Root Cause Analysis of Product Defects in Manufacturing Using Ensembled Bayesian Networks

Karen Wang

A thesis submitted in fulfilment of the requirements for the degree of

Master of Engineering in Mechanical and Mechatronics Engineering

March 2022

Department of Mechanical and Mechatronics Engineering

Faculty of Engineering

The University of Auckland

Abstract

Root Cause Analysis (RCA) of product defects is crucial to improving manufacturing quality and productivity. Nowadays, manufacturers tend to rely on on-site expert knowledge to identify the root cause of product failure. However, manual RCA is extremely difficult and cumbersome, especially in big data environments yielded by the advancement of information technology and sensor technology. While different model-based methods have been introduced in the literature to localise root causes in a data-driven and automated manner, most of them are prone to various limitations in the aspect of robustness, causality discovery, knowledge representation, stochasticity, and sample size. Therefore, we proposed a product-wise framework of the ensembled Bayesian Network (BN) approach to provide a robust, intelligent and human-interpretable probabilistic reasoning method for RCA to circumvent the issues in the existing techniques. BN is adopted to enable interpretable probabilistic reasoning under uncertainty, which provides reliable decision support for RCA in industrial practice. We developed various structure learning algorithms, a parameter learning algorithm and a Bayesian inference algorithm for BN to learn the root causes of product quality issues from historical product defect records. The Ensemble Learning (EL) techniques enhance BN base learners with bootstrapped re-sampling and combine the predictions from multiple structure learning algorithms, ensuring a robust performance of BN. The structure of the framework is modularised by different products to reduce the sample size and to realise high efficiency. As a result, the proposed method can uncover the causal relationship in the industrial data to support manufacturers to make data-driven decisions under the circumstances of product quality failures. To achieve such goals, this project has automatically acquired causal knowledge, identified the root cause with probabilities and predicted quality risks in production. The proposed method has been implemented on real-world data collected from the plastic industry. Experimental results have shown that the ensembled BN framework successfully discovers the root cause along with corresponding probabilities and predicts the poor-quality instances with considerable robustness and accuracy.

Acknowledgements

Firstly, I would like to thank my supervisor, Dr Yuqian Lu, for his guidance, insights and endless support. Without his encouragement and advice, this research would not be possible. He has inspired me to be proactive, work smartly and adopt good working habits. These are invaluable assets for future career and personal development.

I would also like to give thanks to my mentor at Aspect PT, Bob Dedekind, who has imparted comprehensive domain knowledge about real-world manufacturing to me. I am also grateful for his support in scheduling client meetings for the project to bring in more practical insights.

I want to express my appreciation to Chris Rauch from Aspect PT. He has helped me with extracting the required database from AspectPL multiple times. And he is extremely reliable whenever there is a technology problem.

A special acknowledgement to Zengkun Liu, who has provided insights for me to enhance the project and encouraged me to work harder.

I am grateful for my family. My parents, Qingwen Wu and Ke Zhang have given me unconditional love, care, and support throughout my master's research. My brother, Kevin, has also provided me with accountability to stay focused.

Finally, I want to thank all my friends, my colleagues, and the people around me. Your presence and company have driven me to accomplish this project directly and indirectly. Sincere appreciation from the bottom of my heart.

ACRONYMS

ANN	Artificial Neural Network
AUC	Area Under the ROC Curve
BN	Bayesian Network
BIC	Bayesian Information Criterion
BP	Belief Propagation
CED	Cause-Effect Diagram
CMI	Conditional Mutual Information
CPT	Conditional Probability Table
CRT	Current Reality Tree
DAG	Directed Acyclic Graph
DPCA	Dynamic Principal Component Analysis
DPLS	Discriminant Partial Least Squares
EL	Ensemble Learning

FDA	Fisher Discriminant Analysis
FDC-CNN	Fault Detection Classification Convolutional Neural Network
FMEA	Failure Modes and Effects Analysis
FPR	False Positive Rate
FTA	Fault Tree Analysis
GS	Grow-Shrink
HC	Hill-Climbing
IAMB	Incremental Association Markov Blanket
JT	Junction Tree
KL	Kruskal's
MAE	Mean Absolute Error
MES	Manufacturing Execution Systems
MMHC	Max-Min Hill-Climbing
MMPC	Max-Min Parents and Children
MWST	Maximum Weight Spanning Tree

NN	Neural Network
PCA	Principal Component Analysis
PC.stable	Peter and Clark.stable
PLC	Programmable Logic Controller
PLS	Partial Least Squares
RCA	Root Cause Analysis
RMSE	Root Mean Square Error
ROC	Receiver Operating Characteristics
SVM	Support Vector Machine
TS	Tabu Search
WAEL	Weighted Average Ensemble Learning

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

With the surging demand for consumable products, manufacturers urge to increase operational performance to maintain competency. However, product quality failures have been a common problem in manufacturing that can hinder production efficiency dramatically. Root Cause Analysis (RCA) provides a systematic way of identifying the product defect reasons. Concurrently, manufacturers heavily rely on humans to perform RCA, making it a cumbersome task. A variety of scientific methods for RCA have been proposed in the literature. However, they are faced with distinct limitations in industrial practice such as the lack of model robustness, causality discovery, human-understandable knowledge visualisation and stochasticity explanation. Therefore, it is important to develop a robust, intelligent, and human-interpretable probability reasoning method for RCA in manufacturing.

1.1 Product Quality Problem in Manufacturing

The manufacturing sector has been under prominent pressure to elevate operational performance in the growingly competitive market. Product quality is one of the key factors to drive manufacturing success [1]. However, in the global marketplace, the presence of product defects remains to be the main obstacle to achieving production excellence. RCA was therefore used to identify causes of product defects and provide insights for improving production control schemes [2]. It is the process of discovering the causal mechanism of the underlying transition from desirable to undesirable conditions and identifying the root cause of the problem using a structured procedure [3]. Consequently, RCA provides guidance on how to handle product failures and prevent them from recurring in the future.

1.2 RCA Challenges

However, RCA is a difficult and time-consuming engineering problem [2]. Conventional approaches [4], such as Pareto Chart, Cause-Effect Diagram (CED), the Current Reality Tree (CRT), Failure Modes and Effects Analysis (FMEA), and Fault Tree Analysis (FTA), use domain knowledge from on-site experts to recognise product defect root causes [5]. Their reliance on domain knowledge leads to a series of limitations. First, the valuable knowledge is centralised in the mind of some key persons at the factory and cannot be conveniently transferred between the site workers or be accessed in the future [5]. This could leave the factory vulnerable in the absence of key employees. Second, conventional RCA methods require an in-depth understanding of the production system which is time-consuming to be established by manpower [6]. Moreover, human experts could be biased in deriving the causality of a problem, and so inaccurate in their conclusions [5]. Lastly, the development of advanced sensors and information technology leads to a huge volume of high-dimensional data being produced in the production process [7]. The occurrence of big data makes it even harder for humans to comprehend using existing RCA tools [8][9]. Thus, it raises a pressing need for a more robust, intelligent and human-interpretable RCA method in the modern industry.

1.3 Gaps in Existing Scientific Methods

In response to the challenges in traditional RCA methods, scholars have been developing more scientific and data-driven RCA techniques. A popular category of tools for RCA is data-driven multivariate statistical procedures [10]. Studies have shown their capability to isolate the root cause of failed processes in high-dimensional data space with a relatively short sample time [11][12]. Nevertheless, the statistical methods alone are insufficient for an efficient and fully automated RCA. They normally require integration with other techniques such as classifiers [9] and multivariate statistical control charts [13][14]. Furthermore, they are sensitive to sample size [9], and incompetent to represent nonlinear behaviour in the

data [12], making them impractical in industrial cases. On the other hand, machine learning methods [15] have shown higher accuracy than conventional and statistical methods [16][17][18]. Nonetheless, most machine learning algorithms focus on discovering the correlation between variables rather than causality [6]. Their lack of probabilistic properties can overlook the stochasticity involved in RCA. On the contrary, Bayesian Network (BN) stands out with its probabilistic reasoning ability to discover the root cause under uncertainty [6][19] and to support decision-making [20]. Its graphical nature of BN enables interpretable knowledge representation in practice [21]. The aforementioned characteristics make BN an exceptionally strong candidate for RCA. Conversely, implementation of BN could be computational expensive [19]. BN model can also be unrobust as its accuracy might drop evidently with big and sparse datasets [22]. Additionally, its performance tends to vary with the selection of BN structure learning algorithms [23]. However, it has been found that the robustness of the models can be improved by bootstrap sampling [24] and fusing different learning algorithms [25]. Fusing multiple learning algorithms using ensemble learning technique could provide more robust RCA results, inspired by successes of ensemble learning in other technological domains [26][27]. Therefore, this research is going to address the limitations of BN using ensemble learning techniques and a product-wise framework.

In summary, most of the existing methods for RCA lack the capability of causality discovery, understandable knowledge representation and explaining stochasticity in a real-world context. BN is able to provide visualised probabilistic reasoning; however, it is prone to the limitations of stable prediction accuracy, computational efficiency and model robustness. As a result, there is an urgent need for a robust, accurate and interpretable probabilistic reasoning method for RCA.

1.4 Objectives and Proposed Methods

Therefore, we aim to mine the causal relationships from historical production data for a real-world industry to provide robust, accurate and human-interpretable probabilistic reasoning

for RCA. In this way, the acquired insights can help manufacturers to identify the root causes of product defects and mitigate the risks from occurring in the future. To achieve this, we proposed a product-wise framework of ensembled BN where BN is adopted as the fundamental method for RCA to accommodate the stochastic nature of manufacturing process variations [19] and to predict the likelihood of the potential product defect root causes. Ensemble learning techniques are integrated to address the lack of robustness in single BN learners. The framework is modularised by product type to reduce the size of BN, increasing computational efficiency.

Specifically, we work with a plastic manufacturer ABC, to develop an RCA algorithm for establishing causal relations between product quality and production attributes such as operation machines, raw material used and production speed. Historical quality records of every production batch are recorded, together with the machines, raw material, operator and production parameters. Domain experts believe that although the real root cause of product quality may be deeply affected by the real microfabrication process, there may be causal influences between production attributes and product quality. Therefore, our objective is to develop a method to discover causality in product quality from these gathered production records in a robust, intelligent and human-interpretable manner.

1.5 Contributions

We made the following knowledge contributions in this project:

1. Developed an interpretable, data-driven and probabilistic reasoning solution for RCA using BN, allowing manufacturers to engage the causal knowledge visually and efficiently. The probabilistic nature of the results is exceedingly powerful for identifying the root causes of product quality issues.

2. Incorporated ensemble learning methods to address the robustness issues in existing BN models. By aggregating multiple BNs learnt from bootstrapped samples and

combing different structure learning algorithms for BN, the robustness of BN has been increased in identifying root causes.

3. Compared the performance among different structure learning algorithms and different knowledge sources for BN. In our findings, tabu search algorithm exhibits the best performance for both probabilistic root cause reasoning and quality risk prediction. Hybrid knowledge source shows an advantage in inferring root cause probabilities. The results of the comparison offer a direction for model strengthening as well as model selection.

4. An evaluation method for assessing the results of probabilistic reasoning has been designed. It provides a way to quantify the difference between sequences of probabilities. The developed method can measure the deviation between the observation and prediction both in magnitude and in ranking.

1.6 Thesis Outline

The remainder of this thesis is organised as follows. Chapter 2 introduces the background of RCA and presents a review of the work related to the existing RCA methods. In Chapter 3, the problem of this project is defined. The RCA problem is formulated mathematically. The proposed RCA framework is also described. Chapter 4 explains the data preparation process, followed by the demonstration of prediction model construction. A case study is provided to illustrate the implementation of ensembled BN. Chapter 5 shows the evaluation methods and discusses the results of the predicted root-cause probabilities and quality risk prediction. Finally, the achievements of this research and recommendations for future work are concluded in Chapter 6.

Chapter 2

Literature Review

In this chapter, the background of RCA will be introduced. The literature work related to RCA will be reviewed. Then the research gaps identified in the literature review will be summarised. Section 2.1 discusses RCA and its application in real-world manufacturing practice. Section 2.2 elaborates on the existing methods for RCA in the literature. Section 2.3 summarises the gaps in the existing methods.

2.1 Background of RCA

Root cause analysis of product defects is the process of investigating the causal factors that lead to quality deviations [5]. The purpose of RCA is threefold: i) to identify the root cause of a problem, ii) to learn and understand the underlying mechanics of the issue, and iii) to identify appropriate corrective action to systematically rectify the situation. Root cause analysis can be performed with a collection of principles, techniques, and methodologies. Nowadays, manufacturers still perform RCA manually [28] using generic methods, such as Pareto Chart, Cause-Effect Diagram, the Current Reality Tree, Failure Modes and Effects Analysis, and Fault Tree Analysis [29]. However, manual RCA is very restricted in knowledge communication and extraction. Moreover, it is time-consuming to localise the causes manually [4]. Experienced technical experts are needed for RCA, whereas human judgement can be biased and inadequate, resulting in an unsound analysis [5]. In addition, the evolution of information technology has stimulated the growth of big data, making it even harder for human beings to comprehend the rich data to perform RCA [7]. According to Rokach and Hutter [2], the process of localising the root cause is an extremely challenging engineering task, especially in large-scale systems.

2.2 Existing Methods for RCA

This section summarises the scientific methods for RCA that have been proposed in the literature. The researched methods have been categorised into two groups – statistical techniques (Section 2.2.1) and machine learning methods (Section 2.2.2).

2.2.1 Statistical Techniques

Statistical approaches exploit the statistical features in the data to assist the RCA process, including Principal Component Analysis (PCA) [30], Partial Least Squares (PLS) [30], Fisher Discriminant Analysis (FDA) [30], Dynamic Principal Component Analysis (DPCA) with minimax distance classifier [9], Discriminant Partial Least Squares (DPLS) [11]. Among these, FDA has the best performance, followed by DPLS and PCA [30]. Nevertheless, integration of statistical methods with other classifiers [9] or multivariate statistical control charts [13][14] or feature selection techniques [12] is indispensable to empower their ability to identify the root cause. In other words, the statistical methods alone are not sufficient to perform a sophisticated RCA. Moreover, as the data size decreases, their performance deteriorates [9]. A statistical method, fusing Dynamic Principal Component Analysis (DPCA) and minimax distance classifier, was implemented to simultaneously monitor and diagnose an automatically controlled process [9]. It has proven a decent success rate, however, its performance degrades with smaller samples [9]. In the meantime, DPLS was used for RCA on the failure in the Tennessee Eastman chemical plant by maximizing covariance between the predictors [11][30]. The root cause was successfully detected, whereas the assumption of multivariate Gaussian distribution for the control limits of the PCA or PLS-based monitoring indices restricted their validity and adaptability to realistic process data [31]. An application of FDA for RCA in the chemical processing industry revealed FDA's shortcoming in capturing nonlinear behaviour in the data [12]. It led to poor performance with an overall misclassification rate of 38 %. With the aid of feature selection algorithms, the misclassification rate dropped to 17% [12]. In general, statistical RCA approaches can assist RCA in a short run time [11]. However, their performance is not the most competitive [30]. They struggle with non-linear relationship modelling [12], the requirement of large data size

[9] and support algorithms [4][5][12] and lack of interpretability. Therefore, more integrated and automated RCA methods are in demand.

2.2.2 Machine Learning Methods

Machine learning techniques enable automated RCA by pattern mapping and knowledge acquisition from historical product defect records. Algorithms such as decision tree, Support Vector Machine (SVM), Neural Network (NN) and BN have been leveraged to identify the root cause from the historical production data automatically under faulty situations. Decision tree is popular among the machine learning methods for RCA thanks to its nature of generating human-interpretable results [15]. Chen [32] presented a decision tree learning approach to diagnose failures in large internet sites. An improved method of the interactive decision tree was proposed by Detzner to combine experts' domain knowledge into the pattern recognition process in the automotive industry [15]. However, decision tree did not seem to be a well-performed classifier [33] as it required a longer sampling time [2] and was incompetent in handling scarce datasets [32]. Other research has shown that SVM outperforms many conventional classification technologies when it comes to root cause diagnosis [16]. Chiang [12] explored the feasibility of using SVM to determine the root cause of the observed out-of-control status in the chemical processing industry. It turned out that SVM outperformed FDA by three times on the misclassification rate. SVM also tends to run faster [17] and to have a stronger generalization capability with small sample learning problems [16]. Unfavourably, the recognition accuracy of SVM degrades severely when the two crucial structural parameters, penalty factor and kernel function parameter, are not tuned desirably [12]. Han et al. visualised the significant effect of the choice of different parameter pairs on the performance of SVM [17]. Artificial Neural Network (ANN) has proven effective for RCA to recognise patterns in the data easily even over distorted inputs whilst yielding relatively high accuracy [18] and flexibility [34] in classifying the root causes. On the flip side, ANN is subjected to a long training time [34] and the risk of poor convergence with increasing layers [18]. One of the heated discussions about ANN is its black box feature, omitting a logical explanation between inputs and outputs [34]. Accordingly, manufacturers are reluctant to employ it in the real world. Lee addresses this issue by proposing a Fault Detection Classification Convolutional Neural Network (FDC-CNN) model to intentionally

map the artificial variables with substantial features representing process fault [35]. In general, the aforementioned machine learning methods are faced with various issues such as prediction accuracy, data scarcity and causality discovery. Moreover, most of the methods model the RCA problem as a classification problem in determining whether the reason for defects belongs to a class. Such approaches do not account for the probability of multiple root causes as well as the distinct strength of their causal influences on product quality. However, probabilistic reasoning explains the causal influence of the potential root causes with stochasticity. It is an important attribute for RCA in industrial practice as it includes uncertainty and supports decision-making for on-site staff. Another common limitation among the existing RCA methods is the lack of interpretability. Even though the final root causes are identified, it is intractable to explain the causality of the root cause (i.e., why the identified root cause contributes to the issue) without a human-interpretable knowledge representation. This limits the manufacturer's ability to find corresponding actions to solve the problem in the real-world scenario as the results are not explainable and not visualised. Therefore, there is an urgent need for a robust, intelligent, human-interpretable probabilistic reasoning method for RCA.

BN has emerged in the field of RCA with sheering benefits. [5][36]Many studies have proven it effective to address uncertainty [6] [19] where multiple root causes can contribute to the occurred product failure with various probabilities instead of just one defect reason. The probabilities of different root causes inferred from BN can also quantify the strength of their causal influences on the quality issue. Causal relationships can be discovered in BN through modelling conditional dependencies between different variables, making it powerful for reasoning in RCA [5][36]. BN is also a powerful tool for knowledge representation as it displays the relationship amidst different features [21]. Acceptable results can be obtained by BN even with incomplete data [21]. Weidl, G. [20] has further highlighted its advance in decision support with the ability of probabilistic reasoning. Every coin has two sides, implementation of BN is an NP-hard problem. The computational expense increases, as the network size goes up [19]. Furthermore, BN's performance can be worsened without any prior knowledge [6]. The robustness of BN is sensitive to data sparsity [23] and the choice of

different structure learning algorithms [26]. BN models tend to lose robustness from big datasets. The sparsity from big data can make BNs have low independent validation accuracy and be overfit [23]. Moreover, the prediction accuracy of an individual BN model has been demonstrated to be dependent on the selection of the BN structure learning algorithm [26]. This means that BN learnt from a single structure learning algorithm can be insufficient and unrobust for identifying root causes. Comparatively, fusing multiple learning algorithms can improve the robustness of the models [25]. Ensemble learning techniques have the ability to combine different models to improve performance. Yu et al. have used ensemble techniques to integrate Gaussian mixture models to detect and recognize various defect patterns in the semiconductor manufacturing process [37]. Ensemble learning technique has also been applied to extract more robust features, allowing an effective fault diagnosis of a reciprocating compressor [27]. As a result, we are inspired to integrate ensemble learning techniques into the proposed method.

2.3 Summary

In summary, RCA has proven to be a difficult and time-consuming engineering problem using conventional methods. Statistical techniques alone tend to be insufficient for a fully automated RCA. Machine learning methods have shown higher accuracy and efficiency in identifying the root cause. However, most of the papers model RCA as a classification problem. Such approaches do not account for multiple root causes as well as the distinctive strength of their causal influences on product quality. Another common shortage of the existing methods is the lack of interpretability, where the process of root cause identification cannot be explained or visualised. These limitations hinder its practical use in real-world manufacturing. On the other hand, BN has the ability to perform human-interpretable probabilistic learning. Nevertheless, its prediction accuracy is sensitive to scarce data and has the tendency of overfitting with big datasets. The robustness of BN model has also been found to be contingent on the choice of structure learning algorithms and prior knowledge. Therefore, there is an urgent need for a robust, intelligent, human-interpretable probabilistic reasoning method to address the limitations in the existing RCA methods.

Chapter 3

Problem Definition

This chapter provides a definition of the RCA problem. In Section 3.1, the RCA problem is described in the context of the plastic industry with an introduction of the factory's historical production data. It lists three questions that we aim to answer in this project. Section 3.2 formulates the RCA problem mathematically with technical objectives. Section 3.3 presents the proposed method to outline how it can achieve the technical objectives, which guides the process of prediction model construction in Chapter 4.

3.1 RCA Problem Definition

The RCA problem is based on a real production system from a plastic factory, ABC. They have collaborated with APT to collect and store their production performance data. The historical production data contains different production features and observations about the operated jobs. The variables selected from the historical production data are presented in Table 3.1 to impart a basic understanding of the factors involved in the RCA problem.

Table 3.1 Variables selected from the historical production data collected from factory ABC for RCA

Variable Name	Variable Type	Data Description
jobRun	Feature	the number of operations performed for a job
jobStartTime	Feature	the timestamp the job starts
Equip	Feature	the machine a job runs on
jobStartUser	Feature	the operator who starts the job
jobStopUser	Feature	the operator who stops the job
ProductionTime	Feature	the time it takes to complete a job
DownTime	Feature	the downtime rate of a job
jobPLCSetupTime	Feature	the time it takes to set up a job
RejWeight_kg	Feature	the weight of rejected products in kg
rrnDescription	Feature	the names of the potential reject reasons
ActRejectPC	Observation	the reject rate of a job
isRejectFail	Observation	Binary; 1 indicates a job with quality issues, 0 otherwise

In particular, "*isRejectFail*" is the indicator of the quality performance of a job. It is a binary variable, comprising only "0" and "1". "1" indicates that the job has an unusually high reject rate. Intuitively, this job is classified as a risky job with a quality issue; "0" shows that the job has no alarming quality issue. Its value is determined by

$$isRejectFail = \begin{cases} 1, & if\ ActRejectPC > pccRejectPC + tol \\ 0, & otherwise \end{cases} \quad (1)$$

where "$ActRejectPC$" is the reject rate that reflects the quality of a job by counting the number of rejected products in ratio to the total quantity of the produced products during the job; "$pccRejectPC$" is the target reject rate defined by the manufacturers, embodying the ideal reject rate that the factory aims to achieve; and tol represents the tolerance of the reject rate, reflecting the permissible limit of variation in reject rate for the job.

"$rrnDescription$" holds a list of root causes that has historically led to quality failures for a job. These root-cause variables provide a list of root-cause candidates for RCA, guiding the analyst to localise the causal relationship between the root-cause variables and the observations. Other features listed above can also contribute to a quality issue for a job. Knowing the variables from the historical production data gives a clearer direction to define the RCA problem in the manufacturing context.

After understanding some key variables in the historical production data in factory ABC, RCA can be defined more specifically by referring to the variables presented in Table 3.1. RCA is the task of finding the root causes of an issue. It is triggered when an anomaly occurs. For factory ABC, its main issue is product quality. It has been found that their average reject rate ("$ActRejectPC$") is around 10%, which can adversely affect the workflow and the cost level. Therefore, RCA is required to solve the product quality problem in the factory.

The RCA starts with understanding the underlying mechanics of the quality issue. The experts in ABC normally perform RCA on a problematic job (i.e., "$isRejectFail" = 1$) by analysing the historical pattern on an excel sheet to identify the causal factors intuitively. This conventional approach is difficult and time-consuming as mentioned in Section 2.1. Therefore, this project aims to find a more robust, intelligent and human-interpretable method for RCA. By inspecting the historical data with the domain experts from ABC, it appears that some job features in Table 3.1 such as the operator ("*jobStartUser*"), the types of machines ("Equip"), and production time ("*ProductionTime*") can have a causal influence on the reject rate ("$ActRejectPC$") of the jobs. Since this information is captured by AspectPL, this research is inspired to uncover the hidden causal relationships between the variables from these historical production records automatically. Based on the discovered causal relationships between the variables from the historical production data, we also want to know the probability of each root cause causing the quality issue for a job. In this way, the obtained probabilities of each reject reason can help the on-site staff to make data-driven decisions on fixing the quality issue. To take one step further, we would like to foresee what jobs might be subjected to quality issues in the future to prevent the risk in production.

Thus, this problem can be encapsulated into the following three questions with a focus on Q2:

Q1. Given a product, what are the causal relationships in the production system based on the historical data?

Q2. Given a job that has been finished and discovered with quality issues, what are the potential reject causes, and what are their corresponding probabilities based on the known features?

Q3. Given a job yet to be run, is the job going to pose a quality issue (i.e., a high reject rate)?

3.2 Mathematical Formulation of RCA Problem

As the three questions that RCA needs to solve in this study have been identified, we can further formulate the RCA problem mathematically to allow technological models to be built. First, all the elements in the RCA problem such as jobs, historical data, features, observed quality issues and root causes will be defined mathematically. Then, the three questions proposed in Section 3.1 will be translated into the mathematical format. See Table 3.2 for the list of notations.

In a factory, it has produced M distinct types of products $Pd = \{Pd_1, Pd_2, \dots, Pd_M\}$ and operated N jobs $J = \{J_1, J_2, \dots, J_N\}$ on the production floor. The historical dataset D is collected from the records of N jobs. The features of job records, $X = \{X_1, X_2, \dots, X_I\}$, are stored in the dataset D, $X \in D$. X contains a number of I features. Each individual feature is indexed as X_i or (X_i, X_j) in a pair. X encodes the production information about the jobs such as the operator, the type of machines used, and production time as shown in Table 3.1 with the variable type "Feature". Let R_n be the product quality of job J_n, signifying the observational variable "*isRejectFail*" presented in Section 3.1. It is a binary variable, $R_n \in \{0, 1\}$; 1 indicates that job J_n is problematic with quality issues; and 0 indicates that job J_n has no quality issue. The occurred root causes, $Y = \{Y_1, Y_2, \cdots, Y_C\}$ where C denotes the total number of occurred root causes, in the records are regarded as the root-cause candidates leading to product quality issues (i.e., $R = 1$). Each root cause Y_c has a corresponding probability P_c implying the likelihood of root-cause Y_c causing the identified quality issues.

With all the elements involved in RCA defined, the three questions in RCA can be formulated mathematically as follows:

Q1. Given the historical production data D with features $X = \{X_1, X_2, \dots, X_I\}$, how do we build a function h: such that it can satisfy all the components in equation (2), where "1" indicates there is causality between the pair and "0" otherwise. h is developed to detect the existence of causal relationships between the job features X_i and X_j, $h(X_i, X_j) \rightarrow \{0, 1\}$; the existence of

causal relationships between the job feature X_i and the root cause Y_c, $h(X_i, Y_c) \rightarrow \{0,1\}$; the existence of causal relationships between the job feature X_i and the observation R, $h(X_i, R) \rightarrow \{0,1\}$; and the existence of causal relationships between the root cause Y_c and the observation R, $h(Y_c, R) \rightarrow \{0,1\}$.

$$h(X_i, X_j) \rightarrow \{0,1\}; h(X_i, Y_c) \rightarrow \{0,1\}; h(X_i, R) \rightarrow \{0,1\}, h(Y_c, R) \rightarrow \{0,1\} \quad (2)$$

$$\text{for } X_i, X_j \in X, i \neq j, X_i, Y_c \in Y$$

Q2. Given a job J_n that has been finished with quality issues $R_n = 1$, what are the nonempty set of root-cause variables $Z = \{Y_1, Y_2, \cdots, Y_{C'}\}$ for $C' \leq C$, $Z \subseteq Y$, and what are their corresponding probabilities $P = \{P_1, P_2, \cdots, P_{C'}\}$ based on its job feature vector X_n ?

Q3. Given a job $J_{n'}$ that has not been operated yet, what is the value of $R_{n'}$, $R_{n'} \in \{0,1\}$?

Table 3.2 Table of notations for mathematical symbols in RCA problem formulation and proposed method formulation

Symbol	Description
Indices	
m	Index of produced products
n	Index of jobs
i	Index of job features
c	Index of root-cause variables
k	Index of structure learning algorithm
s	Index of bootstrapped samples
Sets	
Pd	Set of products, $Pd = \{Pd_1, Pd_2, \ldots, Pd_M\}$
J	Set of jobs, $J = \{J_1, J_2, \ldots, J_N\}$
X	Set of job features, $X = \{X_1, X_2, \ldots, X_I\}$,
Y	Set of root-cause variables, $Y = \{Y_1, Y_2, \cdots, Y_C\}$
P	Set of probabilities for root-cause variables, $P = \{P_1, P_2, \cdots, P_{C'}\}$
X_i	Job features for job i

D	Set of historical job records for corresponding product, $D = \{D_1, D_2, \dots, D_M\}$
G	Set of BN models learnt by different structure learning algorithms
V	Set of vertices in a BN structure G
A	Set of arcs in a BN structure G
p	Set of conditional probabilities for vertices V in a BN structure G
E	Set of evidence of a to-be-predicted job J_n to be input into a BN structure G for Bayesian inference, $E \leftarrow X_n$
Constants	
M	Number of products
N	Number of jobs
I	Number of features
C	Number of occurred root causes
K	Number of implemented structure learning algorithms
S	Number of total bootstrapped samples
Variables	
X_i	i th production feature of the jobs
Y_c	c th root cause contributing to the quality issues of the jobs
R_n	Binary variable: 1 indicating job J_n has quality issues; 0 otherwise

3.3 Proposed Method Formulation

This section presents the proposed product-wise framework of ensembled BN. It illustrates how the proposed method can solve the previously formulated RCA problems mathematically. The fundamental BN model has also been defined to lay the foundation of model construction in the next chapter.

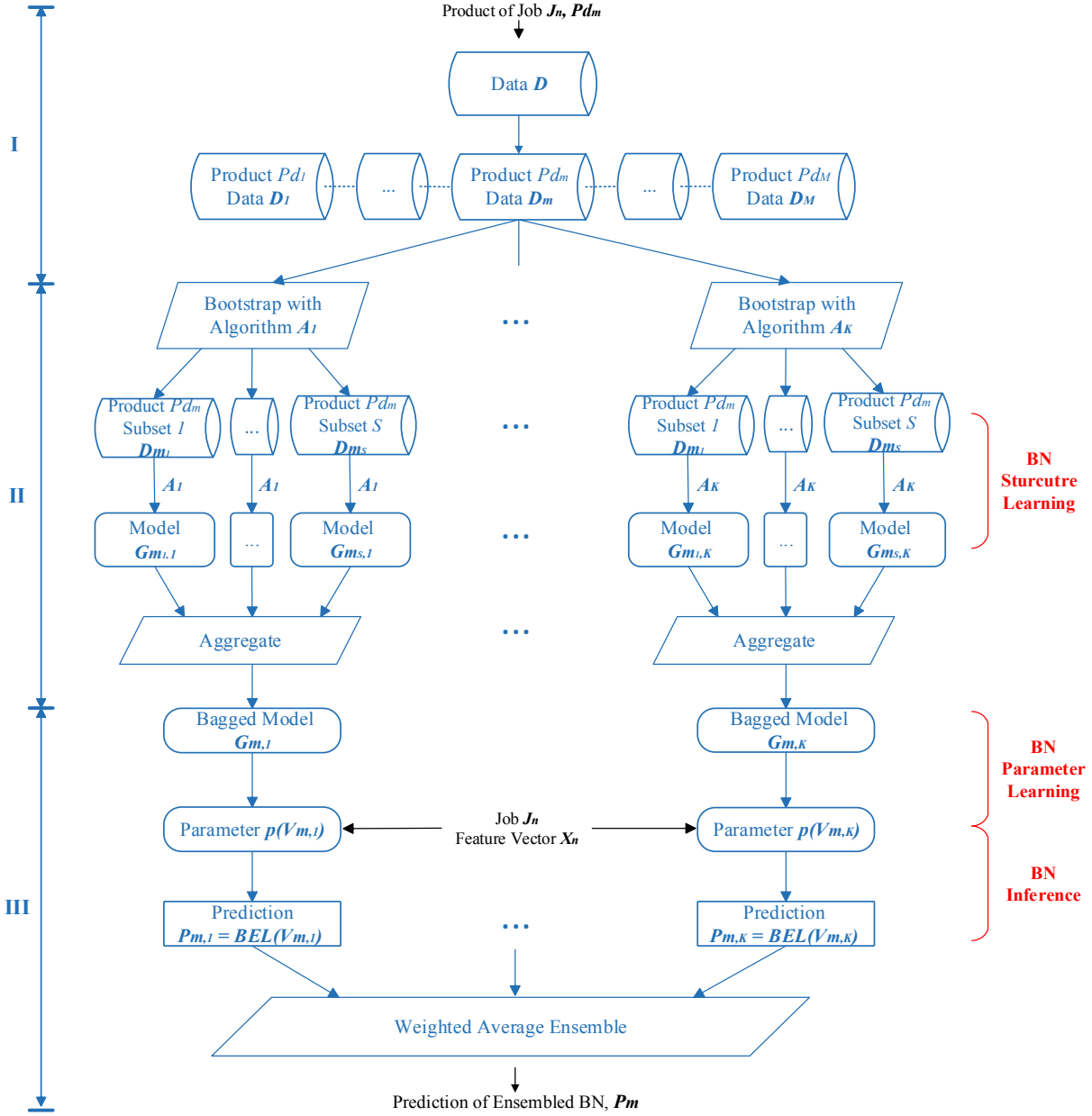


Fig. 3.1 The proposed product-wise framework of ensembled BN for finding the root cause relations between job features X, potential root causes $Y$and the observation of product quality R for product Pd_m based on its historical data D_m

The proposed method employs a product-wise framework of the ensembled BN model to find the causal relationships between job features X, potential root causes Y and observation of product quality R for a product Pd_m based on its historical production data D_m. Once the causal network is discovered, the job features of the to-be-predicted job J_n, X_n will be introduced as external evidence E into the causal network to allow probability inferencing and risk predictions. The proposed framework consists of three steps outlined in Fig. 3.1: I. Modularise Data by Products, II. Construct bagged BN models, and III. Combine BN predictions using the Weighted Average Ensemble Learning (WAEL) technique. The three steps will be described following the mathematical formulation of the RCA problem in Section 3.2. The notations of the mathematical symbols mentioned in the section are displayed in Table 3.2.

In the first step, given a job J_n producing product Pd_m, the proposed method modularises the historical data D into M (i.e., the number of products) small product-wise data samples $\{D_1, D_2, \ldots, D_M\}$. Each dataset captures the historical job records that produce the corresponding product. This strategy reduces the sample size for BN. Hence, it increases BN learning efficiency and avoids sparsity that often occurs in big datasets [23]. As a result, the historical data sample for job J_n will be D_m according to its product Pd_m.

Based on the modularised historical record D_m, in the second step, the structures of BN will be learnt using a specific structure learning algorithm A_k with bagging ensemble technique. In total, there are K different structure learning algorithms, $A_k \in \{A_1, A_2, \ldots, A_K\}$, implemented to learn BN models. Initially, D_m is bootstrapped evenly into S subsets for all the K algorithms as shown in part II of Fig. 3.1. Each bootstrapped sample D_{m_s} is a subset of D_m, $D_{m_s} \subseteq D_m$. Then, the structure learning algorithm A_k, $A_k \in \{A_1, A_2, \ldots, A_K\}$, will learn a BN model $G_{m_s,k}$ for each subset D_{m_s} resulting in a total of S models $\{G_{m_1,k}, G_{m_2,k}, \ldots, G_{m_S,k}\}$ for algorithm A_k. The implementation of different structure learning algorithms A_k learning BN model $G_{m_s,k}$ will be illustrated in Section 4.2.3.2. At the end of Step 2, a number of S learnt BN models $\{G_{m_1,k}, G_{m_2,k}, \ldots, G_{m_S,k}\}$ are aggregated into one BN structure $G_{m,k}$ using bagging

ensemble technique. The bagging ensemble technique is integrated to account for sample variations and to reduce the risk of scarce data by fusing the models from the various S bootstrapped samples, enabling robust BN structure learning. The detailed implementation of model aggregation using bagging ensemble is shown in Section 4.2.2. After the bagged BN structure $G_{m,k}$ is constructed with distinct structure learning algorithm A_k, parameter learning and inferencing of BN structures $G_{m,k}$ need to be conducted in Step 3 to infer a set of root-cause probabilities. To present these processes, the properties of BN structures $G_{m,k}$ need to be defined.

The BN structures $G_{m,k}$ learnt from Step 2 or in general form G are directed acyclic graphs (DAGs). Mathematically, BN can be encoded as BN: $G = (V, A)$ where V denotes vertices ($v_i \in V$), corresponding with the variables selected from the job features X, root-cause variables Y and quality risk indicator R mentioned in Section 3.3, $V \subseteq \{X, Y, R\}$. In the case of RCA for product Pd_m, $\{X, Y, R\} \subseteq D_m$. For example, the vertices $V = \{v_1, v_2, v_3, v_4\}$ in Fig. 3.2 represent the quality issue indicator R, the potential root causes Y_1, Y_2 occurred in D_m and the job feature X_1 respectively. Each node v_i has different states s_i, embodying the values that the specific variable can take (e.g., v_1 in Fig. 3.2 has two states "0" and "1" indicating the existence of quality issues of a job). A denotes the set of arcs that directly link the vertices, signifying the conditional dependency between the connected random variables (v_i, v_j). The main task of BN structure learning is to determine the existence of the arcs between random variables (v_i, v_j). The strength of causal dependency relations is quantified by conditional probabilities $p(v_i|v_j)$ [19]. Accordingly, each vertex is associated with a Conditional Probability Table (CPT) that defines probabilities for the distinct states of the node given the states of its parents [19]. The procedure of parameter learning in BN is to obtain the CPT of each vertex, $p(v_i|v_j)$ or $p(v_i)$ if v_i has no parent.

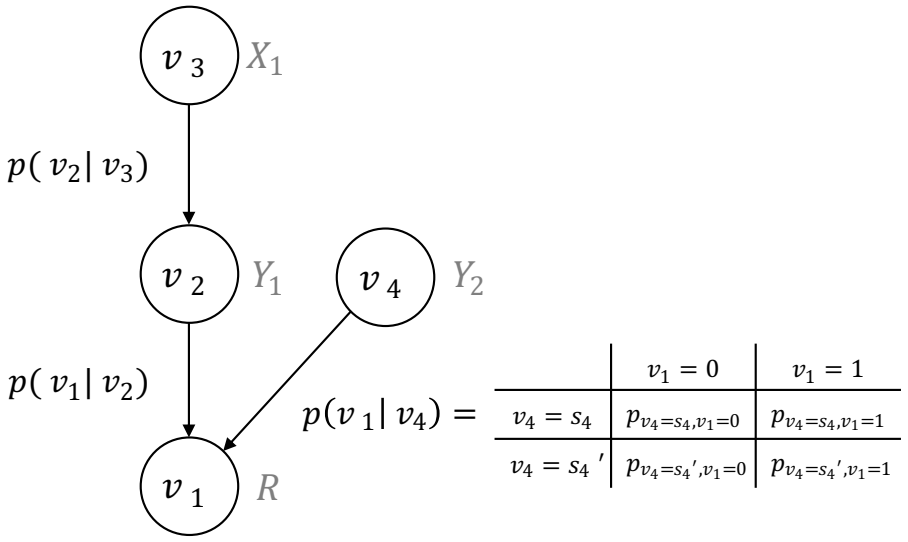


Fig. 3.2 An abstract BN consists of variables $\{v_1, v_2, v_3, v_4\}$

Knowing the formulation of bagged BN structure G_{mk} from Step 2, parameter learning and inferencing can be illustrated on each bagged BN model $G_{m,k}$ to obtain a set of predicted root-cause probabilities $P_{m,k} = \{P_{m,k_1}, P_{m,k_2}, \cdots, P_{m,k_C}\}$. As shown in part III of Fig. 3.1, Step 3 starts with BN parameter learning where $p(v_i)$ for each v_i from all the vertices $V_{m,k}$ in model G_{mk} is estimated, $v_i \in V_{m,k}$. The explicit parameter learning method will be elaborated in Section 4.2.4. Then inference introduces the job features of the to-be-predicted job J_n as external evidence E into the BN model G_{mk} (i.e., $E \leftarrow X_n$). Bayesian inference then updates the belief distribution for each $v_i \in V_{m,k}$, $BEL(v_i)$ based on the new evidence E using junction tree algorithms (explained in Section 4.2.5). $BEL(v_i)$ stands for the conditional distribution of v_i, given all the associated evidence in the network. When v_i corresponds the root-cause variable Y_c, $BEL(v_i)$ contains the probability of root cause Y_c contributing to quality failures, P_C. Similarly, the probabilities can be inferred for any potential defect root cause $Y \subseteq D_m$, obtaining $P = \{P_1, P_2, \cdots, P_C\}$. The resulting probabilities are denoted as $P_{m,k} = \{P_{m,k_1}, P_{m,k_2}, \cdots, P_{m,k_C}\}$ in the context of using structure learning algorithm A_k based on dataset D_m. Finally, the sets of predicted probabilities $\{P_{m,1}, \dots, P_{m,K}\}$ from different BN models $\{G_{m,1}, G_{m,2}, \dots, G_{m,K}\}$ learnt by distinct algorithms $\{A_1, A_2, \dots, A_K\}$ are fused into a single set of root-cause probabilities $P_m = \{P_{m_1}, P_{m_2}, \dots, P_{m_C}\}$

through WAEL technique to enhance prediction accuracy and robustness, which is further explained in Section 4.2.6. The resulting root-cause probabilities P_m can answer Q2 for RCA. The same procedures are taken to predict the quality risk $R_{n'}$, of a job $J_{n'}$ to answer Q3.

In general, the proposed solution comprises two functional modules, BN models and ensemble learning techniques. BN is the fundamental model of our proposed method. It is a probabilistic graphical model for reasoning under uncertainty [23]. BN development process goes through three procedures namely, structure learning, parameter learning, and inference to provide probabilistic graphical reasoning. Structure learning algorithms for BN uncover the causal relationships between the variables $V \subseteq \{X, Y, R\}$ from the historical data D and construct human-interpretable graphical networks accordingly. Parameter learning estimates the CPT for each vertex v_i, which is an important attribute for inference. Inference updates the belief in the network by passing the messages regarding probability distributions throughout the network to infer the probabilities of different root causes leading to the event of interest. As a result, intelligent and human-interpretable probabilistic reasoning is achieved by BN. On the other hand, ensemble learning techniques are incorporated to reinforce the robustness of the constructed BN models. Bagging ensemble techniques are applied during BN structure learning to counter BN's sensitivity to data sparsity. The weighted average ensemble learning technique is integrated after the BN inferencing. It fuses the predictions from the structures of BN models learnt by different learning algorithms to alleviate the deficiencies in the accuracy and stability of a single BN model, ensuring robustness. The implementation of the ensembled BN models will be explained explicitly with a case study in Section 4.2 in the sequence of bagging ensemble learning, BN structure learning, parameter learning, Bayesian inference and WAEL, which follows the workflow of the proposed RCA framework.

Chapter 4

Prediction Model Construction

To build the models, data and job features need to be collected and pre-processed. Then, the proposed ensembled techniques and BN models are implemented to provide a robust, intelligent and human-interpretable RCA method. Section 4.1 explains the processes of data collection and data pre-processing. Section 4.2 explains the design and the construction of the ensembled BN framework for automated RCA. A case study will be presented as an example to demonstrate the process and to answer the proposed RCA questions.

4.1 Data Preparation

As shown in the proposed method in Section 3.3, job features are essential to the construction of ensembled BN models for RCA. Therefore, the production data need to first be collected from the machines. Then, data pre-processing is needed to prepare and extract job features from the data for the prediction model construction. Sequentially, Section 4.1.1 presents the process of data collection. Section 4.1.2 demonstrates the process of data pre-processing, including data cleaning, feature selection and data discretisation.

4.1.1 Data Collection

Data are generated by various sensors stalled on the machines in factory ABC. Programmable Logic Controller (PLC) offers a digital connection between the physical system and the computational system to collect data. It is a special-purpose computer designed to withstand industrial conditions (e.g., extended temperature ranges, electrical noises, and various vibrations and impacts.). Its fundamental role in system control is achieved by executing programmed control functions on corresponding physical devices. Fig. 4.1 demonstrates the overall model of a PLC-controlled manufacturing system. An input device measures and transmits data from the manufacturing system into the PLC [38]. On the other hand, output devices receive commands from the PLC to execute a specific operating action.

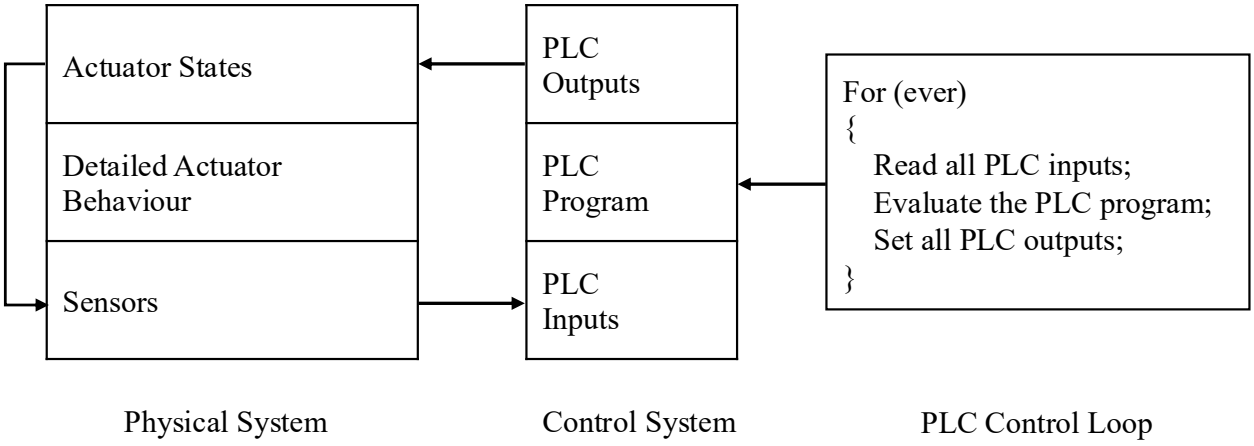


Fig. 4.1 The overall model for a PLC controlled manufacturing system [42]

In this research, the data sample of the plastic industry is extracted from AspectPL software. AspectPL uses an advanced PLC, Beckhoff Embedded PC Automation Controllers, for data collection to realize a larger memory space and more economical use. Machine connectivity is supported with EtherCAT for a wide range of connections at a high speed. AspectPL was designed with the ISA-95 standard, an international standard for building an automated interface between enterprise and control systems. This provides manufacturers with consistent terminology and information operation models globally. A snapshot of the data collected from ABC industry has captured in Fig. 4.2. It contains the raw data with all the job features.

isRejectFail	rejContributor	rrnDescriptio	rejQuantity	ToolID	matID	jobID	jobRun	equID	jobStartUser	jobStopUser	jobStartTime	ProductCount	ActCycleTime	RejWeight_kg	ActRejectPC	ActDowntimePC	jobPLCDownTime	jobPLCSetupTime	ProductionTime
0	1	Setup Rejec	205	1213	PPB0045	129	1	BM06	23	26	12/11/14	7646	24.85	10865	2.68	25.75	65880	59640	255840
0	1	Setup Rejec	202	1213	PPB0045	569	1	BM07	26	6	16/02/15	6720	26.18	10706	3.01	8.03	15360	11580	191280
0	0.29	Setup Rejec	256	1213	PPB0045	728	1	BM07	23	29	9/03/15	11194	25.89	46322	7.81	19.38	69660	23280	359520
0	0.71	Yield Updat	618	1213	PPB0045	728	1	BM07	23	29	9/03/15	11194	25.89	46322	7.81	19.38	69660	23280	359520
0	0.24	Setup Rejec	58	1213	PPB0045	969	1	BM01	6	6	20/04/15	13145	25.61	12985	1.86	1.54	5280	3540	341880
0	0.76	Yield Updat	187	1213	PPB0045	969	1	BM01	6	6	20/04/15	13145	25.61	12985	1.86	1.54	5280	3540	341880
0	0.12	Setup Rejec	91	1213	PPB0045	1121	1	BM07	6	29	15/05/15	13670	26.14	40810	5.63	2.79	10260	3720	367560
0	0.88	Yield Updat	679	1213	PPB0045	1121	1	BM07	6	29	15/05/15	13670	26.14	40810	5.63	2.79	10260	3720	367560
0	0.01	Setup Rejec	4	1213	PPB0045	1228	1	BM07	30	30	11/06/15	13538	26.02	33814	4.71	3.26	11880	3360	364140
0	0.99	Yield Updat	634	1213	PPB0045	1228	1	BM07	30	30	11/06/15	13538	26.02	33814	4.71	3.26	11880	3360	364140
0	0.06	Setup Rejec	73	1213	PPB0045	1383	1	BM01	23	23	5/07/15	12892	25.7	67946	9.94	5.98	21060	14640	352440
0	0.94	Yield Updat	1209	1213	PPB0045	1383	1	BM01	23	23	5/07/15	12892	25.7	67946	9.94	5.98	21060	14640	352440
0	0.16	Setup Rejec	49	1213	PPB0045	2914	1	BM01	23	26	23/03/16	5476	25.14	16748	5.77	12.41	19500	7800	157140
0	0.84	Yield Updat	267	1213	PPB0045	2914	1	BM01	23	26	23/03/16	5476	25.14	16748	5.77	12.41	19500	7800	157140
0	1	Yield Updat	553	1213	PPB0045	3260	1	BM01	29	29	18/05/16	5713	25.89	29309	9.68	17.64	31680	0	179580
1	0.27	Setup Rejec	209	1213	PPB0045	4368	1	BM01	25	26	23/11/16	5943	25.2	41499	13.18	5.81	9240	7980	159000
1	0.73	Yield Updat	574	1213	PPB0045	4368	1	BM01	25	26	23/11/16	5943	25.2	41499	13.18	5.81	9240	7980	159000
0	15	Setup Rejec	75	1213	PPB0045	5321	1	BM01	26	23	26/04/17	134	27.97	265	3.73	87.75	26640	25920	30360
0	-14	Yield Updat	-70	1213	PPB0045	5321	1	BM01	26	23	26/04/17	134	27.97	265	3.73	87.75	26640	25920	30360

Fig. 4.2 A snippet of raw historical production data collected from factory ABC

4.1.2 Data Pre-processing

This section demonstrates the process of manipulating the manufacturing data collected from the plastics factory ABC. The extracted raw data goes through data cleaning (Section 4.1.2.1), feature selection (Section 4.1.2.2) and data discretisation (Section 4.1.2.3) to be prepared for predictive model construction later in Section 4.2.

4.1.2.1 Data Cleaning

Some erroneous and unusual data points might occur in the raw data extracted from the factory. The data cleaning process will first inspect the quantitative data to exclude irrational data points. Then a filter process will be undertaken to ensure a sufficient sample size for model construction using the proposed method.

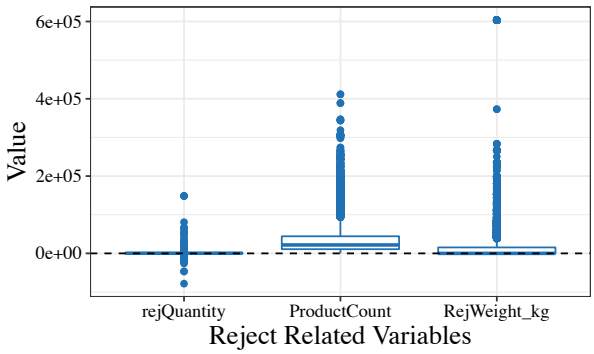


Fig 4.3 (a) Data distribution of reject related variables before data cleaning

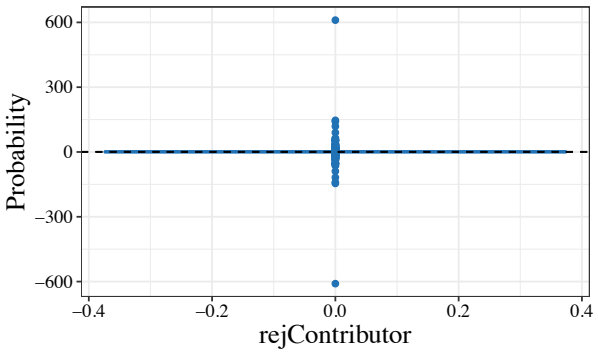


Fig 4.3 (b) Data distribution of "*rejContributor*" before data cleaning

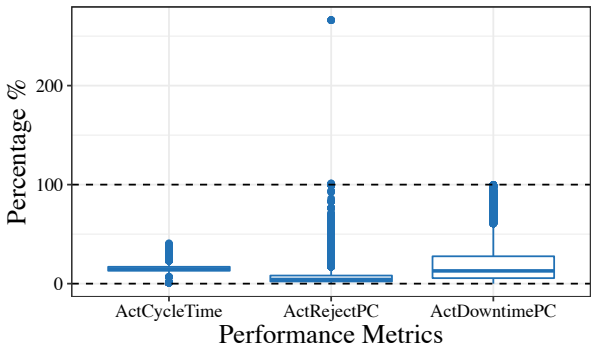


Fig 4.3 (c) Data distribution of performance metrics before data cleaning

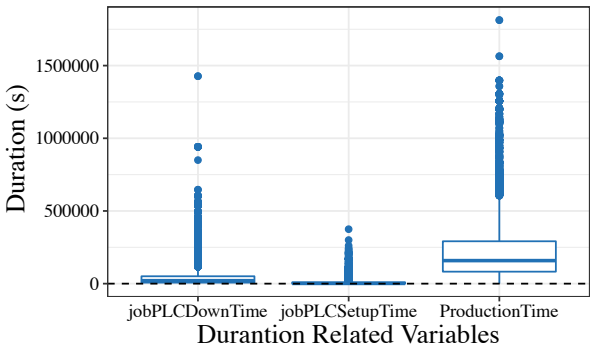


Fig 4.3 (d) Data distribution of duration variables before data cleaning

Fig. 4.3 Data distribution of quantitative variables from the historical production data before data cleaning

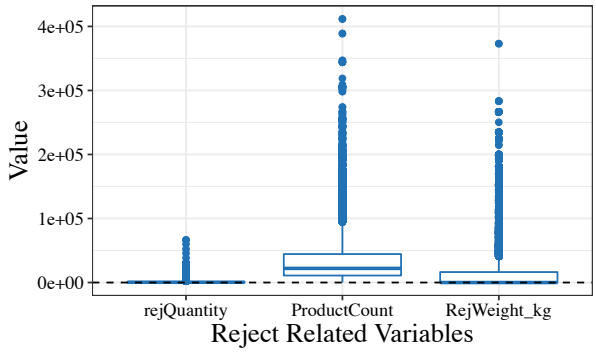


Fig 4.4 (a) Data distribution of reject related variables after data cleaning

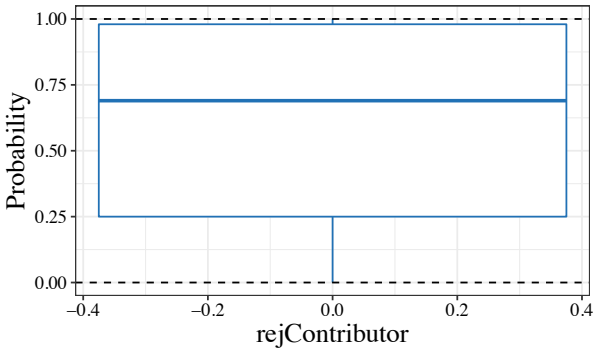


Fig 4.4 (b) Data distribution of "*rejContributor*" after data cleaning

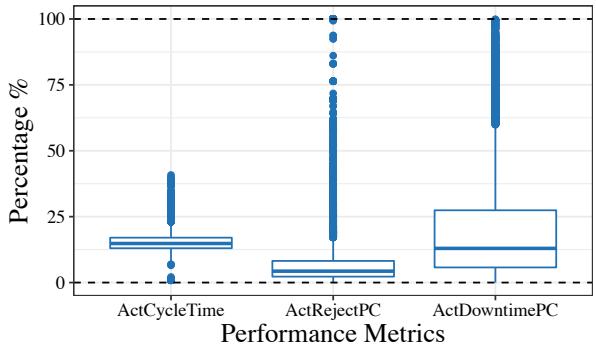


Fig 4.4 (c) Data distribution of performance metrics after data cleaning

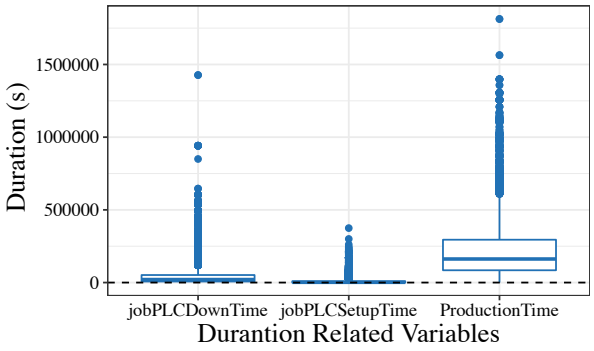


Fig 4.4 (d) Data distribution of duration variables after data cleaning

Fig. 4.4 Data distribution of quantitative variables from the historical production data after data cleaning

Firstly, the quantitative variables are inspected to guarantee the rationality of the data. The difference in data distribution for different numeric attributes before and after data filtering has been shown in Fig. 4.3 and Fig. 4.4 respectively. In common sense, the reject quantity ("*rejQuantity*") of a job should be equal to or above zero. However, a few negative values occur in the first boxplot on the left in Fig. 4.3. This could be caused by workers' delayed action on starting product count in AspectPL system for the job. In AspectPL, reject quantity is calculated as the difference between the number of produced materials and the number of packed good products. If the software receives an indication of a job starting later than reality, the total product count for the job will end up much lower than the actual number, while the number of good products remains unchanged, resulting in a possibility of negative reject values. Therefore, such values are incorrect and should be excluded from the dataset.

Instinctively, some invalid data are expected in the probability of reject causes ("*rejQuantity*") and the reject rate ("*ActRejectPC* ") of the jobs as these two variables are formulated in accordance with reject quantity. These incorrect values are also discarded to ensure that the sample falls inside the reasonable range after cleaning as displayed in Fig. 4.4.

The data is then filtered to make sure that each material included in the sample has more than 10 historical job records. This is to accommodate the overall framework of product-wise BNs, which has been described in Section 3.3. Consequently, it reduces the data size to 199 products with 6721 job instances.

4.1.2.2 Feature Selection

This section explains the process of selecting relevant and valid features from the perspective of composite variables and correlated variables. Feature selection is to ensure the attributes used to build the model will not hinder the model performance.

4.1.2.2.1 Composite Variables

To avoid repetitive information and multicollinearity induced by including both composite variables and their constituent variables in the model, some individual features need to be eliminated.

$$Reject\ Rate\ = \frac{Reject\ Quantity}{Product\ Count} \tag{3}$$

As displayed in equation (3), the reject rate of a job ("*ActRejectPC*") is calculated from reject quantity ("*rejQuantity*") and product count ("*ProductCount*"), which implies multicollinearity between these attributes. This means that the outcome of "*rejQuantity*" and "*ProductCount*" can alternate the probabilistic information on the outcome of "*ActRejectPC*", and vice versa. According to Song et al.'s statement [39], composite variables are widely applied in practice to control Type I error rate (i.e. false positive rate), circumvent multicollinearity, or condense useful information. Therefore, the composite variables will be kept in the models and their

respective constituent attributes will be excluded. Accordingly, "*rejQuantity*" and "*ProductCount*" will not be considered as inputs of the model. Reject rate is chosen also because it reflects the quality level of a job, which is an important causal factor to be included. Following the same philosophy, "*jobPLCDownTime*" and "*ProductionTime*" are also discarded and the composite feature "*ActDowntimePC*" has been conserved in the model as a predictor.

4.1.2.2.2 Correlated Variables

It is well known that correlated features in regression analysis can lead to an inflation of Type I Error, whereas such issue is inclined to persevere in BN [40]. If random variables in the model are correlated, it indicates that changes in one variable are associated with shifts in another. This is because the change in unison among multiple independent attributes makes it harder for the model to establish the relationship between each independent variable and the dependent variable separately. Bae et.al [40] have given mathematical proof and straightforward experiment results in Table 4.1 regarding the adverse effect of neglecting correlation on false positive rate during network structure learning regardless of the model selection metrics.

Table 4.1 Experiment results on the effect of ignoring the correlation [40]

	IID Data			Correlated Data		
	BIC	AIC	LRT	BIC	AIC	LRT
No. of times the true network was selected (out of 1000 simulations)	79	363	386	71	246	301
False Positive Rates	0.0033	0.1432	0.0490	0.0136	0.1986	0.0893

The same problem is likely to occur in the inference process of BN. For example, two features highly correlated with each other and with y, might both be identified as insignificant in an inference model, potentially missing an important explanatory signal. Furthermore, removing highly correlated features allows more accurate relations to be found, and facilitates computation storage and run speed. Therefore, it is generally recommended to trim out highly correlated features.

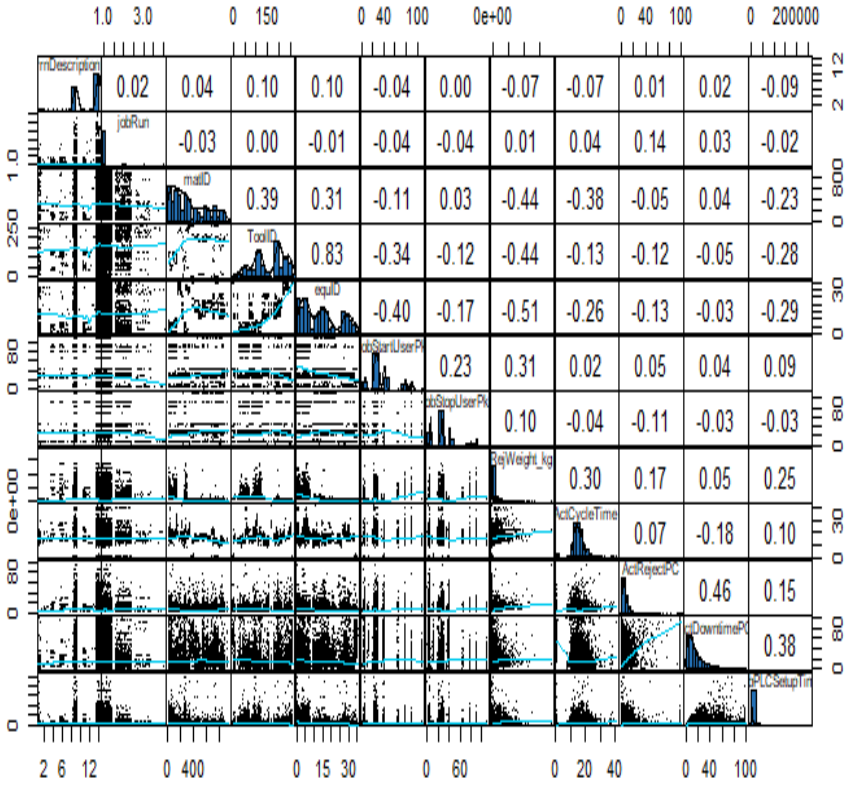


Fig. 4.5 Correlation matrix of all job features

With acknowledgement of the necessity to drop high correlation, a pairwise correlation test is conducted in Fig. 4.5. It suggests a high correlation between the tool used for the job ("*ToolID*") and the machine that the job runs on ("*EquID*"), while all other job features seem not to be highly correlated so these features will be kept. The occurred high correlation between the tool and the machine features makes sense because the tool of the job is the corresponding mould of its produced materials. The mould has to adapt to specific operating machines, making it highly correlated. Accordingly, one of the attributes needs to be

removed. To retain as much valuable information as possible, the variable with higher variation will be selected as it is likely to have higher predictive importance. One way to measure the variability for a categorical variable is entropy. This concept was introduced by Shannon [41] who has defined that for a categorical random variable X with the value of $\{x_1, x_2, \dots, x_n\}$ and probabilities $p(x_i)$, its entropy is formulated as

$$H(X) = -\sum_{i=1}^{n} p(x_i) \log p(x_i) \tag{4}$$

where $p(x_i)$ is the probability of value x_i, n is the total number of possible values, and H(X) represents the Shannon diversity index. The higher the index, the more variations there is within the variable. Fig. 4.6 below displays the distribution of the entropy of tools and equipment within different products in our data. It is obvious that equipment lies at a higher entropy level than the tool. It also encloses a larger area under the curve. This implies a higher variability in the equipment variable; therefore, the tool variable will be discarded.

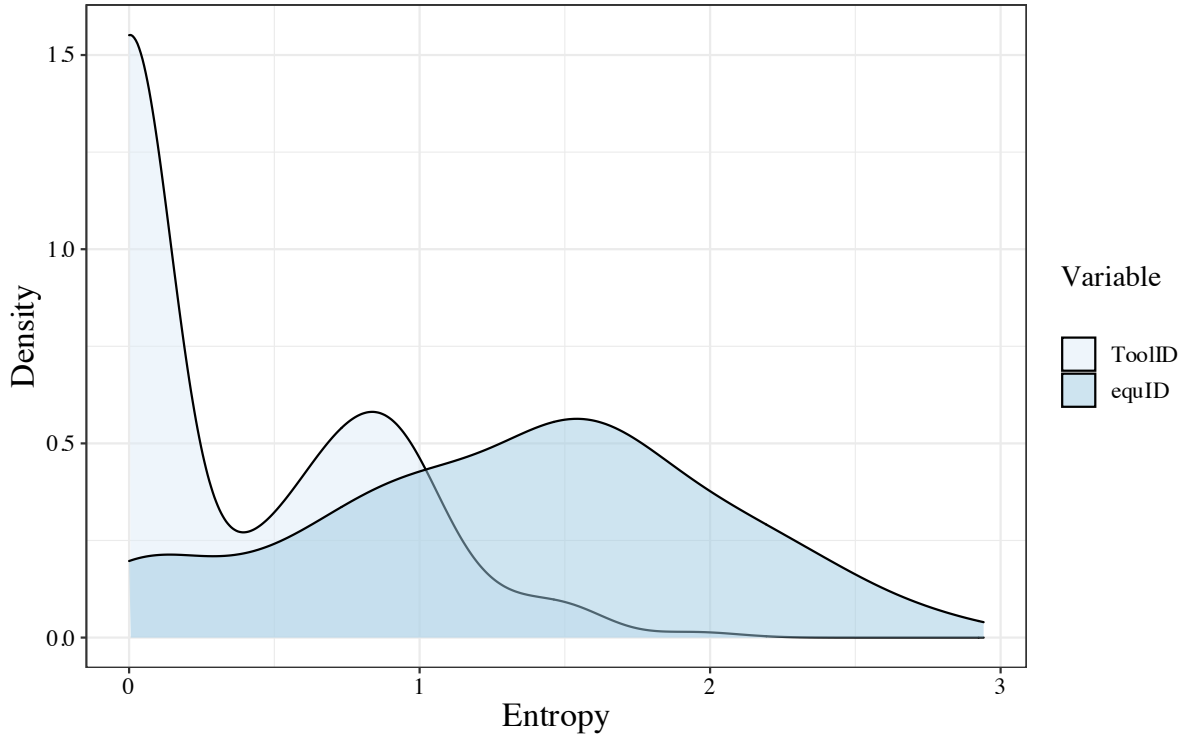


Fig. 4.6 Entropy distribution of tools and equipment

4.1.2.3 Data Discretisation

A broad background of Bayesian network structure learning theory and algorithms assumes all the random variables to be discrete [42]. However, it is common to have continuous and discrete variables coexist in practical circumstances [43]. In our case, there are both discrete variables such as equipment, Tool, and continuous ones (start time, downtime, etc.) in Manufacturing Execution Systems (MES) data. Whereas these continuous attributes in the practical operation do not always fulfil the gaussian hypothesis [44], which can violate the assumption of Gaussian Bayesian Network involving continuous parameters. In our case, most of the attributes are heavily skewed and do not follow a normal distribution. Therefore, data discretisation needs to be considered. Furthermore, data binning is widely practised to mitigate variable interactions preliminary to Bayesian network structure learning, since implicit complexity can be induced by the interactions and dependencies between continuous variables in the networks [45]. It is also easier for BN classifiers to handle discrete values thanks to their simplicity. As a result, data discretisation leads to a more efficient model development process. Lastly, one of the main goals of this research is to provide insights to assist factory workers with decision-making. Thus, the results must be interpretable for human beings to read and process. However, it is less likely for continuous attributes to correlate with the response variable because of infinite degrees of freedom. Hence, models with continuous features tend to be more difficult to interpret. After discretising the features, the resulting groups corresponding to the target can be explained more easily. Therefore, discretised Bayesian network is recommended with regard to the compatibility, efficiency, and interpretability of the resulting models even though discretisation may cause some loss of information.

There are discretisation algorithms that discretise the data automatically without any knowledge about the features in the data. However, the focus of our study is to support manufacturing staff to make data-driven decisions. Thus, the obtained features must be

interpretable, and the discretisation methods ought to be flexible toward distinct factories. Hence, the choice of discretisation policy is determined adaptively according to field-specific expertise. "*ActCycleTime*" and "*ActDownTime*" are aggregated into two groups, normal and abnormal as demonstrated in Fig. 4.7. For example, if a job's downtime exceeds the tolerance set by the manufacturer, then it will be classified as Abnormal. Moreover, additional information is extrapolated through discretisation of the Date Time variable inspired by expert knowledge. According to on-site staff with domain knowledge, the season and the shift during which production takes place also play a significant role in the quality of the job. Therefore, we have derived two categorical attributes, season and shift, from the job start time column. Finally, equal-width interval discretization is used to split reject weight ("*RejWeight_kg*") and job set-up time ("*jobPLCSetupTime*") in Fig. 4.7. This method is chosen due to its ability to preserve the probability distribution of each input variable. It divides the domain of a continuous variable x, into k intervals with a uniform width, where k is a predetermined parameter [46]. The width of the interval, W, is calculated by:

$$W = \frac{X_{max} - X_{min}}{k} \tag{5}$$

where $X_{max} = max\{X_1, X_2, \ldots, X_n\}$ and $X_{min} = min\{X_1, X_2, \ldots, X_n\}$.

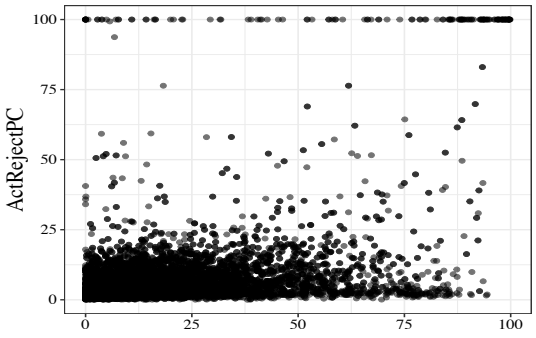


Fig. 4.7.1 (a) "*ActDowntimePC*" before binning

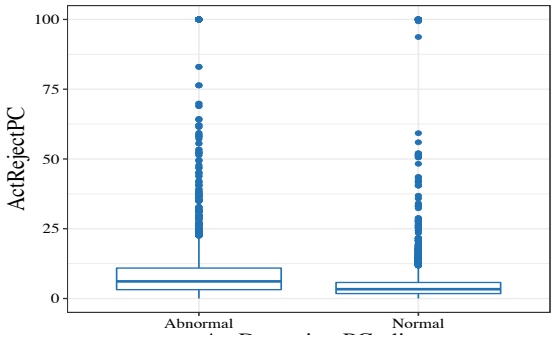


Fig.4.7.1 (b) "*ActDowntimePC*" after binning

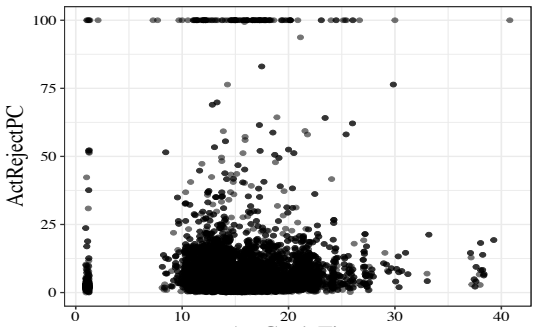


Fig. 4.7.2 (a) "*ActCycleTime*" before binning

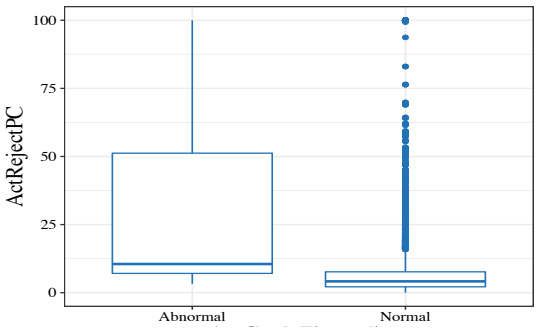


Fig.4.7.2 (b) "*ActCycleTime*" after binning

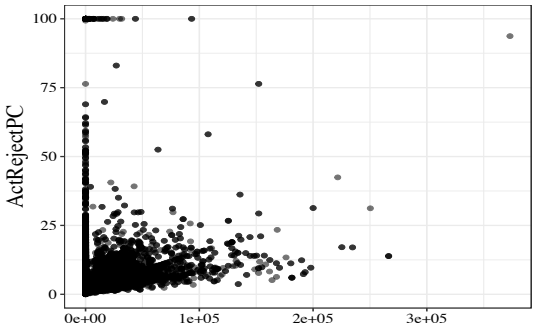


Fig. 4.7.3 (a) "*RejWeight_kg*" before binning

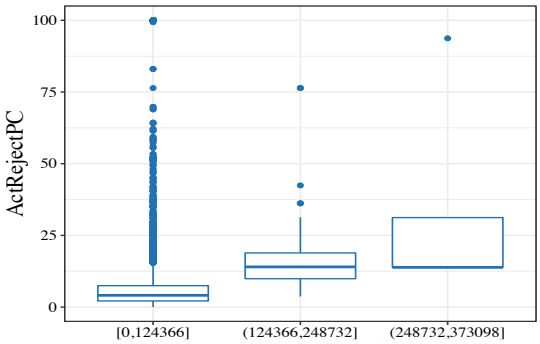


Fig.4.7.3 (b) "*RejWeight_kg*" after binning

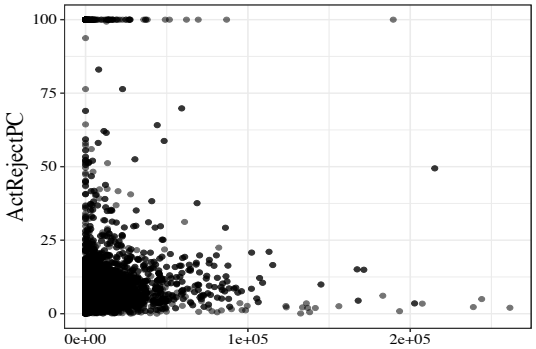


Fig. 4.7.4 (a) "*jobPLCSetupTime*" before binning

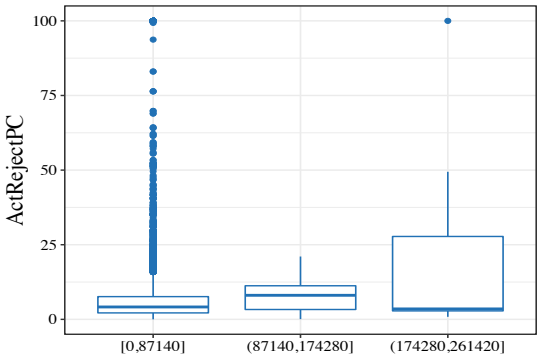


Fig.4.7.4 (b) "*jobPLCSetupTime*" after binning

Fig. 4.7 Relationships between "ActRejectPC" and four quantitative features

4.2 Prediction Model Construction

Once the job features are prepared, ensembled BN models can be built to perform RCA. This section will demonstrate how ensembled BN models are constructed and conducted based on a case study, which will be described in Section 4.2.1. We will follow the workflow of the proposed RCA framework introduced in Section 3.3 to explain the development of the functional modules in the proposed methods and to solve the questions raised in Section 3.1. The answers to the questions will be presented concisely at the end of this section. Firstly, the bagging ensemble learning technique is illustrated in Section 4.2.2 so that it can be incorporated into the following procedure of BN structure learning to ensure model robustness. Then structure learning is undertaken in Section 4.2.3 to unveil the causal relationships from the historical job records and to construct graphical models accordingly. The learnt causal graphs of BN offer human-interpretable knowledge representation and aid to approach Q1. Different structure learning methods and knowledge sources are used to learn BN structures so that various BN models can be obtained and fused later using WAEL technique to circumvent the existing deficiencies in accuracy and stability in a single BN model. After that, the parameters of BN are learnt by Bayesian parameter estimation in Section 4.2.4 to allow Bayesian inference. Consequently, Bayesian inference is developed in Section 4.2.5 to infer the probabilities of reject root causes as well as the classification of quality risk for a job. Lastly, the predictions from different BN models are combined using WAEL technique in Section 4.2.6, resulting in a set of root-cause probabilities for Q2 and a single quality risk prediction for Q3. In this way, the proposed method answers the three defined RCA questions, summarised in Section 4.2.7. And it provides a robust, intelligent and human-interpretable probabilistic reasoning method for RCA.

4.2.1 Description of the Case Study

A case study has been used to showcase the development of the proposed method. It also provides an example to demonstrate how and how well the proposed method answers the three questions raised in Section 3.1. The job features, historical production records, and potential root causes related to the case are described in this section. The observed ground truth of the case regarding the three questions is also presented, serving as a benchmark for the predictions obtained by the proposed model.

Selected Case:

A job with ID '5077' producing the product 'ABC500' is selected as the testing case for the problem (i.e., J_n = 5077, $Pd_m = $ 'ABC500')

Job Features X_{5077}:

Table 4.2 Job Features X_{5077} for job '5077'

Tool ID	Equip ID	Season	Shift	Job Start User	Job Stop User	Is RejectFail R_{5077}	Act Downtime PC	RejWeight_ kg	Job Setup Time
1216	BM01	Autumn	Night	25	29	1	Abnormal	150306	70200

Historical records:

The proposed method will uncover the causal relationships from the historical job records that have also produced the product 'ABC500', D_{ABC500}. It has been found that the product with ID 'ABC500' has a historical record of 30 jobs, among which 6 are problematic with high reject rates. This case will follow the proposed product-wise RCA framework of ensembled BN to illustrate the model construction process and to answer the three questions.

Ground truth for the three questions that we are trying to solve:

1. It is hard to define the ground truth for a causal graph, however, a causal network solely suggested by the experts at the plastic plant has been obtained as a reference in Fig. 4.8.

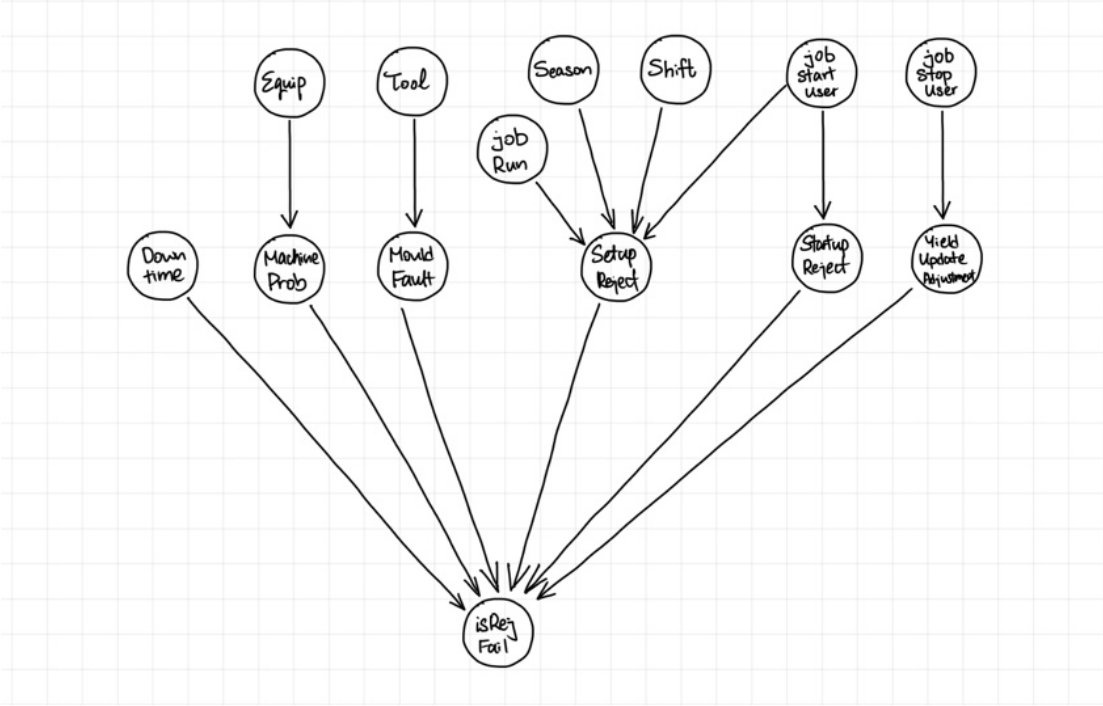


Fig. 4.8 Manually constructed BN

2. True probabilities of different reject reasons causing the quality issues are listed below.

   Setup Reject: 0.08
   Yield Update Adjustment: 0.92
   i.e. $Y_{5077} = \{SetupReject,\ YieldUpdateAjustment\}$
   $P_{5077} = \{0.08,\ 0.92\}$

3. This job is a problematic job with quality issues, so the result we are expecting in prediction is "*isRejectFail*" = 1 ($R_{5077} = 1$).

4.2.2 Bagging Ensemble Learning for BN

Bagging ensemble learning technique is developed first so that it can be used to reinforce the structure learning process of BN in the following section. It is the first functional module in the proposed RCA framework. Much research has proven the advantage of bagging EL in improving overall performance [47]. Li et al. [26] integrated the bagging method with three different BN learning algorithms, revealing bagging EL's excellent generalization capability and its stable performance among different BNs. Another study has shown that the bagging learning method outperforms the single classifier widely as well as the boosting method [48]. Bagging technique accounts for sample variations and reduces the chance of a poor BN model induced by sparse data, enabling robust BN structure learning. As a result, the bagging method is chosen in our project. Bagging is also known as bootstrap aggregating [49]. It resamples the original data with replacement and implements homogenous learners on the varying resulting samples [48]. Eventually, the predictions are aggregated by average voting. Following bagging procedures, we bootstrapped 10 subsets from the training data D_{ABC500} of the target product (i.e., S = 10). Then, BN structure learning algorithms are performed on each of the 10 samples, $\{D_{ABC500_1}, D_{ABC500_2}, \dots, D_{ABC500_{10}}\}$. Ultimately, the learnt structures are combined by averaging the arc strength of each BN model, resulting in a bagging ensembled Bayesian structure.

Algorithm 1 Bayesian Network Structure Learning with Bagging Ensemble

---

Input: Training data for product Pd_m, D_m, Structure learning algorithm A_k;
Bootstrap rounds, S.

Output: Bagged Bayesian Structure $G_{m,k}$.

for *each* D_{m_s} in $\{D_{m_1}, D_{m_2}, \dots, D_{m_S}\}$: **Step 1 Bootstrapping**

  $G_{m_s,k} = A_k(D_{m_s})$ **Step 2 Training**

end

$G_{m,k} = \frac{1}{S}\sum_{s=1}^{S} G_{m_s,k}$ **Step 3 Aggregating**

Following the bagging ensemble process in Algorithm 1, the effect of bagging ensemble learning is visualized in the case study as shown in Fig. 4.9. The network on the right is learnt on a single sample, while the structure on the left is the aggregated model of 10 Bayesian networks learnt from varying bootstrapped training sets. As the red lines indicated, many arc directions are reversed in the bagged BN. More importantly, the link between reject weight ("*RejWeight_kg*") and season ("*Season*") is eliminated in the bagged structure. This could be because the sample in a single run can coincide to have a dependency between these two nodes by chance. However, the bagging ensemble can account for sampling variability and impair the bias that could occur in a single sample, resulting in a more representative structure.

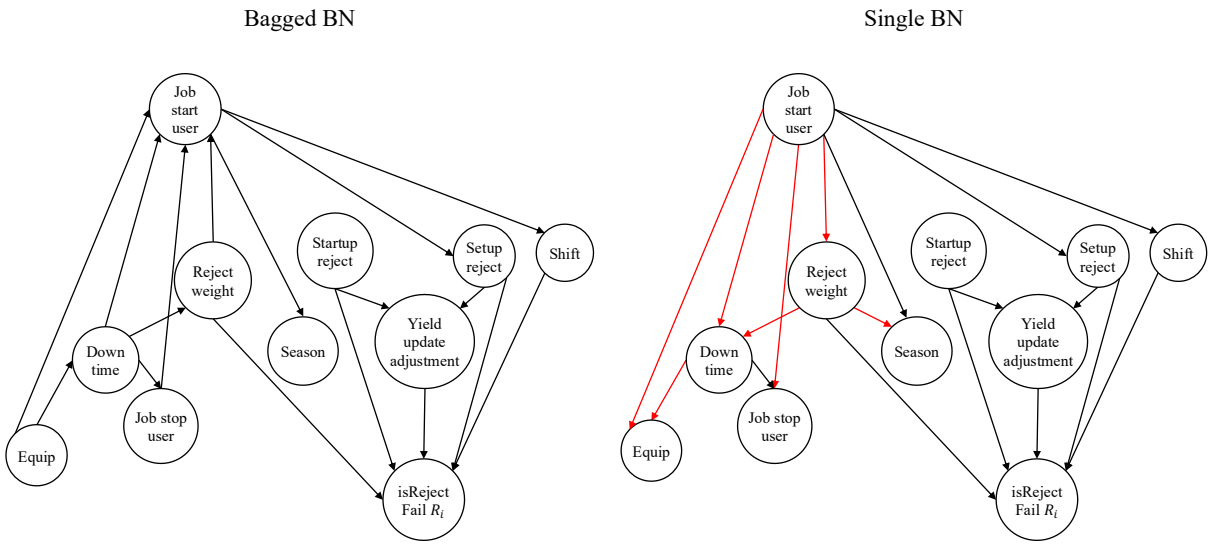


Fig. 4.9 Comparison between a single BN and a bagged BN using tabu search with hybrid knowledge for product 'ABC500'

4.2.3 Structure Learning

The structure of BNs will then be learnt by adopting the bagging ensemble technique to provide a causal network that describes the causal relationships in the production system of the industry. These learnt causal graphs help answer Q1 of the questions raised in Section 3.1. They also contribute to one of the elements of the objectives – to be human-interpretable because the causal graphs visualise the causal relationships in the production system. This section will present the causal structures learnt from the historical production data of the case study. Moreover, different knowledge sources and structure learning algorithms are used to learn the BN structures. This is to allow the weighted average ensemble learning to fuse the predictions from the structures learnt by different learning algorithms to ensure a robust solution. The performances of the individual structure learning algorithms and knowledge sources can also be compared to the performance of the proposed ensembled BN in Chapter 5. Therefore, this section will demonstrate how BN models can be learnt by different types of knowledge sources and structure learning methods.

4.2.3.1 Knowledge Sources

This section describes different knowledge sources from which the BN structures are learnt. In the context of BN, knowledge represents the causal relationships between different variables v_i. When a causal relationship exists between v_i and v_j, an arc linkage a_{v_i,v_j} is established in the corresponding BN model, $G = (V, A)$. a_{v_i,v_j} can be referred to as causal knowledge. If the arc linkage a_{v_i,v_j} in the network is known ahead of structure learning, a_{v_i,v_j} is called prior knowledge. They are normally discovered by human instinct or experience. Structure learning is the process of deducing the structure of BN from the

dependency relations in the data, aided by any prior knowledge as constraints [5]. In our experiment, there are three different knowledge sources that the BN can learn from, namely "Data", "Hybrid" and "Human". When a structure is purely learnt from the data without any prior knowledge, its knowledge source is labelled as "Data"; When the entire structure is solely built on human knowledge without any structure learning algorithm, it is classified as "Human" knowledge source; When structure learning algorithms learn the BN structure with prior knowledge, it is defined as a "Hybrid" knowledge source as it involves both prior knowledge obtained from human and structure learning from the data. In this study, the prior knowledge is found by the data analysts in the factory presented in Fig. 4.10. It points to the variables that embody the potential reject reasons, towards the quality failure node, abbreviated as "*isRejectFail*". These arcs reflect the directed causality from root-cause variables to quality failure.

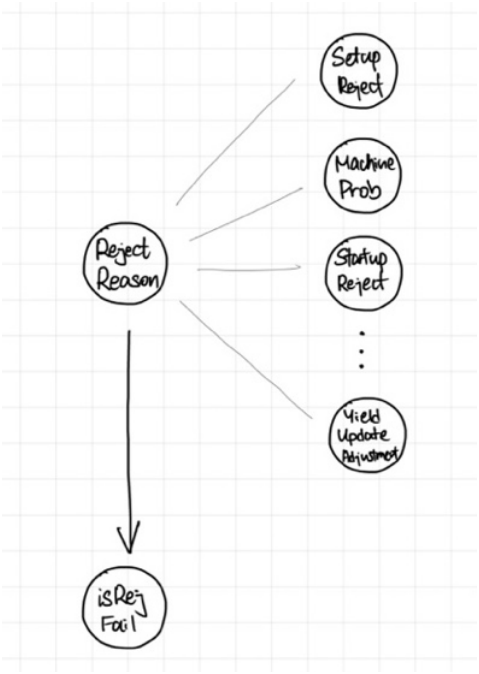


Fig. 4.10 Hybrid knowledge indicating directed causality from root-cause variables to quality issues

Furthermore, a BN constructed purely from human knowledge is obtained in Fig. 4.11. The causal relationships in the graph are elicited by the on-site experts from the plastic industry. It implies some causal influences from job features to the root-cause variables (e.g., the type of "*Shift*" can cause the occurrence of "*Setup Reject*"). As a result, we will build BNs using these three different knowledge sources. Particularly, "Data" and "Hybrid" knowledge sources can be integrated with different structure learning algorithms. A comparison will also be conducted in chapter 5 to evaluate the performance of BN structures built from different knowledge sources. For demonstration purposes, the following Section 4.2.3.2 will only present the structure learnt from hybrid knowledge for each of the structure learning algorithms that have been used.

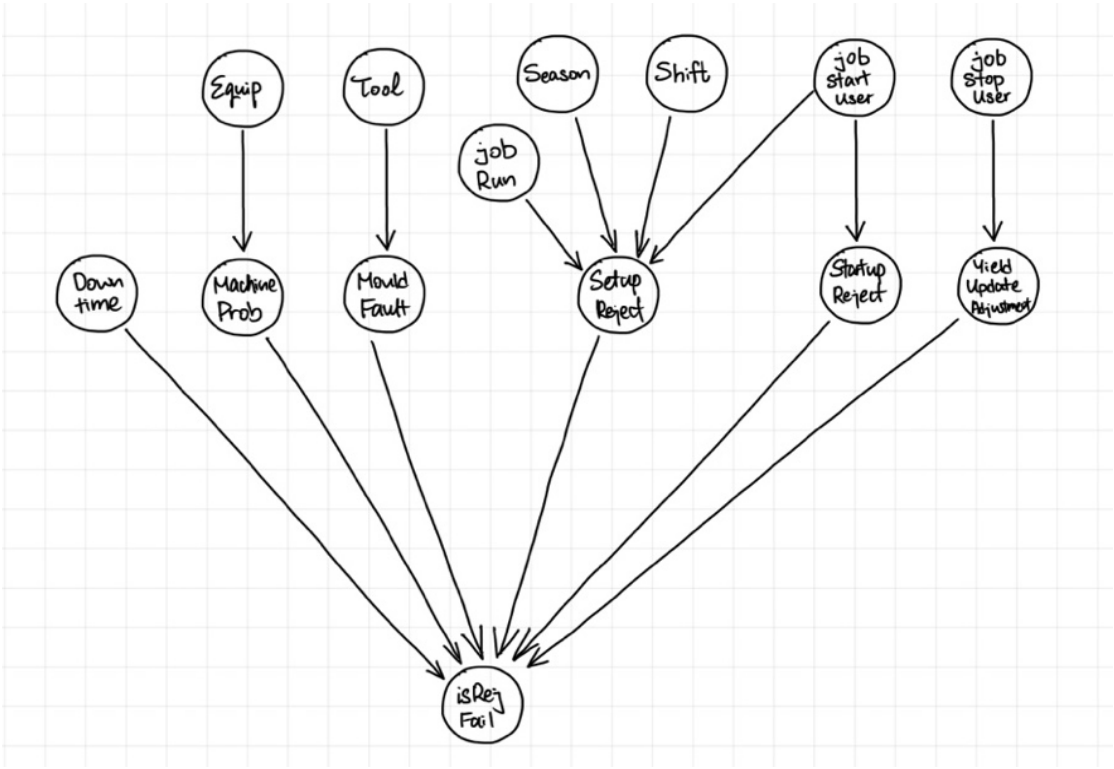


Fig. 4.11 Structure purely learnt from Human knowledge indicating directed causality from tail node to head node

4.2.3.2 Structure Learning Algorithms

In BNs, predicted probabilities are inferred based on a causal structure. However, the structure of Bayesian network is not often known. In the context of manufacturing, normally only a few arcs, representing causal relationships, in the network could be established based on expert knowledge. To achieve a more intelligent and efficient RCA process, the structure needs to be learnt automatically. Therefore, structure learning algorithms are used to automatically deduce the structure of BN from the dependency relations in the data, aided by any prior knowledge as constraints [5]. The task of learning the structure of a BN from a dataset of D_m for Product Pd_m can be defined as determining a set of directed arcs A for the DAG $G = (V, A)$ to achieve some criterion used for evaluating the goodness of fit of the model. As formulated in Section 3.2, V denotes vertices ($v_i \in V$). Here it stands for the variables selected from the historical job records, including the job features X , root-cause variables Y and quality risk indicator R, $V \subseteq \{X, Y, R\}$. Since this research aims for a robust RCA, different structure learning algorithms will be explored to evaluate and compare their performances. Furthermore, the results of different learners will be optimised through the weighted average ensemble to overcome the existing shortages of model stability in a single BN learning algorithm (explained in Section 4.2.6). This study has categorised the BN structure learning approaches into four groups: score-based learning, constraint-based learning, hybrid learning, and pairwise mutual information algorithms. And each learnt causal graph can serve as an answer for Q1.

4.2.3.2.1 Score-based Learning

The score-based approach construes the problem of structure learning as an optimization problem [50]. These algorithms score each BN candidate based on a predefined function. Then, the heuristic search algorithm will be applied to search for the structure with the maximal score. The scoring function that numerically measures the fit of models and the

search algorithm to traverse the search space of possible models are the two essential elements of score-based learning. In this study, we have chosen the Bayesian Information Criterion (BIC) as the scoring function. It approximates the posterior probability of the network candidate based on the available data with the assumption of a uniform prior probability distribution on the entire search space [51]. The lowest BIC is selected by the metrics, whose mathematical formulation is computed in equation (6). The first term with the log-likelihood for the model expresses the accuracy (i.e., how well the model fits the training data). The second regularization term penalizes the complexity of the model in favour of sparsity to avoid overfitting data. In combination, the BIC metric favours accurate and simple models [52].

$$BIC(G;D) = -\log P(D|\Theta) + \frac{d}{2}\log n \tag{6}$$

where D denotes the data, Θ signifies the parameters of the models, d is the number of parameters in the structure, n is the sample size.

For the heuristic search of the optimal structure, Hill-Climbing (HC) and Tabu Search (TS) algorithms have been considered in the experiment. They both adopted the greedy search technique for the steepest descent. HC is a local greedy heuristic that repeatedly performs single-edge manipulations to the structure and accepts the changes that improve BIC. Random restarts are implemented to avoid the search space being trapped in local optima. On the other hand, tabu search is a metaheuristic that directs a local greedy search out of local optima using a memory list. It monitors the progress of the search and stores the search history to adaptively modify the permitted neighbourhood networks to guide the solution space into more promising areas of interest, in particular, beyond local optimality. The search is directed by making tabu (i.e., banning) the moves that have been performed recently recorded on the list. The implementation of the two selected score-based structure learning algorithms follows the logic presented in Algorithm 2.

Algorithm 2 Score-based Structure Learning [53]

Input: a training dataset D_m, an initial empty DAG G, a score function $BIC\ (G, D_m)$.
Output: the DAG G_{max} that optimises $BIC\ (G,\ D_m)$.

Step 1 Score: Compute the score of G, $S_G = BIC\ (G, D_m)$, and $S_{max} \leftarrow S_G\ ,\ G_{max} \leftarrow G$.
Step 2 Hill-Climbing:
while $S_{max} > S_{max}'$ **do**
# compute the score for every possible arc addition, deletion or reversal in G_{max}
for *each* $O \in \left\{A(G_{max})\backslash\{a_{v_i,v_j}\}, A(G_{max}) \cup \{a_{v_i,v_j}\}, re(A(G_{max}))\right\}$ **do**
Arc operation on $G, G^* = O(G)$
$S_{G^*} = BIC(G^*, D_m)$
if $S_{G^*} > S_{max}$ and $S_{G^*} > S_G$ **then**
$G \leftarrow G^*,\ S_G \leftarrow S_{G^*}$
end
end
if $S_G > S_{max}$ **then**
$S_{max} \leftarrow S_{max}',\ S_{max} \leftarrow S_G,\ G_{max} \leftarrow G$
end
end
Step 3 Tabu Search:
# pick the DAG with highest S_G that's been unvisited in the last t_1 steps regardless of S_{max}
for $t < t_0$ **do**
while $S_{max} > S_{max}'$ **do**
for *each* $O \in \left\{A(G_{max})\backslash\{a_{v_i,v_j}\}, A(G_{max}) \cup \{a_{v_i,v_j}\}, re(A(G_{max}))\right\}$ **do**
Arc operation on $G, G^* = O(G)$
$S_{G^*} = BIC(G^*, D_m)$
if $S_{G^*} > S_{max}$ and $S_{G^*} > S_G$ **then**
$G \leftarrow G^*$ *such that* $G^* \notin TL(t_1)$
$S_G \leftarrow S_{G^*}$
end
end
if $S_G > S_{max}$ **then**
$S_{max} \leftarrow S_{max}',\ S_{max} \leftarrow S_G,\ G_{max} \leftarrow G$
Update TL
end
end
end
Step 4 Random Restart:
for 1: r **do**
perturb $G' = O(G_{max})$ such that
$O \in \left\{A(G_{max})\backslash\{a_{v_i,v_j}\}, A(G_{max}) \cup \{a_{v_i,v_j}\}, re(A(G_{max}))\right\}$
end
Search from **Step 2**
Return G_{max}

The structures using the score-based learning method from our example are obtained in Fig. 4.12. The graph shows that the skeletons learnt by HC and TS are the same despite the orientation of the two arcs highlighted in red. This makes sense since the two algorithms resemble each other by both using the greedy algorithm with only the difference in the memorised list. The learnt causal graphs express some causal relationships with the tail node being the causes and the head node being the root. For example, "*Shift*" → "*isRejectFail*" indicates that the states of shift (i.e., day shift or night shift) can contribute causally to the quality issues. By interpreting the learnt graphs, Q1 can be answered.

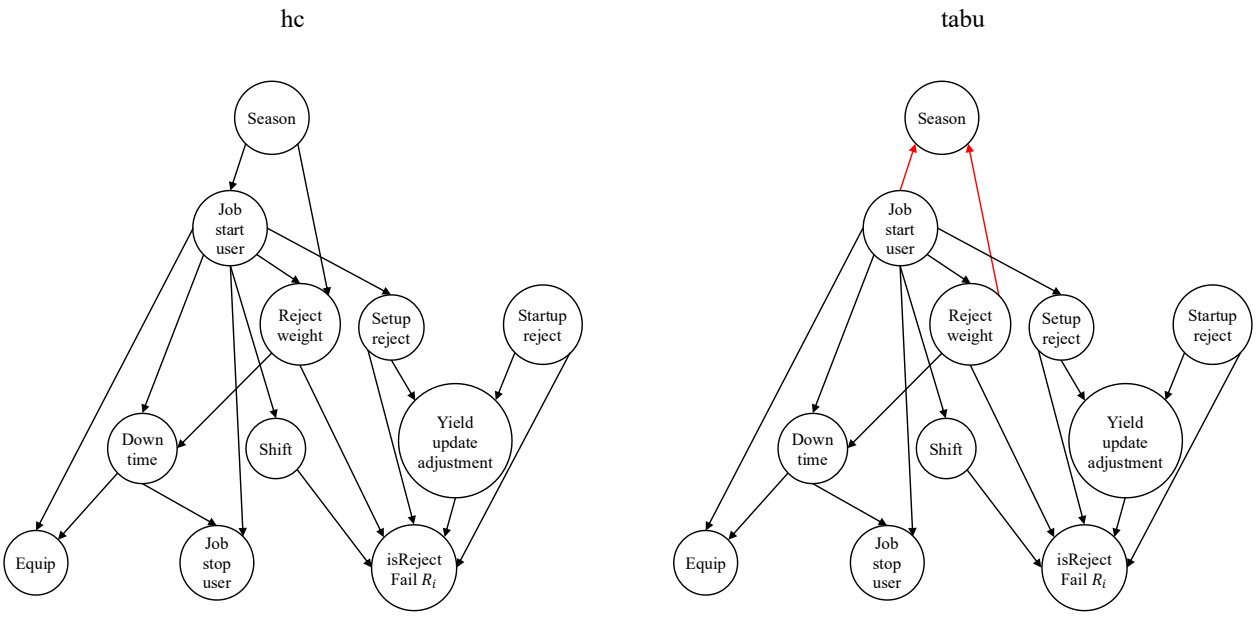


Fig. 4.12 Bagged BN structures learnt by hill-climbing and tabu search using hybrid knowledge for product 'ABC500' showing high similarity except for reversed arc direction highlighted in red

4.2.3.2.2 Constraint-based Learning

The constraint-based structure learning algorithms learn independence within the network structure by performing a series of conditional hypothesis tests between pairwise variables. Then the DAG will be constructed according to the inductive causation (conditional independence tests). There are a few constraint-based learning algorithms selected for structure learning in our experiment, the Peter and Clark stable algorithm (PC.stable), Grow-Shrink (GS) algorithm and Incremental Association Markov Blanket (IAMB) algorithm. PC.stable algorithm is the modern adoption of the first practical implementation of causal graphical models originated by Pearl (1991). The algorithm requires the ordering of the edges to consider the correct links early in the search in avoidance of an exhaustive search. The process of PC.stable implementation is encoded in Algorithm 3. First, it starts with a fully connected undirect graph. Then inductive causation is conducted heuristically in Step 2 to test if there is any (conditional) independence between any pair of the adjacent nodes in ascending order. If the hypothesis is accepted, the correspondent edge will be removed from the structure. Step 3 identifies the v-structures among all the pairs of non-adjacent nodes with a common neighbour. In the end, it sets the directions of all other arcs in a way to satisfy the acyclicity constraint. The BN structure learnt by PC.stable algorithm based on historical data D_{ABC500} is shown in Fig. 4.13. Different from the structures learnt by score-based algorithms, the structure learnt by PC.stable algorithm is quite sparse. No job feature is linked to the quality failure node, indicating that the quality issues are only caused by the root-cause variables.

Algorithm 3 The Peter and Clark Stable Algorithm

Input: a training data set D_m, a conditional independence test $Test(v_i, v_j \mid S;\ D_m)$,
Output: Partially oriented DAG G.

Step 1 Initialisation: a complete undirected graph G
Step 2 Inductive Causation:
for *each* pair (v_i, v_j), $i \neq j$, $order(v_i) = order(v)_{min}$ **do**
$Test\ (v_i, v_j \mid S;\ D_m)$ # conditional independence test on v_i and v_j
# if v_i and v_j are independent given set S, set S as the separating set of (v_i, v_j),
if $v_i \perp\!\!\!\perp v_j \mid S$ **then**
$S_{v_i v_j} \leftarrow S$
$G \setminus \{v_i - v_j\}$
# move onto the next pair if there is no other subset S with the same ordering
else
if $\exists\ S$ such that $order(S) = order(v_i)$ **then**
go to $Test\ (v_i, v_j \mid S;\ D_m)$
end
end
end
Step 3 Learn Arc Directions:
for *each non-adjacent* (v_i, v_j), $\exists v_k$ such that $v_i - v_k$, $v_j - v_k$ and $v_k \in / S_{v_i v_j}$ **do**
# Replace the triplets with a v-structure
$\{v_i - v_k - v_j\} \leftarrow \{v_i \rightarrow v_k \leftarrow v_j\}$
end
# If v_i is adjacent to v_j and there is a strictly directed path from v_i to v_j,
if $v_i - v_j$ and $\exists\ path(v_i \rightarrow v_j)$ **then**
$\{v_i - v_j\} \leftarrow \{v_i \rightarrow v_j\}$ # to avoid introducing cycles
end
if $\neg\ (v_i - v_j)$, $v_i \rightarrow v_j$ and $v_k - v_j$, **then**
$\{v_k - v_j,\} \leftarrow \{v_k \rightarrow v_j\}$
end
Return G

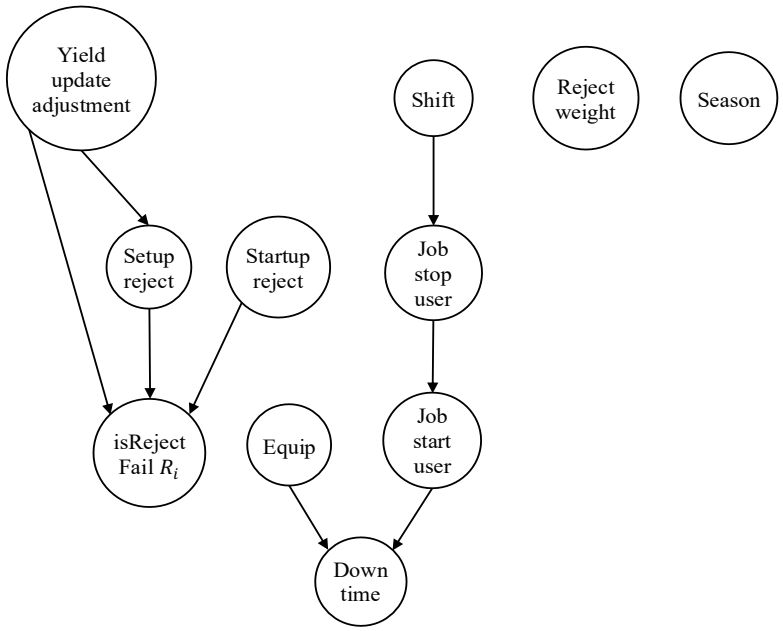


Fig. 4.13 A bagged BN learnt by PC.stable algorithm with hybrid knowledge for product 'ABC500'

Grow-shrink algorithm and incremental association Markov blanket algorithm follow a similar procedure as the PC.stable algorithm, except that it utilises the Markov blankets [6] to reduce the number of candidate DAGs early on so that quicker heuristics is achieved. Grow-shrink algorithm starts structure learning by resolving the Markov blanket for each node $MB(v_i)$ in order. It is achieved in two phases, grow phase and shrink phase. In grow phase, if v_i and v_j are dependent, v_j is added to the blanket of v_i, and shrink phase removes extraneous variables that are supposed to be outside the blanket as shown in Algorithm 4. The rest of the algorithm follows the regular routine of the constraint-based learning approach. It looks for a d-separating set for every variable pair. Then, the edges are oriented whenever the shared neighbour creates a dependency satisfying the DAG conditions.

Algorithm 4 Grow-Shrink Algorithm

Input: a training data set D_m, a conditional independence test $Test(v_i, v_j \mid S; D_m)$,
$MB(\cdot)$: the Markov blanket information for node $v_i \in V$

Output: Partially oriented DAG G.

Step 0 Learning Markov blankets:
for *each* variable v_i **do**
  learn Markov blanket $MB(v_i)$:
  while $\exists\ v_j \in V \backslash \{v_i\}$ such that $I(v_i, v_j \mid MB(v_i))$ **do**
    $MB(v_i) \cup \{v_j\}$ **[Grow Phase]**
  end
  while $\exists\ v_j \in MB\{v_i\}$ such that $I(v_i, v_j \mid MB(v_i) \backslash \{v_j\})$ **do**
    $MB(v_i) \backslash \{v_j\}$ **[Shrink Phase]**
  end
end
The rest follows **Algorithm 4** The Peter and Clark Stable Algorithm

The BN structure learnt by GS algorithm based on historical data D_{ABC500} is shown in Fig. 4.14. The structure becomes slightly denser than the one from pc algorithm. Extra job features such as "*Season*" and "*Reject weight*" have direct causality towards quality issues.

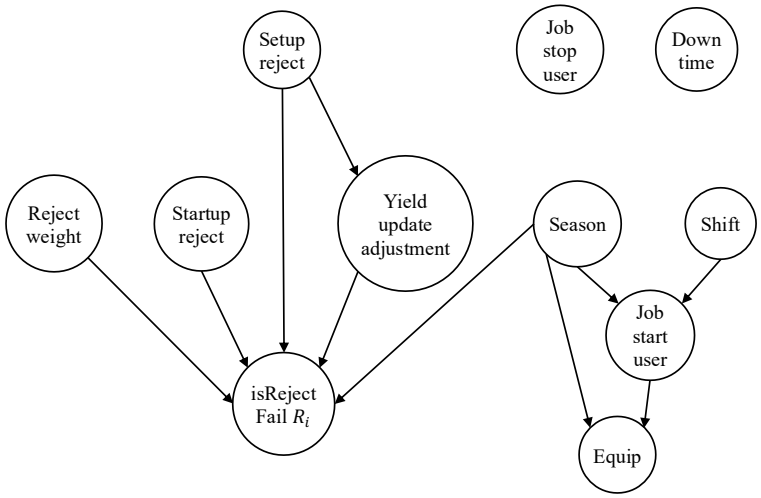


Fig. 4.14 A bagged BN learnt by grow-shrink algorithm with hybrid knowledge for product 'ABC500'

IAMB algorithm resembles GS learning with two phases of Markov blanket discovery, a forward phase and a backward phase. With the application of the Markov blanket, IAMB restricts the subset of variables to test for independence in advance, enabling more efficient

learning. For each variable v_i, the algorithm keeps track of the hypothesis set of v_i, signifying the potential set of parents of v_i. In the forward phase, it finds the variables with a strong association with v_i to be added to $MB(v_i)$. The dependence is measured by an information-theoretic heuristic function called Conditional Mutual Information (CMI). Consequently, it determines the variables in the Markov blanket. The backward phase discards all the false positive variables from the hypothesis set, leaving the true $MB(v_i)$. The implementation of IAMB described above is encoded into pseudo-code in Algorithm 5, and the resulting BN structure is presented in Fig. 4.15. From the BN structures learnt by the constraint-based algorithms, it has been found that the networks are generally sparse with fewer links than the score-based algorithms. This could be a result of the pairwise dependence test with the restriction of acrylic graphs. Furthermore, the structures learnt by the constraint-based algorithms contain undirected arcs. In this study, the orientation is automatically extended to satisfy the constraints of DAGs.

Algorithm 5 Incremental Association Markov Blanket Algorithm

Input: a training data set D_m, a conditional independence test $Test(v_i, v_j \mid S;\ D_m)$,
$MB(\cdot)$: the Markov blanket information for node $v_i \in V$
Output: Partially oriented DAG G.

Step 0 Learning Markov blankets:
Forward Phase: Add true positives to MB(T)
while $MB(v_i)' = MB(v_i)$ **do**
  $MB(v_i)' = MB(v_i)$
  Find v_{max} in $V \backslash MB(v_i) \backslash \{v_i\}$ such that maximizes $CMI(v_{max};\ v_i \mid MB(v_i))$
  if $v_{max} \perp v_i \mid MB(v_i)$ **then**
    $MB(v_i) = MB(v_i) \cup \{v_{max}\}$
  end
end
Backward Phase: Remove false positives from MB(T)
for *each* $v_j \in MB(v_i)$ **do**
  if $v_j \perp v_i \mid MB(v_i) \backslash \{v_j\}$ **then**
    $MB(v_i) = MB(v_i) \backslash \{v_j\}$
  end
end
The rest follows **Algorithm 4** The Peter and Clark Stable Algorithm

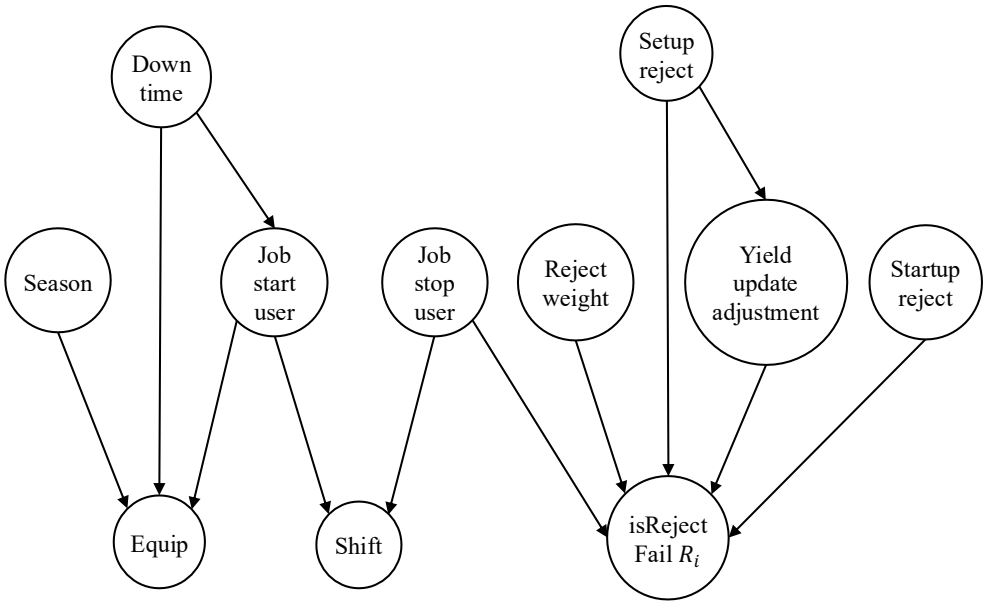


Fig. 4.15 A bagged BN learnt by IAMB algorithm with hybrid knowledge for product 'ABC500'

4.2.3.2.3 Hybrid Learning

Max-Min Hill-Climbing (MMHC) algorithm is adopted in this experiment for structure learning. It is a hybrid algorithm that marries the ideas from both constraint-based and search-and-score techniques. Max-min hill-climbing algorithm begins with reconstructing the skeleton of a BN using a local discovery algorithm, Max-Min Parents and Children (MMPC) to restrict the search space demonstrated in Step 1 Algorithm 6. After that, the direction of the edges is determined by a hill-climbing algorithm, a greedy Bayesian-scoring search. This hybrid method offers the advantage of a sound skeleton identification process and parameter tuning [54]. The BN structure acquired by MMHC algorithm based on the historical records D_{ABC500} is demonstrated in Fig. 4.16. In this case, the structure tends to be separated into two subgraphs. One contains the job features and the other consists of potential root causes plus the "*Reject Weight*" feature. Indicating that the job features might not have strong causal influences on quality issues.

Algorithm 6 Max-Min Hill-Climbing Algorithm

Input: a training data set D_m, Max-Min Parents and Children algorithm $MMPC$.
Output: a Bayesian Structure G.

Step 1 Restrict Search Space:
for *each* node $v_i \in V$ **do**
$PC_{v_i} = MMPC\ (v_i, D_m)$
end
Step 2 Learn Arc Directions:
Initialisation: an empty graph G
while $S_{max} > S_{max}'$ **do**
for *each* $O \in \left\{A(G_{max}) \setminus \{a_{v_i,v_j}\}, A(G_{max}) \cup \{a_{v_i,v_j}\}, re(A(G_{max}))\right\}$ **do**
Arc operation on $G, G^* = O(G)$
$S_{G^*} = BIC(G^*, D_m)$
if $S_{G^*} > S_{max}$ and $S_{G^*} > S_G$ **then**
$G \leftarrow G^*,\ S_G \leftarrow S_{G^*}$
end
end
if $S_G > S_{max}$ **then**
$S_{max} \leftarrow S_{max}',\ S_{max} \leftarrow S_G,\ G_{max} \leftarrow G$
end
end

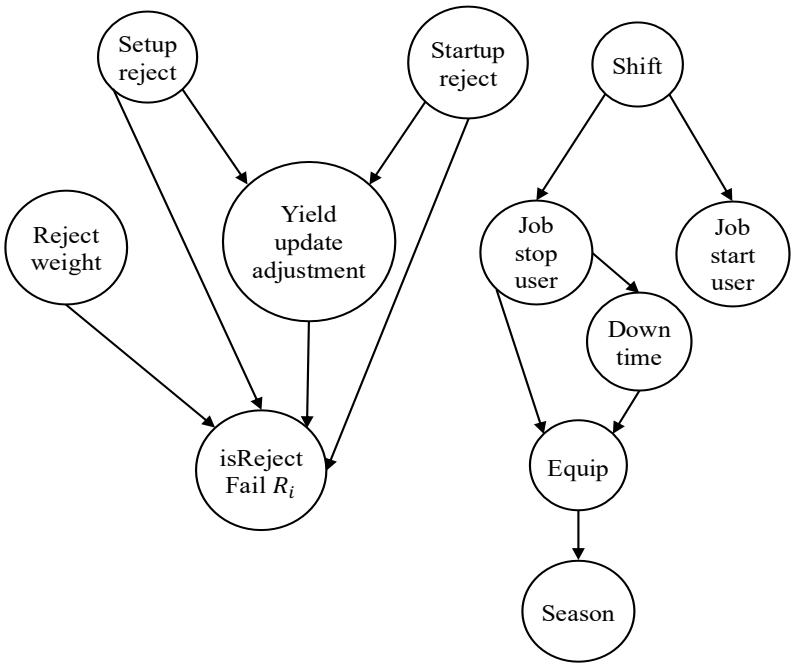


Fig. 4.16 A bagged BN learnt by max-min hill-climbing algorithm with hybrid knowledge for product 'ABC500'

4.2.3.2.4 Pairwise Mutual Information Algorithms

Finally, the study includes chow.liu structure learning algorithm to explore its performance on learning Bayesian structure. It is a tree search-based application that seeks the maximum-likelihood spanning tree structure closest to the true one. This approach is exceptionally time-efficient at finding the DAG thanks to its tree structure nature. The implementation process of chow.liu algorithm is encoded in Algorithm 7. Step 1 calculates pairwise mutual information measures between the variables. In Step 2, Kruskal's (KL) algorithm is applied to construct the Maximum Weight Spanning Tree (MWST). In the end, the root node is picked, and the direction of the edges is radiating outward from the root node. Fig. 4.17 shows the BN structure learnt by chow.liu algorithm. As expected, it exhibits a tree structure. All the nodes are connected in the graph. However, the arc direction between the quality issue node and the potential root causes seems to be problematic. For example, the arc pointing from "*isRejectFail*" to "*Startup Reject*" might not be reasonable. The learnt direction signifies that the quality issue node should lead to the root cause node, which does not align with common sense. Normally, the orientation should be reversed such that the rejects which occurred at the starting up period of a job can cause the job to have quality problems. The reason for the irrational directionality from chow.liu algorithm could be due to its last step of assigning directions, where a root node is picked randomly to blindly set all the edges points away from the root node. This step does not incorporate the statistical features from the data, making the arc orientation likely to be unrealistic.

Algorithm 7 Chow.liu Algorithm

Input: a training data set D_m, mutual information measure M(v_i , v_j), Kruskal's algorithm KL

Output: a Bayesian Structure G.

Step 1 Compute Pairwise Mutual Information:

for *each* pair (v_i , v_j), $i \neq j$ such that $order(v_i) = order(V)_{min}$ **do**

  $M(v_i , v_j)$

end

Step 2 Find MWST:

for *each* $a(v_i , v_j) \in A$ in $DES\left(w_{v_i , v_j}\right)$ **do**

  Find T_{opt} such that $KL(P, P_T) \leq KL(P, P_t)$

end

Step 3 Pick the Root Node:

$v_R \leftarrow \forall v_i$

for *each* $\{v_i - v_j\} \in A(T)$ **do**

  assign $\{v_i - v_j\}$ direction such that $\{v_i \leftarrow v_j \leftarrow v_R\} \mid \{v_j \leftarrow v_i \leftarrow v_R\}$

end

Return T

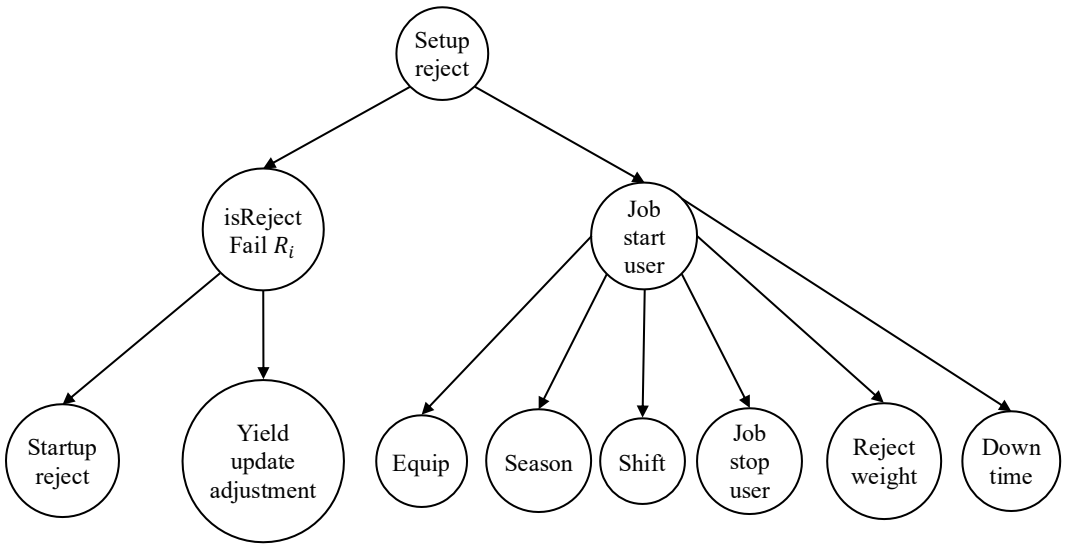


Fig. 4.17 A bagged BN learnt by chow.liu algorithm with hybrid knowledge for product 'ABC500'

4.2.4 Parameter Learning

After the structures of BN are learnt, the conditional probability tables between each pair of parent and child nodes in the network need to be obtained to allow probability inferencing. Parameter learning entails computing the CPTs of each node given its parent nodes. Implicitly, CPTs can be obtained by counting the occurrence of each state of the variables from the data. However, the implicit method becomes exponentially computationally expensive as the level of variable states increases. Hence, Bayesian parameter estimation is applied to estimate the values of CPTs in the experiment. $p(v_i)$ denotes the probability density function of an observable random variable v_i, reflecting the contribution to the quality issue in our case. With its distribution depending on the unknown parameter θ, $p(v_i|\theta)$ represents the prior probability density function for variable v_i, given θ. When new evidence $E = \{v_1, v_2, \dots, v_n\}$ is found for variable v_i in the experiment, the goal of parameter learning is to compute $p(v_i|E)$ so that its value approaches the unknown $p(v_i)$ as close as possible. It is noted that the parameter θ is modelled as a random variable following distribution $p(\theta)$. Then the probability density function of v_i given a set of evidence E can be inferred as follows:

$$p(v_i|E) = \int p(v_i, \theta|E)\, d\theta \tag{7}$$

$$= \int p(v_i|\theta, E)\, p(\theta|E)\, d\theta$$

$$= \int p(v_i|\theta)\, p(\theta|E)\, d\theta$$

As $p(v_i|\theta)$ is known before obtaining new evidence E, the posterior probability density function for parameter θ after E, $p(\theta|E)$, needs to be obtained. This is achieved by adapting Bayes' theorem as shown in equation (8).

$$p(\theta|E) = \frac{p(E|\theta)\, p(\theta)}{p(E)} = \frac{p(E|\theta)\, p(\theta)}{\int p(E|\theta)\, p(\theta)\, d\theta} \tag{8}$$

where $p(\theta)$ is the prior distribution, and $p(E|\theta)$ is the likelihood function. $p(\theta)$ represents the knowledge of the parameter prior to engaging the information from the data, while $p(\theta|E)$ updates the distribution succeeding the introduction of new evidence. Consequently, the probability distribution of node v_i can be estimated. For example, the manual structure has undertaken the Bayesian parameter estimation on 'ABC 500' data, and the estimated CPTs are shown in Fig. 4.18 below. The arcs in the network signify the existence of causality from the tail node to the head node. For example, the arc of "*DownTime*" → "*isRejectFail*" indicates that the states of "*DownTime*" (i.e., "normal" or "abnormal") have causal influences on the states of "*isRejectFail*" (i.e., "0"-normal or "1"-quality issue). For a node v_i, $v_i \in V$, if v_i has a parent, its CPT will be conditional probability distribution based on the states of its parent nodes $p(v_i|pa(v_i))$ shown in black tables in Fig. 4.18; else, its CPT will be its own probability distribution $p(v_i)$ illustrated in blue. These conditional probabilities also embody the strength of causal dependency relations between a pair of variables (v_i , v_j). For instance, $p(\text{isRejectFail} = 1|\text{DownTime} = \text{"abnormal"}) = 0.18$ implies that given the feature "*DownTime*" is "abnormal", it has a probability of 0.18 to cause the "*isRejectFail*" to be "1" (i.e., problematic with quality issues). These learnt CPTs can now be used in Bayesian inference to infer the probabilities of reject root causes.

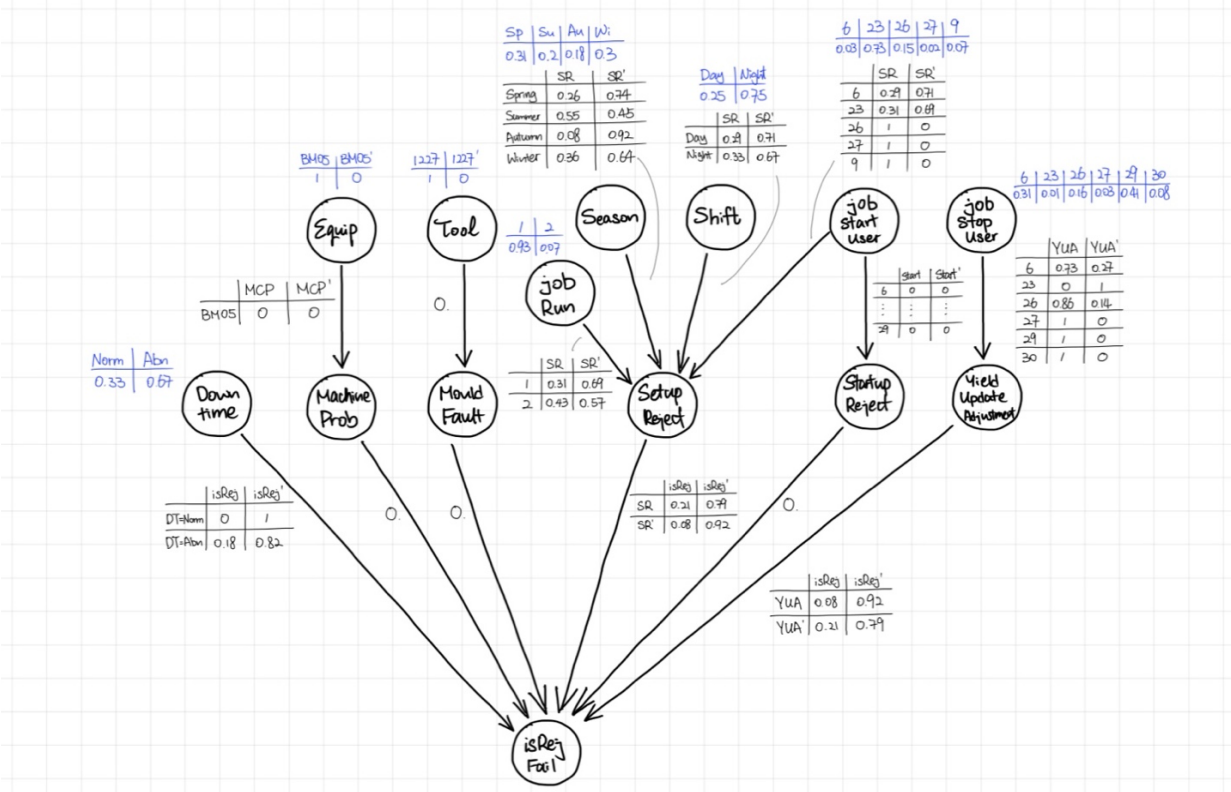


Fig. 4.18 Manual BN structure with parameters learnt from D_{ABC500} using Bayesian parameter estimation where each CPT represents the probability distribution of a node v_i.

4.2.5 Inference

Once the parameters of the BN models are learnt, Bayesian inference then needs to be performed to obtain the root-cause probabilities from the learnt BN models. The purpose of a Bayesian network is to enable the efficient computation of updated probability distributions for a set of events in the Bayesian network, given the evidence of the newly observed cases (e.g., the job feature X_{5077}). If our Bayesian network is a tree structure, then Belief Propagation (BP) can be applied to infer the probability of interest automatically. However, as displayed previously, most of the learnt structures are more complicated than trees. Therefore, Junction Tree (JT) algorithm needs to be implemented in our study to infer the probability of each reject cause from the complex structures that are learnt previously. The core idea of the junction tree algorithm is to turn a graph into a tree of clusters that are amenable to Belief Propagation. We start with a Bayesian structure with its corresponding

parameters learnt from above, and then undergo the following steps for a JT inference: (1) Moralize the graph (2) Triangulate the graph (3) Build a junction tree (4) Apply Belief Propagation. For presentation purposes, a simplified Bayesian network $G = (V, A)$ is used for demonstrating the process of JT algorithm.

Step 1 Moralisation

In this step, the parents of each node are connected as demonstrated in red in Fig. 4.19. Then the directionality of the arcs is dropped so a uniform treatment of directed and undirected graphs is possible.

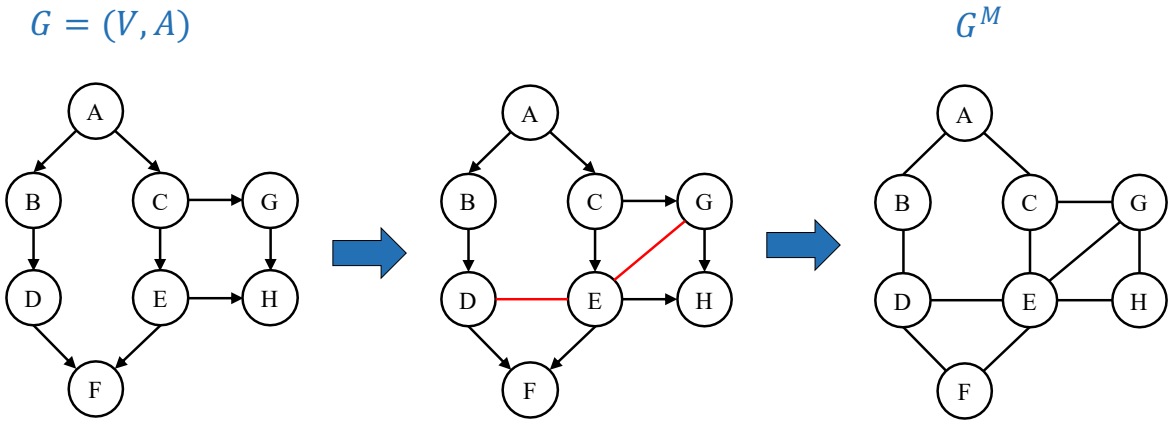


Fig. 4.19 Moralisation of BN to connect parents and undirect the graph, adapted from [55]

Step 2 Triangulation

Triangulation adds chords into the moral graph G^M such that any cycle of more than three vertices short in the graph is cut short. In Fig. 4.20, the cycle {A-B-C-D-E} is identified, and two chords A-D and A-E are added to avoid cycles with more than three vertices.

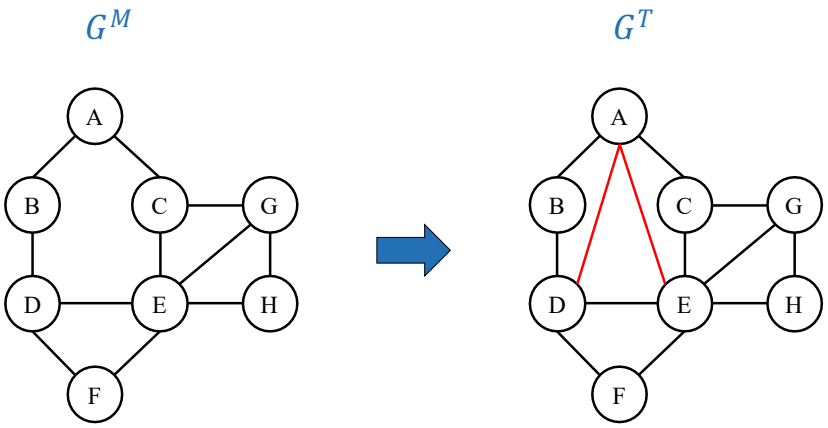


Fig. 4.20 Triangulation of BN to remove cycles with nodes ≥ 3, adapted from [55]

Step 3. Construct Junction Tree

Based on the triangulated graph G^T, a junction tree can be built by forming a maximal spanning tree from the cliques. First, all the maximal cliques are identified from G^T in Fig. 4.21.

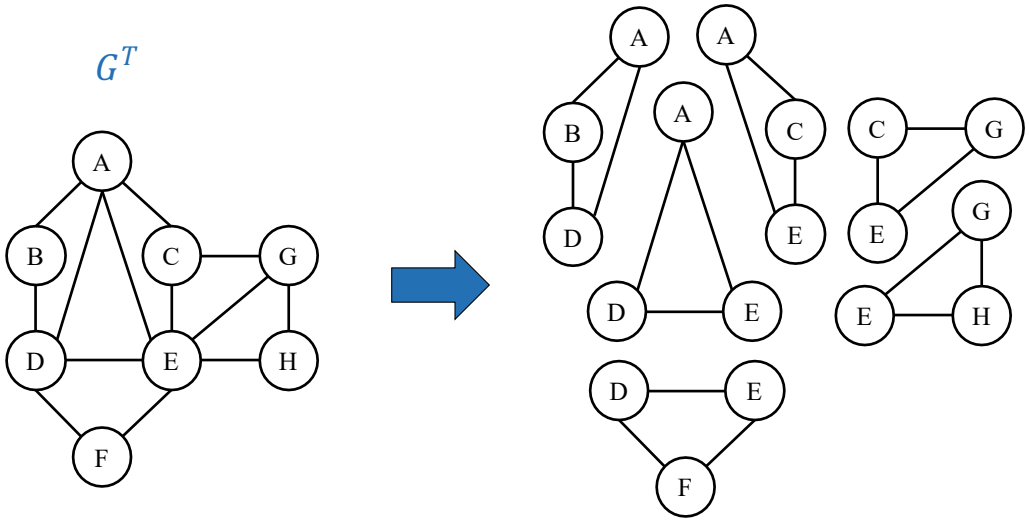


Fig. 4.21 Identifying cliques in triangulated graph, adapted from [55]

Then, a junction graph G^J presented in Fig. 4.22 is reconstituted with nodes representing the cliques identified from G^T. If two cliques intersect, they are joined in the junction graph by an edge labelled with their intersection, called separators.

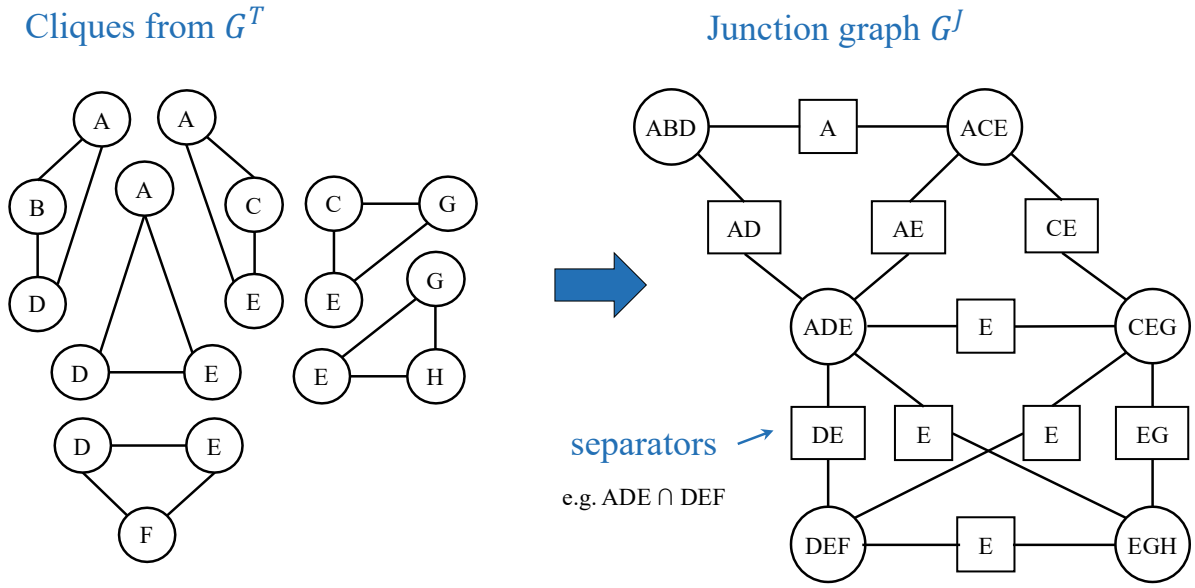


Fig. 4.22 Junction graph construction, adapted from [55]

After that, a junction tree G^{JT} is extracted from the junction graph in Fig. 4.23 such that the tree contains all the cliques (spanning tree) and satisfies the running intersection property:

For each pair of nodes X, Y, all nodes on the path between X and Y contain $X \cap Y$.

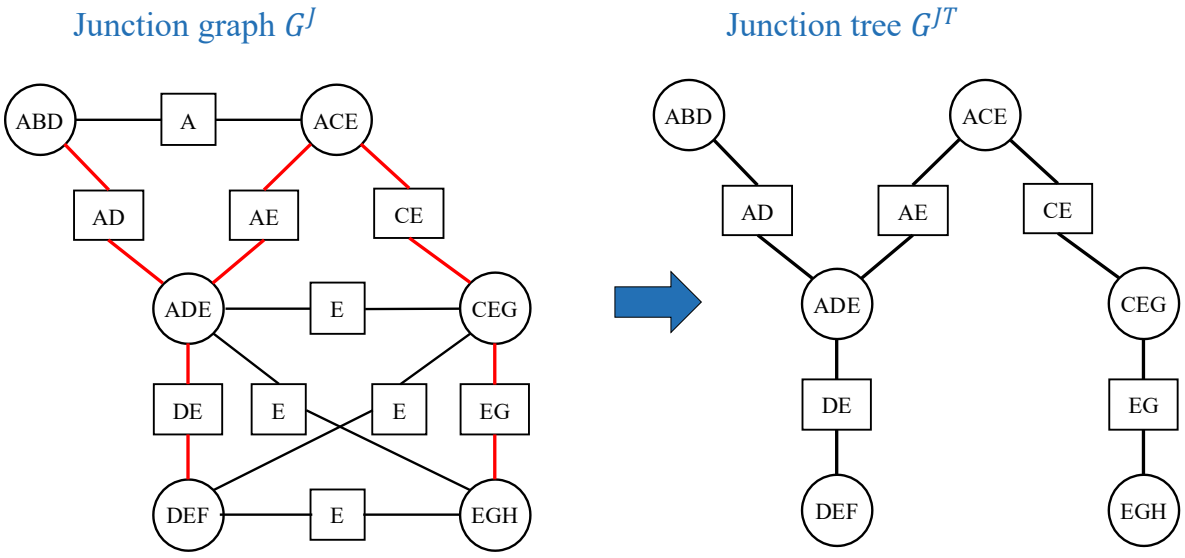


Fig. 4.23 Junction tree construction, adapted from [55]

Finally, the probability distributions of the cliques (nodes) and separators (edge labels) in the junction tree G^{JT} need to be transferred from the conditional probability distribution of the original Bayesian network G using potentials. The process is carried out as follows. For each (conditional) distribution from the BN, create a node potential:

$$P(v_i|par(v_i)) \Rightarrow \phi_i(v_i, par(v_i)) \tag{9}$$

where $par(v_i)$ is the parent of node X_i, ϕ_i signifies the potential between the nodes.

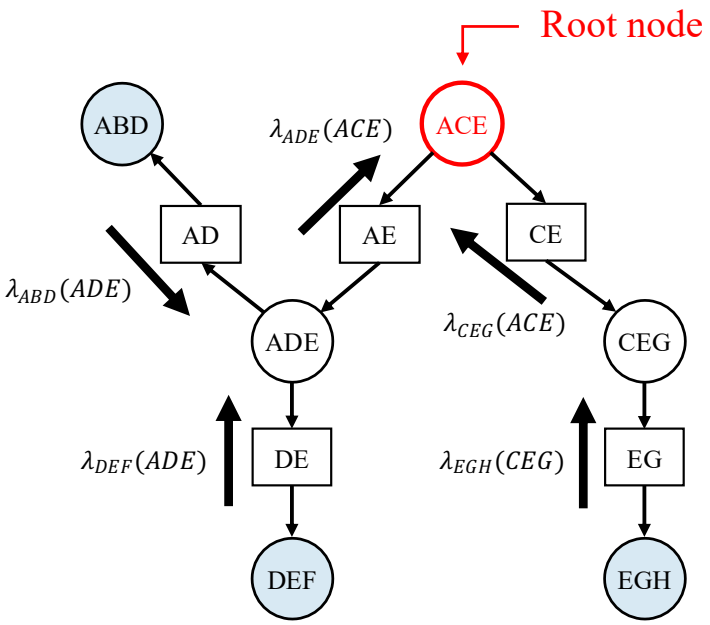
Assign each node potential to its associated clique C, and compute the clique potential ϕ_C for C as the product of its assigned node potentials:

$$\phi_C = \prod_{\phi_{v_i}} \phi_{v_i} \tag{10}$$

Such that $\{v_i\} \cup par(v_i) \subseteq v_C$

Step 4. Belief Propagation

Belief propagation, also identified as sum-product message passing, is a message-passing algorithm for delivering inference on tree-like structures. Since the learnt Bayesian networks have been converted into tree structures by JT algorithms, now BP can be used to infer the probability of each reject reason from the junction trees, conditional on any observed nodes with the external evidence. In addition, belief propagation is a generalisation of the Forward-Backward method and consists of two passes, one from the leaf nodes to the root node, and the other from the root node to the leaf nodes. The logical application of BP is demonstrated according to the two passes, Pass 1 (Upward Pass) and Pass 2 (Downward Pass).



Pass 1: Upward Pass

Fig. 4.24 Belief propagation upward pass

Pass 1 starts from the leaf nodes highlighted as light blue in Fig. 4.24 to the root node coloured in red. For each node v_i that the algorithm is processing on, BP takes the evidence

at the node v_i (e.g., node 'DEF') and message from its child node if there is any to compute the message $\lambda(v_i)$ to pass it upwards to its parent node (e.g., node 'ADE'). $\lambda(v_i)$ embodies the conditional probability of observing future evidence, given the outcomes of node v_i. Its calculation is illustrated in Fig. 4.25. When the message reaches its parent node $par(X)$, information received by $par(v_i)$ becomes $\lambda_{v_i}(par(v_i))$ where the conditional probability matrix M_{v_i} is incorporated into the message. In the case of a junction tree, M_{v_i} will be transferred to the potential $\phi\big(v_i, par(v_i)\big)$ from the original conditional probability tables. The upward message-passing process iterates through all nodes until it reaches the root node. Then all the new evidence has been included to update the probabilities in one direction. The logic flow of BP Pass 1 is also encoded in Algorithm 8.1.

Algorithm 8.1 Belief Propagation – Pass 1

Input: a Bayesian Structure G, condition probability table for each node $v_i \in V_{M_{v_i}}$

Output: Inferred probability $BEL(v_i)$.

for $v_i \in V_{M_{v_i}}$ **do**
  if $v_i \in L$ **then**
    Q.push (v_i)
  else
    count $(v_i) = |C_{v_i}|$
  end
end
while length$(Q) \neq 0$ **do**
  $v_i = Q$.pop ()
  $\lambda(v_i) = \theta(v_i)$
  for $c \in C_{v_i}$ **do**
    $\lambda(v_i) = \lambda(v_i) \odot \lambda c(v_i)$
  end
  if $X \neq R$ **then**
    $\lambda_{v_i}(pa(v_i)) = M_{v_i}\lambda(v_i)$
    count $(pa(v_i))$ = count $(pa(v_i))$ – 1
    if count $(pa(v_i)) = 0$ **then**
      Q.push $(pa(v_i))$
    end
  end
end

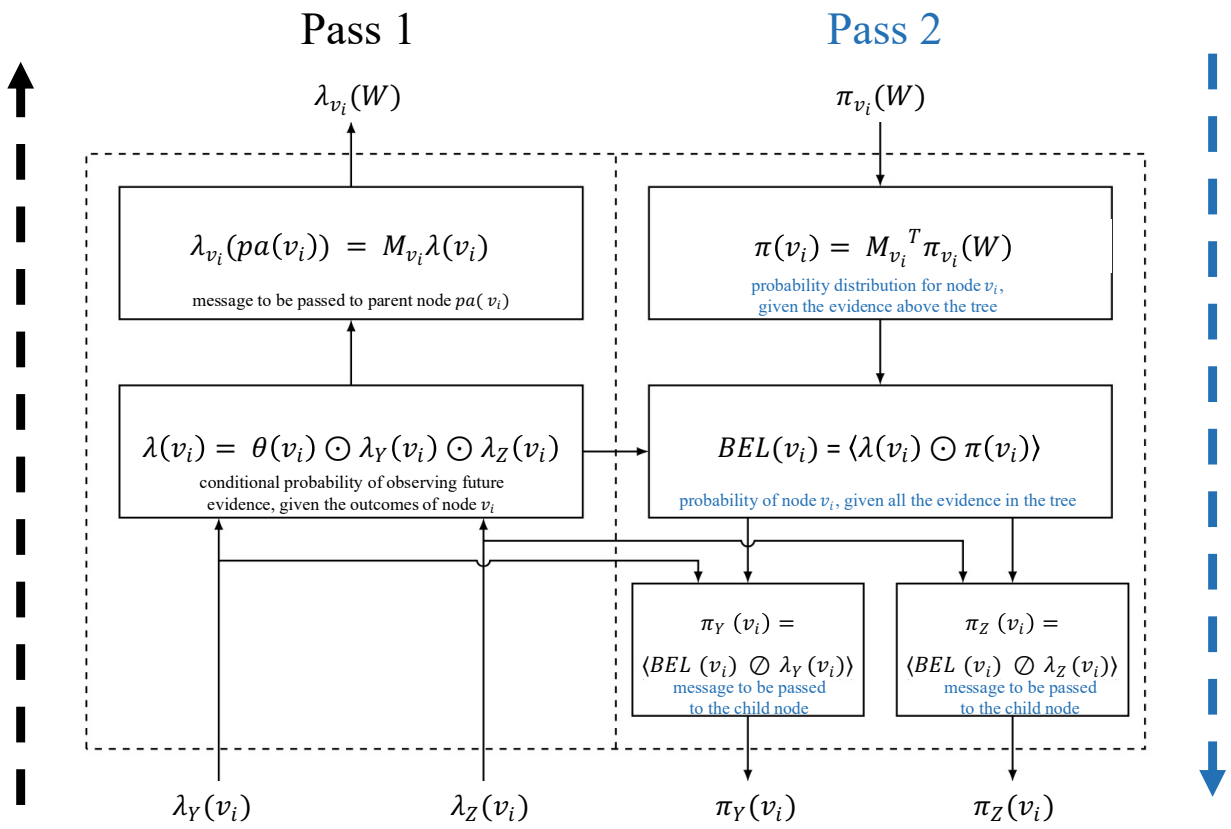
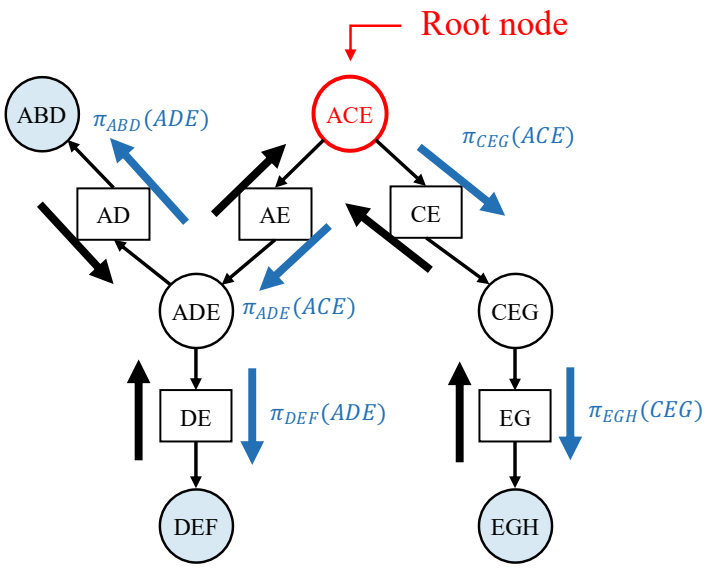


Fig. 4.25 Messages passed in pass 1 and pass 2 in belief propagation

After all the messages holding the new information have been delivered in Pass 1, Pass 2 starts in the opposite direction from the root node to the leaf nodes following the blue arrows in Fig. 4.26. At each iteration, BP takes the prior distribution of the current node v_i and the message from its parent nodes if there is any to compute the message $\pi(v_i)$ to pass it downwards to its child nodes. $\pi(v_i)$ represents the conditional distribution for variable v_i, given the evidence preceding node v_i in the networks. Since the node v_i has already captured both messages from its children and parents, now the belief probability $BEL(v_i)$ can be calculated by combining these messages, $\lambda(v_i)$ and $\pi(v_i)$. $BEL(v_i)$ encapsulates the conditional distribution for a node v_i, given all the associated evidence in the network. The message containing the belief probabilities keeps being passed down to its child nodes with the elimination of duplicate information to update the belief of its child nodes. Once it reaches all the leaf nodes, all the belief probabilities $BEL(v_i)$ in the network have been inferred. The logic flow of BP Pass 2 is also encapsulated in Algorithm 8.2.



Pass 2: Downward Pass

Fig. 4.26 Belief propagation downward pass

When the inferred variable is a root-cause variable (i.e., $v_i \Rightarrow Y_c$), $BEL(v_i)$ expresses the probability of a root cause Y_c leading to quality failures, given all the evidence in the network, which is our probability of interest for Q2. If the variable is the quality risk indicator $v_i \Rightarrow R$, the belief probability $BEL(v_i)$ encodes the probability of the quality risk indicator R being "1", implying whether the job will have quality issues or not. If the probability is greater than the classification threshold of 0.5, the job is predicted to be problematic with quality issues. Otherwise, the job is projected to be normal. In this way, Q3 can also be solved. The inferred root-cause probabilities and risk prediction for each of the BN models learnt by different learning algorithms and knowledge sources based on the case sample are displayed in Table 4.3 at the end of the section. However, we are not using the predictions from one of the BN models to answer Q2 and Q3. Instead, the predictions will be fused into aggregated results using WAEL to answer the research questions. This is to mitigate the risk of the existing shortages in prediction accuracy and stability of a single BN model [26]. Accordingly, the implementation of WAEL is presented in the next section.

Algorithm 8.2 Belief Propagation – Pass 2

Input: a Bayesian Structure G, condition probability table for each node $v_i \in V_{M_{v_i}}$

Output: Inferred probability $BEL(v_i)$

Q.push (R)
while length(Q) $\neq$ 0 **do**
  $v_i = Q$.pop ()
  if $X = R$ **then**
    $\pi(v_i) = {M_{v_i}}^T$
  else
    $\pi(v_i) = {M_{v_i}}^T \pi_{v_i}(pa(v_i))$
  end
  BEL (v_i) = $\langle \lambda(v_i) \odot \pi(v_i) \rangle$
  for $c \in C_{v_i}$ **do**
    $\pi_C(v_i) = \langle BEL(v_i) \oslash \lambda_C(v_i) \rangle$
    Q.push (C)
  end
end

4.2.6 Weighted Average Ensemble Learning

Although the probabilities of reject root causes and quality risk predicted based on distinct BN models are available to answer the RCA questions, they might be insufficient regarding accuracy and stability. It has been found that a single BN learner is faced with deficiencies in prediction accuracy and model robustness [26]. Such phenomenon is also observed in Section 4.2.3.2. where some of the learnt BN structures are sparse and contain unreasonable arc directions. This entails the risk of poor causality discovery and inaccurate prediction of a single BN model. For the purpose of providing a robust and accurate RCA solution, the predictions from different BN models will be fused to achieve a more accurate and stable prediction. WAEL has been used in a broad range of research fields to aggregate multiple models [56]. It has also been demonstrated with better performance than constituent regression or classification models alone [57]. Thus, we have integrated the WAEL with BN models to improve accuracy and model robustness. WAEL is a voting ensemble method that combines the predictions from multiple models by taking the weighted sum of the predictions for regression models or selecting the class with the largest weighted sum of

predicted probabilities for classification models. The choice of weight is an important aspect of the weighted average ensemble. The weight ought to reflect the skill of each model. In our case, the weights of the BN models are determined by their robustness, which is reflected by their likelihood of being the best classifier. The likelihood is computed as the number of times that a BN model outputs the most accurate prediction (optimal frequency) in ratio to the total number of times that a prediction has been performed (total frequency). Accordingly, the weights are formulated as below:

$$W_{G\,m,k} = f_k / \sum_{i=1}^{K} f_i \tag{11}$$

where $W_{G\,m.k}$ is the weight assigned to Bayesian learning algorithm A_K for Product Pr_m, $\mathrm{f_k}$ is the optimal frequency for algorithm k, and K is the total number of BN learning algorithms.

To calculate the values of the weights, the optimal frequency of each model needs to be obtained. We have compared all the BN models learnt by different structure learning algorithms and knowledge sources on the entire dataset to summarise the frequency of each learner being the best-performed model as shown in Fig. 4.27.

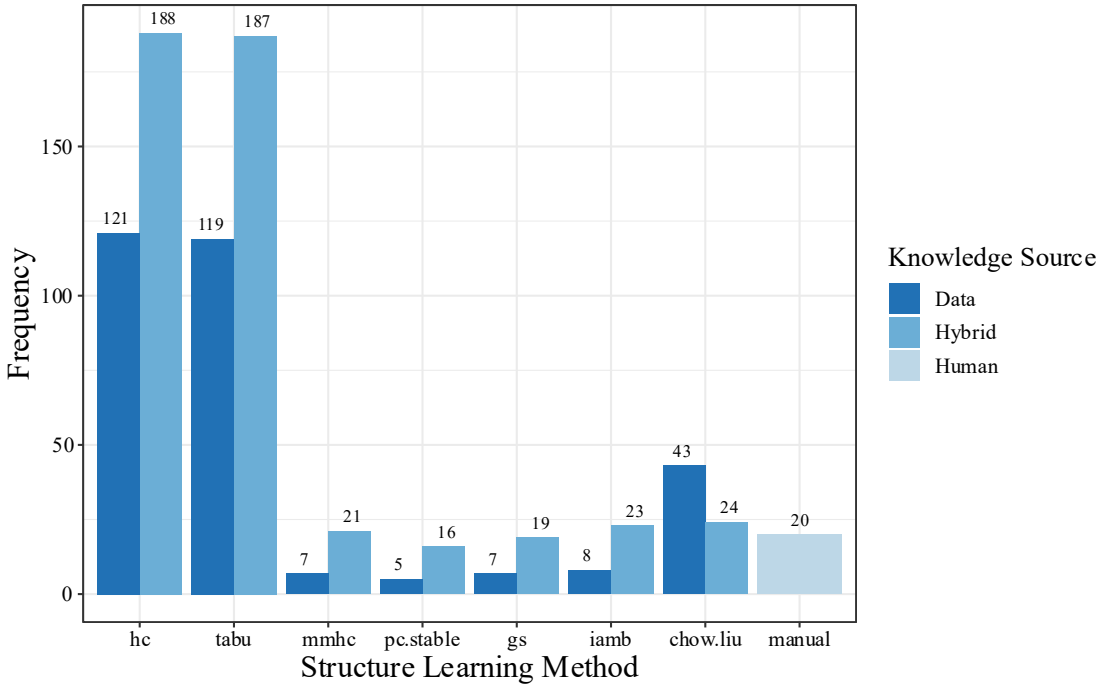


Fig. 4.27 Optimal model frequency for different structure learning methods and different knowledge sources

Subsequently, the WAEL technique follows Algorithm 9 to obtain the fused predictions for ensembled BN. It first gathers the BN structures learnt by different learners in Step 1. Then the predictions are performed through parameter learning and inference on the collected distinct structures in Step 2. After that, each prediction is assigned a weighted according to equation (11). Lastly, the weighted predictions are summed up to obtain the aggregated results. In this way, WAEL combines the predictions from distinct BN models into one set of root-cause probabilities and a single risk prediction for a job as displayed in Table 4.3.

Algorithm 9 Bayesian Network Prediction with Weighted Average Ensemble Learning

Input: Testing data for product Pr_m, D_m;
a list of bagged models for product Pr_m, $G_{m,k} \in \{G_{m,1}, \dots, G_{m,K}\}$;
weights of each bagged model, $W_{G\,m,k} \in \{W_{G\,m,1}, \dots, W_{G\,m,K}\}$.
Output: Weighted average prediction, P_m

for *each* $G_{m,k}$ in $\{G_{m,1}, \dots, G_{m,K}\}$ **do** **Step 1 Get Models**
  $P_{m,k} = G_{m,k}(D_m)$ **Step 2 Prediction**
end for
$P_m = \sum_{k=1}^{K} W_{G\,m,k}\ P_{m,k}$ **Step 3 Weighted Sum**

4.2.7 Answering the RCA Questions

Since the predictions from different BN models have been fused into a robust aggregated solution, the RCA questions raised in Section 3.1 can be answered. For Q1, the casual relationships have been discovered from the historical production data D_{ABC500} for the testing instance, job '5077' producing product 'ABC500'. And the acquired causal knowledge has been presented in a human-interpretable form of graphical BN structures using various structure learning algorithms in Section 4.2.3.2.

For Q2, the parameters of the BN models are learnt, and the job features of the testing instance X_{5077} are input into the learnt BNs during Bayesian inference. By following the procedures of the junction tree algorithm, the probabilities of potential reject causes are inferred from different BNs shown in Table 4.3 below. Lastly, the root-cause probabilities predicted from different BN models are fused into an aggregated set of probabilities using WAEL to output a robust solution. This ensembled solution is highlighted in red in Table 4.3. It is used to answer Q2 as follows: the probability of "*Setup Reject*" causing the job to have a quality issue is 0.08, and the probability of "Yield Update Adjustment" causing the job to have a quality issue is 0.92. From the table, we can also see the predictions for the example case from various BN models. The results have shown that besides the proposed ensemble method, TS and HC algorithms also tend to provide accurate probabilistic reasoning even when the structures are purely learnt from the data. Whereas, constraint-based and hybrid techniques, as well as manual graphs, seem to underperform in identifying the root cause with stochasticity.

For Q3, we assume that job '5077' has not been run yet, so the quality risk of the job is what we want to know. In this question, some features that can only be obtained at the end of a job operation are not known such as reject weight quantity and downtime. Therefore, only the job features that can be obtained prior to the job operations are introduced into different BN models to predict the quality risk of job '5077'. Similar to Q2, the results are ensembled using WAEL to achieve a stable solution. The quality risk obtained using the proposed method is "1", $R_{5077} = 1$, indicating the test case job '5077' has quality issues. The results of other stand-along BN models are also presented in Table 4.3 below. It has been shown that all methods exhibit a strong ability in predicting the high-reject jobs except the manual and PC.stable models. It is worth noting that these results are only based on one specific product, more experiments need to be conducted to draw a reliable and representative conclusion. Therefore, Chapter 5 evaluates the performance of the proposed ensembled BN method as well as other constituent BN models on a sample of 6721 jobs (199 products).

Table 4.3 Inferred root cause probabilities and risks using different methods

Network Type:		**Q2: Reject Reason Probability**		**Q3: Risk Prediction**
Structure Learning Alg.	Knowledge Source	Setup Reject P_{SR}	Yield Update Adjustment P_{YUA}	isRejectFail R_{5077}
Ground Truth	**/**	**0.08**	**0.92**	**1**
Manual	Human	0	0.97	0
HC	Data	0.08	0.92	1
TS	Data	0.08	0.92	1
PC.stable	Data	0.15	0.85	0
GS	Data	0.15	0.85	1
IAMB	Data	0.15	0.85	1
MMHC	Data	0.15	0.85	1
chow.liu	Data	0.07	0.89	1
HC	Hybrid	0.08	0.92	1
TS	Hybrid	0.08	0.92	1
PC.stable	Hybrid	0.07	0.93	0
GS	Hybrid	0.07	0.93	1
IAMB	Hybrid	0.07	0.93	1
MMHC	Hybrid	0.07	0.93	1
chow.liu	Hybrid	0.07	0.93	1
WAEL	WAEL	0.08	0.92	1

Chapter 5

Evaluation and Discussions

This chapter evaluates the results obtained from the proposed RCA framework on the entire dataset, compares the performance of different BN learning techniques, and discusses the robustness of the ensembled BN models. Section 5.1 focuses on the inferred probabilities of different reject causes for RCA regarding Q2. Section 5.2 examines the accuracy of the predicted product failure risk reflecting the performance of the proposed method in answering Q3.

5.1 Evaluation Methods for RCA

In this study, the proposed product-wise RCA framework of ensembled BNs is applied to the processed data with a sample size of 6721 jobs (199 products). The k-fold cross-validation scheme is used to train and test the models. Different metrics have been chosen and designed to evaluate the performance of the proposed model for both predicting the probabilities of the reject causes for Q2 in Section 5.1.1, and classifying the quality risk of scheduled (i.e., to-be-run) jobs for Q3 in Section 5.1.2.

5.1.1 Evaluation Methods for Risk Prediction

Question 2 in Section 3.1 is to identify a list of potential reject reasons and their corresponding probabilities for a job with quality issues. This means that the predicted result of an instance will be a sequence of probabilities. The ordering of the root causes in the sequence is important, indicating which reject causes are the dominant reasons. Therefore, both the accuracy of each predicted probability and their ranking in the sequence need to be evaluated to have a comprehensive judgement on the predicted lists of probabilities. To realise this, a prediction error metric is designed to measure the accuracy score for a sequence of probabilities. There are two elements in the prediction error metric, Mean Absolute Error (MAE) and ranking error, where MAE is to quantify the difference between

the predicted and observed probabilities for a group of root causes on average, and ranking error is to identify the ranking difference between prediction and observation as sequences. The calculation of prediction error includes 3 steps:

1. Calculate MAE for the list of predicted probabilities
2. Compute the ranking error of the sequence
3. Compute the final prediction error

Step 1 MAE

MAE is calculated by

$$MAE = \frac{1}{n}\sum_{i=1}^{n} | \widehat{y_i} - y_i | \tag{12}$$

where $\widehat{y_i}$ is the predicted probability for root cause i in the sequence, y_i is the observed probability, and n is the length of the probability list (i.e., the number of identified root causes in the list for a job). In our study, MAE is chosen over the Root Mean Square Error (RMSE) because RMSE is more specialised in describing large errors that are particularly undesirable. However, in our case, probabilities are predicted, and the difference and the variance are normally quite small. Therefore, MAE is favoured to capture the nuance in probability differences.

Step 2 Ranking error

Before computing the ranking error of the predicted sequence, all the noises in the probabilities need to be removed so that the ranking metric will not be oversensitive toward considerably small probabilities. This is achieved by truncating the probabilities to 2 significant figures. Then each probability in the list is assigned a rank according to the magnitude of the predicted probabilities. The ranking difference between the observed and predicted sequences can be quantified. Eventually, the ranking difference in ratio to the possible maximum ranking difference is computed as ranking error. The possible maximum ranking difference is formulated as follows:

$$ranking\ error_{\ max} = \begin{cases} \frac{k^2}{2} & if\ k\ is\ even \\ \frac{k^2-1}{2} & if\ k\ is\ odd \end{cases} \quad (13)$$

where k is the length of the predicted list.

Step 3 Prediction Error

Lastly, MAE and ranking error are combined by a weighted sum to comprise the prediction error. In this study, both MAE and ranking error are designated with a weighting of 0.5.

$$\varepsilon_{prediction\_error} = w_{MAE}\ \varepsilon_{MAE} + w_{ranking\_error}\ \varepsilon_{ranking\_error} \quad (14)$$

In this way, both the accuracy of the predicted probability values and their ranking in the sequence can be captured in the prediction error metric.

5.1.2 Evaluation Methods for Risk Prediction

The third question (Q3) proposed in Section 3.1 is a classification problem where the jobs with quality issues need to be predicted prior to the execution. Hence, a confusion matrix can be incorporated. Subsequently, accuracy, sensitivity and specificity can be derived from the confusion matrix to evaluate the performance of the proposed method. The formula for the accuracy metric is illustrated in Fig. 5.1. So as the equations for sensitivity and specificity. They are commonly used to measure the performance of a predictive model.

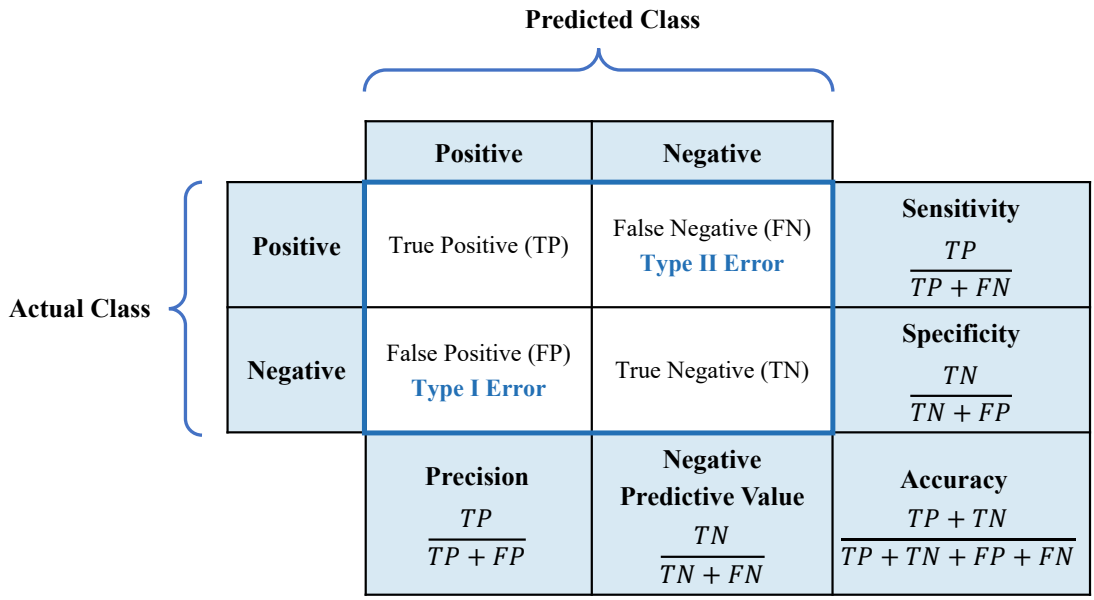


Fig. 5.1 Confusion matrix with classification metrics

In the context of manufacturing, we are more concerned with Type I Error (False Positives). Since the system should not distract the staff often with a false alarm, it needs to guarantee all the alarms that went off are correct and worth taking note of. Otherwise, the alert system will lose its credibility among the factory workers. As a result, the evaluation system will emphasise False Positive Rate (FPR), which is the proportion of identified positives (i.e., jobs predicted to be problematic) among the normal jobs. This is also defined as 1-specificity. The Receiver Operating Characteristics (ROC) curve provides a good way to visualise the true positive rate (or sensitivity on the y-axis) against the false positive rate (or "1-specificity" on the x-axis). It also gives a picture of the classifier's performance across all possible probability thresholds. Additionally, the Area Under the ROC Curve (AUC) provides an aggregate measure of performance across the whole spectrum of classification thresholds. One way of interpreting AUC is as the ability of a classification model to distinguish 1s from 0s. Lastly, the robustness of different algorithms will be assessed by the likelihood of generating the worst predictions.

5.2 Predicted Probabilities of Reject Causes for RCA

This section shows the results for question Q2, predicting the probabilities of each root cause for each job. The following passages will compare the prediction performance between different learning techniques in Section 5.2.1 and evaluate the robustness of the proposed method in Section 5.2.2.

5.2.1 Accuracy Comparison Between Different Methods

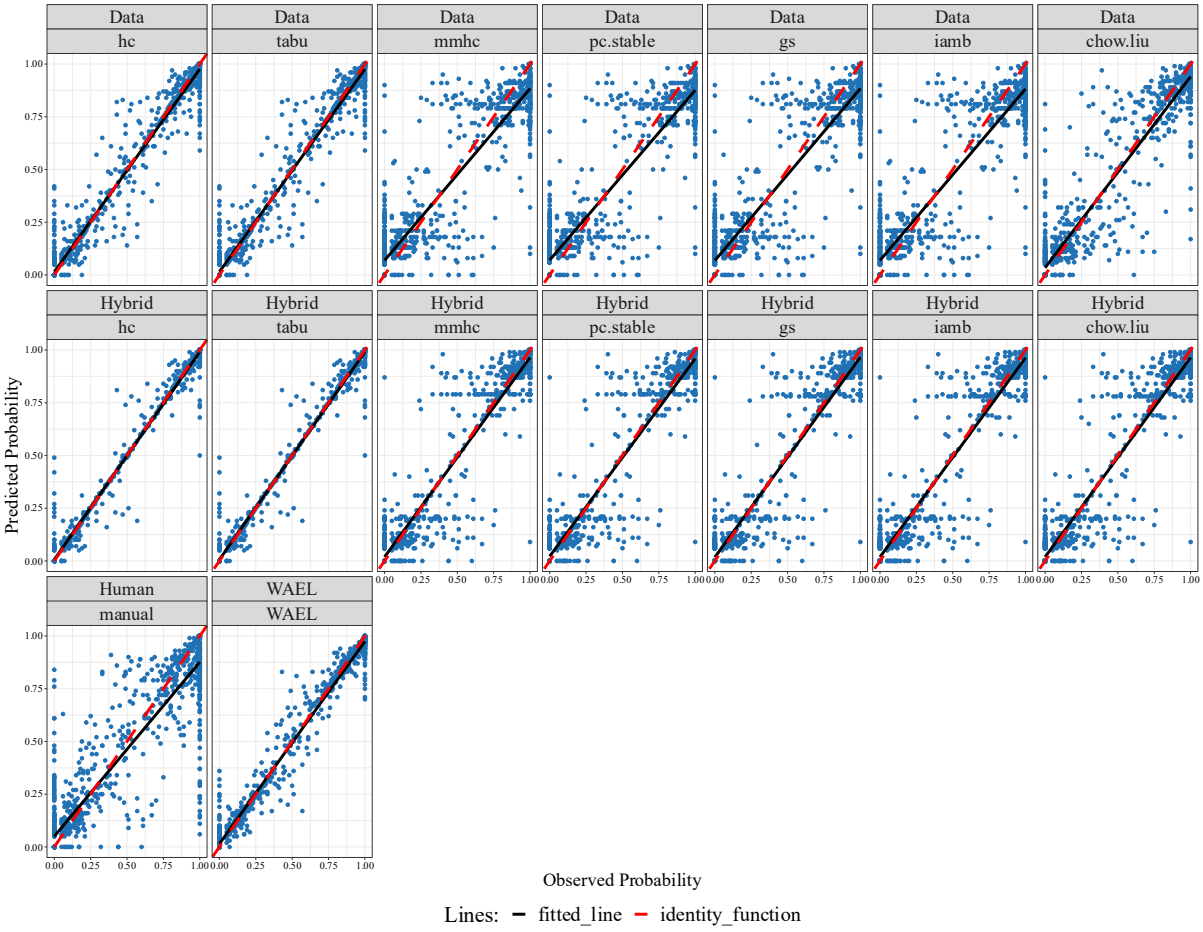


Fig. 5.2 Faceted scatter plots of predicted vs observed probabilities for different reject reasons by distinct knowledge sources and structure learning methods

The faceted scatter plot in Fig. 5.2 gives a straightforward illustration of the association between prediction and observation. It plots the probabilities of all the identified reject reasons for every single job in the dataset. The observed values lie on the x-axis, while the y-

axis shows the predicted probabilities for the reject causes. The regression line fitting the data is presented as the dashed red line. And the black diagonal line in the centre with a gradient of 1 acts as a reference for a perfect prediction (i.e. 100% accuracy). Directly, we can see that the projected dashed lines in the first row deviate away from the reference line except for score-based algorithms ("hc", "tabu"). Whereas the regression lines for the models in the middle row are closer to the diagonal line, suggesting a better prediction. The manual model seems to exhibit relatively poor performance as it diverges further away from the reference line. Last but not least, the weighted average ensemble learning technique fits the data reasonably well, indicating a strong correlation between the model's predictions and its actual results.

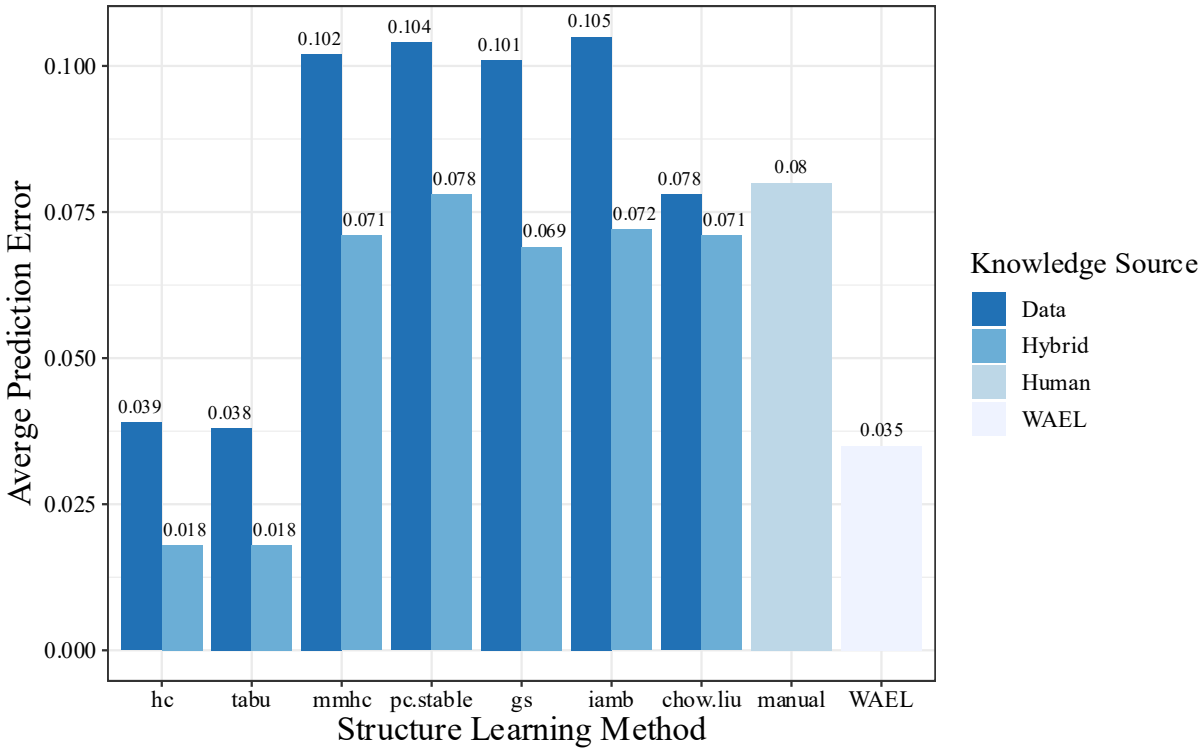


Fig. 5.3 Bar plot of averaged prediction error for RCA over different structure learning methods with different knowledge sources

The bar graph in Fig. 5.3 portrays the average prediction error of the reject reason probability using different structure learning methods and involving different levels of prior knowledge. Among all the structure learning methods, tabu search ("tabu") and hill-climbing ("hc") algorithms seem to lead to the lowest mean prediction error despite the level of prior

knowledge. In particular, tabu search reaches a slightly smaller error than hill-climbing algorithm when the structure is purely learnt from the data. Their prediction variances are almost identical as shown in the boxplot in Fig. 5.4. This is because they belong to a greedy search score-based method with the only difference in the solution memorisation step. Moreover, the BNs learnt by "hc" and "tabu" are very similar in the case study demonstrated in Fig. 4.12, which explains the similar performance of the two algorithms. Marco et al.'s study also agrees with our findings on tabu search being the overall most accurate method than constraint-based algorithms [58]. This phenomenon is caused by the scarcity in the structures learnt by the constraint-based algorithms. According to the structure learning processes of the constraint-based algorithms shown in Algorithm 3-5 in Section 4.2.3.2, we have found that they share similar structure learning procedures in which they only accept an arc when there is a conditional dependence between two nodes under the constraint that the introduced edge will not create cycles in the graph. This strategy largely reduces the chance of node linkage in the graph, hence bringing about a sparse connection and an inadequate accuracy.

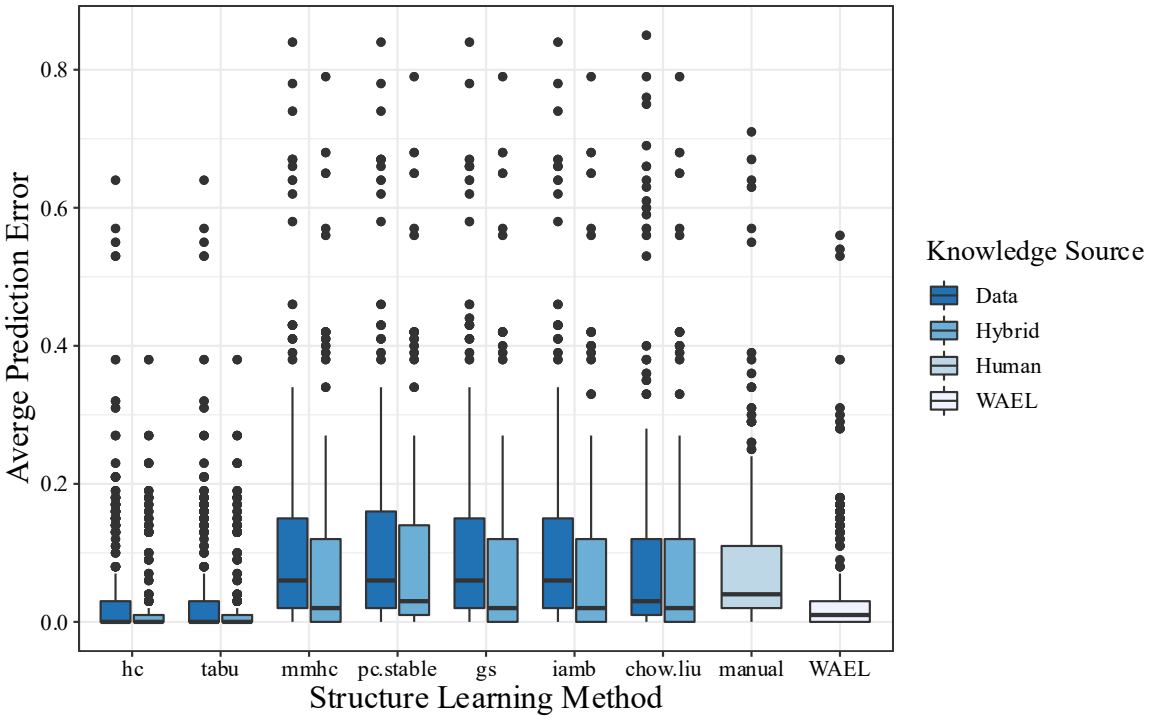


Fig. 5.4 Boxplot of averaged prediction error for RCA over different structure learning methods with different knowledge sources

By shifting the focus onto the effect of different knowledge sources on RCA performance, we found a distinctive favour on the models learnt with hybrid knowledge as their mean prediction error is significantly lower than models with other knowledge sources. This aligns with the expectation that extra knowledge provides insights and guidance to structure learning so resulting in a better prediction. However, BN purely built on expert knowledge exhibits a worse prediction than hybrid knowledge. It contrasts with the expectation of the more guidance, the better the prediction. This could be because the exclusively human-made network makes the structure insensitive to any relationship that occurs in the data. As a consequence, it misses the features and causality that are hidden in the data. The full human knowledge also brings rigidity to the structure that restricts the parameter learning and inference processes, resulting in a performance shabbier than expected. On the other hand, hybrid knowledge mines the causal relationships mainly from the data with a hint of expert insights. By incorporating both knowledge sources, it integrates flexibility by learning from the stochastic industrial data and the realistic reasoning from the real-world experience, therefore, outputting better performance in RCA.

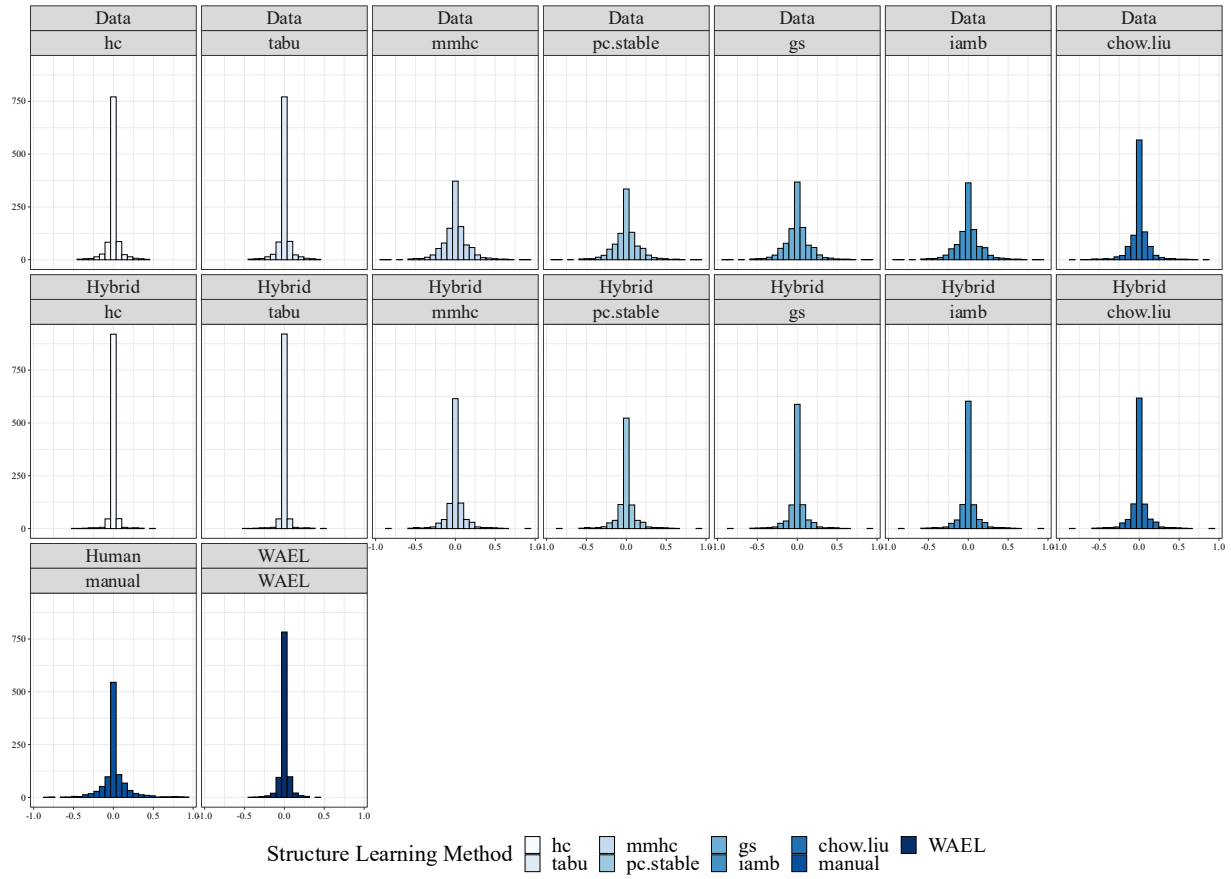


Fig. 5.5 Residual histogram plot over different structure learning methods from different knowledge sources

Residuals are also inspected to check if the model is appropriate and trustworthy for the data. They are the estimates of experimental error obtained by subtracting the observed probabilities from the predicted probabilities for the root cause. Fig. 5.5 illustrates the faceted residual histogram plot for different structure learning methods. Its y-axis shows the frequency of each residual value that has occurred in the experiment. From the plot, we can see that the overall patterns of the residuals for all the models approach a bell shape, signifying a normally distributed variance. Thus, the normality assumption is likely to be true.

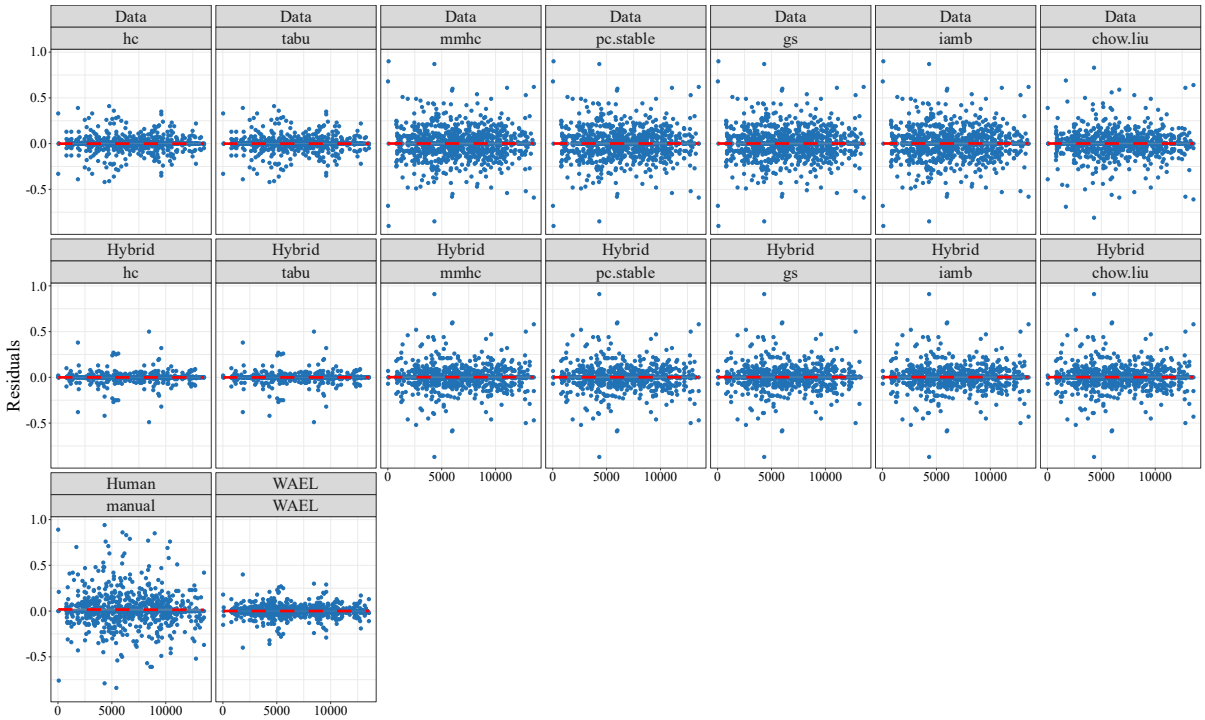


Fig. 5.6 Residual scatter plot over different structure learning methods from different knowledge sources

The residuals are further investigated in a scatter plot based on the models using different structure learning methods and knowledge sources. The residual plot not only helps to check the validity of a regression model but also provides guidance on how to improve it. As displayed in Fig. 5.6, no special pattern such as a curve or a fluctuating pattern is speculated in the plots. This implies that the error term is random so the independent variables in our models have well explained the underlying patterns in the data. Correspondingly, the models are well behaved.

5.2.2 Robustness of Ensembled BN

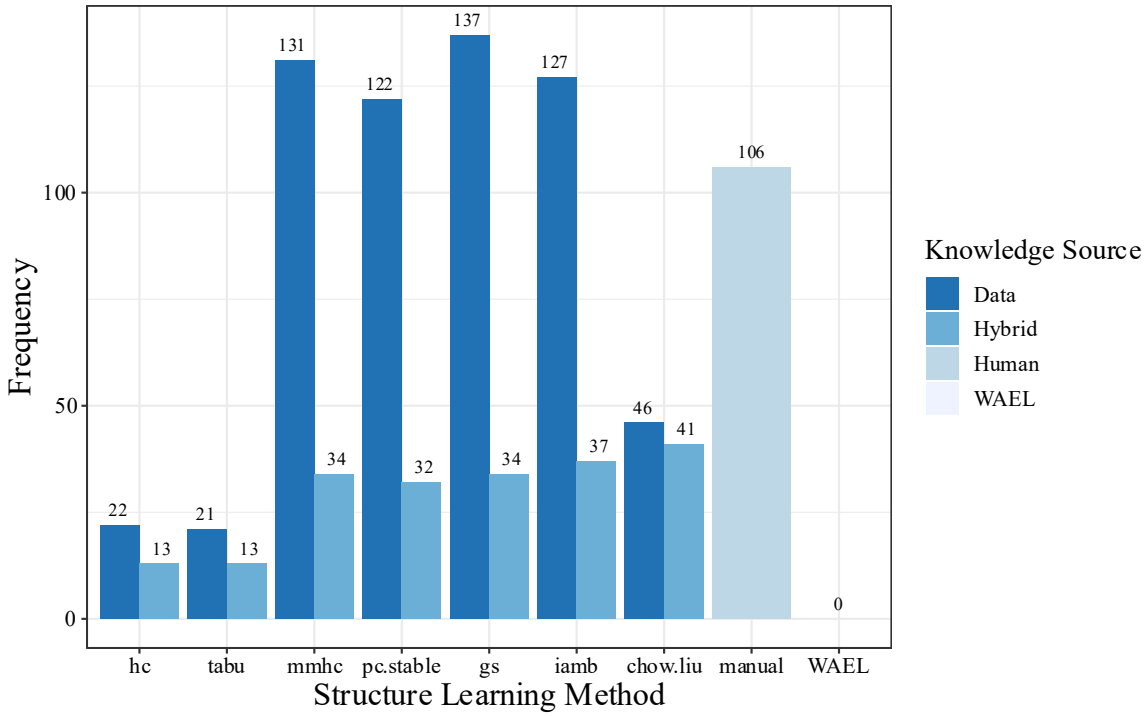


Fig. 5.7 Worst model frequency for RCA over different structure learning methods and different knowledge sources

Finally, the robustness of the models is assessed as it is crucial to industrial practice in manufacturing. The accuracy of the predicted root cause probabilities requires to be stable among different products and jobs under various configurations. Frequent prediction swings will diminish the value of the models, confuse the workers, and hinder the production process. Therefore, the stability of the models is vital. This study has tested the robustness across the 6792 jobs in the data, and it is assessed by counting the occurrence of each model outputting the worst prediction for each job. The higher the frequency, the less robust the model. In Fig. 5.7, we can see that the proposed WAEL method proves to be the most stable technique among all the algorithms. It has never appeared to be the worst model for any job instance. This is mainly due to its voting nature of putting more weights on the more stable algorithms as elaborated in Section 4.2.6. This weighting method alleviates the risk of the existing deficiencies in accuracy and stability in stand-alone algorithms [26]. As a result, it achieves our goal of providing a robust RCA model.

5.3 Defect Risk Prediction

This section shows the results for question Q3, predicting whether a scheduled job will be a risk in the aspect of product quality. Section 5.3.1 compares the prediction performance of the models using different learning techniques. Then, the robustness of the proposed method is assessed against the constituent algorithms in Section 5.3.2.

5.3.1 Accuracy Comparison between Different Methods

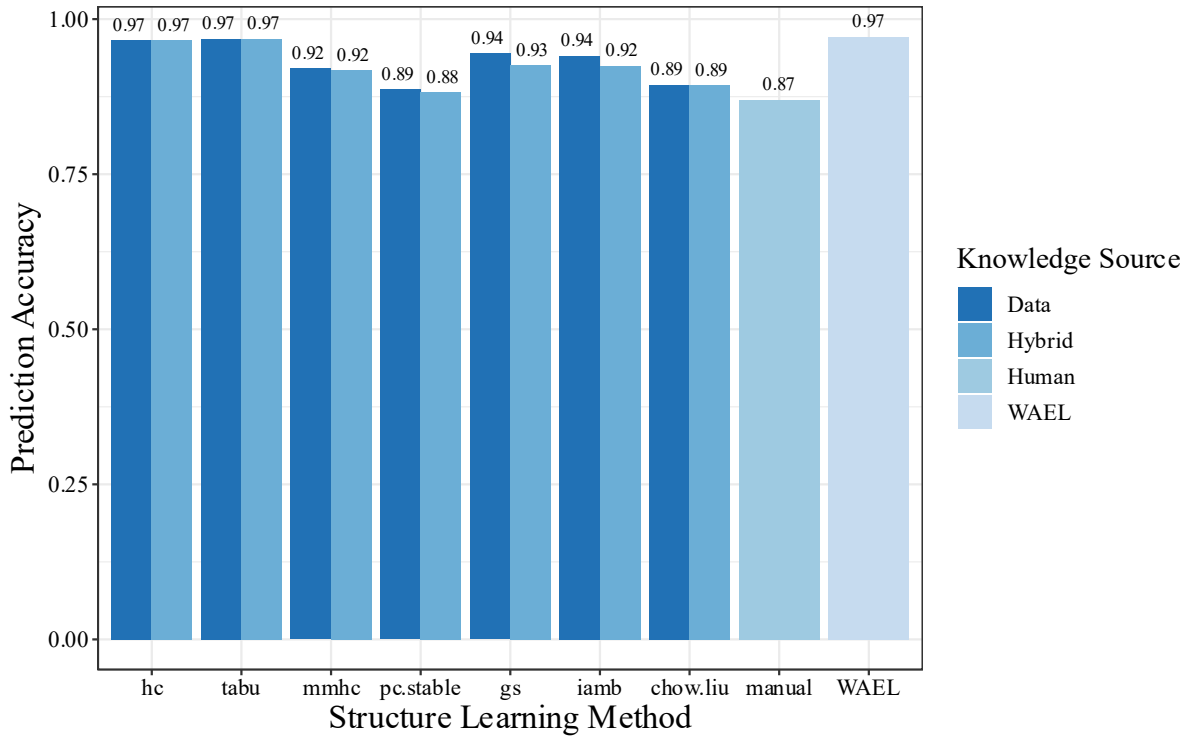


Fig. 5.8 Classification accuracy over different structure learning methods grouped by knowledge source for risk prediction

In the task of predicting a risky job, the graph above (Fig. 5.8) shows the accuracy of job classification using different structure learning algorithms based on different knowledge sources. In general, all methods seem to exhibit a high accuracy for prediction, among which score-based methods ("hc", "tabu") still yield the best results. This matches the observation from probabilistic reasoning in Section 5.2.1 (Q2). Comparatively, knowledge acquired from

hybrid resources (both data and human knowledge) draws little advantage in prediction accuracy between score-based structure learning methods. Surprisingly, models learnt purely from data outperform other knowledge sources in prediction accuracy by a slight amount for constraint-based structure learning methods ("pc.stable", "gs", "iamb"). Such phenomenon occurs possibly because of the lack of evidence during inferencing for the nodes in the added knowledge substructure (prior knowledge) from the hybrid knowledge source (exemplified in Fig. 4.10). In the case of risk prediction, the jobs are yet to be run so there will be no external information introduced to the observational variables such as the occurrence of reject cause reasons. This makes the added links (i.e., knowledge) from the reject cause nodes to job failure nodes redundant in the hybrid models. Conversely, hybrid methods can also introduce noises into the inference process due to the extra connection from the prior knowledge with null parameters in the risk prediction. As a result, the accuracy of the hybrid algorithms will be adversely impacted as Oniśko et al. stated that the diagnostic accuracy of Bayesian network models suffers from imprecision in parameters of zeros [2].

On the other hand, the proposed WAEL model continues to show a prominent advantage in job classification as it does in probabilistic reasoning in Section 5.2.1. The overall accurate predictions of all the algorithms also contribute to the excellent performance from the WAEL method as it applies the weighting to the probabilities of the predicted class obtained by various algorithms and determines the riskiness of the job using the weighted sum of the likelihood against the classification threshold of 0.5.

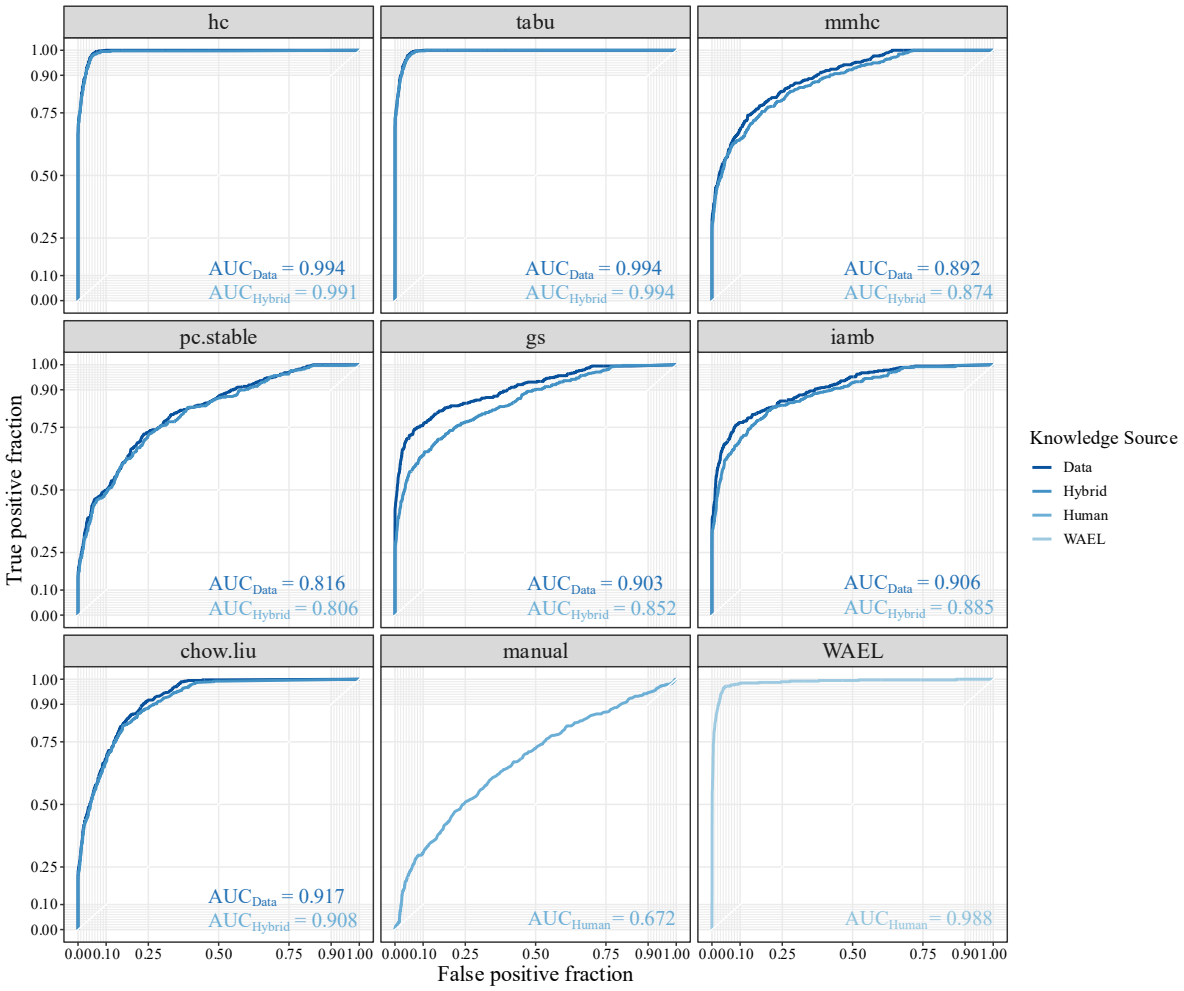


Fig. 5.9 ROC over different structure learning methods from different knowledge sources for risk prediction

The ROC curve above reveals the performance of Bayesian network classifiers developed by different structure learning methods and knowledge sources at all discrimination thresholds. Similar to the outcome from the accuracy chart, the networks built solely from data bring about more excelling performance than other knowledge sources, except for score-based algorithms. In addition, tabu search and hill-climbing continue to dominate the model performance amid all structure learning methods across all possible classification thresholds with an AUC of 0.994. Whilst human-built model leads to poor performance at an AUC of 0.672. Peculiarly, WAEL rises a relatively large AUC of 0.988 following the score-based algorithms (Fig. 5.9). This hints that WAEL is an accurate and robust classifier across a wide range of classification cut-offs.

5.3.2 Robustness of Ensembled BN

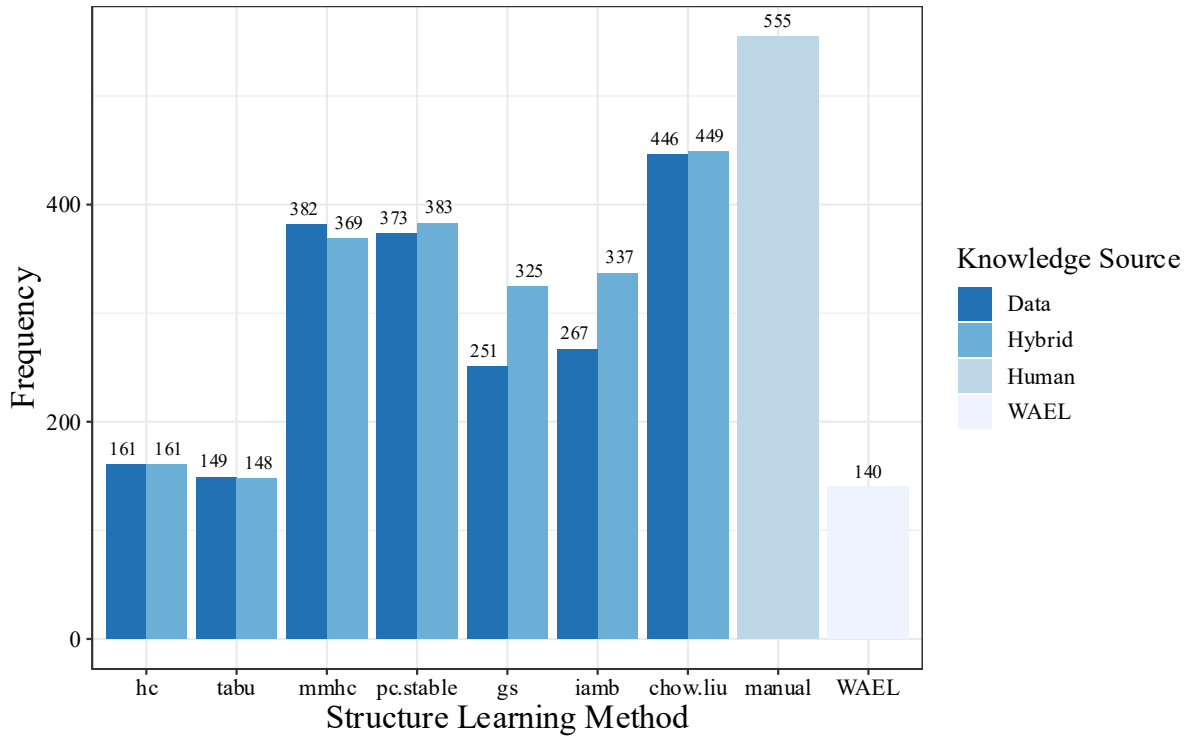


Fig. 5.10 Worst model frequency for risk prediction over different structure learning methods and different knowledge sources

As reasoned in Section 5.2.2, stable and robust models are essential and mandatory in real-world industry. Thus, the robustness of the models is also examined in the task of classifying whether a job will be high-risk (high-reject rate) given the assignable attributes. Again, the models of interest are put through vigorous testing across the data sample of 6792 jobs to check the strength of different models. Here, the model strength is reflected through the number of times that each model produces an incorrect classification. The more frequent the occurrence of misclassification, the less reliable the model. As Fig. 5.10 displays, WAEL remains to be the most robust model. Whereas tabu search algorithm also follows tightly as a strong candidate for classification problems. Therefore, a Bayesian network using tabu search or WAEL technique will be recommended for a classification problem.

Chapter 6

Conclusions and Future Work

This chapter summarises the achievements and the findings of this project in Section 6.1. We have also suggested possible future work for this research in Section 6.2.

6.1 Conclusions

This project addresses the product quality problem in manufacturing where RCA is in urgent demand to identify the reasons behind a large number of defects for quality and productivity improvement. However, RCA remains to be a challenging task in the concurrent industry due to the heavy reliance on expert knowledge from conventional methods. A variety of modern RCA methods have been developed by scholars. Whereas most of them lack the ability to offer robust, intelligent and illustrative probabilistic reasoning. As a result, a product-wise framework of ensembled Bayesian networks has been proposed in this study with the objective to provide a robust, intelligent, and human-interpretable probabilistic reasoning for RCA in manufacturing.

To fulfil the objective, three research questions for RCA have been asked. This research has then implemented the proposed method on the data from a real-world industry to answer these questions. Firstly, interpretable causal graphs have been elicited automatically from the historical data to provide insights for factory workers (Q1). Secondly, root-cause analysis has been performed on the problematic jobs to identify the reject reasons with probabilities (Q2). Lastly, we have predicted the potential high-reject jobs prior to execution to evade risk proactively (Q3).

By conducting the experiments on a sample of 6791 jobs, we have found out that the proposed method of ensembled BNs shows a prominent advantage in robustness and accuracy amidst all the models in inferring accurate probabilistic reasoning. It has also been discovered that among all the constituent structure learning algorithms, tabu search exhibits the highest accuracy in RCA followed by hill-climbing algorithm. The methods using hybrid knowledge sources outperform the ones using purely human or purely data knowledge in inferring root-cause probabilities. In the task of classifying risky jobs, the proposed method remains to be the strongest and most stable model, and tabu search persists to dominate the performance accuracy.

In summary, this project is considered successful in that it achieved the goal of providing a robust, accurate and interpretable probabilistic reasoning method for RCA to support manufacturers with data-driven decision-making under the circumstance of quality failures. This research has also overcome the issue of unrobust BN models in the existing methods using ensemble learning techniques. A comparison between different learning algorithms for BN has been presented to provide guidance on model selection and enhancement. Lastly, an evaluation method has been designed to assess the accuracy of predicted sequences of probabilities.

6.2 Future Work

Future work can be undertaken to further reinforce this research. Additional features of machinery measurements (e.g., vibrations) and environmental conditions (e.g., humidity) can be involved to provide a more comprehensive knowledge representation of the industry. Research direction in developing delicate parameter calibration models can be explored to fine-tune the parameters for BN and ensemble learning techniques such as the resampling time of bagged BN and the weights of WAEL. Lastly, more product-tailored prior knowledge can be introduced to fit each model adaptively.

References

[1] F. Psarommatis, S. Prouvost, G. May, and D. Kiritsis, "Product Quality Improvement Policies in Industry 4.0: Characteristics, Enabling Factors, Barriers, and Evolution Toward Zero Defect Manufacturing," *Front. Comput. Sci.*, vol. 2, no. August, pp. 1–15, 2020, doi: 10.3389/fcomp.2020.00026.

[2] L. Rokach and D. Hutter, "Automatic discovery of the root causes for quality drift in high dimensionality manufacturing processes," *J. Intell. Manuf.*, vol. 23, no. 5, pp. 1915–1930, 2012, doi: 10.1007/s10845-011-0517-5.

[3] D. Okes, "Improve your root cause analysis. Manufacturing Engineering," in *Root Cause Analysis : the Core of Problem Solving and Corrective Action*, 2005, pp. 171–178.

[4] Y. Y. Wee, W. P. Cheah, S. C. Tan, and K. Wee, "A method for root cause analysis with a Bayesian belief network and fuzzy cognitive map," *Expert Syst. Appl.*, vol. 42, no. 1, pp. 468–487, 2015, doi: 10.1016/j.eswa.2014.06.037.

[5] L. et Al., "Root cause analysis of failures and quality deviations in manufacturing using machine learning." 2018.

[6] J. Yu and M. M. Rashid, "A novel dynamic bayesian network-based networked process monitoring approach for fault detection, propagation identification, and root cause diagnosis," *AIChE J.*, vol. 59, no. 7, pp. 2348–2365, Jul. 2013, doi: 10.1002/aic.14013.

[7] S. He, Z. He, G. Alan, and L. Li, "Quality Improvement using Data Mining in Manufacturing Processes," *Data Min. Knowl. Discov. Real Life Appl.*, no. January, 2009, doi: 10.5772/6459.

[8] A. K. Choudhary, J. A. Harding, and M. K. Tiwari, "Data mining in manufacturing: A review based on the kind of knowledge," *J. Intell. Manuf.*, vol. 20, no. 5, pp. 501–521, 2009, doi: 10.1007/s10845-008-0145-x.

[9] F. Tsung, "Statistical monitoring and diagnosis of automatic controlled processes using dynamic PCA," *Int. J. Prod. Res.*, vol. 38, no. 3, pp. 625–637, Jan. 2000, doi: 10.1080/002075400189338.

[10] J. Yu and S. J. Qin, "Statistical MIMO controller performance monitoring. Part II: Performance diagnosis," *J. Process Control*, vol. 18, no. 3–4, pp. 297–319, 2008, doi: 10.1016/j.jprocont.2007.09.003.

[11] R. D. B. L.H. Chiang, E.L. Russell, "10.8 Fault Diagnosis," in *Fault Detection and Diagnosis in Industrial Systems*, 2001, pp. 146–165.

[12] L. H. Chiang, M. E. Kotanchek, and A. K. Kordon, "Fault diagnosis based on Fisher discriminant analysis and support vector machines," *Comput. Chem. Eng.*, vol. 28, no. 8, pp. 1389–1401, 2004, doi: 10.1016/j.compchemeng.2003.10.002.

[13] A. C. Raich and A. Çinar, "Multivariate statistical methods for monitoring continuous

processes: assessment of discrimination power of disturbance models and diagnosis of multiple disturbances," *Chemom. Intell. Lab. Syst.*, vol. 30, no. 1, pp. 37–48, 1995, doi: 10.1016/0169-7439(95)00035-6.

[14] A. AlGhazzawi and B. Lennox, "Monitoring a complex refining process using multivariate statistics," *Control Eng. Pract.*, vol. 16, no. 3, pp. 294–307, 2008, doi: 10.1016/j.conengprac.2007.04.014.

[15] A. Detzner, R. Ruckschlos, and M. Eigner, "Root-Cause Analysis with Interactive Decision Trees," *Proc. Int. Conf. Inf. Vis.*, vol. 2020-September, pp. 322–327, Sep. 2020, doi: 10.1109/IV51561.2020.00060.

[16] J. Yang, Y. Zhang, and Y. Zhu, "Intelligent fault diagnosis of rolling element bearing based on SVMs and fractal dimension," *Mech. Syst. Signal Process.*, vol. 21, no. 5, pp. 2012–2024, 2007, doi: 10.1016/j.ymssp.2006.10.005.

[17] T. Han, D. Jiang, Q. Zhao, L. Wang, and K. Yin, "Comparison of random forest, artificial neural networks and support vector machine for intelligent diagnosis of rotating machinery," *Trans. Inst. Meas. Control*, vol. 40, no. 8, pp. 2681–2693, 2018, doi: 10.1177/0142331217708242.

[18] S. Yoshimura and A. S. Jovanovic, "Analyses of Possible Failure Mechanisms and Root Failure Causes in Power Plant Components Using Neural Networks and Structural Failure Database," 1996. [Online]. Available: http://asmedigitalcollection.asme.org/pressurevesseltech/article-pdf/118/2/237/5645057/237_1.pdf.

[19] S. Dey and J. A. Stori, "A Bayesian network approach to root cause diagnosis of process variations," *Int. J. Mach. Tools Manuf.*, vol. 45, no. 1, pp. 75–91, Jan. 2005, doi: 10.1016/j.ijmachtools.2004.06.018.

[20] D. Kasper, G. Weidl, T. Dang, G. Breuel, A. Tamke, and W. Rosenstiel, "Object-oriented Bayesian networks for detection of lane change maneuvers," *IEEE Intell. Veh. Symp. Proc.*, no. February 2014, pp. 673–678, 2011, doi: 10.1109/IVS.2011.5940468.

[21] A. Alaeddini and I. Dogan, "Using Bayesian networks for root cause analysis in statistical process control," *Expert Syst. Appl.*, vol. 38, no. 9, pp. 11230–11243, Sep. 2011, doi: 10.1016/j.eswa.2011.02.171.

[22] B. G. Marcot and T. D. Penman, "Advances in Bayesian network modelling: Integration of modelling technologies," *Environ. Model. Softw.*, vol. 111, no. March 2018, pp. 386–393, 2019, doi: 10.1016/j.envsoft.2018.09.016.

[23] G. Weidl, A. L. Madsen, and S. Israelson, "Applications of object-oriented Bayesian networks for condition monitoring, root cause analysis and decision support on operation of complex continuous processes," *Comput. Chem. Eng.*, vol. 29, no. 9, pp. 1996–2009, 2005, doi: 10.1016/j.compchemeng.2005.05.005.

[24] H. Njah and S. Jamoussi, "Weighted ensemble learning of Bayesian network for gene regulatory networks," *Neurocomputing*, vol. 150, no. PB, pp. 404–416, 2015, doi:

10.1016/j.neucom.2014.05.078.

[25] T. G. Dietterich, “Ensemble methods in machine learning,” in *Multiple Classifier Systems*, 2002.

[26] M. Li, R. Zhang, and K. Liu, “A New Ensemble Learning Algorithm Combined with Causal Analysis for Bayesian Network Structural Learning,” vol. 12, 2020, doi: 10.3390/sym12122054.

[27] Y. Zhang, J. Ji, and B. Ma, “Fault diagnosis of reciprocating compressor using a novel ensemble empirical mode decomposition-convolutional deep belief network,” *Meas. J. Int. Meas. Confed.*, vol. 156, 2020, doi: 10.1016/j.measurement.2020.107619.

[28] W. E. Wong, R. Gao, Y. Li, R. Abreu, and F. Wotawa, “A survey on software fault localization,” *IEEE Trans. Softw. Eng.*, vol. 42, no. 8, pp. 707–740, 2016, doi: 10.1109/TSE.2016.2521368.

[29] D. A.M., “A statistical comparison of three root cause analysis tools,” *J. Ind. Technol.*, vol. 20, no. November, p. 20, 2004, [Online]. Available: http://www.scopus.com/inward/record.url?eid=2-s2.0-2942703888&partnerID=40&md5=2a3c06d507fa5c75fd16892b4319b952.

[30] L. H. Chiang, E. L. Russell, and R. D. Braatz, “Fault diagnosis in chemical processes using Fisher discriminant analysis, discriminant partial least squares, and principal component analysis,” *Chemom. Intell. Lab. Syst.*, vol. 50, no. 2, pp. 243–252, 2000, doi: 10.1016/S0169-7439(99)00061-1.

[31] J. Yu, “A particle filter driven dynamic Gaussian mixture model approach for complex process monitoring and fault diagnosis,” *J. Process Control*, vol. 22, no. 4, pp. 778–788, 2012, doi: 10.1016/j.jprocont.2012.02.012.

[32] M. Chen, A. X. Zheng, J. Lloyd, M. I. Jordan, and E. Brewer, “Failure diagnosis using decision trees,” *Proc. - Int. Conf. Auton. Comput.*, pp. 36–43, 2004, doi: 10.1109/ICAC.2004.1301345.

[33] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, “Classification and regression trees,” *Classif. Regres. Trees*, pp. 1–358, Jan. 2017, doi: 10.1201/9781315139470/CLASSIFICATION-REGRESSION-TREES-LEO-BREIMAN-JEROME-FRIEDMAN-RICHARD-OLSHEN-CHARLES-STONE.

[34] L. Xu and M. Y. Chow, “A classification approach for power distribution systems fault cause identification,” *IEEE Trans. Power Syst.*, vol. 21, no. 1, pp. 53–60, Feb. 2006, doi: 10.1109/TPWRS.2005.861981.

[35] K. B. Lee, S. Cheon, and C. O. Kim, “A convolutional neural network for fault classification and diagnosis in semiconductor manufacturing processes,” *IEEE Trans. Semicond. Manuf.*, vol. 30, no. 2, pp. 135–142, May 2017, doi: 10.1109/TSM.2017.2676245.

[36] L. Abele, M. Anic, T. Gutmann, J. Folmer, M. Kleinsteuber, and B. Vogel-Heuser, *Combining knowledge modeling and machine learning for alarm root cause analysis*, vol.

46, no. 9. IFAC, 2013.

[37] J. B. Yu, X. L. Lu, and W. Z. Zong, "Wafer defect detection and recognition based on local and nonlocal linear discriminant analysis and dynamic ensemble of gaussian mixture models," *Zidonghua Xuebao/Acta Autom. Sin.*, vol. 42, no. 1, pp. 47–59, Jan. 2016, doi: 10.16383/J.AAS.2016.C150311.

[38] S. Qin and G. Wang, "A Study of Fault Detection and Diagnosis for PLC Controlled Manufacturing System," *Commun. Comput. Inf. Sci.*, vol. 326 CCIS, no. PART 1, pp. 373–382, Oct. 2012, doi: 10.1007/978-3-642-34381-0_43.

[39] M. K. Song, F. C. Lin, S. E. Ward, and J. P. Fine, "Composite Variables: When and How," *Nurs. Res.*, vol. 62, no. 1, p. 45, Jan. 2013, doi: 10.1097/NNR.0B013E3182741948.

[40] H. Bae, S. Monti, M. Montano, M. H. Steinberg, T. T. Perls, and P. Sebastiani, "Learning Bayesian Networks from Correlated Data," *Sci. Reports 2016 61*, vol. 6, no. 1, pp. 1–14, May 2016, doi: 10.1038/srep25156.

[41] M. Altermann and D. Kuhn, "A Mathematical Theory of Communication," *Bell Syst. Tech. J.*, vol. 27, no. 3, pp. 379–423, Jul. 1948, doi: 10.1002/J.1538-7305.1948.TB01338.X.

[42] J. P. Pellet and A. Elisseeff, "Using Markov blankets for causal structure learning," *J. Mach. Learn. Res.*, vol. 9, pp. 1295–1342, 2008.

[43] B. R. Cobb, R. Rumí, and A. Salmeroń, "Bayesian Network Models with Discrete and Continuous Variables," *Stud. Fuzziness Soft Comput.*, vol. 213, pp. 81–102, 2007, doi: 10.1007/978-3-540-68996-6_4.

[44] Y. Wang, Z. Wang, S. He, and Z. Wang, "A practical chiller fault diagnosis method based on discrete Bayesian network," *Int. J. Refrig.*, vol. 102, pp. 159–167, Jun. 2019, doi: 10.1016/J.IJREFRIG.2019.03.008.

[45] Y. C. Chen, T. A. Wheeler, and M. J. Kochenderfer, "Learning discrete Bayesian networks from continuous data," *J. Artif. Intell. Res.*, vol. 59, pp. 103–132, 2017, doi: 10.1613/jair.5371.

[46] R. Dash, R. L. Paramguru, and R. Dash, "Comparative analysis of supervised and unsupervised discretization techniques," *Int. J. Adv. Sci. Technol.*, vol. 2, no. 3, pp. 29–37, 2011, [Online]. Available: https://www.researchgate.net/profile/Rajashree_Dash/publication/266058863_Comparative_Analysis_of_Supervised_and_Unsupervised_Discretization_Techniques/links/55727c6b08aeacff1ffacde8.pdf.

[47] Z. Li, D. Wu, C. Hu, and J. Terpenny, "An ensemble learning-based prognostic approach with degradation-dependent weights for remaining useful life prediction," *Reliab. Eng. Syst. Saf.*, vol. 184, no. December 2017, pp. 110–122, 2019, doi: 10.1016/j.ress.2017.12.016.

[48] Y. Peng, "A novel ensemble machine learning for robust microarray data classification," *Comput. Biol. Med.*, vol. 36, no. 6, pp. 553–573, 2006, doi: 10.1016/j.compbiomed.2005.04.001.

[49] L. B.-M. learning and U. 1996, "Bagging predictors," *Springer*, vol. 45. pp. 5–32, 2001, [Online]. Available: https://link.springer.com/article/10.1007/BF00058655.

[50] S. Beretta, M. Castelli, I. Gonçalves, R. Henriques, and D. Ramazzotti, "Learning the structure of Bayesian networks: A quantitative assessment of the effect of different algorithmic schemes," *Complexity*, vol. 2018, 2018, doi: 10.1155/2018/1591878.

[51] M. Scanagatta, A. Salmerón, and F. Stella, "A survey on Bayesian network structure learning from data," *Progress in Artificial Intelligence*, vol. 8, no. 4. Springer Verlag, pp. 425–439, Dec. 01, 2019, doi: 10.1007/s13748-019-00194-y.

[52] N. Cruz-Ramírez, H. G. Acosta-Mesa, R. E. Barrientos-Martínez, and L. A. Nava-Fernández, "How Good Are the Bayesian Information Criterion and the Minimum Description Length Principle for Model Selection? A Bayesian Network Analysis," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 4293 LNAI, pp. 494–504, Nov. 2006, doi: 10.1007/11925231_46.

[53] M. Scutari, C. E. Graafland, and J. M. Gutiérrez, "Who learns better Bayesian network structures: Accuracy and speed of structure learning algorithms," *Int. J. Approx. Reason.*, vol. 115, pp. 235–253, Dec. 2019, doi: 10.1016/j.ijar.2019.10.003.

[54] I. Tsamardinos, L. E. Brown, and C. F. Aliferis, "The max-min hill-climbing Bayesian network structure learning algorithm," *Mach. Learn.*, vol. 65, no. 1, pp. 31–78, Oct. 2006, doi: 10.1007/s10994-006-6889-7.

[55] S. Renooij, "ALGORITHMS FOR DECISION SUPPORT Bayesian networks."

[56] C. Hu, B. D. Youn, P. Wang, and J. Taek Yoon, "Ensemble of data-driven prognostic algorithms for robust prediction of remaining useful life," *Reliab. Eng. Syst. Saf.*, vol. 103, pp. 120–135, 2012, doi: 10.1016/j.ress.2012.03.008.

[57] N. Mahendran *et al.*, "Sensor-Assisted Weighted Average Ensemble Model for Detecting Major Depressive Disorder," 2019, doi: 10.3390/s19224822.

[58] M. Scutari, C. E. Graafland, and J. M. Gutiérrez, "Who learns better Bayesian network structures: Accuracy and speed of structure learning algorithms," *Int. J. Approx. Reason.*, vol. 115, pp. 235–253, 2019, doi: 10.1016/j.ijar.2019.10.003.