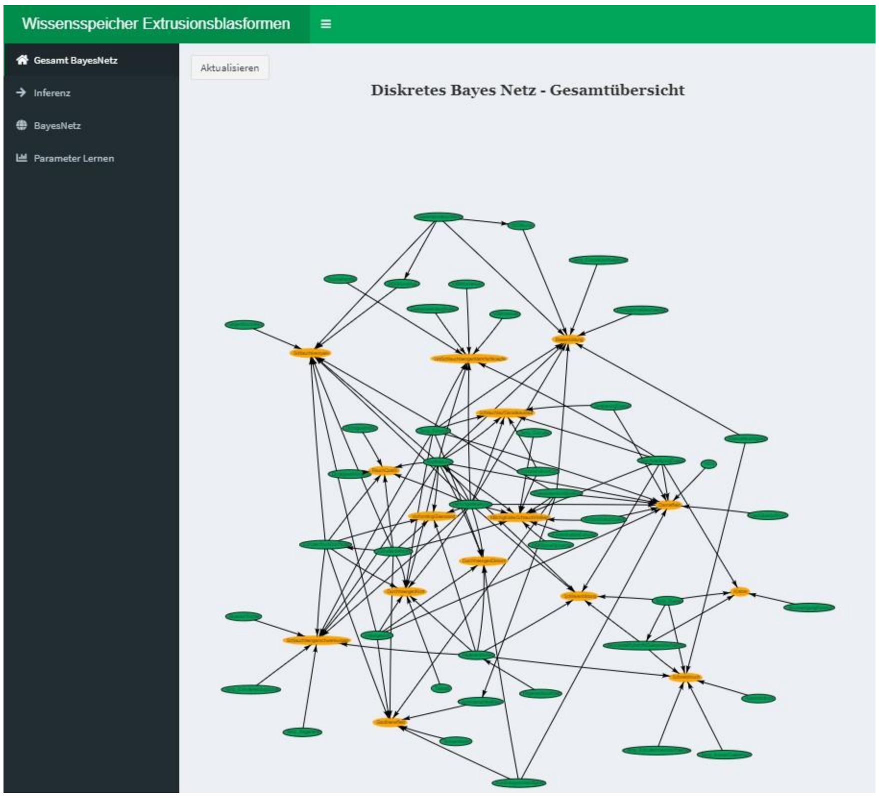
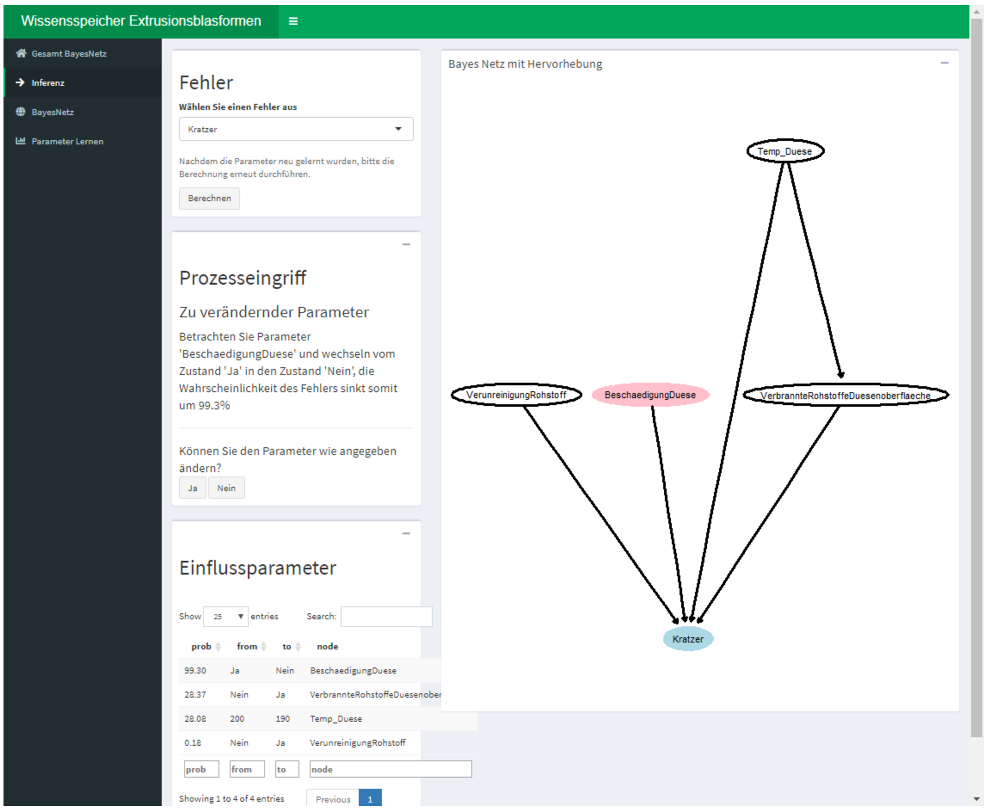

Wissensspeicher Extrusionsblasformen
Gesamt BayesNetz
Inferenz
BayesNetz
Parameter Lernen
Aktualisieren
Diskretes Bayes Netz - Gesamtübersicht

Figure 5.1: Tab "Total Bayes Net

Figure 5.1 shows the overview of the created main network. The user has the possibility to operate the graph interactively. In addition to zooming in and out and moving the network, it is possible to select individual nodes.
"grab" and thus move them. If you want to highlight dependencies to a fault, this can be done by clicking on the respective node. Those nodes which have an edge to the selected one remain unchanged in their color, others are grayed out and thus moved into the background.
The "Inference" tab is used to predict the cause of the error (see Figure 5.2).


Wissensspeicher Extrusionsblasformen

Gesamt BayesNetz
Inferenz
BayesNetz
Parameter Lernen

Fehler

Wählen Sie einen Fehler aus

Kratzer

Nachdem die Parameter neu gelernt wurden, bitte die Berechnung erneut durchführen.

Berechnen

Prozesseingriff

Zu verändernder Parameter

Betrachten Sie Parameter 'BeschaedigungDuese' und wechseln vom Zustand 'Ja' in den Zustand 'Nein', die Wahrscheinlichkeit des Fehlers sinkt somit um 99.3%

Können Sie den Parameter wie angegeben ändern?

Ja Nein

Einflussparameter

Show 25 entries Search:

prob	from	to	node
99.30	Ja	Nein	BeschaedigungDuese
28.37	Nein	Ja	VerbrannteRohstoffeDuesenober
28.08	200	190	Temp_Duese
0.18	Nein	Ja	VerunreinigungRohstoff

Showing 1 to 4 of 4 entries Previous 1

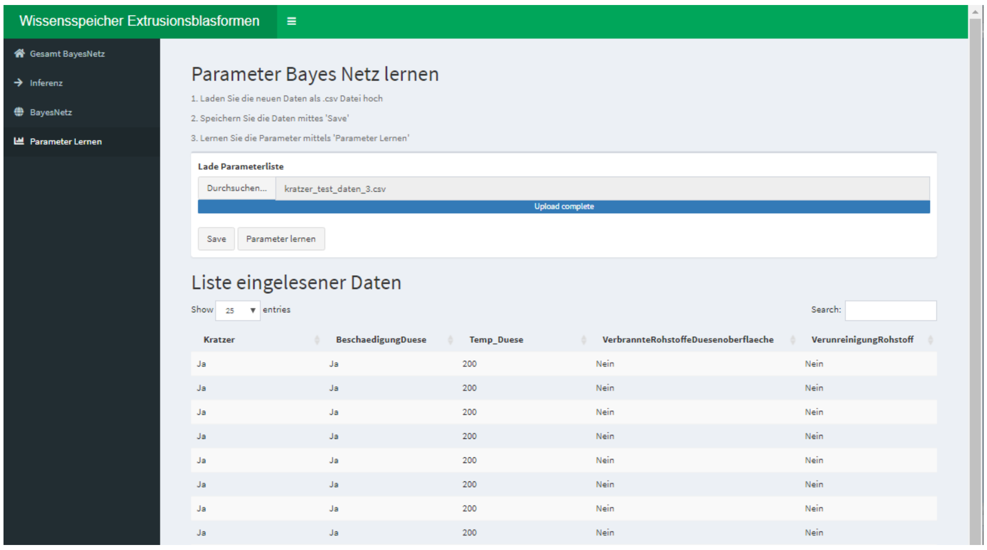
Bayes Netz mit Hervorhebung

Figure 5.2: Tab "Inference", Prediction of Error Cause for Error *Scratches*

The operator selects the pending fault pattern in the upper section and then presses the *Calculate* button. In the right section of the window, the Bayesian network of the selected fault is displayed. The node that was predicted as the most probable cause of the fault is highlighted in red. On the left side, the node to be changed and its value change to be made are displayed as text. Below there is the possibility to tell the assistance system whether the intervention is possible or not. Since the prediction of the knowledge store works without connection to the machine, there is the possibility that a value change is suggested, which cannot be implemented in reality, since the machine is already in this state. If it is possible to change the node as recommended, feedback must then be given as to whether the error has been eliminated or whether a further advance is necessary.

must be output. In the lower area, the setter receives an overview of the calculated predictions with probabilities and value changes.

Tab "Parameter learning" can be seen in Figure 5.3. There the operator has the possibility to read in new data sets and then to train the network further. If new data are available, which are to be added to the knowledge memory, proceed as follows.
The data set must have the specified structure and be available as a csv file. It is possible to select and upload a file via an Explorer window. Once the upload is complete, a table appears with the new data. The next step is to save this table and then to train the Bayes net with the new data set. The training has to be executed actively and is not started automatically. After the network has been trained, the operator is informed that the parameters have changed and the predictions must be recalculated for an error that has occurred.



Parameter Bayes Netz lernen

1. Laden Sie die neuen Daten als .csv Datei hoch

2. Speichern Sie die Daten mittes 'Save'

3. Lernen Sie die Parameter mittels 'Parameter Lernen'

Lade Parameterliste

kratzer_test_daten_3.csv

Liste eingelesener Daten

Kratzer	BeschaedigungDuese	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

Figure 5.3: "Parameter learning" tab

At the current time, no real production data are available. Synthetic data have been generated for the sub-network *Scratch,* but these are not close to reality. They were needed to test the training and prediction of the fault cause. These tests were performed using the *Kratzer* fault. On-site implementations or tests with real data have not taken place at this time. The customer/industry partner will begin data collection and acquisition in the near future.

6 ASSESSMENTANDOUTPUNCTS

In this chapter, open points and an outlook are addressed. Chapter 5 described that only defects for the *preform* phase were processed and structures were created. Here, the structure must be completed further, and the *finished product* phase and any errors that occur there must be processed.
The next step is to look at data collection, as this has not yet been collected. However, a certain number of data sets is needed to learn and train the Bayesian network. The institute must provide support in finding ways to obtain both categories of data. The measures for this must be carried out by the industry partner.

Due to the lack of data, no test runs could be carried out with the current status of the assistance system. These must be carried out as soon as possible, as soon as sufficient data has been collected. In this course, an on-site implementation as well as tests during the real setup process should also be carried out.

The knowledge store calculates the predictions and recommendations based on "offline data". The current machine state is unknown and can only be determined by the operator. Because of this, cases can occur in which the assistance system issues recommendations that are not compatible with the current machine state. Here it is necessary to establish a "live connection" to the machine and thus to make the predictions and recommendations *depending on the state.* In addition to querying the current status, it is possible to integrate data acquisition into the process. Since there are some error causes that the operator has to enter manually at the current time for data acquisition, it is worth discussing whether additional sensor technology can increase the degree of automation of data acquisition. In addition to increasing the degree of automation in data collection, the susceptibility to errors is reduced at the same time.
The last open point to mention is the user interface, which must be fully programmed in its function and appearance. Here, the previously shown drafts have been created as a possible solution.

Part II

A N H A N G

A

THEORIEANDVERTIFICATION

a.1 blow molding (*extrusion blow molding*)

The project description calls for a knowledge repository for an extrusion blow molding machine. No knowledge in the field of blow molding is available from the studies. With the help of an *in-house* presentation and a Bachelor's thesis on extrusion blow molding (processes and machines), as well as public literature, initial insights and know-how in the subject area are gained.

Blow moulding in general describes a family of processes for the production of plastic wood bodies. This report is restricted to ex- trusion blow moulding and only provides a brief overview of this subject area. In recent years, stretch blow moulding has come to the fore alongside extrusion blow moulding. These two processes have in common that a thermoplastic preform is inflated in a negative mold, the plastic solidifies by cooling and a solid hollow body can be removed. [13]

The extrusion blow molding process itself is again divided into different un- categorised according to the procedure used.

- Conventional extrusion blow moulding
- Conventional extrusion blow moulding with squeezing over
- Manipulation procedure
- Suction blow molding
- Insertion procedure

The following figure A.1 shows the different extrusion blow molding processes. In each case, the preform and the tool and the varying complexity of the possible hollow body shapes are shown.

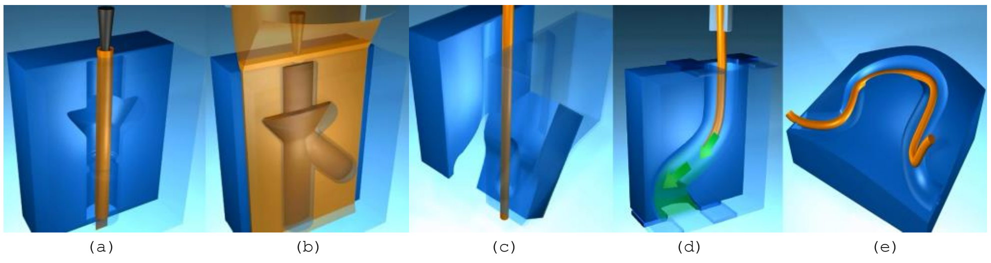


Figure A.1: Extrusion blow molding (a) conventional, (b) conventional with over- squeeze, (c) manipulation process, (d) suction blow molding, (e) Insertion procedure from [2] p.32ff.

In the project, a machine for extrusion blow molding with squeezing is considered, therefore this process is dealt with again individually in the following. However, in general the process is identical for all methods, differences are only contained in individual points. In addition to differences in the process, the methods show significant differences in resource consumption and also in the possibility of the complexity of the 3D hollow bodies.

General process of extrusion blow molding

1. Heat plastic in an extruder and plasticize.
2. Diverting the melt from a horizontal to a vertical movement and producing a tubular preform.
3. Close tool (two half shells) around preform and squeeze upper and lower ends.
4. Retract the blow pin/ blowing needle into the mould.
5. Inflate the preform and cool and solidify it on cooled walls of the mould.
6. Venting and opening of the tool halves.
7. Removal of the hollow body and removal of over-squeezing and slugs.

[13]

The following figure (A.2) shows in graphics the individual steps of extrusion blow molding. After the melt has been heated and plasticized, it is conveyed by the extruder via a screw into the head. A horizontal movement is then transformed into a vertical one. The thermoplastic is pressed out of the head through a nozzle and formed into a preform. If the preform is the right length, the mould halves are closed and the upper and lower ends are squeezed in. After the blow pin has been retracted, the preform is inflated and pressed against the inner wall of the mould until the thermoplastic solidifies. Finally, the mould is opened, the hollow plastic part is removed from the mould and any slugs are removed. In a final step it is possible, for example, to carry out a leak test.

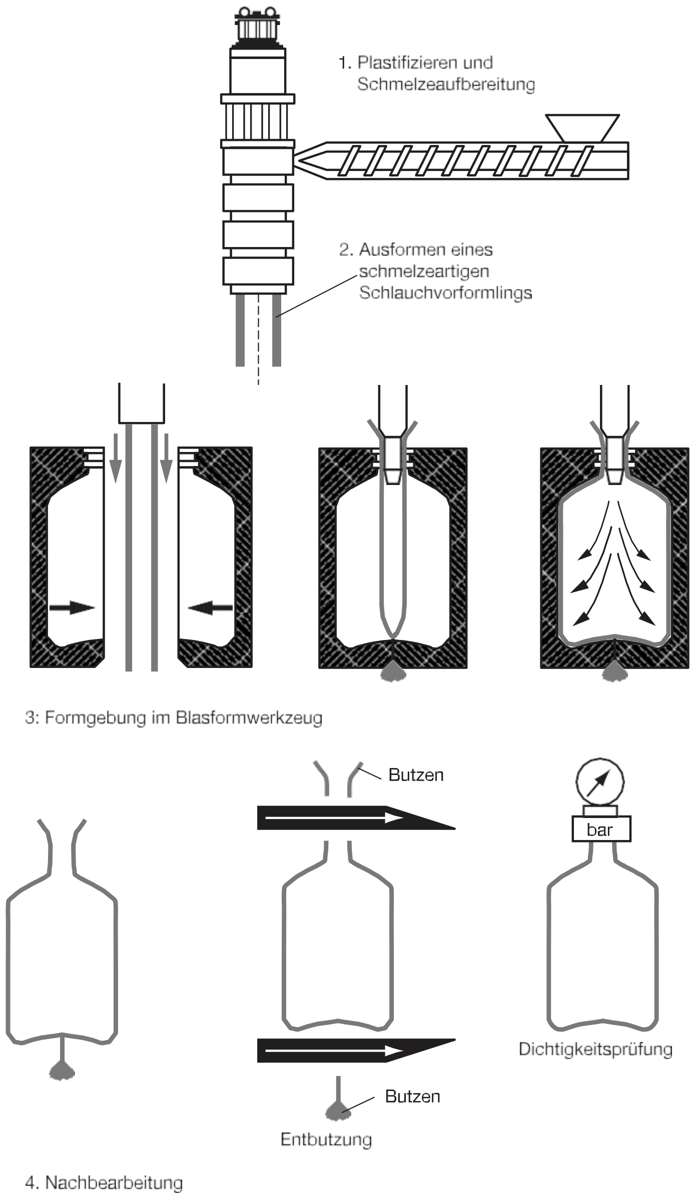

1. Plastifizieren und Schmelzeaufbereitung
2. Ausformen eines schmelzeartigen Schlauchvorformlings
3: Formgebung im Blasformwerkzeug
Butzen
bar
Dichtigkeitsprüfung
Butzen
Entbutzung
4. Nachbearbeitung

Figure A.2: Process plastic extrusion blow molding from [13] p.16

a.1.1 *Extrusion blow molding with squeeze over*

Conventional extrusion blow molding with squeezing is the simplest manufacturing process. Here, the material input is not resource efficient. As shown in figure A.1(b), a preform is required which has the width of the later hollow body. In addition to the high material consumption (large extruder required), high closing forces are needed for the squeezing process. At the points of over-squeezing, the wall thickness is at its maximum and is used for forming.

fe is minimal. After the hollow body has been formed, it has a completely circumferential pinch seam, which requires reworking as well as a weak point. The advantages of this are low moulding costs and short set-up and adjustment processes[9].

a.1.2 *Continuous/ discontinuous extrusion blow moulding*

A difference within a process that should not be neglected is the way in which the melt for the preform is ejected. A distinction is made between *continuous* and *discontinuous ejection* of the melt. In the case of continuous ejection, thermoplastic melt flows continuously out of the head and thus forms the tubular preform. The ejection speed is strongly dependent on the extruder speed. This method is mostly used for short and tough thermoplastics, which at the same time have a small size, thus an uncontrolled tube elongation or wall thickness reduction can be prevented. In the case of discontinuous ejection, the entire melt for a preform is collected in a reservoir and then conveyed outside in an impact-like manner. This process is used when a long or heavy parison is required or when a thermoplastic is involved which has only a low toughness.[2] The melt is ejected in a discontinuous process.

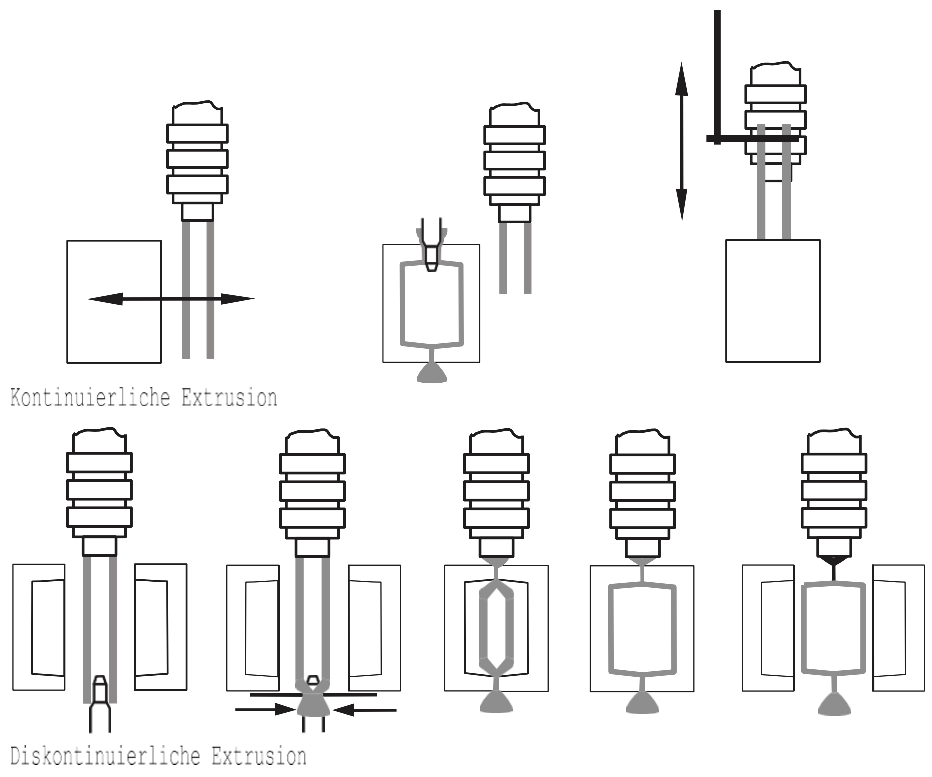


Figure A.3: Continuous and discontinuous extrusion according to [13] p.40 and p.42

a.2 bayes networks

Bayes nets are used for data analysis and the creation of probability models. Often there is a large amount of data and one wants to make predictions. Data is usually stored and saved in tabular form. A row describes an observation and a column the values of a specific attribute. A row has a value for each attribute. Bayes nets offer the possibility to link attributes, to establish dependencies and to predict the occurrence of events.[8] Bayes nets can be used to predict the occurrence of events.

The graphical representation of a Bayesian network uses the *directed acyclic graph (DAG)* method. Here, a finite set of *nodes are connected* by directed *edges* (arrows), hence the name *directed*. In addition to the directed edges, it is important that no loops/cycles are allowed to occur, hence *acyclic* see Figure A.4.[11]

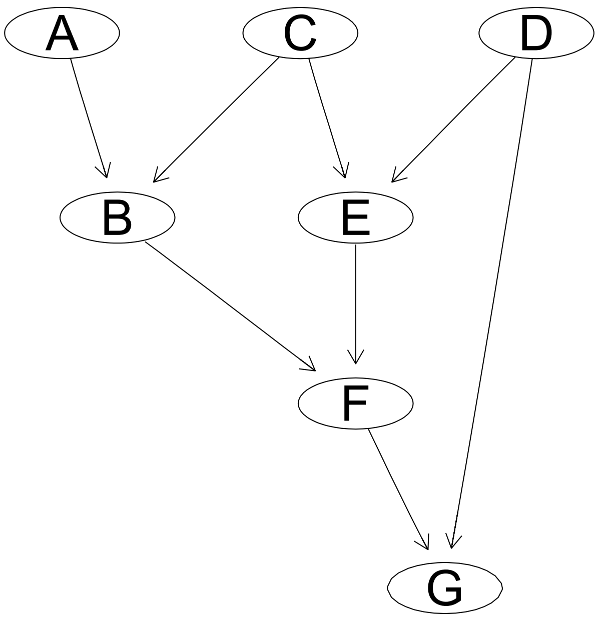


Figure A.4: Bayes net as *directed acyclic graph (DAG)*

Each node has at least two states/levels. A distribution is assigned to a state. The sum of all states of a node results in 1. If a node X has subsequent nodes, then these are called *children of X, e.g. children(A)=B*. The predecessors of a node X *are* called parents, e.g. *parents(B)=A,C*. By means of *inference* one can determine probabilities of an attribute under *given* conditions. Among other things, *Bayes'* theorem $P(B/A) = \frac{P(A|B)\cdot P(B)}{P(A)}$
[6]
In Figure A.4 it is possible to determine the probability for node F under given *A and C by* means of inference. Mathematically this is expressed as follows: $P(F/A,\ C)$. The explanation of the applied mathematics and the calculations of the Bayes nets will be explained in this report.

are not described in detail. If necessary, reference is made to the specified literature.

a.3 introduction to programming with r

In the mechatronics course, content is taught in programming with *C* and *C++.* Later, the programming language Python is introduced in the robotics specialisation. However, knowledge of programming with **R is** not covered.
Before starting with the later creation of a program code, a basic knowledge is learned with the literature of U. Ligges "Programmieren mit **R**" [10] and D. Obszel- ka "Statistisches Programmieren mit **R**" [1]. Both works offer examples for reprogramming or self-programming. The programming language **R** is an "open source" software and optimized for data analysis and graphics. Methods and technologies can be added by adding add-on packages (*packages*). [10]
R-Studio is used for programming. This program has a *source code window*, a *console*, a *variable overview* and a *navigation area,* where plots can also be output. **R-Studio** supports programming and gives the programmer suggestions for completing his code line.

The works are used to learn the syntax and the basic handling and use of the **R** language. In addition to small mathematical examples (calculating the *body mass index* or *solving a quadratic equation*), functions are defined and applied.
The following code Listing B.1 shows two ways to calculate the Fibonacci sequence. If in the first step the equation for the c a l c u l a t i o n i s completely written out for each "n", then in the second step a function is defined, which only receives the parameter "n" and returns the results. In the last section a comparison is performed, which returns a *TRUE* as result. Further examples will be omitted due to the scope.

a.4 graphical creation of bayes nets

a.4.1 *By means of BayesianFusion GeNIe*

The **R-package** *bnlearn* offers the possibility to build a Bayes net with "pure" code. This can be done faster, more intuitively and more easily with a graphical interface. *BayesFusion, LLC1* offers the possibility to build a Bayes net graphically with their software *GeNIe.* The software is available for

1 Homepage *BayesianFusion*
https://www.bayesfusion.com/?gclid=CjwKCAjwzt6LBhBeEiwAbPGOgXh_nTLNoyEVWKk9IC7O7_c_mFh4tSVxXYlqVAx8Fq_vn73Ya56EfhoC32MQAvD_BwE

available for the common operating systems and does not require any special resources.

After starting the software and opening a new network, you can drag and drop the individual components for the later Bayes network into the workspace.
After the required nodes have been inserted and dependencies have been created by arrows/ edges, the properties of the individual nodes are adjusted. States and probability distributions are entered. Finally, the Bayes net is calculated and the distributions are output. By changing the layout the distributions are displayed in a bar chart at each node, so it is possible to get a quick overview. Figure A.5 shows a completed Bayes net using GeNIe. The distributions entered are based on *fictitious and invented* values. The example was created for the industry partner in order to explain the topic of Bayes nets and their function.

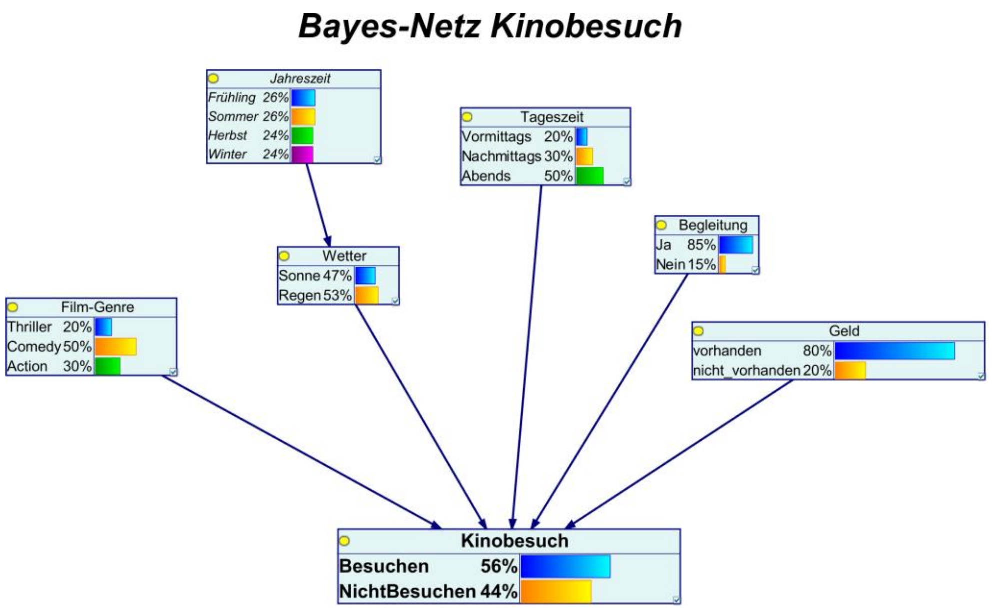
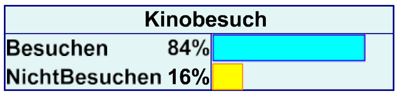


Figure A.5: Bayesian network cinema visit

The example network (Figure A.5) examines which parameters have an influence on the decision whether to visit the cinema or not. The *film genre*, *weather*, *time of day*, *company* and *money are* listed as influencing parameters, whereby the weather is dependent on the *time of year.* No dependencies are created among each other, so the Bayes net remains clear. Basically, when creating a Bayes net, only add dependencies that are relevant and important. Dependencies that cannot occur are removed. With the given probabilities it results that a visit to the cinema takes place to 56%.

GeNIe offers the possibility to specify states of a node and thus to determine a visit to the cinema for given conditions. If you specify that a *thriller is playing*, it is *raining* and the film is playing in the *evening,* the probability of going to the cinema increases to 84% (Figure A.6).



Kinobesuch		
Besuchen	84%	🟦
NichtBesuchen	16%	🟨

Figure A.6: Bayes net cinema visit, condition *thriller, rain and evening*

In addition to creating a Bayesian network, the software also offers the possibility to read in data and determine parameters. Thus, the distributions for the nodes could be determined from a survey based on the data. This is called "parameter learning". If you save the created Bayes net in the format "Hugin File .net", you can load it with *bnlearn* into **R** and use it further.

However, since this software is a paid pro-
an alternative was sought. The alternative is explained in the following section. There will be no further in-depth discussion of the *GeNIe* software.

a.4.2 *By means of SamIam*

SamIam[2] stands for *Sensitivity Analysis, Modeling, Inference and More*. It is an "open source" software of the *Automated Reasoning Group, University of California*.

Since it is a free product and also offers the possibility to create Bayesian networks graphically, the focus in the following is placed on *SamIam.* Figure A.7 shows the started software. The operation is comparable to that of the GeNIe software.

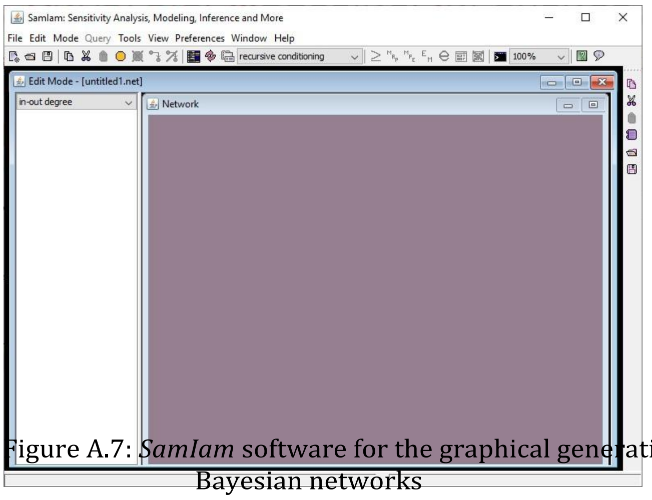

SamIam: Sensitivity Analysis, Modeling, Inference and More
File Edit Mode Query Tools View Preferences Window Help
recursive conditioning
100%
Edit Mode - [untitled1.net]
in-out degree
Network

Figure A.7: *SamIam* software for the graphical generation of Bayesian networks

Drop" to create individual nodes and arrows/edges for dependencies.

In the properties of a node one creates the states with their respective distributions.

2 http://reasoning.cs.ucla.edu/samiam/

Figure A.8 shows another example Bayes net, in this case created with *SamIam. This is* an example from *bnlearn*. Later on, this example is used to test the linkage between *SamIam* and **R. The example examines the influence parameters of a prominent brine x-ray.** The example examines the influence parameters on a *conspicuous chest x-ray* and *respiratory distress*. Nodes that can be directly influenced/modified are the "Visit to Asia" node and the "Smoking" node. *Tuberculosis, bronchitis and lung cancer* are consequences of the two previous nodes. The leaf nodes are a *conspicuous chest x-ray* (chest_X_ray) and *shortness of breath* (dysnoea).

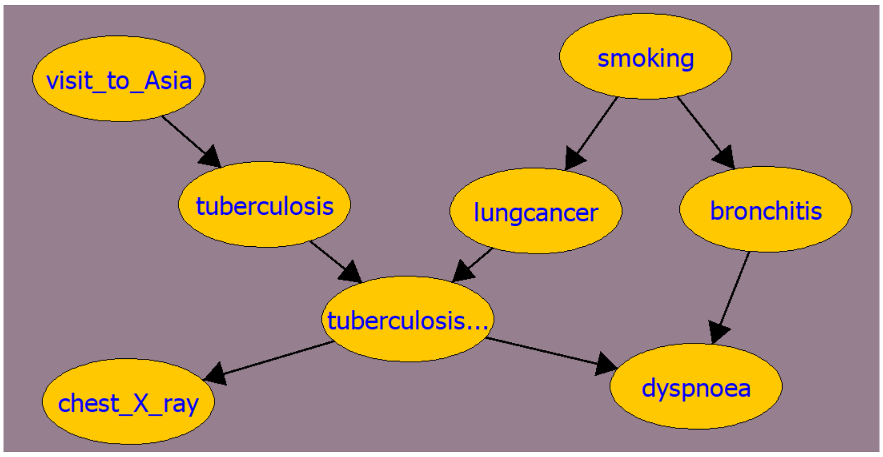


Figure A.8: Bayesian network "Asia" created with *SamIam*

It is also possible to display the probabilities of a conspicuous chest X-ray for the conditions present.
The visual representation is somewhat more pragmatic and simpler compared to *GeNIe.* However, the functions and representations are sufficient for the project.

After learning how to use *SamIam* to create a structure of a Bayes net, the next step is to look at and test the link between *SamIam* and *bnlearn.*

a.5 link samiam and r

The idea for the project is that the structure of the Bayes net or several Bayes nets are created graphically with *SamIam* and saved as a *.net (Hugin Net File).* The format used makes it possible to read in and further process a Bayes net using *bnlearn.*
In this section, an example from the *bnlearn* documentation is used, since data is available for it and thus a real case can be simulated.

At the beginning, the Bayes net is read in and stored as a variable. To check the structure, the Bayes net is displayed graphically in the next step s. Figure A. 9. Figure A.9. For a simplified overview and reduced typing, the nodes are represented with abbreviations. Figure A.8 shows the structure in plain names.

To learn the parameters, one needs a Bayes net that is not yet complete (structure sufficient). Listing B.2 shows the program code for the processing. In the first step, the formal equation of the structure is extracted from the read Bayes net. In the next step, this is used to create a pure structure (dag). The package *bnlearn* provides the data for the example *Asia*. These are read in and divided into two parts. The first part is used for learning the parameters, whereas the second part is used for testing. Table A.1 shows an excerpt of the data. The data set contains a total of 5000 surveys.

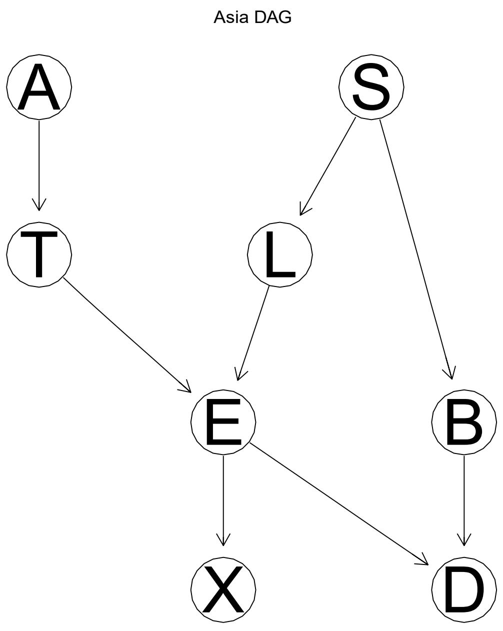


Figure A.9: Structure (DAG) Example *Asia*

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

Table A.1: Extract of data *Asia*

The parameters for the Bayesian network are learned from the training data. In this step, each node receives its possible states with probability distribution. The class of the resulting variable is now "bn.fit". The command *coef(bn.asia) outputs* the distributions of each node. It is also possible to look at the probabilities for a node, for this one passes as parameter not the whole net but only the desired node. Table A.2 shows the distribution for a conspicuous *chest X-ray (X)* given its direct dependence on *tuberculosis* or *lung cancer or bronchitis (E).*

X \ E	Yes	No
Yes	0.994	0.046
No	0.006	0.954

Table A.2: Probability distribution for a conspicuous chest X-ray Example *Asia*

After the parameters are learned, the Bayes net is complete and ready for "use". Besides calculations of inferences for specific cases, it is also possible to predict a node in its value.
The read-in data is divided into two parts. The test data is used in the following to predict whether the chest X-ray shows abnormalities or not. When comparing with the original data, one can determine the certainty of a correct prediction. It is predicted for 2500 further observations by the network whether the chest x-ray is *positive* or *negative.* The comparison with the original data shows that only in 94 cases a wrong prediction is made. In 96.24% the Bayes net can predict the correct result. The value can be improved by using more number of data for learning. Ana- log for introduction to *bnlearn* inferences can be computed.

B R-CODE

b.1 fibonacci sequence

```
# Fibonacci sequence
(an)n>=1 # general
formula
fibonaccifolge <- function(n){
                a <- (1 / sqrt(5)) * (((1+sqrt(5))/2)^n - ((1-
                    sqrt(5))/ 2)^n)

                return(a)
}

# different sequences 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)

# output of the
sequence a_man <-
c(a8,a9,a10) a_man

# Call over function n
= c(8,9,10)
a_aut <- fibonaccifolge(n)
a_aut

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

Listing B.1: Fibonacci sequence programmed in **R**

b.2 read in and learn parameters bayes net example 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: Reading and Editing Bayes Mesh in R, Example *Asia*

b.3 algorithm for merging individual sub-bayes networks

```
## 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("scratch", "fusion_break"," streak_formation", "smoke_quall","
    bubble_formation"," curtain_effect")
bayesnets <- readIn_multiNet(bn_names)
bigstring <- buildBNstring(bayesnets)
modelstring <- collaps_modelstring(bigstring)
dag <- modelstring_to_dag(modelstring)
```

Listing B.3: Reading several subnets into R and merging them into one

LITERATURE

[1] Daniel Obszelka and Andreas Baierl. *Statistical Programming with R. A detailed, clear, exciting, and field-tested introduction*. Ed. by Sybille Thelen. 1st ed. Wiesbaden, Germany: Springer Fach- medien Wiesbaden GmbH, 2020.

[2] M. Erfert. *Fundamentals of manufacturing processes with system analysis of the ma- schine and process*. Presentation, internal and trustworthy. 2017.

[3] Fraunhofer. *Fraunhofer-Gesellschaft zur Förderung der angewandten Forschung e.V.* 2021. url: https://www.fraunhofer.de/.

[4] Fraunhofer IWU. *Fraunhofer Institute for Machine Tools and Forming Technology IWU*. 2021. url: https://www.iwu.fraunhofer.de/.

[5] Fraunhofer Institute IWU and industrial partners. *Development of a knowledge-based system for the parameterization of plastic blow molding machines*. Project description, internal document, trustworthy. 2021.

[6] Uffe B. Kjærulff and Anders L. Madsen. *Bayesian Networks and Influ- ence Diagrams: A Guide to Construction and Analysis*. 2nd ed. New York: Springer Science+Business Media New York, 2013.

[7] MBK Maschinenbau Koetke. *Factors influencing blow* molding *processes.*

[8] Rudolf Kruse, Christian Borgelt, Christian Braune, Frank Klawonn, Christian Moewes and Matthias Steinbrecher. *Computational Intelligence A Methodological Introduction to Artificial Neural Networks, Evolutionary Algorithms, Fuzzy Systems and Bayesian Networks*. Edited by Prof. Dr. Wolf- gang Bibel, Prof. Dr. Rudolf Kruse and Prof. Dr. Bernhard Nebel. 2nd ed. Wiesbaden, Germany: Springer Fachmedien Wiesbaden GmbH, 2015.

[9] D. Kötke. *Technical guide for extrusion blow molding machines*. Bache- lor work, internal document, trustworthy. 2015.

[10] Professor Dr. Uwe Ligges. *Programming with R. Statistics and their Evaluation*. 3rd ed. Berlin und Heidelberg, Germany: Springer-Verlag Ber- lin Heidelberg, 2008.

[11] Marco Scutari. *Understanding Bayesian Networks with Examples in R*. Department of Statistics University of Oxford.

[12] Marco Scutari and Jean-Baptiste Denis. *Bayesian Networks With Examples in R*. 2nd ed. New York: CRC Press Taylor & Francis Group, 2021.

[13] Dr.-Ing. Michael Thielen, Prof. Dr.-Ing. Peter Gust and Dr.-Ing. Klaus Hartwig. *Blow moulding of hollow plastic parts*. Ed. by Jörg Strohbach. 2nd ed. Munich, Germany: Carl Hanser Verlag GmbH & Co. KG, 2020.