D 2021

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

OVERLAP IN AUTOMATIC ROOT CAUSE ANALYSIS IN MANUFACTURING

EDUARDO LUÍS DE MEIRELES E OLIVEIRA
TESE DE DOUTORAMENTO APRESENTADA
À FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO EM
ENGENHARIA E GESTÃO INDUSTRIAL

UNIVERSIDADE DO PORTO

DOCTORAL THESIS

Overlap in Automatic Root Cause Analysis in Manufacturing

Author:
Eduardo Luís de Meireles E OLIVEIRA

Supervisors:
Prof. Vera L. MIGUÉIS
Prof. José L. BORGES

A thesis submitted in fulfillment of the requirements
for the degree of Doctor of Philosophy in Industrial Engineering and Management

in the

December 15, 2021

"The impediment to action advances action. What stands in the way becomes the way. "

Marcus Aurelius Antoninus

Abstract

The Manufacturing sector is highly competitive, and the management of its operations can be very complex. A critical part of managing manufacturing operations is to solve the problems that occur as quickly and efficiently as possible, while making sure the problems are solved permanently. Root Cause Analysis (RCA) is a process through which one can find the true origin of a problem. RCA is necessary to make sure that problem solving is focused on the real causes of problems, and not on their symptoms. RCA is not a trivial problem, requiring extensive system and execution analysis, which makes the process slow and inefficient. However, recently some studies have developed Automatic Root Cause Analysis (ARCA) solutions, that make use of the increasing data collection and analysis performed in manufacturing companies to improve the efficiency of the process.

This work focuses on the development of ARCA solutions, specifically on solutions with the objective of locating the root cause in a manufacturing process. It focuses on overlap, an issue that occurs when analyzing the data of the logistics of a manufacturing process. Overlap happens when all products that go through a certain machine in a manufacturing step are all processed in a given machine in a later manufacturing step. This manifests itself in the data with synchronised factors, and leads to a scenario where one is not able to distinguish what were the most influencing factors in the occurrence of a problem, as the data is perfectly synchronised. Such situation is problematic as it becomes extremely hard to detect the true root causes from the data, and constitutes a hindrance to the development of ARCA solutions. To tackle this issue, this work defines overlap, and proposes three ways of measuring it, with increasing levels of refinement. This led to several approaches that can be used to develop ARCA solutions that are resilient to overlap. The three measures are based on: i) Data Mining/Machine Learning; ii) Information Theory; iii) Causal Inference.

To validate the proposed approaches, several experiments were conducted using simulated data and data from a real case-study in semiconductor manufacturing. The developed approaches have been shown to be able to greatly reduce the number of possible root causes, and even of finding the single true root cause when the data available makes it possible to do so.

The contributions of this work are: i) an overview and conceptualization of the ARCA approaches in manufacturing; ii) the identification and definition of overlap, providing three ways of measuring it; iii) the development of several validated ARCA solutions based on the proposed measures; iv) the identification of the conditions where it is possible to identify the location of the true root cause in the presence of overlap; v) the identification of the effects of noisy data on the performance of ARCA solutions.

Resumo

A manufatura é uma indústria altamente competitiva, e a gestão das suas operações pode ser muito complexa. Algo crítico na gestão das operações de manufatura é a rápida e eficiente resolução dos problemas, assegurando que estes são resolvidos de forma permanente. A Análise de Causas Raiz (ACR) é um processo que permite descobrir a verdadeira origem do problema. A ACR é necessária para assegurar que a resolução dos problemas se foca nas causas reais, e não apenas nos seus sintomas. Contudo, a ACR requer uma análise exaustiva dos sistemas e da sua execução, o que torna este processo lento e ineficiente. Todavia, estudos recentes desenvolveram soluções de Análise Automática de Causas Raiz (AACR), aproveitando as crescentes quantidades de dados existentes nas empresas de manufatura, de modo a melhorarem a eficiência do processo.

Este trabalho foca-se no desenvolvimento deste tipo de soluções, nomeadamente em descobrir a localização das causas raiz de problemas no processo de manufatura. Concentra-se ainda numa situação que ocorre ao analisar os dados logísticos dos processos de manufatura. Denominada de sobreposição, esta situação ocorre quando todos os produtos que passaram por uma determinada máquina numa etapa do processo são todos eles processados numa outra máquina, noutra etapa de manufatura. Isto manifesta-se nos dados com fatores que se tornam sincronizados, levando a um cenário onde não é possível distinguir quais os fatores com mais influência no problema. Isto é prejudicial, uma vez que se torna extremamente difícil detetar causas raiz a partir dos dados, o que constitui um obstáculo para o desenvolvimento de soluções de AACR. Para abordar esta questão, este trabalho propõe três formas de medir a sobreposição, com níveis crescentes de refinação, levando ao desenvolvimento de abordagens AACR resilientes à sobreposição. Estas três perspetivas são baseadas em: i) Mineração de Dados; ii) Teoria da Informação; iii) Inferência Causal.

Para validar as abordagens propostas, foram realizadas várias experiências usando dados simulados e reais, baseados num estudo de caso na manufatura de semicondutores. As abordagens desenvolvidas demonstraram capacidade de reduzir consideravelmente o número de causas raiz que têm de ser analisadas, sendo inclusivamente capazes de encontrar a verdadeira causa raiz, quando os dados disponíveis o permitem.

As contribuições deste trabalho são: i) uma estruturação e conceptualização das abordagens de AACR sobre manufatura; ii) a identificação da questão da sobreposição, providenciando três métricas; iii) o desenvolvimento e validação de várias soluções de AACR baseadas nas métricas propostas; iv) a identificação das condições onde é possível determinar a localização da verdadeira causa raiz mesmo na presença de sobreposição; v) a identificação dos efeitos do ruído nos dados na performance das soluções de AACR.

Acknowledgements

I would first like to thank my supervisors, Prof. Vera and Prof. José Luís for the motivation and guidance provided, as well as the patience demonstrated in this somewhat long journey with many twists and turns.

I would also like to give thanks to all the professors who, despite not being supervisors, helped me with their advice along the way, namely Prof. Henriqueta Nóvoa, Prof. Rui Camacho and Prof. Marco Reis.

I would like thank the company of my colleagues in INESC and DEGI, who reminded me I was not alone in this process. A special thanks to Sara for her advice as a more experienced PhD student, and the members of the DEGI Club/IEMS committee for the fun times and help in organizing the events of the department. Also a special thanks to my friend Thomy, with whom I have shared in many conversations the many woes and joys of being a PhD student.

I would like to thank Helena, Isabel, Soledade and the rest of the staff in both INESC TEC and DEGI for the support on the not so pleasant, but required, bureaucracies I had to face.

A big thanks to all my friends, who provided a much needed emotional support throughout this long journey. I will not mention their names to make sure I do not do an injustice to anyone, and because I was told I should keep my acknowledgements to a single page.

Finally, and most importantly, a deep gratitude to my parents, who have supported me for many years, and guided me through a path of search for knowledge and wisdom with their example and words.

This work was financed by National Funds through the Portuguese funding agency, FCT - Fundação Ciência e Tecnologia within project: SFRH/BD/138228/2018. It was also partially done under Project "TEC4Growth - Pervasive Intelligence, Enhancers and Proofs of Concept with Industrial Impact/NORTE-01-0145-FEDER-000020", which was financed by the North Portugal Regional Operational Programme (NORTE 2020), under the PORTUGAL 2020 Partnership Agreement, and through the European Regional Development Fund (ERDF)

Contents

Abstract iii

Resumo v

Acknowledgements vii

1 Introduction 1
- 1.1 General Context 1
- 1.2 Motivation and General Objective 3
- 1.3 Specific Objectives 5
- 1.4 Thesis Structure & Outline 6

2 Background Knowledge 11
- 2.1 Introduction 11
- 2.2 Root Cause Analysis 11
  - 2.2.1 Traditional Techniques 12
  - 2.2.2 Discussion 17
- 2.3 Data Mining & Machine Learning 17
  - 2.3.1 Data Preparation & Exploration 18
  - 2.3.2 Supervised Learning 24
  - 2.3.3 Unsupervised Learning 30
  - 2.3.4 Discussion 34
- 2.4 Control Charts 35
- 2.5 Causal Inference 38
- 2.6 Discussion & Conclusions 43

3 Problem & Case Study Definition 45
- 3.1 Introduction 45
- 3.2 Problem Definition 45
- 3.3 Case Study 48
  - 3.3.1 Semiconductor Industry 48
  - 3.3.2 Semiconductor Packaging Process 49
  - 3.3.3 Redistribution Layer (RDL) Process 51
- 3.4 Stochastic Simulator 54
- 3.5 Discussion & Conclusions 58

4 Automatic Root Cause Analysis in Manufacturing: An Overview & Conceptualization 59
- 4.1 Introduction 59
- 4.2 Types of Data and Root Cause Levels 62
- 4.3 Methodologies 67
  - 4.3.1 **Factors ⇒ Problem** 68

4.3.2 **Factors ⇒ RC** . . . 71
4.3.3 **RC ⇒ RC** . . . 72
4.3.4 Summary . . . 72
4.4 Evaluation Measures . . . 75
4.5 Gaps & Research Opportunities . . . 78
4.6 Managerial Insights . . . 80
4.7 Conclusions . . . 81

5 On the Influence of Overlap in Automatic Root Cause Analysis in Manufacturing **83**
5.1 Introduction . . . 83
5.2 Overlap Operationalisation . . . 84
5.3 Proposed Solution & Problematic Moment Identification Stage . . . 86
5.4 Factor Ranking Algorithms . . . 87
5.4.1 Co-Occurrences . . . 88
5.4.2 Chi-Square . . . 89
5.4.3 Random Forest . . . 90
5.5 Experimental Setup . . . 91
5.5.1 Mockup Data . . . 92
5.5.2 Stochastic Simulation Data . . . 93
5.5.3 Real Case Study Data . . . 94
5.6 Experimental Results . . . 94
5.6.1 Mockup Data . . . 94
5.6.2 Stochastic Simulation Data . . . 96
5.6.3 Real Case Study . . . 98
5.7 Discussion . . . 100
5.8 Conclusions . . . 102

6 Overlap in Automatic Root Cause Analysis in Manufacturing: An Information Theory-based Approach **105**
6.1 Introduction . . . 105
6.2 Proposed Methodology . . . 106
6.2.1 Motivation . . . 106
6.2.2 Proposed Measure . . . 106
6.2.3 Proposed Approach . . . 109
6.3 Evaluation . . . 111
6.3.1 Evaluation Setup . . . 111
6.3.2 Evaluation Results . . . 112
6.4 Discussion . . . 114
6.5 Conclusions . . . 116

7 Understanding Overlap in Automatic Root Cause Analysis in Manufacturing Using Causal Inference **117**
7.1 Introduction . . . 117
7.2 Methodology . . . 118
7.3 Evaluation . . . 123
7.3.1 Evaluation Setup . . . 123
7.3.2 Evaluation Results . . . 125
7.4 Discussion . . . 128
7.5 Conclusions and Future Research . . . 128

8 Conclusions and Future Research **131**

8.1 Research Question Conclusions 131

8.2 Contributions 135

8.3 Managerial Insights 136

8.4 Limitations & Future Research 138

Bibliography **141**

List of Figures

2.1 An example of a Pareto chart. 12
2.2 An example of an Ishikawa diagram. 14
2.3 An example of a Failure Mode and Effects Analysis. 15
2.4 An example of a Fault Tree Analysis. 16
2.5 An example of the principle components resulting from PCA, and the amount of variance they explain. 23
2.6 An example of a decision tree after training. 25
2.7 An example of a FP tree. 31
2.8 An example of a CUSUM control chart for upward drift. 36
2.9 An example of an EWMA control chart for upward drift. 37
2.10 Example of a causal graph/model. 39
2.11 Example of the causal graph in Figure 2.10 after the intervention $do(X = x_i)$. 40

3.1 An illustration of a process that generates Location-Time type of data. 46
3.2 High level representation of the semiconductor manufacturing industry. 48
3.3 Fabricated wafer. Each square in the round wafer corresponds to a die (image property of Amkor Portugal). 49
3.4 Schematics of a cross-section of a chip packaged trough Fan-Out WLP (image property of Amkor Portugal). 50
3.5 Wafer-Level Packaging Process. 50
3.6 The different layers of the RDL process. 51
3.7 The different sub-processes of each layer of the RDL process. 53
3.8 Diagram of the stochastic simulator structure. 55

4.1 The types of data/root cause levels that can occur when developing ARCA solutions for manufacturing. 62

5.1 Flowchart detailing the proposed ARCA solution. 86
5.2 Root Causes defined in the datasets obtained through stochastic simulation. 94
5.3 Results of the Mockup experiments when the root cause is one of the overlapped factors. 95
5.4 Results of the Mockup experiments when the root cause is not one of the overlapped factors. 95
5.5 Comparison between Overlap$_{Avg}$ and Overlap$_{Count}$ behavior with the data from the Real case study experiment. 101

6.1 Detail of an example of visualization of PMI overlap among different tuples in Stochastic Simulation data. 111
6.2 Results of the Mockup experiments when the root cause is in one of the overlapped factors. 112

6.3 Results of the Mockup experiments when the root cause is not in one of the overlapped factors. 112

7.1 Causal Model of overlap. 119
7.2 Overlap model of the example in Figure 3.1. 120
7.3 Causal Model of overlap used to generate mockup datasets. 123

List of Tables

2.1 Example of dataset for supervised learning. 24
2.2 Example of a dataset for association rule mining. 30

3.1 Example of data in a Location-Time type of problem. 47
3.2 Parameters of the stochastic simulator. 56

4.1 Table detailing the number of papers per manufacturing industry. . . . 61
4.2 Classification of the types of data used in each paper, divided by industry. 64
4.3 Summary of the number of papers using each technique, per solution category. 68
4.4 Classification how each paper extracts root causes from the techniques, divided by category and industry. 73
4.5 Summary of what techniques are used by each paper. 74
4.6 Classification of the papers based on the types of data used and category of solutions. 74
4.7 Classification of the papers by type of evaluation measures. 76
4.8 Classification of the papers by evaluation type used and category of solutions. 78
4.9 Classification of the papers by type of data and the evaluation type used. 79

5.1 Table used to compare the occurrence of problems in a Step-Machine combination. 89
5.2 Parameters' values of the mockup datasets generated. 92
5.3 List of the problematic moments identified for each of the three stochastic simulation dataset. 96
5.4 Table with the results of the RCA algorithms for the Stochastic Simulation data. 98
5.5 Table summarizing the problematic moments identified in the Real case study dataset, as well as the common root-causes among the different algorithms proposed. 99

6.1 Example of Table 3.1 transformed with one-hot encoding, before computing the PMI values. 109
6.2 Matrix with the PMI values of the interactions between tuples in the example of Tables 3.1 and 6.1. 109
6.3 Table with the results of the RCA algorithms with Stochastic Simulation data. 113
6.4 Table summarizing the problematic moments identified in the Real Case Study dataset, as well as the common root-causes among the different algorithms proposed. 115

7.1 Parameters' values of the mockup datasets generated. 124
7.2 Example of a small mockup dataset. 124
7.3 Metadata of the Simulated and case study datasets. 125
7.4 Results of the interventional queries proposed and of the PMI method, when varying the values of ϵ_u. 125
7.5 Results of the interventional queries proposed and of the PMI method, when varying the values of ϵ_l. 126
7.6 Results from the Simulation and case study datasets. 127

List of Abbreviations

AOI	**A**utomatic **O**ptic **I**nspection
ARCA	**A**utomatic **R**oot **C**ause **A**nalysis
AR	**A**ssociation **R**ules
BD	**B**ack-**D**oor (criterion)
CUSUM	**Cu**mulative **Sum**
DM	**D**ata **M**ining
DT	**D**ecision **T**ree
EWMA	**E**xponentially **W**eighed **M**oving **A**verage
I/O	**I**nput/**O**utputs
ML	**M**achine **L**earning
NN	**N**eural **N**etwork
NOC	**N**ormal **O**perating **C**onditions
LBS	**L**aser **B**all **S**oldering
OSAT	**O**utsourced **S**emiconductor **A**ssembly and **T**est
PCA	**P**rincipal **C**omponent **A**nalysis
PCM	**P**rocess **C**ontrol **M**onitoring
PMI	**P**ositive **M**utual **I**nformation
RCA	**R**oot **C**ause **A**nalysis
RDL	**R**e**d**istribution **L**ayer
RF	**R**andom **F**orest
RIE	**R**eactive **I**on **E**tching
SVM	**S**upport **V**ector **M**achine
WLP	**W**afer **L**evel **P**ackaging

Chapter 1

Introduction

1.1 General Context

Manufacturing is any industry that makes products through the transformation of raw materials or assembly of components, with the use of manual labor and/or machinery, on a large scale. The manufacturing sector plays a major role in the global economy, specially in developing countries. In the European Union, almost 2 millions manufacturing enterprises employ more than 28,5 million people (European Commission, 2020). However, manufacturing has been undergoing changes, especially in developed and heavily industrialized countries, with an increasing surge of digitisation of the manufacturing environments, and the advent of Industry 4.0 (Alcácer and Cruz-Machado, 2019).

Industry 4.0 refers to a German manufacturing strategy to bring forth a new industrial revolution, in which the focus is to increase competitiveness by reducing costs and increasing the production flexibility and product customization (Alcácer and Cruz-Machado, 2019). This is achieved through the use of digital technologies that provide a plethora of tools to increase the amount of information and knowledge about the manufacturing processes. The knowledge obtained enables better control of the processes, increased automation and better response time to both costumers' needs and to problems that may surge in the manufacturing process. It is within this last aspect that this doctoral work is framed, as its aim is to develop solutions that improve the efficiency of diagnosis, and therefore the response time to problems.

Even when it is not possible to immediately advance to an Industry 4.0 solution, it is possible to use an Industry 3.5 framework, a hybrid strategy between Industry 3.0 and to-be Industry 4.0 (Chien, Hong, and Guo, 2017; Chien, Lin, and Lin, 2020; Ku, Chien, and Ma, 2020). The recent increase in data availability and the development of solutions that make use of such data is motivated by heavy competition (Choudhary, Harding, and Tiwari, 2009) and decreasing profit margins, which leads companies to search for solutions that reduce costs of their manufacturing processes or add value to their products.

A major component in reducing costs and at the same time improving customers' satisfaction is to reduce the number of defective products. This is one of the concerns of quality management, which aims to ensure that a consistent product is delivered. By reducing the number of defective products, not only those which reach the customers but also those within the manufacturing process, it is possible to cut costs in both wasted materials and time spent by human resources dealing with those defects.

In order to reduce the number of defective products caused by the occurrence of unexpected problems, it is necessary to solve problems as quickly as possible, while at the same time making sure they do not reoccur. To do so, it is crucial to find the true origin of the problem, i.e., the root cause. Root Cause Analysis (RCA) is a process of analysis to understand the causal mechanism behind a change from a desirable state to an undesirable one, in order to keep a problem from recurring (Ong, Choo, and Muda, 2015). RCA allows companies to focus on how to fix the problem rather than on its symptoms (Radziwill, 2019).

However, RCA can be a very complex process, requiring extensive system and execution analysis (Steinhauer et al., 2016). This makes the process very time-consuming, and can lead to a long time between a problem occurring and it being permanently solved. As such, it is desirable to increase the efficiency of the RCA process. Such has led to the development of solutions that use the increased digitisation and data availability in manufacturing companies to automatize at least part of the process, making the task of the human analyst easier, more focused, and more efficient. These solutions can be denominated Automatic Root Cause Analysis (ARCA).

Project

This thesis arises in the context of a project in semiconductor manufacturing. The aim of that project was to develop solutions that reduced the time spent by the analysts in finding the causes for problems in the manufacturing process. As the quality levels required in the semiconductor manufacturing sector are very high, it is necessary that the number of defective products is very low, and also that any systematic decrease in the quality is detected and solved as quickly as possible.

The production process in semiconductor manufacturing is composed by several steps, in which machines use lithography and other chemical reactions to generate very complex circuitry. The manufacturing process in this project is organised in a cellular layout, but the work developed on this thesis is also applicable in manufacturing processes with a process or product layout.

The main problem affecting the process at the time of project was the problem of overkill. Given the amount of product that is produced and the existing high quality standards, automatic optic inspection is used to filter the products according to their quality. Only products with an indication of a possible defect are manually inspected. However, sometimes a product can have no defect, and still be selected as a potential defect by the automatic optic inspection. This is named overkill. It can be caused, for example, by the product becoming darker than usual, which when compared with the standard used to control quality, leads to false positives. An abrupt increase in the number of false positives, as was seen in the company prior to the project, leads to a a rise in the amount of manual inspections, which in turns increases the cycle time of production. As such, it is desirable to keep overkill to a minimum, in order to minimize human resource costs allocated to inspection, and also the cycle time.

As the company was also undergoing an improvement in their sensorization capability, the project that contextualizes this thesis arises as a necessity to reduce the workload on the specialized human resources that are responsible for discovering the root causes of the problems. However, as the improvement in sensorization was in an embryonic stage, the only type of data available about the whole process pertained the routes that each product went trough in the manufacturing process, namely in which machines were performed each step, and when. The case study that represents this project is described in greater detail in Section 3.3.

1.2 Motivation and General Objective

The doctoral work described in this thesis is developed within the context described above. The development of ARCA solutions that are more efficient and are flexible enough to deal with diverse scenarios is critical to the improvement of the management of manufacturing operations.

Due to the increase in data and complexity of production processes, it is increasingly hard for experts to rely only on their knowledge of the process or on traditional engineering solutions, as these are not adequate to process such high volume of information, which is also highly dimensional. Traditional RCA solutions (e.g., Ishikawa diagram, Failure Mode and Effects Analysis) focus on organizing and presenting data in a way that facilitates reading and reasoning about it. However, these solutions still require considerable user input, and do not scale well with the number of factors to consider. ARCA solutions, based on data mining and machine learning algorithms, focus on automatically discovering and associating factors and data patterns to problems. These approaches scale well with the number of factors, and can heavily reduce the amount of information to consider by the analyst, which improves the speed and efficiency of finding the root cause of problems.

As such, this work has as a general objective the understanding on how ARCA solutions work, how can they be improved, not only in terms of efficiency and root cause detection performance, but also how to develop solutions that can be used in a vast array of situations, where the sources of data are limited, or the data is noisy.

Such work is of interest in order to provide ARCA solutions to more manufacturing companies, with different levels of digitisation of their manufacturing processes.

In the case study in semiconductor manufacturing that originated this doctoral work, there was only access to data about where the products were processed in each step, and when they were processed. However, the manufacturing process had enough complexity to generate a problem that traditional RCA methods could not find a root cause efficiently. This led to the need to develop solutions that increased the efficiency of the RCA, but which were limited in the data that could be used.

Three types of data exist when developing ARCA solutions for manufacturing. The data can be Location-Time data, Physical data, and Log-Action data (these are described in more detail in Chapter 4). This work is focused on understanding and developing ARCA solutions when the only data available is Location-Time data. Such type of data pertains the internal logistics of the manufacturing process: at which set of machines each product was processed and at which time. Location-Time is the the first level of data that is possible to obtain via digitisation, and therefore it is easier to obtain, and solutions using this type of data can be applied to more manufacturing companies.

Some factories do not have the required infrastructure in terms of sensors to collect the data that enables the development of solutions that automatically detect the physical nature of the root cause. However, most factories have information on the flow of the product through the manufacturing process. Data about this flow can be used to determine the location of root causes. When the manufacturing process is particularly complex (e.g., in semiconductor manufacturing (Lima et al., 2021)), even this data of the production flow can become highly dimensional and complex to analyse, which motivates the use of data mining techniques in the development of solutions to aid in diagnosis.

While researching towards the general objective described above, the presence of a certain issue became evident. Most of the times, location-time data from real scenarios have situations when groups of products go through the exact same machines in certain steps of the manufacturing process. This creates a dataset where it is impossible to distinguish the influence of each individual machine on the quality of the product. In other words, there are occurrences of local synchronicity in the manufacturing process that lead to overlap in the data. This makes it impossible to detect which of the machines in that group is the real root cause of the problem. This issue was denominated overlap, as the data on the machines "overlap" each other, and they cannot be separated within the dataset. It is clear that the presence of overlap can be a critical impediment to the process of RCA, specially when trying to develop automatic solutions for it, as these need to take into consideration such phenomenons, in order to surpass them. As such, the focus of this doctoral work became to study overlap, what it is, how does it influence the development of ARCA solutions, and how can the issue be dealt with.

1.3 Specific Objectives

In more detail, the objectives of this work can be divided in three research questions:

- **R.Q. 1**: How can we structure and conceptualize the solutions/approaches on ARCA in manufacturing?
- **R.Q. 2**: What is overlap, and how can it be defined and measured?
- **R.Q. 3**:What challenges does overlap bring to the development of ARCA solutions using Location-Time data?

Research question **R.Q. 1** focuses on the need to understand how ARCA solutions in manufacturing can be organised in a way it is possible to understand the commonalities and dissimilarities among the different approaches. It is necessary to understand how these solutions/approaches are related to each other, in order to have a robust guidance in the development of ARCA solutions. This structured knowledge is not available in the literature. The knowledge of what type of solution one can develop with the data available, what are the techniques available to do so, and how can root causes be extracted from these techniques are all relevant in the development of ARCA solutions in manufacturing. This research question is tackled in Chapter 4, where the existing approaches are structured and conceptualised, in order to provide guidance to the remaining of the doctoral work, and to future works in this topic. Please note that the terms "ARCA approach" and "ARCA solution" are used interchangeably throughout this thesis.

The question **R.Q. 2** represents the main focus of this doctoral work, i.e., the study of overlap. It is intended to define what this issue represents, and how it can be measured. As overlap has never been identified before in the literature, there is a need to define it in such a way that its meaning is clear, and that it is possible to measure it, in order to understand when and how it occurs. It is also necessary to be able to implement this definition and these measures in automatic solutions, in order to improve the efficiency of RCA as much as possible. This question is tackled in Chapters 5, 6, and 7, where in each of these chapters overlap is considered from a different perspective, and the definition and measures of overlap become increasingly refined with the advancement of the chapters.

Finally, **R.Q. 3** directions the study to the practical effects of overlap in the development of ARCA using Location-Time data. Which are the limitations it introduces, and in which conditions can we detect root causes even in its presence. Given the identification of overlap and the first notions of the impact that it can have on automated diagnosis solutions, it is of the utmost importance to fully understand the effects that overlap can have in the context of ARCA in manufacturing. This is required in order to grasp how should the solutions be developed in order to make them more robust to the issue of overlap and its consequences. This question is answered in Chapters 5, 6, 7. Chapter 5 presents the challenge created by overlap in the use of classifiers for diagnosis in Location-Time data, and proposes the use of factor ranking algorithms to circumvent this issue. Chapter 6 introduces refinements to the previous measure, which lead to improved performance of the developed solutions. Chapter 7 studies in detail how overlap hinders the diagnosis of the true root cause of a problem, by determining the overlap limit after which it is impossible to determine the true root cause of a problem.

1.4 Thesis Structure & Outline

The first chapters contain the background knowledge and problem definition, followed by the main chapters of the thesis that answer the research questions mentioned in the previous section. The main chapters are Chapters 4, 5, 6, and 7, and these chapters originated four papers. All chapters have a discussion section that summarizes the main conclusions, and frames the chapter within the doctoral work, and its contributions to the literature.

Chapter 2 introduces the background knowledge required to understand this thesis. The first section (after the introduction) explains the basics of root cause analysis, as well as the traditional techniques that are usually used. The shortcomings of these techniques are discussed, as well as some ideas used in the main chapters of the thesis that were based on the traditional techniques. Then, an overview of data mining and machine learning techniques used or mentioned in this doctoral work is presented in Section 2.3. It is also taken the opportunity to discuss the existing types of analytic categories. Such discussion is relevant for this work, as the category that it focuses on is different from the categories of the techniques that are generally used. Such analysis needs to be taken into consideration in order to use the techniques correctly. In the following section, it is expounded the background knowledge on control charts, an integral part of the ARCA solutions proposed in the main chapters, as well as some discussion on the choices made pertaining these techniques. Finally, Section 2.5 presents some notions of causal inference that are required to fully understand Chapter 7. In each of the sections in this chapter, it is mentioned if the knowledge presented is used in this doctoral work, or if it appears mentioned in other related works, which are referenced in the literature review.

In Chapter 3, the problem of this doctoral work is defined. The problem is formally defined in the second section of this chapter, with a detailed explanation of the problem of locating root causes in a manufacturing process, and the definition of overlap. Section 3.3 details the case study in semiconductor manufacturing that served as the motivation and the base for this doctoral work. Section 3.4 presents the stochastic simulator used in this doctoral work to generate more data for analysis. This was necessary, as it was not possible to obtain the root causes of the problems that occurred in the case study. As such, it was required to generate data where it is possible to define the root causes, in order to perform a more robust validation of the approaches/solutions developed. The datasets generated using the stochastic simulator described in this section are then used in the following chapters.

Chapter 4 is a literature review on ARCA in manufacturing. The objective of this chapter is to analyse the existing literature on this topic, and conceptualize about the ARCA definition and types. The commonalities observed in the works in this area are also highlighted. The research done in this chapter was motivated by the need to develop a structure of the existing literature that previously did not exist. Each work discussed in this chapter was published in different areas and niches, each with its own vocabulary and different objectives. To achieve the proposed structure, the literature was divided into three types of data, the methodologies used in each ARCA solution were classified in terms of techniques, three perspectives of use (or how to determine the root cause using a specific technique), and four types of evaluation measures that could be adopted to validate the ARCA solutions. This structuring and conceptualization of the literature allowed for a proper framing of the problem that this doctoral thesis tackled, helping in defining the scope and its place in the literature. A paper based on this chapter has been submitted and is currently under review.

The first chapter in which it is discussed how to tackle overlap is Chapter 5. In it, overlap is defined, and the first way to measure it is proposed. It is discussed why overlap is a critical issue, and why most approaches based on a predictive analytic perspective are ill-prepared to face overlap. The proposed measure is based on the strength of association between factors, which comes from statistics and data mining literature. Then the overall approach to develop ARCA solutions that are resilient to overlap is presented. Both stages of this approach, namely the problematic moment identification and the factor ranking algorithms used are discussed. The datasets used, not only in this Chapter but in the following ones, are detailed. The results of the experiments validating the existence of overlap and the approaches based on this first way to measure overlap are also presented. The discussion of these results is also complemented with a discussion on the different types of labels that are used in the development of ARCA solutions, and some decisions pertaining the way to measure overlap. This chapter also originated a paper that has been submitted and is currently under review.

Chapter 6 identifies some enhancements done to the measure of overlap proposed before. To implement such improvements, it is proposed a measure based on information theory concepts, namely positive mutual information. A novel approach based on this new way of measuring overlap is also proposed. In addition, a visualization of the issue of overlap is presented, which can make the task of understanding and detecting overlap easier. This information theory based approach is then validated, and the results reveal improvements in relation to the previous proposed measure, described in Chapter 5 . The proposed approach is capable of identifying overlap and locating the root causes with good performance, while being robust to overlap. The possible root cause locations are presented as a list of most likely root causes, where the overlapped locations are presented as having the same importance.

In Chapter 7, the final main chapter of this doctoral work is presented. The previous two chapters provide a list of most likely root causes, where the overlapped root causes have the same importance. In this work, we attempted to go further, and see if it is possible to identify the true root cause from a group of overlapped locations. For the cases in which this identification is not possible, the overlap threshold that limited this possibility is identified. For this purpose, it was necessary to use causal inference, in order to develop a graphical definition of overlap, which in turn allows for the computation of the probability that an overlapped location is the actual cause of a problem. Such graphical definition and probabilities allowed for the definition of a theoretical threshold that, when surpassed, makes it impossible to distinguish the influence of overlapped locations. The novel graphical definition and the causal inference probabilities also led to the development of a new ARCA approach. This approach and the proposed threshold are validated in the experiments that were used in the previous two chapters. It was possible to validate the existence of the threshold, and conclude that the new approach based on causal inference can achieve a similar performance to the one based on information theory, proposed on the previous chapter. In the discussion of this chapter, some considerations about the effect of noise on the diagnosis process are discussed.

In the last chapter, this doctoral work is summarised, with its main conclusions being discussed, as well as some managerial insights. The limitations of this work and future research avenues are also presented in this last chapter.

To recapitulate, this thesis presents and proposes the first definition of overlap, which is an issue that occurs when performing diagnosis with Location-Time data from a manufacturing process. In order to develop solutions that are robust to overlap, three measures are proposed, in increasing degree of refinement. The first one is based the chi-square test and Cramér's V, and overlap is measured as the strength of association between factors. The second is based on information theory, and overlap is measured as the positive mutual information between two factors. This new measure improved the sensitiveness of the overlap measurements. The third overlap measure is based on causal inference, and proposes a graphical definition of overlap, which allows for the computation of interventional probabilities that can help determine the true root cause. This last measure allowed to define an overlap limit, after which it is not possible to detect the true root cause of a problem, and it is only possible to reduce the number of possible root causes. These three measures originated different solutions that are robust to overlap, and allow for the discovery of root causes, or at least identify a reduced group of possible root causes. The solutions developed based on positive mutual information and causal inference have the best root cause detection performance.

Chapter 2

Background Knowledge

2.1 Introduction

In this chapter, the aim is to introduce the background knowledge required for the full comprehension of this doctoral work.

In the first section, it is going to be expounded the basic notions of Root Cause Analysis (RCA), and the traditional techniques that are currently used in this process. In Section 2.3, some concepts and algorithms of Data Mining (DM) and Machine Learning (ML), which are mentioned in the next chapter are detailed. The next section focuses on control charts, a quality management technique that can be used to identify the data to analyse, that is an integral part of the approaches proposed in Chapters 5, 6, and 7. Finally, Section 2.5 introduces knowledge on causal inference, which is essential for the understanding of Chapter 7.

2.2 Root Cause Analysis

The general aim of this doctoral work is to develop solutions that help automatize RCA. We can define RCA as a process of analysis to understand the causal mechanism behind a change from a desirable state to an undesirable one, in order to keep a problem from recurring (Ong, Choo, and Muda, 2015). To perform RCA in a manufacturing process means that there is an identified problem (e.g., systematic loss of product quality, machine breakdown) with an unidentified cause, and there is a need to find the cause, in order to remove/repair it, and prevent that problem from happening again.

RCA is a quality management process, and as such has as general objectives improving customer satisfaction, and improving the manufacturing process to reduce waste (American Society for Quality, 2019a). It can be considered an integral part of continuous improvement, as RCA is the starting point to prevent problems from reoccurring. A successful RCA process makes sure the organization is focusing on the true cause of problems, instead of just eliminating the symptoms.

Given its importance in quality management and continuous improvement, several approaches and techniques have been developed to aid in the RCA process. The process of finding a root cause has a clear objective (finding the cause of a problem), but this objective can be achieved in different ways. Problems can arise in many contexts, and the data available may vary and be very complex to analyse. As the problem is likely ongoing, it is desirable that RCA is performed as quickly as possible. Given the complexity of performing RCA, the techniques developed to aid the RCA process focus on guiding the analysis of the information available in an organized and structured way, in order to achieve an analysis that is at the same time as complete and as fast as possible.

Five of the most common traditional RCA techniques are (SixSigma, 2017; Kambilonje, 2020): Pareto Chart, Five Whys, Ishikawa Diagram, Failure Mode and Effects Analysis (FMEA), and Fault Tree Analysis, which are detailed below.

2.2.1 Traditional Techniques

Pareto Chart

The Pareto chart is a tool that is based on the principle with the same name. The Pareto principle states that, for many outcomes, around 80% of the problems come from around 20% of the causes (Tague, 2004). As such, identifying and focusing on those fewer causes will have the greatest impact in terms of consequences.

An example of a Pareto chart is shown in Figure 2.1. It is composed of an histogram and a line graph, where the causes are ordered from the most frequent to the least. On the left axis there is a count of the number of defective products that passed through a certain step-machine combination, which measures the height of the histogram's bars. The right axis is linked to the line graph, which represents the cumulative percentage of all the defective products.

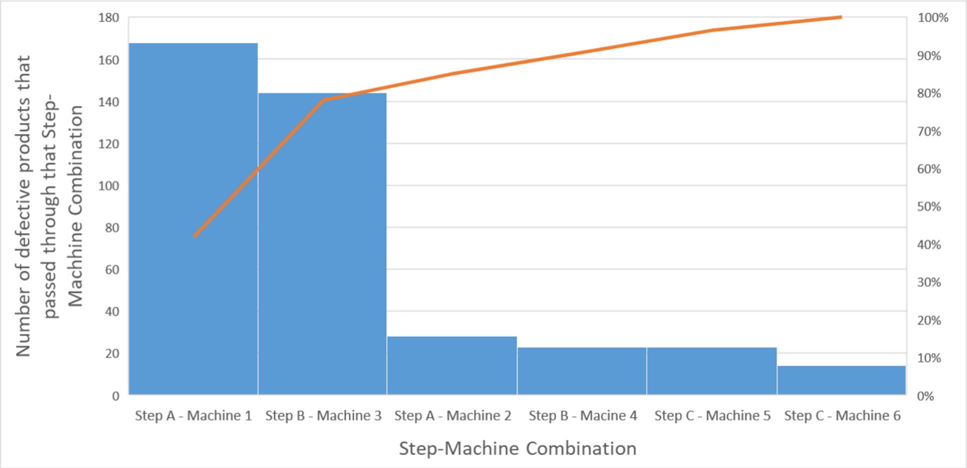


FIGURE 2.1: An example of a Pareto chart.

The example in Figure 2.1 does reflect the Pareto principle, as two of the Step-Machine combinations have around 80% of the total number of defective products pass through them.

In a RCA process, Pareto charts are usually used at the beginning, as they enable the analysts to quickly sort the possible causes in relation to their consequences, allowing priorities to be established. These priorities then guide the remaining of the process, by focusing on the possible causes that most likely have a greater impact.

Five Whys

The five whys is an iterative technique to establish cause-and-effect relations, starting from a problem, and ending in the root cause of that problem. This technique was originally developed by Sakichi Toyoda (American Society for Quality, 2019d), and is part of the Toyota Production System.

The technique is fairly simple, yet it can serve as an important guiding principle when performing RCA. As the name implies, the technique consists in asking five times why, in sequence, in order to understand how the events and/or variables are causally linked, and how these led from the root cause to the problem.

Consider the following example:

1. The cycle time increased. **Why?**
2. High values of overkill lead to many manual inspections. **Why?**
3. All products that go through Machine X have high overkill. **Why?**
4. The flow of cleaning fluid in that machine is too high. **Why?**
5. The flow valves corroded, due to lack of maintenance. **Why?**

   The maintenance plan does not check the flow valves as frequently as it should.

The technique starts from the problem (increased cycle time), and in each Why, we link each stage of the problem, establishing a cause-and-effect relationship between the stages, until we reach the true cause. The five whys is a technique to be used in tandem with other techniques. For example, Pareto charts could have been used to establish that most products that had overkill went through Machine X, therefore linking Whys numbers two and three.

Therefore, the five whys is an important technique to guide the RCA process on a higher level, and is complemented by other techniques. The reasoning behind this technique is similar to the one used in Section 4.2 (Chapter 4) to establish the different types of data, and the type of root cause that it is possible to obtain.

Ishikawa Diagram

An Ishikawa, or fish-bone, diagram is closely related to the five whys, and depicts the links between an event (e.g., a problem) and its possible causes (American Society for Quality, 2019c), in such a way that the hierarchy between cause and effect is visible. It was created by Kaoru Ishikawa , and the hierarchical structure that is reflected in the diagram makes it look like the bones of a fish.

An example of this is Figure 2.2. The problem is high overkill levels. This problem can originate from different aspects of the manufacturing process: the equipment, the operators, the environment, or even the process itself. For each of these groups of causes, more refined causes can be established. In terms of process, the waiting time between each step can be too long or too short, and/or the recipe parameters can be inadequate. For the equipment, the chemical cleaning can be too strong. In this particular cause, two more refined factors can be the root cause of the problem. The chemical may be too corrosive because there is too much flow of the chemical reaching the product, or it could be that the chemical itself is being mixed incorrectly.

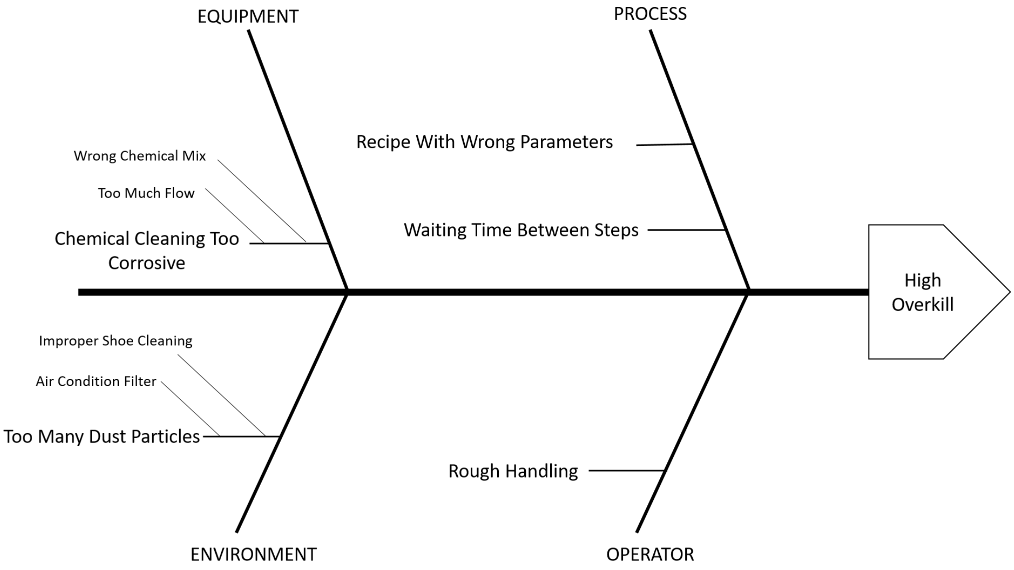


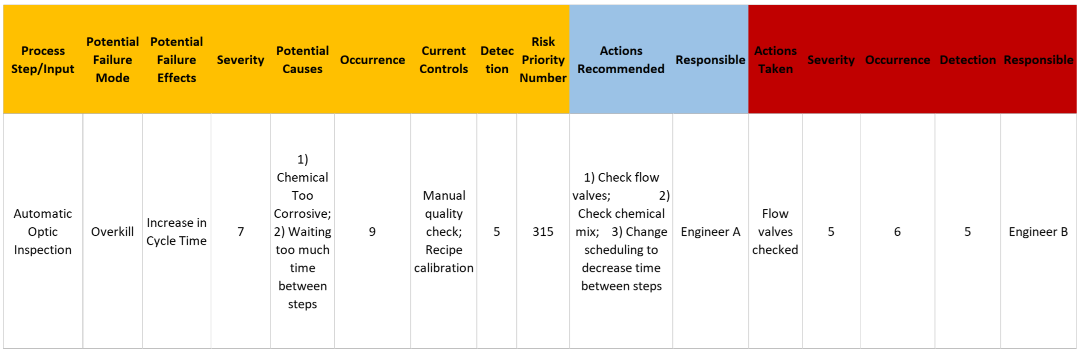
FIGURE 2.2: An example of an Ishikawa diagram.

The Ishikawa diagram allows the analysts to organize the information in a way that the hierarchy of cause-and-effect is clearly depicted, structuring the RCA process by showing the hypothesis that can be investigated and focusing the endeavors.

Failure Mode and Effects Analysis

Failure Mode and Effects Analysis (FMEA) is is a step-by-step approach for identifying all possible failures in a design, a manufacturing or assembly process, or a product or service (American Society for Quality, 2019b). Historically it originates from the U.S. military in the late 1940's. The part "Failure Mode" represents the several ways in which something might fail, while "Effects Analysis" refers to the consequences of those failure. It consists of an exhaustive list of all components and sub-components, and how these can fail, registering how likely is the failure to occur, and how serious are the consequences of the failure.

An example of the result of a FMEA process is Figure 2.3. An FMEA chart can be divided in three main regions. In the first nine columns in the chart (in yellow in Figure 2.3), there is the initial development of FMEA. These identify (from left to right): where the problem may occur; what is the failure; what are the effects of the failure; how severe is the failure; what are the potential causes; how likely it is to occur; what mechanisms currently exist to detect the existence of the failure; how likely it is for the problem to be detected before it affects the process or the client; and what is the priority when dealing with that failure. In blue are the two columns that pertain the improvement activities that are recommended for that failure mode, should it occur, and who is responsible to enact them. Finally, the five red columns represent the post-improvement activities, and the values of the three rating ("Severity", "Occurrence", "Detection") after the improvement occurred. The three ratings vary from 0 to 10, where 10 means more criticality. "Risk Priority Number" is the result of the multiplication of the three ratings (Juran, 2018). The higher this value is, the higher the priority this failure mode has in relation to other failure modes.



Process Step/Input	Potential Failure Mode	Potential Failure Effects	Severity	Potential Causes	Occurrence	Current Controls	Detection	Risk Priority Number	Actions Recommended	Responsible	Actions Taken	Severity	Occurrence	Detection	Responsible
Automatic Optic Inspection	Overkill	Increase in Cycle Time	7	1) Chemical Too Corrosive; 2) Waiting too much time between steps	9	Manual quality check; Recipe calibration	5	315	1) Check flow valves; 2) Check chemical mix; 3) Change scheduling to decrease time between steps	Engineer A	Flow valves checked	5	6	5	Engineer B

FIGURE 2.3: An example of a Failure Mode and Effects Analysis.

FMEA is more useful to increase the efficiency of future RCA processes, than to find the root causes of a novel problem. As it exhaustively lists how a process/equipment can fail and its consequences, it can be used to quickly diagnose a problem than reoccurs, or a problem that occurs for the first time, but that could be predicted from the construction/assembly of the product or how the process is designed. Due to its characteristics, FMEA is used more for equipment maintenance than to perform RCA of problems that affect the quality of products. Other RCA techniques may be used to help develop a FMEA chart.

Fault Tree Analysis

Fault Tree Analysis (FTA) is a graphical tool where an undesired state of a system is analyzed using Boolean logic to combine a series of events. Instead of focusing on all possible system failures, like the FMEA, it focuses on a single problem at a time, and tries to relate all sub-problems that could lead to main problem. It is a tool that enables to establish the connection between different problems, which can share common root causes. It was originally developed in the Bell laboratories in the early 1960's (Hessing, 2020).

In FTA, rectangles represent state of systems or component events, while circles represent primary or basic failure events. The logic gates AND (a rectangle with a semi-circle on top), and OR (a rectangle with a semi-circle removed from below) represent the logic operations between events.

Figure 2.4 is an example of a Fault Tree Analysis. In this example, high overkill can be caused either by the chemical cleaning being too corrosive, or by there being too many dust particles in the manufacturing environment. The chemical being too corrosive can be caused either by the flow being too much, or because the wrong chemical mix was used. In relation to the dust particles, for it to be problematic, two events must occur: improper shoe cleaning, and inappropriate functioning of the air conditioning. Only one of the events would not be enough to cause an exaggerated amount of dust particles in the air, but both at the same time do cause the problem.

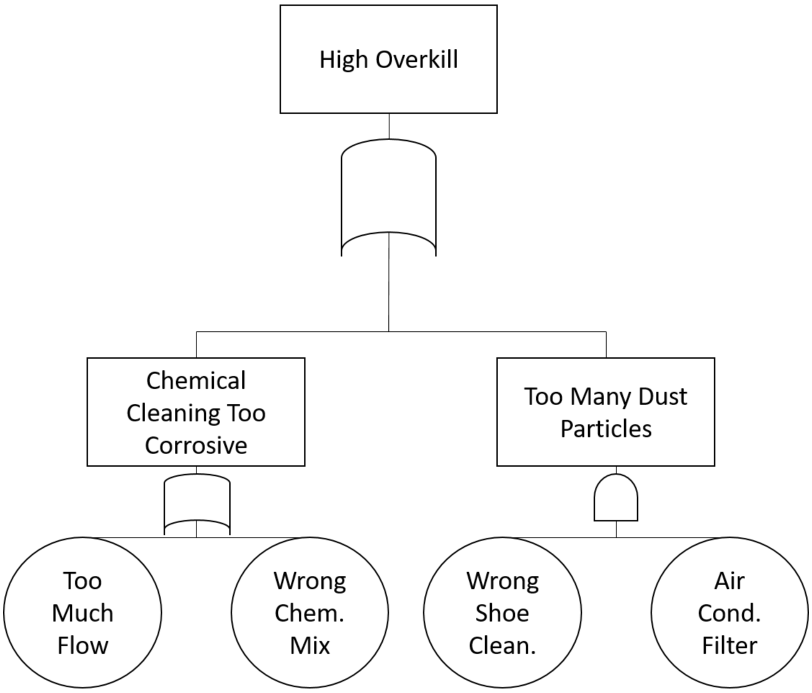


FIGURE 2.4: An example of a Fault Tree Analysis.

Probabilities can be assigned to basic events (the circles), and these can be used to do a quantitative FTA, that can allow for the estimation of risks of failures.

As the previous techniques, this one focuses on structuring the information and events, and is based on human input through the analysts, who codes the chain of events using the elements in this technique. This has the advantage of being able to provide a quantitative analysis.

2.2.2 Discussion

RCA is a critical process for continuous improvement and the maintenance of quality in a manufacturing process. The techniques introduced in this section are the most used, and can even be called traditional.

Their objective is to help sort, organize and structure information, in order for the search for a root cause to be guided by objective principles. This structuring of the information enables a more efficient and focused investigation, with higher chance of success in discovering the true root cause of the problem that occurred.

These techniques, however, require much human input, and most of them are of qualitative nature. With the increase in the complexity of manufacturing processes, and the growth in the amount of data generated by these processes, the traditional techniques have more difficulty in handling the high quantity of data. As such, new solutions are being developed (such as this doctoral work), which use data mining and machine learning techniques to reduce the dependence on human input, and improve the efficiency of the process. Such techniques are introduced in the next section.

2.3 Data Mining & Machine Learning

Data Mining (DM) and Machine Learning (ML) techniques were developed in order to make it easier to extract knowledge from large amounts of data. DM is a subarea of business analytics, and pertains the activity of exploring existing data to discover new patterns and associations that can provide useful insights about a certain topic. ML is a subarea of artificial intelligence, which focuses on how can a computer automatically learn relevant information from data available, and generalize it for future data. They are similar in their purpose of extracting knowledge from data, but differ in the amount of human intervention that is desired. While in DM the focus is in the knowledge obtained, independently of the amount of human intervention, in ML the objective is the creation of algorithms which are as autonomous as possible.

In the context of this doctoral work, DM and ML techniques aid in the development of novel solutions for RCA, as traditional techniques cannot handle large amounts of data efficiently, requiring human input, while DM and ML focus on scouring large amounts of data for the required insights, with as little human input as possible.

In order to properly frame the use of DM and ML techniques in RCA, we have to consider how these techniques can be used and lead to different types of analysis.

There are four types of data analytic approaches (Maydon, 2017; Michigan State University, 2019):

1. **Descriptive Analytics** focuses on what finding *what* the actual scenario is, and describing the data and the process that originated it.

2. **Diagnostic Analytics** attempts discovering *why* something happened, or how did the process reached a certain state.
3. **Predictive Analytics** tries to predict what will happen or *when* will a certain event (e.g. a machine breakdown) take place.
4. **Prescriptive Analytics** aims at proposing solutions or strategies on *how* to solve current or future problems.

Considering the above-mentioned approaches, RCA is clearly within the **Diagnostic Analytics** type, as its focus is to determine the root cause of a problem, and how the problem came to be. However, most DM and ML techniques were developed with a descriptive or predictive aim. Data exploration and unsupervised techniques usually focus on describing the dataset. Supervised techniques were originally aimed at prediction problems. As such, it is necessary some caution when adapting these techniques to a diagnosis process.

After discussing the types of analytics possible, and how RCA is framed within these types, we will now discuss in more detail the types of DM and ML techniques available. These techniques can be divided in three major groups: Data preparation & exploration, supervised learning, and unsupervised learning.

2.3.1 Data Preparation & Exploration

The input for all projects involving DM and ML is data, in the form of one or more datasets. In order to understand the knowledge that is possible to extract from these datasets, they have to be explored and prepared to be used by other DM and ML techniques. This involves looking into the data, understanding the state it is in, which variables can and should be used, if they are nominal or numerical, and possible relations among these variables. This falls within the scope of descriptive analytics, as the objective is to describe the data in order to understand as clearly as possible what exists, and how that can be expanded.

Two groups of techniques can be considered. First, feature selection focuses on determining which variables should be used based on the relations among variables. Second, feature engineering goes one step further by creating new variables that can bring further insights to the project.

Feature Selection

Feature selection techniques are simpler than feature engineering, in the sense that the focus is simply selecting a subset of variables that improves the performance of DM and ML techniques. These can be divided into filter methods, wrapper methods, and embedded methods (Guyon and Elisseeff, 2003).

Filter Methods

Filter methods are independent of the DM and ML techniques we wish to apply. The intuition behind these methods is to use a statistic measure to score each variable and select the ones that have a stronger relation to the target variables, i.e., the variables that are the main focus of the project. The most common statistic measures in this context are Pearson correlation, Chi-Square correlation, and mutual information.

Pearson correlation is used when we have numerical data, and measures the linear dependence between two variables. It is given by Expression (2.1). The numerator is the covariance matrix between the two variables, while the denominator is the product between the standard deviation of each variable.

$$\rho_{X,Y} = \frac{\text{cov}(X,Y)}{\sigma_X \sigma_Y} \tag{2.1}$$

Its values vary between -1 and 1, where 0 implies no correlation, 1 a perfect positive correlation (when X increases, Y also increases), and -1 a perfect negative correlation (when X increases, Y decreases). It is used to measure the strength of the correlation between each variable and the target variable, and variables with a weak correlation are discarded.

Related to Pearson correlation is Partial correlation, which is adopted in several studies. These studies are further explored in Chapter 4. Partial correlation measures the degree of association between two variables, while removing the effect of a set of controlling variables. It is useful when there is the suspicion that one set of variables may be having a strong influence on the correlation of other variables. Formally written as $\rho_{XY \cdot Z}$ (where Z is a set of controlling variables), partial correlation is the correlation between the residuals e_X and e_Y resulting from the linear regression of X with Z and of Y with Z.

When dealing with nominal data, the most common statistic is the Chi-Square statistic. It is based on the Chi-squared independence test, which tests the existence of association between two nominal variables. The Chi-Square statistic is given by Expression (2.2).

$$\chi^2 = \sum_{i=1}^{R} \sum_{j=1}^{C} \frac{(o_{i,j} - e_{i,j})^2}{e_{i,j}}, \tag{2.2}$$

where i and j are indexes of the levels of each of the two nominal variables being analysed, $o_{i,j}$ is the observed frequency at the level i of the first variable, and the level j of the second variable, and $e_{i,j}$ the is the expected frequency, for those levels. The expected frequency is computed by Expression (2.3):

$$e_{i,j} = \frac{\text{level } i \text{ total} * \text{level } j \text{ total}}{\text{all levels total}} \tag{2.3}$$

These expressions mean that we are comparing the observed values with the corresponding expected values if all levels in the two variables were completely independent/random. Expression (2.2) gives the sum of the squared differences divided by the expected values. In the Chi-squared independence test, the χ^2 is compared with the critical value in the chi-square distribution with $(R-1)(C-1)$ degrees of freedom (R the total number of rows, and C the total number of columns).

As with the Pearson correlation, the aim is to compare each variable with the target variable, and eliminate those which are independent, as they likely have little influence in the analysis.

Sometimes after performing a statistical test like the chi-square, if the test is rejected, it is relevant not only to understand the importance of the variable, but also how relevant is each individual level within that variable. To understand the importance of each level, a post-hoc test can be performed, where the contingency table representing the two variables being compared is analysed to determine which levels has a greater impact in rejecting the test, i.e., in making the two variables dependent.

The test used in this thesis (in Section 5.4.2) is the Fisher exact test. It is a test that is usually applied when the samples are small, and its denominated exact as the p-value can be calculated exactly, rather than using approximations that become close to exact when the sample size is large enough. However, the Fisher test is more accurate than chi-square test on variables with few levels (two or three), while the chi-square is better on variables with more levels (Berkson, 1978). Still, it is recommended as a post-hoc test (Shan and Gerstenberger, 2017).

The rationale behind the Fisher test is similar to the chi-square test, as it compares the observed values with the expected values that would occur if the variables were independent. The difference is that the distribution that is used to decide if the variables are independent or not is the hypergeometric distribution, instead of the chi-square distribution.

For comparison between each level of the two variables, a test is performed for each possible pair of levels from the two variables (e.g., i as a level of the first variable and j as a level of the second variable). This way, in the test for a single pair, each variable now only has two possible values: either it assumed the value of the level in the pair being tested, or it did not. This way, the contingency table counting the occurrences of each level becomes a two by two table, which is appropriate for a Fisher test. The values of these tests can then be compared to see which levels have more influence in making the two variables dependent on each other.

As we are performing multiple statistical tests at the same time, the probability of observing a rare event increases, and with it the probability of incorrectly rejecting a hypothesis (Bland and Altman, 1995). To counteract this, it is possible to use a correction on the decision threshold on what is significant as the number of tests increase. The correction used in this test is the Bonferroni correction (Dunn, 1961).

The rationale is to make a correction to the significance level by dividing this level by the number of tests to be performed.

While the Pearson statistic works only with numerical data and the chi-square statistics only with nominal data, mutual information (Shannon, 1948) can be used with both numerical and nominal features. Mutual information measures the amount of information it is possible to extract from one variable using another variable. Expression (2.4) represents how to compute such measure.

$$I(X;Y) = \sum_{x}\sum_{y} p(x,y) log \frac{p(x,y)}{p(x)p(y)}, \tag{2.4}$$

where the $p(x)$ is the probability of level x of variable X occurring, $p(y)$ is the probability of level y of variable Y occurring, and $p(x,y)$ the probability of both levels occurring at the same time. Mutual information varies between 0 and 1, where 1 means the highest association between variables. As with the previous two statistic measures, the objective is to compare each variable with the target variable and eliminate those which are independent.

Wrapper Methods

The aim of wrapper methods is to obtain the set of variables that optimizes the performance of a DM/ML technique that is going to be used. To do so, the technique is trained with several sets of variables, and the one which achieves the best performance is selected. Given these characteristics, wrapper methods are dependent on the DM/ML technique to be used.

In order to increase the efficiency of the wrapper methods, two heuristics can be used:

Forward Selection is an iterative method that trains the model starting with just one variable, and increases the amount of variables considered by one in each iteration. In each iteration, all variables that have not been yet added definitely to the final subset of variables are tested using new models that contain the previously selected variables and the new variable being tested. The best variable in each iteration will stay for the next iterations. It iterates until adding a new variable leads to a decrease in performance.

Backward Elimination Is also an iterative method, but works in inverse to forward selection. It starts with all the variables, and in each iteration it tries to eliminate the variable whose elimination leads to the greater improvement in performance. It iterates until the removal of any remaining variables leads to a decrease in performance.

Embedded Methods

Embedded methods are methods where the algorithms themselves have feature selection capabilities. This way, qualities of both filter and wrapper methods can be used. Examples of methods that have this embedded feature selection are regression algorithms that perform regularization, such as the LASSO (Tibshirani, 1996) and RIDGE models (Hoerl and Kennard, 1970). Regularization means adding a penalty to using a large number of variables in the model. This is done to avoid overfitting (lack of generalisation), by making sure only the most relevant variables are selected.

Feature Engineering

With feature engineering, the objective is to generate new more meaningful features that can provide better insights when analysed through additional DM and ML techniques.

The simplest example is to adapt a nominal dataset to be used with a technique that only works with numerical data. There are two common approaches to do this. Integral encoding, where each level in a nominal variable is associated with a number. For example, in a variable with the levels "Machine ABC", "Machine XPT", and "Machine RTX", each time an instance has the level"Machine ABC" in that variable, it is replaced by 1, if it has "Machine XPT" it would be replaced by 2, and the remaining level would be replaced by 3. However, this may introduce an artificial order among the levels, which is sometimes undesirable. Considering the previous example, some methods could try to compute an average of the now numerical variable, where a value like 1.5 is not easily interpretable, as it could mean that there is an equal amount of instances of "Machine ABC" and "Machine XPT", or, alternatively, it could mean that the majority of instances are "Machine ABC", with a minority of "Machine RTZ" instances. Also, this makes the techniques unstable, as a different initial order of the levels in that factor can lead to different results, simply because of a different coding. One-hot encoding, on the other hand, creates a new variable for each level in a nominal variable. Then, each instance has the value of 1 in the variable corresponding to the level it had, and 0 in the other levels' variables (an example of this is Table 6.1). This way, no order is artificially introduced in the data, although it makes the data sparser.

Besides the conversion from nominal to numerical data, there is also the possibility to create new variables from the original ones, which can contribute with more insights, as well as reduce the number of variables to consider.

The most commonly used technique to do this is Principal Component Analysis (PCA). This technique works with numerical data, and tries to find the combinations of variables that better explain the variance in the data. PCA can be performed with the following steps:

1. **Standardize Data** The numerical values should be standardized, as PCA is very sensitive to differences in ranges.
2. **Compute Covariance Matrix** Creates a symmetric nxn matrix (n being the number of variables) that stores the covariance between all variables.
3. **Identify Principal Components** This is done through the computation of the eigenvalues and eigenvectors of the covariance matrix.

By performing these steps, the principal components are obtained, which are a linear combination of the old variables. They are then sorted by amount of variance explained in a Pareto chart, like in Figure 2.5. In this particular case, the first two components can explain more than 80% of the total variance. This means that we can use these two components as variables, without losing much information. The components may even reflect latent variables, that can provide further insights when interpreted.

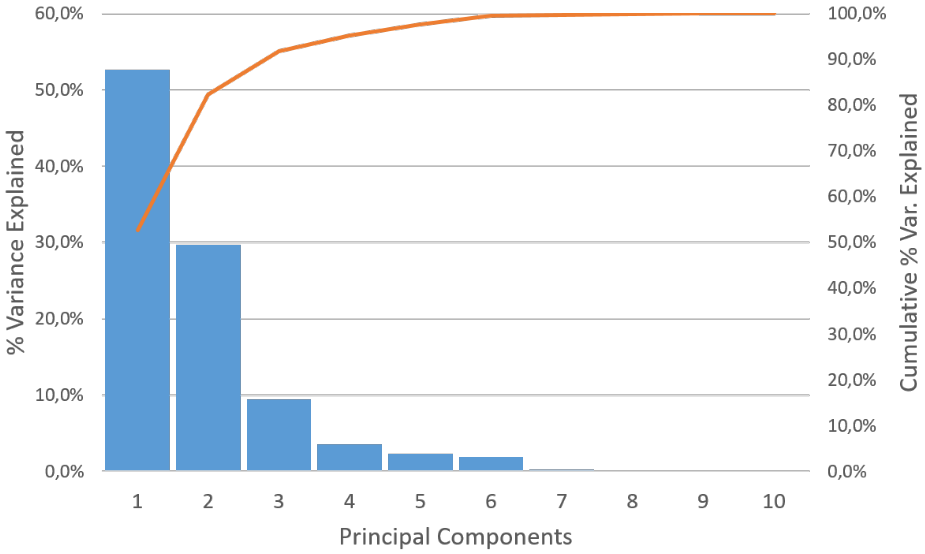


FIGURE 2.5: An example of the principle components resulting from PCA, and the amount of variance they explain.

PCA is widely used in the literature together with other techniques. However, it focuses solely on linear relations among variables. There are a few alternatives for nonlinear feature engineering and reducing the number of features.

Kernel PCA follows the same principles than PCA, but adds an additional step between steps 1 and 2, in which data is transformed into a higher-dimensional space using a kernel. The final results are then projected back into the original space.

Another alternative for feature engineering is Isomap (Tenenbaum, Silva, and Langford, 2000), which connects all points of data into a neighborhood graph using geodesic distances (the distance between two points through the path available). It is then able to compute the shortest distance between two nodes, and uses this to compute a lower-dimensional embedding into which to project the data.

Another possible feature engineering method is t-distributed Stochastic Neighbor Embedding (t-SNE) (Maaten and Hinton, 2008). This technique allows for the visualization of very high dimensional data in a two or three-dimensional space. It is composed of two stages: 1) it constructs a probability distribution in the original high-dimensional space, such that similar instances have a higher probability of ocurring while dissimilar instances have a lower probability of occurring together; 2) it defines a similar probability distribution in the new low-dimensional space, and then minimizes the Kullback–Leibler divergence (KL divergence) between the two distributions in terms of the locations of the points in each dimension. This can be useful in exploring and describing a dataset, and the new low-dimensional space can be used with other techniques.

2.3.2 Supervised Learning

Supervised Learning refers to ML techniques in which the aim is to discover associations among the inputs and outputs in the data. Both inputs and outputs can be together in a single dataset, but when using supervised techniques, it is necessary to clearly identify which of the variables in the dataset are the inputs, named factors (or independent variables), and which are the outputs, named labels (or dependent variables). Table 2.1 is an example of a dataset for supervised learning. Each column represents a factor, except for the last one which is the label. Each row represents an instance that is used to either train or test the model.

TABLE 2.1: Example of dataset for supervised learning.

Factor 1	**Factor 2**	**Factor 3**	**...**	**Label**
Value 1	Value 2	Value 3	...	Label 1
Value 4	Value 5	Value 6	...	Label 2
Value 7	Value 8	Value 9	...	Label 3
...	...	...	...	...

When applying supervised learning techniques, the usual procedure is to train the technique's model using part of the data, fitting the model to that data, and then testing in another part of the data. Labels are required when training, and are not used when testing. This way it is possible to see if the model can predict the label based only on the factors. Given the way supervised learning techniques work, it is usually framed within the predictive analytics, as the aim is to predict the label value of unlabeled data. In this doctoral work, supervised learning techniques are used for diagnosis, as they can be used to associate factors to problems. However, some caution should be used when applying supervised learning techniques to diagnosis, as their main focus is prediction. As the main objective of diagnosis is to discover why something happened, when using classification algorithms, it is important to be able to understand which factors are used by the model to associate each instance to a label, as these factors are most likely the reasons for the label to take a certain value. As such, supervised techniques used in diagnosis have to be easily interpretable.

The supervised learning techniques can be divided according to the labels' variable type: if it is a nominal variable, a classification technique is used; if it is a numerical variable, a regression technique is used.

Classification

In classification, the label to be predicted is a nominal variable with many categories/levels. The aim is to have a model that can predict each instance of a similar problem, using a dataset with the same structure. Several classification techniques exist, each of them with a different model as a base.

In this doctoral work, the most used classification algorithm is the Decision Tree (DT). This technique produces a model that is easily interpretable, which is required when using classification techniques for diagnosis. DT induction algorithms aim at creating a structure similar to the one in Figure 2.6. Each factor is represented by a circle, and the value of each label by a rectangle. The value each factor can take is written above the arrows. For a label with the levels "High" and "Low", the model associated the factors "Step 04" and "Step 01". If the instance has the value "Equipment 14" in factor "Step 04", the instance is predicted to have the value "High". If it has any of the other values in "Step 04", it then checks the value of "Step 01" factor, and if the value is "Equipment 03", the instance is predicted as "High". If not, the instance is predicted as "Low".

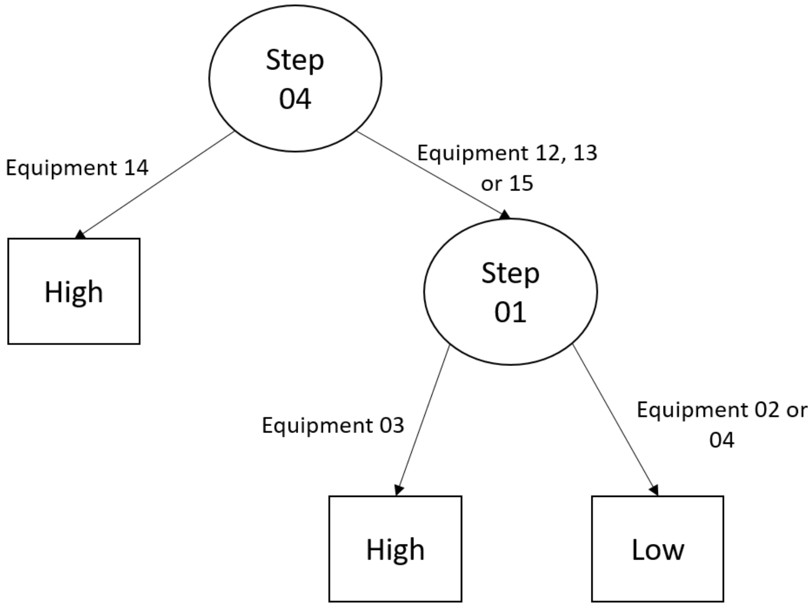


FIGURE 2.6: An example of a decision tree after training.

From a predictive analytics perspective, the most relevant aspect is that the model can accurately classify most of the instances with the correct label. From a diagnosis perspective, the relevant part is that, by looking at the model, it is possible to conclude that “Step 04 - Equipment 14” and “Step 01 - Equipment 03” are factors that most likely led to the label being “High”.

The structure shown above can be achieved by many types of DT algorithms. The DT can be shorter, with less nodes connected vertically, or deeper. Deeper DT have the tendency to overfit, i.e., to have less generalisation capacity. One of the most used DT algorithms in the literature, and that is used in this doctoral work is the C4.5 algorithm proposed by Quinlan (Quinlan, 1993). This algorithm is based on the concept of information entropy. Entropy of a variable is the uncertainty in relation to the values of that variable. It is given by Expression (2.5) (Shannon, 1948).

$$H(L) = -\sum_{i=1}^{n} P(l_i) \log P(l_i), \tag{2.5}$$

where l_i represents each category the variable L can have, and $P(l_i)$ represents the probability of that level occurring. This value can vary between 0 and 1, where 1 represents the maximum entropy, ergo maximum uncertainty.

The basic idea behind the C4.5 algorithm is to, in each iteration, select the factor that reduces uncertainty the most. This increase in certainty is usually called information gain, and can be computed using Expression (2.6).

$$IG(L,F) = H(L) - H(L|F), \tag{2.6}$$

where $H(L|F)$ is the conditional entropy of L given the knowledge of F (given by Expression (2.7)). In other words, if L represents the label and F represents a set of factors, $IG(L,F)$ is how much more information we have about the value of the label by knowing the value of the set of factors.

$$H(L|F) = \sum_{l \in L, f \in F} P(l,f) \log \frac{P(f)}{P(l,f)}, \tag{2.7}$$

The C4.5 algorithm has the following steps:

1. For each factor f, find the information IG in relation to the label.
2. For the factor with the best IG, create a decision node that splits on that factor.
3. Continue splitting the data iterating through all the other factors until no more factors left, or the information gain becomes negligible. Always keep the best factors from previous iteration.

Other types of classifiers, which are not used in this doctoral work, but appear mentioned in Chapter 4 are:

Artificial Neural Networks are a model composed of connected nodes called artificial neurons, which exchange information between themselves, and are used to create complex functions that link the inputs to the output. They use numerical data, and consequently nominal data need to be converted to numerical using, for example, one-hot encoding. These models can achieve impressive predictive accuracy. However, they are black-box models, and are not easily interpreted.

Support Vector Machines (SVM) aim at finding the frontier that separates two categories of the label the most. In order to work with high dimensional spaces, it uses kernels to project the high-dimensional data into a bi-dimensional one, and then defines the frontier that maximizes the distance to the support vectors. These support vectors are the points of each category that are closer to the frontier. After the frontier is determined, it is once again projected in the high-dimensional space, using the kernel. This technique also uses numerical data and is considered a black-box model.

Bayesian Networks are probabilistic graphical models that represent a group of variables and their conditional dependencies. The models are composed by a Directed Acyclic Graph (DAG) and conditional probabilities tables of the variables that are conditionally dependent on each other. These together define the joint probability distribution. A Bayesian network can be constructed using expert knowledge or inferred through structure learning. It works with nominal data, and its models can answer a variety of probabilistic queries, such as "What is the probability of this outcome given that this factor has a specific value?", or "Given this outcome, what is the probability that this factor has a specific value?". This greatly improves the interpretability of the model.

Ensembles

Ensemble supervised learning uses multiple supervised learning models to achieve a better predictive performance. By combining several smaller models, an ensemble technique can achieve better robustness to noise, and capacity for generalization. There are two common types of ensembles, i.e. boosting and bagging.

Boosting is a group of ML techniques that uses several weak learners to create a composite learner that is able to achieve a better performance than a classifier based on a single model. A weak learner is a classifier which is only slightly better than random guessing. But by combining several weak learners, it is possible to obtain a strong learner, which has a very high accuracy. The weak learners can be any type of technique.

The most famous boosting technique is AdaBoost (Schapire, 1999). The basic intuition is to train several weak classifiers in sequence, where the outputs of these learners are combined into a weighted sum. After each weak learner is trained and tested, the instances that were misclassified by that learner are given more weight in the training of the next classifier, in order for it to focus on the misclassified instances. This way, each classifier focuses on a part of the dataset in which it is stronger, and the contribution of all the weak classifiers adds up to a very accurate prediction.

Boosting also has the advantage of being able to deal with high dimensional datasets, as each classifier can focus on specific and relevant factors, eliminating factors that do not improve predictive power.

Bagging, also called bootstrap aggregating, is designed to improve the predictive accuracy of ML algorithms, while at the same time reduce variance and avoid overfitting (or lack of generalization capacity). While boosting trains its weak classifiers in sequence, bagging trains its classifiers in parallel. It generates several sub-datasets by sampling with replacement (also known as bootstraping), and trains each classifier in one of these samples. The outcome (classification) of each of these weak classifiers is aggregated using a voting system.

One of the most used bagging algorithms, and one that is used in this doctoral work, is Random Forest (RF) (Breiman, 2001). In RF, the weak classifiers used are short DT. Deep DT tend to overfit the data and have a lot of variance. However, using short DT together with bagging allows for a model with high predictive accuracy and low variance. Random forests, in addition to sampling instances, also sample factors. What this means is that, for each split in a DT, a sub-set of factors is randomly selected. This is done in addition to normal bagging, because, in normal bagging, if one or more factors are very strong predictors for the label, these factors will be selected in many of the short DT, which creates correlation between these weak classifiers. By sampling factors instead, RF ensures independence between the classifiers.

RF can also be used to determine factor importance. This is done by computing the out-of-bag error after having permuted factors among the training data. As each DT is trained on a different sample, out-of-bag error is the error predicting the values of the instances that were not selected by the sample used to train that DT. The importance of each factor is computed by permuting the factors among the training data, and computing the out-of-bag error again. The importance of each factor is the average of the differences in the out-of-bag errors, before and after permuting that factor.

Regression

Regression, as mentioned previously, tries to predict a numerical label, using other factors. It tries to establish a function between the inputs and the outputs, which can be graphically translated into a line or hyper-plane (depending on the number of dimensions).

The simplest form of regression is the linear regression, in which the end result is a linear equation with the form of Expression (2.8).

$$y = X\beta + \alpha, \tag{2.8}$$

where y represents the label array, X the factors matrix, β is the coefficient/parameter array, and α the error term. Through matrix operations, it is possible to obtain the α and β values for a given dataset, and obtain a regression that can be applied to other instances with the same factors.

For non-linear regressions, several DM and ML techniques exist. Some of the most used are:

Artificial Neural Networks can create complex functions based on smaller elements represented by their neurons. In the case of regression, the network ends in a single neuron representing the label. As explained before, they work with numerical data, and they produce a black-box model, where its parameters are not easily interpretable. Several types of neural networks exist, with specific objectives, such as dealing with time series data. Some of the most used neural networks are long short-term memory for considering past values of time series, or convolutional neural networks, mostly used for image analysis.

Support Vector Regression uses the same theory that bases SVM, but instead of finding the frontier that best divides two categories in the label, it focuses on finding the best hyper-plane that fits the data. The best hyper-plane is the one that has a maximum number of points included within a certain distance of the hyper-plane. As mentioned before, a kernel is used to transform high-dimensional spaces into lower spaces.

Ensembles both RF and AdaBoost have formulations that focus on regression instead of classification. In this situation, instead of each weak classifier voting, the average of all the proposed values is computed.

Autoregressive Integrated Moving Average (ARIMA) is used with time series data. When predicting values using time series data, it is essential to be able to use instances that report to a time series past values. An ARIMA model is basically a linear regression, but that uses its own lagged values as factors, and it is necessary to decide how many lagged values it is desirable to use. However, regression works better when the factors are not correlated. To do this, ARIMA makes a time series stationary by considering the differences between the values at each time.

2.3.3 Unsupervised Learning

Contrary to supervised learning, in unsupervised learning the variables are not divided between factors and labels. The objective with this type of technique is to describe and understand if there is a relation between variables, and to discover if there is any underlying structure, or if the data can be represented in a more compact form.

Given the above description, unsupervised learning techniques are usually framed within the scope of descriptive analytics, as they are mainly used to understand the current state of the data. These techniques can also be used in tandem with other techniques, for example to create an automatic way of labeling the data, or reduce the dimensionality of the data used for better predictive performance and/or for better insights.

Two types of unsupervised learning appear in this doctoral work, mainly in the literature review of Chapter 4. Those are association rules and clustering.

Association Rules

Association rule mining techniques find rules that describe interesting relations between variables in large datasets. The first application of association rules was in finding consumption patterns in transaction data at supermarkets (Agrawal, Imielinski, and Swami, 1993). For example, the rule $\{\text{diapers}\} \Rightarrow \{\text{beer}\}$ means that people who buy diapers usually also buy beer as well.

To perform association rule mining, the data is usually presented in a format like in Table 2.2. In association rule mining, each instance or row is denominated a transaction, and each variable is an item, for example “diapers”, or “beer”. A group of items is denominated itemset. If an item appears in a transaction, a 1 appears in the cell, otherwise its value is 0.

TABLE 2.2: Example of a dataset for association rule mining.

Transaction	Item 1	Item 2	Item 3	Item 4	Item 5
1	1	1	0	0	1
2	0	0	1	1	0
3	1	0	0	1	0
...	...	...	...	...	...

In order to obtain interesting rules, several concepts have been formulated:

- **Support** indicates the frequency that the itemset appears in the dataset.
- **Confidence** indicates how often the rule has been observed to be true.
- **Lift** is the ratio of the observed support for a given rule over the support expected if the items in the rule were independent.

Of the algorithms that mine association rules from datasets, the two most relevant are the Apriori algorithm (Agrawal and Srikant, 1994), and the FP-Growth algorithm (Han, Pei, and Yin, 2000).

The Apriori algorithm is structured as follows:

1. A threshold of how frequent an item/itemset needs to be before it is considered interesting is established.
2. Count the support for each item individually.
3. For all items above the pre-defined threshold generate itemsets composed of those items, and count their support. Filter the itemsets with frequency below the threshold
4. Repeat the iteration, creating itemsets one item larger than before, until no frequent itemsets can be determined.

Albeit historically important, the Apriori algorithm needs to scan the database many times, which makes it inefficient. Also, it needs to generate and count many candidate itemsets, which can be inefficient if the database is very large.

The FP-growth algorithm attempts to fix these issues. FP stands for frequent pattern, and these can be created without generating candidate itemsets. Instead, the FP-growth algorithm represents the datasets as a pattern tree. An example of a FP tree is represented in Figure 2.7. Each node in the tree represents an item. The most frequent items are represented on top, and the nodes below represent the occurrence of that item together with all the items connected to it on top. In this example, there are five transactions with {Item 1}, of which three with the itemset {Item1, Item2}, and one of those with the itemset {Item 1, Item 2, Item 5}.

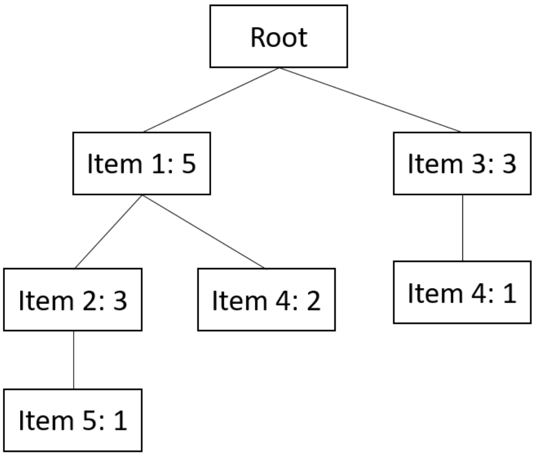


FIGURE 2.7: An example of a FP tree.

The FP-growth algorithm follows these steps:

1. Scan the database to find the occurrences of the items in the database (same as Apriori).
2. Start constructing the FP tree. Examine the first transaction and update the tree with all the itemsets in that transaction.
3. Do the previous step for all transaction, adding new itemsets as they appear, and increasing the counts on the already existing ones.

Both the FP-growth and the Apriori algorithms have as output a set of association rules. However, the FP-growth algorithm has the advantage of only requiring two scans of the database, while the Apriori algorithm requires a scan in each iteration, making FP-growth more efficient.

Clustering

Clustering techniques focus on grouping together objects (instances or variables) that are more similar between themselves than to the elements of the other groups. As other unsupervised learning techniques, its focus is on describing and exploring the dataset.

There exist several techniques to cluster data, for example based on the distance between objects, density in areas of the data space, or the existence of a particular statistical distribution.

Connectivity-based

The most used clustering technique based on the connection between points is the hierarchical clustering, which creates a hierarchy of clusters based on distance, which visually translates into a dendrogram. Hierarchical clustering can be done in two different ways:

- **Agglomerative**, where each object starts as its own cluster, and then the clusters grow by adding the objects of the cluster that are closer to them.
- **Divisive**, where all objects start in one cluster, and then this cluster is split in order to decrease the distance between objects within each cluster

The most common metrics to determine the distance between objects are the Euclidean distance, the Manhattan distance, and the Mahalanobis distance. To determine the distance between two clusters, several linkage criteria can be used, such as the complete-linkage, where the distance between clusters is measured by the maximum distance between two points of each cluster, and the single-linkage, where the distance between clusters is measured by the minimum distance between two points of each cluster.

Centroid-based

When clustering is based on centroids, each cluster is represented by a point in the data space. That point is not necessarily an object from the dataset to be clustered. Such point is called a centroid. The most used centroid-based clustering technique is the k-means, where the number of clusters is pre-defined, and the objective is to find the k centroids that minimize the squared distance of each object in the cluster to the centroid that represents that cluster.

This is an optimization problem that is NP-hard, and as such the focus is on finding approximate solutions. This is usually done with Lloyd's algorithm (Lloyd, 1982), an iterative algorithm which has the following steps:

1. **Assignment step**, where each object is assigned to the cluster with the centroid closer to it.
2. **Update step**, where each centroid's position is computed once again based on the objects in its cluster.

By iteratively following these two steps, the changes in the centroids' positions become progressively smaller, as each centroid converges to a position that minimizes the distance of all points to its closer centroid. This result is a local optimum.

Density-based

Clusters in density-based clustering correspond to areas with a high density of objects. As objects in sparse areas are considered noise, this type of clustering can be used even in the presence of outliers or with noisy datasets.

The most commonly used density-based algorithm is the Density-Based Spatial Clustering of Applications with Noise, or DBSCAN (Ester et al., 1996). Its main idea is to find areas that have a minimum density, where points are very close to each other. The dense areas are separated by sparse areas with few points.

To find these areas, the algorithm uses two parameters: i) number of required neighbors $minPts$; ii) radius ϵ.

A point with at least $minPts$ neighbors within an ϵ distance of it is considered a core point. All its neighbors are part of the same cluster. If these neighbors are also core points, it is considered that the neighborhoods belong to the same dense area, and are considered as a single cluster. Non-core points in a cluster are called border points. Points which do not belong to any cluster, as they do not have the minimum number of neighbors, and are not a neighbor of a core point, are considered noise (Schubert et al., 2017).

Probabilistic Clustering

The idea behind probabilistic clustering is that a group of points can represent a statistical distribution. If it is assumed that the data is composed of several distributions of the same type (e.g., Normal/Gaussian distribution), it is possible to identify clusters of objects in the data that can closely approximate such statistical distributions.

One commonly used method is the Gaussian mixture model, which uses the expectation-maximization algorithm (Dempster, Laird, and Rubin, 1977) to find the mixture of Gaussian distributions in the data that better represent the structure of the data. The expectation-maximization algorithm is an iterative algorithm, where each iteration consists in two steps, which have some similitude with the k-means algorithm:

- **Expectation step**, which uses the current parameters of the distributions to determine the expectation of each object belonging to each distribution.
- **Maximization step**, which computes the parameters that maximize the expected log-likelihood of the distribution of objects found in the expectation step.

Neural Network-based

Artificial neural networks may also be used in unsupervised learning. They are called Self-Organizing Maps (SOM) (Kohonen, 1982), and they produce a low-dimensional (usually two dimensions) representation of the dataset, called a map. As it reduces dimensions, in addition to reveal similarities in the data, it can also be considered a dimensionality reduction technique. It is also sometimes useful for visualizations.

The goal of training a self-organizing map is to make the different units of the network respond similarly to certain data patterns. Competitive training is used to discover the parameters of each unit. When a new object is considered by the network, its distance is computed to all the weighted vectors that represent each unit. The winning neuron gets the object in its cluster, and is considered the best matching unit. The weights of that winning unit and its neighbors are updated. This is repeated for all data in the training dataset. The more data it is fed into the network, the more robust the SOM will be to the possible patterns it needs to consider.

2.3.4 Discussion

In this section, the background knowledge in terms of DM and ML techniques was exposed. The most relevant aspect to consider when analyzing all these techniques is that none of these techniques was developed with a diagnostic analytics perspective, while the main goal of this doctoral work is precisely diagnosis.

Data preparation and exploration techniques, as well as unsupervised techniques mostly focus on descriptive analytics, while supervised techniques focus on predictive analytics. This reinforces the idea that all techniques used with a diagnostic analytics perspective were not conceived with this purpose, and as such, careful consideration and application cannot be forgotten. This is one of the main discussions that happen in this doctoral work, starting in Chapter 5. In fact, the mindset ingrained in the use of DM and ML techniques may not be adequate for diagnosis.

When the original objective of the techniques is prediction, for example, it is still possible to use that technique for extracting relevant associations, but it is necessary to consider possible bias introduced by the original predictive purpose.

Some of the mentioned techniques, such as Decision Trees, Random Forest, one-hot encoding, and Chi-Square statistic, are used directly in this doctoral work. The other techniques mentioned, although not used directly in this work, are mentioned in other works in the area, namely in Chapter 4, and their brief description here should ease the comprehension of those works.

2.4 Control Charts

This doctoral work focuses on developing new tools to improve quality by making it more efficient to find the root causes of problems. Another quality management tool that can be used to handle a large amount of data efficiently is control charts. These are used to monitor variables' data and see when such variables present anomalous behaviour, indicating that the system requires intervention to restore the desired quality in the products.

In this work, control charts are used to identify problematic moments in what regards a systematic loss of quality at the end of a manufacturing process. Then, the data pertaining these moments is analysed to discover the root causes, in order to solve the problem permanently, and restore the manufacturing process to the desired state.

To identify these problematic moments, we use control charts based on moving averages: Cumulative-Sum (CUSUM) and Exponentially Weighted Moving Averages (EWMA). These algorithms are capable of identifying sustained small/medium shifts in a manufacturing process by not only using information from the most recent sample of products, but also previously processed samples. These two algorithms are used to control the proportion of problematic products in a lot.

Among the different existing alternatives (e.g., rule-based, moving averages, distance-based charts), we focused on control charts based on moving averages, as we have noticed that the problematic intervals have a "burst-like" or "spike-like" shape, in the sense that the effects of a problem are manifested very quickly, but also can disappear very quickly. An example of this can be seen in Figure 2.9. The capacity of the moving average methods to quickly detect small and medium-sized deviations is adequate for this kind of problems. Also, the data is highly imbalanced, as there are much fewer problematic products than normal ones. This increases the need of methods that are able to quickly detect small changes, as problematic products are much less frequent than normal ones.

CUSUM (Page, 1954) controls the cumulative sum of the differences between the observed proportions and a target value. It signals an out-of-control proportion by identifying (upward and/or downward) drifts in this cumulative sum, beyond a boundary. Expression (2.9) represents the difference between the proportion of problems in the i^{th} lot and the target, and Expressions (2.10), (2.11) represent the lower and higher cumulative sums respectively. $\bar{x}_i$ is the i^{th} proportion value, $\bar{\bar{x}}$ the target value, and $\hat{\sigma}_{\bar{x}}$ is the standard deviation of the proportion values. k represents the size of the shifts that we want to identify, and is set equal to $\hat{\sigma}_{\bar{x}}$. The values of S_{Li} and S_{Hi} are compared to a boundary $\pm h$, which is set to $3\hat{\sigma}_{\bar{x}}$.

$$z_i = \frac{\bar{x}_i - \bar{\bar{x}}}{\hat{\sigma}_{\bar{x}}} \tag{2.9}$$

$$S_{Li} = -max(0, (-z_i - k) + S_{Li-1}) \tag{2.10}$$

$$S_{Hi} = max(0, (-z_i - k) + S_{Hi-1}) \tag{2.11}$$

h and k are parameters set after analyzing the data. In the case of this doctoral research, they were defined considering the case study data, as to enable the algorithm to identify problematic moments without overshooting because of noise. $\hat{\sigma}_{\bar{x}}$ is estimated based on the dataset used, and the distribution of the lots' problematic products proportions. Fig. 2.8 displays an example of a CUSUM control chart, i.e. a depiction of the algorithm. The horizontal axis displays the lot number, while the vertical axis displays the CUSUM statistics, namely S_{Hi} and h (the horizontal line).

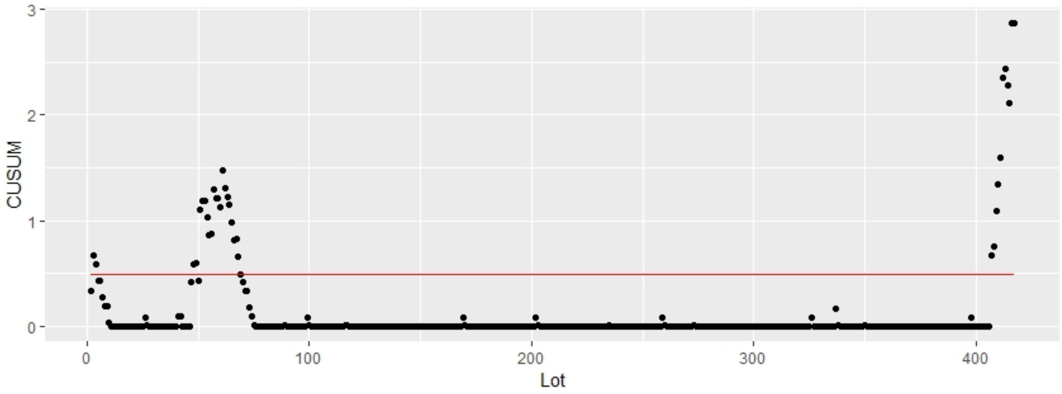


FIGURE 2.8: An example of a CUSUM control chart for upward drift.

EWMA (Roberts, 1959) focuses on controlling a weighted average, where the most recent lots have more influence. Expression (2.12) represents the moving average at lot i, and Expressions (2.13), (2.14) represent the lower and upper boundary. When the moving average z_i crosses these boundaries, a problematic moment is identified. The only parameter in EWMA is λ, which represents the weight of the most recent lot. After analyzing the data from the case study, it was set to $\lambda = 0.4$, as this enables the algorithm to identify problematic moments without overshooting because of noise.

$$z_i = \lambda \bar{x}_i + (1 - \lambda) z_{i-1} \tag{2.12}$$

$$LCL = \bar{\bar{x}} - 3\hat{\sigma}_{\bar{x}} \sqrt{\frac{\lambda}{2-\lambda}[1-(1-\lambda)^{2i}]} \tag{2.13}$$

$$UCL = \bar{\bar{x}} + 3\hat{\sigma}_{\bar{x}} \sqrt{\frac{\lambda}{2-\lambda}[1-(1-\lambda)^{2i}]} \tag{2.14}$$

Fig. 2.9 displays an example a a EWMA control chart, i.e. a depiction of the algorithm. The horizontal axis displays the lot number, while the vertical axis displays the EWMA statistics, namely z_i and UCL.

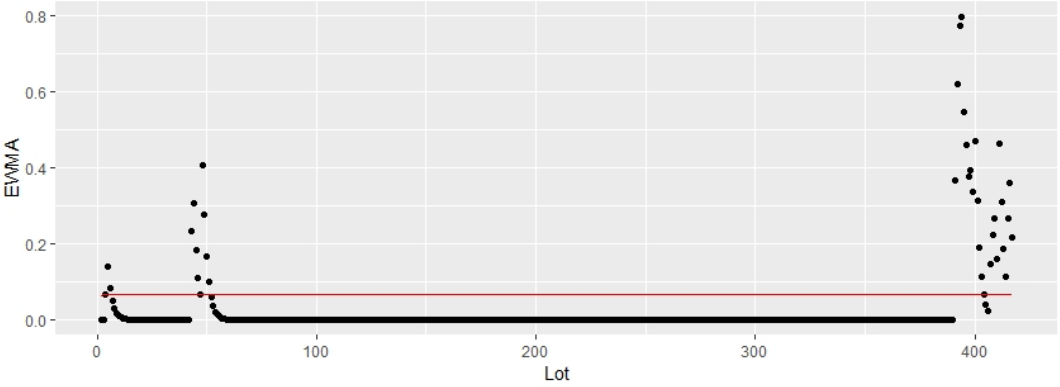


FIGURE 2.9: An example of an EWMA control chart for upward drift.

In the context of this doctoral work, EWMA was chosen as the problematic moment identification to use in Chapters 5, 6, and 7. This choice is based on two considerations. First, during the exploratory phase, EWMA identified the intervals in which a problem occurred correctly, while keeping a desirable granularity of intervals (i.e., not dividing the intervals in several sequential sub-intervals). Second, while considering some effect of the preceding stages can be desirable to understand if the manufacturing process is evolving towards less or more defects, it can also become an issue if done excessively. Such situation happened when using the CUSUM chart, as the effect of the preceding stages was too pronounced, and the identified intervals became too large, which means that the intervals contained data from moments that were not problematic. On average, the intervals identified by the CUSUM chart were 23 time units longer than supposed, as opposed to the EWMA chart, in which the intervals were better aligned with supposed intervals (average of 4 time units longer).

2.5 Causal Inference

In this section, the background theory on causal inference is detailed, as it is necessary for the full comprehension of Chapter 7.

A brief review of causal inference concepts is presented in this section. For a more comprehensive read, please refer to Pearl (2009). Causal inference aims at identifying cause-effect relationships among factors or events. As the aim of this doctoral work is to find the location of problems' root causes, it makes sense to delve into how to adequately relate causes to effects (in this case, the effects are the problems). To do this, we focus on a general theory of causation based on Structural Causal Models (SCM), a theory that subsumes and unifies other approaches to causation, and provides a coherent mathematical foundation for the analysis of causes and counterfactuals (Pearl et al., 2009).

While statistics aims at identifying associations among variables, being able to estimate beliefs or probabilities of past and future events, causal analysis aims to infer not only beliefs and probabilities under static conditions, but also under changing conditions, such as changes induced by external interventions (Pearl et al., 2009). Because the laws of probability do not dictate how one property of the distribution changes when another property is modified, this information must be provided by causal assumptions which identify relationships that remain invariant, or stable, when external conditions change (Pearl et al., 2009). Understanding how we can identify and model these stable relationships allows us to model overlap, and its effect on the manufacturing process. This understanding also allows us to pose causal queries, i.e., questions to extract information of a causal model, namely the probabilities of a variable causally affecting another, given certain conditions. These queries make it possible to understand the causal effect of a product passing through a tuple on its final quality.

As the name indicates, SCM requires a structure that allows us to identify the relations that remain invariant and that we are trying to study. Such structure is provided by causal graphs/models, such as the example in Figure 2.10. The nodes represent variables, and the directed edges represent the possibility of stable causal relationships, that indicate a physical relationship between variables. In this example, X is directly influenced by its parent U, and in turn directly influences its child W. Moreover, it is important to highlight the edges lacking: in this example, the lack of edge between X and Y indicates an assumption that X does not influence Y directly. To relate this example with RCA, we could consider X and W as tuples a product can go trough, Y as the quality of the final product, and U as representing all unknown factors that could affect both X and Y, but we cannot measure. We can also translate a causal model into functional causal models, where each relation is represented by a function, e.g., Expressions (2.15). These expressions detail how exactly one variable influences the other (while the edges only indicate that there is an influence).

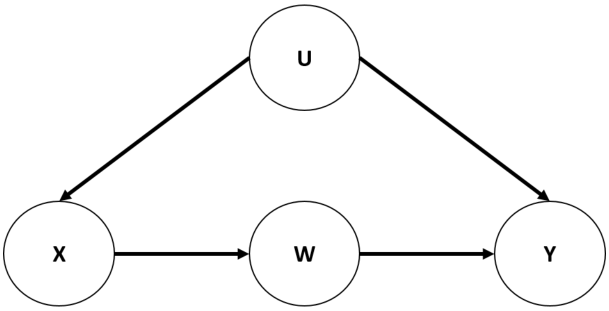


FIGURE 2.10: Example of a causal graph/model.

We can also translate these causal models into functional causal models, where each relation is represented by a function, like the expressions below:

$$\begin{aligned} U &= f_1(\epsilon_1) \\ X &= f_2(U, \epsilon_2) \\ W &= f_3(X, \epsilon_3) \\ Y &= f_4(U, W, \epsilon_4) \end{aligned} \tag{2.15}$$

Regarding Expressions (2.15), f_i (f_1, f_2, f_3, and f_4) can be any function. These functions represent the stable mechanisms through which a variable affects another, e.g., if U occurs, so does X, or for each unit increase in U, X increases by 0.5. In this example, we can see that the arguments of each function correspond to the parents of each variable, and a ϵ_i, which represents noise or uncertainty. These functions are necessary to answer counterfactual queries, while causal graphs are required by both interventional (see Section 2.5 for more details) and counterfactual queries (relating to "what would have happened if" questions, more details in Section 2.5).

Causal graphs are required to answer causal queries because they encode the stable relationships that are possible to assume. These stable relationships allow us to make a change in a relationship, with the reasonable expectations that the change will not affect the other relationships. This modularity opens the possibility of interventional queries, where when we change something in the model, this does not result in a change in the rest of the model (Pearl, 2009). Hence we can "simulate" interventions, just like we would do in a randomized experiment, and obtain the probabilities of such interventions using observational data (i.e., data which results from passive observation, and not from experimental manipulation and control). Such probabilities can be translated as causal effects.

Interventional Queries

An interventional query is a type of causal query where we compute the effect of how changing or intervening in a variable affects another variable. These queries assume the form of Expression (2.16).

$$P(Y = y|do(X = x_i)) \tag{2.16}$$

Expression (2.16) means the probability of $Y = y$, knowing that we perform an intervention "forcing" X to assume the value of x_i (indicated by the $do(\cdot)$ operator). This can also be interpreted as the causal effect of X on Y. This probability is different from a regular conditional probability $P(Y = y|X = x_i)$, because the latter assumes that no changes to the causal structure are made, while the intervention indicates that the influence of the parent variables is disconnected, because the value of X is not influenced by its parents, but is "forced" upon the variable by the intervention. The new graph resulting from the intervention is depicted in Figure 2.11. As can be seen in Figure 2.11, X and U are no longer connected by an edge, and instead the constant x_i is imposed on variable X.

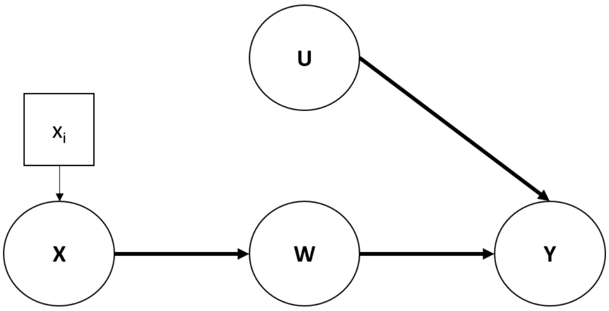


FIGURE 2.11: Example of the causal graph in Figure 2.10 after the intervention $do(X = x_i)$.

In order to estimate the effect of interventions, it is relevant to introduce the concept of d-separation. D-separation helps understand graphically when it is possible to compute the causal effect of a variable on another one, and which other variables we need to know to do so. It is a graphical rule for understanding when are two variables independent from each other given a set of other variables, and which variables one needs to control in order to make a correct judgement without being affected by spurious correlations. From Pearl (2009), a set of variables Z is said to d-separate X from Y if and only if Z blocks every path from node X to node Y. A path is blocked when: i) it contains a chain $i \rightarrow m \rightarrow j$ or a fork $i \leftarrow m \rightarrow j$ such that the middle node m is in set Z, or ii) it contains an inverted fork (or collider) $i \rightarrow m \leftarrow j$ such that the middle node m is <u>not</u> in set Z and such that no descendant of m is in set Z. When Z d-separates X and Y, we can say that X and Y are independent given set Z. In the example in Figure 2.10, X and Y are d-separated by the set of variables U, W, which blocks all paths and spurious correlations from Y to X. Please note that the empty set $\{\emptyset\}$ can d-separate two variables. In such case, X and Y are fully independent.

To compute the effect of an intervention using observational data there are two possible ways relevant for this work: i) adjustment for direct causes, and ii) back-door criterion. The basic idea in both is that, if we control a set of variables that d-separates X and Y, it is possible to compute the effect of the intervention using preintervention probabilities.

The adjustment for direct causes is based on the following: if we control all the direct causes (or parents) of the possible cause X, we eliminate all spurious correlations that could influence X. Ergo, if pa is the set of direct causes of X, and Y is not equal to X or any pa_i, the effect of the intervention $do(X = x_i)$ on Y is given by:

$$P(Y|do(X = x_i)) = \sum_{pa_i} P(Y|X = x_i, pa_i)P(pa_i), \tag{2.17}$$

where $P(Y|X = x_i, pa_i)$ and $P(pa_i)$ represent preintervention probabilities (Pearl, 2009).

The adjustment for direct causes works well in the scenarios where we know and can measure all parents of X. However, such is not always possible. For example, in Figure 2.10, we may not know or be able to measure variable U. In such case, it would not be possible to compute the effect of the intervention by adjusting for the direct causes. However, the parent variables of X are not the only set of variables that d-separates X and Y. If we are able to find such a set Z that: i) no node in Z is a descendant of X, and ii) Z blocks all paths between X and Y that contains an arrow into X, Z can be used instead of the direct causes. This is called the Back-Door (BD) criterion (Pearl, 2009). And, if a set Z satisfies the BD criterion relative to (X, Y), then the causal effect of X on Y is identifiable and is given by:

$$P(Y|do(X = x_i)) = \sum_{z} P(Y|X = x_i, z)P(z), \tag{2.18}$$

where $P(Y|X = x_i, z)$ and $P(z)$ represent preintervention probabilities (Pearl, 2009). Therefore, we can "replace" the control on the direct causes (that we may not know), by a control on a set Z, from which all variables are observed, and that block all spurious correlations.

Counterfactual Queries

Counterfactual queries relate to "what would have happened if" questions. They estimate what would have happened in a world slightly different from the one observed, knowing what actually happened in the observed world. Counterfactual queries are not directly addressed in this doctoral thesis, but they are still relevant to understand concepts on probability of causation. The counterfactual sentence "Y would be y, in situation e, had X been x" is interpreted as the equality $Y_x(e) = y$, with $Y_x(e)$ being the potential response of Y to $X = x$ (Pearl, 2009). In other words, given that we have observed e (which could be $e = (Y = \overline{y}, X = \overline{x})$), what would happen to the value of Y in a world where $X = x$ occurred instead of the observed $X = \overline{x}$ (considering x and y to be binary variables, $\overline{x}$ and $\overline{y}$ mean that the variables are not active, or have value equal to zero).

To compute such counterfactual queries from observational probabilities, we need: i) observed probabilities, ii) a causal model, iii) a functional causal model, i.e., what would be the behavior of the variables if we change something in the model.

The conditional probability $P(y_x|e)$ of a counterfactual sentence "If it were x then y", given evidence e, can be evaluated using the following three steps (Pearl, 2009):

1. **Abduction** - Update the joint probability of all variables $P(u)$ by the evidence e to obtain $P(u|e)$.

2. **Action** - Modify the model by the action $do(X = x)$.

3. **Prediction** - Use the modified model to compute the probability of $Y = y$, the consequence of the conterfactual.

In other words, we use our observations to compute the actual value of the exogenous values (u), i.e., the variables that may influence the variables of the model but that we do not observe, and then keep the value of the exogenous variables in a modified model, subject to the intervention we are trying to estimate the effect of. To perform the abduction step, it is required to know exactly how the variables are connected. It is for this reason that a functional causal model is required.

When considering the probability of causation, Pearl (1999) introduces three types of causation, based on counterfactual queries. As the topic of this thesis is root cause analysis, it is pertinent to look into these types of causation in more detail. The three types of causation are: i) necessary cause, ii) sufficient cause, and iii) necessary and sufficient cause.

A necessary cause relates to the Probability of Necessity (PN), which measures how necessary the cause is for the production of the effect. If we consider X and Y as binary variables, PN can be defined as follows:

$$PN \triangleq P(\overline{y}_{\overline{x}}|x, y) \tag{2.19}$$

In other words, what is the probability that x not occurring would have stopped y from occurring, given that both x and y effectively occurred.

A sufficient cause relates to the Probability of Sufficiency (PS), which measures if the cause alone is sufficient for the production of the effect. PS can be defined as follows:

$$PS \triangleq P(y_x|\overline{x},\overline{y}) \tag{2.20}$$

In other words, what is the probability that y would have occurred if x had occurred, given that both x and y did not occur.

The Probability of Necessity and Sufficiency (PNS) measures how likely an instance is affected both ways. Expression (2.21) shows how to compute PNS:

$$PNS \triangleq P(y_x,\overline{y}_{\overline{x}}) \tag{2.21}$$

PNS, PN, and PS are related according to Expression 2.22 (Pearl, 2009):

$$PNS = P(x,y)PN + P(\overline{x},\overline{y})PS \tag{2.22}$$

2.6 Discussion & Conclusions

The background knowledge described in this chapter is either used in the doctoral work, or mentioned by other studies referenced in the literature review.

The traditional RCA techniques mentioned in Section 2.2 establish the starting point of the RCA process, and some of them serve as guiding principles to the development of certain concepts presented in this thesis. For example, the idea behind the five whys is similar to the one used in conceptualizing the types of data in ARCA solutions, in Chapter 4, Section 4.2.

Of all the DM and ML techniques presented in Section 2.3, decision trees are frequently used in the literature, as they are an easily interpretable technique. In this thesis, it is used as a benchmark to compare classifiers to the novel proposed approaches. The chi-square statistics is the basis of the first measure of overlap, presented in Chapter 5. Random forest is also used in Chapter 5 for its capacity of estimating the importance of factors. The other techniques presented in this section, although not used directly in this work, are referenced in the literature review, and should help in better understanding Chapter 4. In this section it is also of relevance the discussion of the different types of analytics that exist, and the fact that most techniques focus on descriptive or predictive analytics, while this doctoral work focuses on diagnosis. This notion of discrepancy between the techniques original purpose and the purpose intended in diagnosis is critical to understand why overlap is an important issue, and needs to be tackled.

The control charts presented in Section 2.4 are an integral part of the proposed approaches in Chapters 5, 6, and 7, as they are used to determine which data represents a problematic moment that requires an RCA process.

Finally, the knowledge in Section 2.5 is necessary for the comprehension of the approach proposed in Chapter 7.

Chapter 3

Problem & Case Study Definition

3.1 Introduction

The aim of this chapter is to formally define the problem that this work is tackling, frame it within the literature, and provide additional information about its real life context.

This chapter starts by formally defining the problem as the development of ARCA solutions using Location-Time data, and using the **Factors** ⇒ **Problem** type of methodology (as defined in Chapter 4). We also present overlap, and why it is a critical issue and the central focus of this doctoral work. Then, the semiconductor manufacturing process that is the base of the case study that originated the study of this problem is detailed. In addition, this chapter introduces a stochastic simulator developed to emulate the real case study, and to have more data and more capacity to test different scenarios, which increases the robustness of the validation of the proposed methodologies. As such, the structure of the stochastic simulator is presented. This chapter is concluded by presenting a brief discussion of the materials presented here, and how they impact the following chapters of the thesis.

3.2 Problem Definition

The central problem of this doctoral work is to develop ARCA solutions using data on the flow of products through the manufacturing process. Also, the products are labeled according to their quality (the products can be labeled as problematic or normal). As such, it is possible to frame this work in the literature as analysing data of the Location-Time type, and that its methodology is of the **Factors** ⇒ **Problem** type (both these terms are defined in the conceptualization presented in Chapter 4). When developing solutions within this scope, the objective is to locate the root cause of a problem within the manufacturing process.

Figure 3.1 illustrates a manufacturing process flow that generates a dataset that can be used to locate a root cause. In a manufacturing process, products (identified in

Figure 3.1 by the circles with P_i) go through several manufacturing steps (e.g., machining, coating), and in each of those steps they are processed in a certain machine. All products go through the same steps, and some machines can perform more than one step. While the example in Figure 3.1 is a "still frame" of the manufacturing process, in the data generated by the process, each row depicts the full route that a product went through in the manufacturing process. In this illustrative example, P1 was already processed and it is being monitored, P2 is being processed in Machine 3, and P3 is waiting to be processed before Step B. There is a problem causing a systematic loss of quality. The root cause of the problem is in Step A, Machine 1, indicated by a warning sign. This manufacturing process flow can be originated in both job shop (where machines are multi-purpose) and flow-type production environments (where machines are specialized in a single step), as long as there is more than one possible machine for each step. At the end of the process, the products' quality is monitored. If the quality lowers significantly for a given time period, it means that a systematic problem occurred in the manufacturing process, and RCA is required to help finding the origin of the problem.

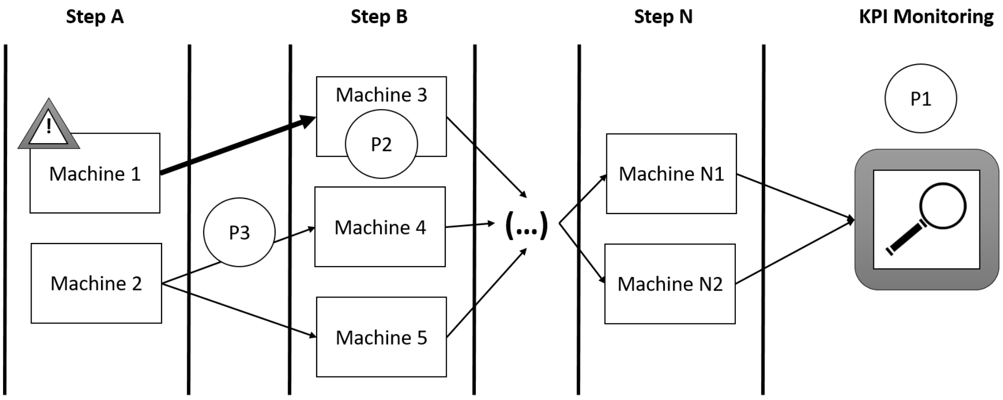


FIGURE 3.1: An illustration of a process that generates Location-Time type of data.

The manufacturing process generates data like the one in Table 3.1, in which each row represents a product, and each column represents a manufacturing step as attributes or factors. For each product, the machine that was used in each step is known. In each step/column is registered the machine used. The "Problem" column registers whether the product had a problem or not. When performing RCA with such data, the aim is to determine the tuples of steps and machines that are the most likely root causes.

When overlap occurs, it creates a situation where it can become impossible to distinguish the effects of one tuple from another, hence the name overlap. In the example in Figure 3.1 and Table 3.1, all products that go trough Machine 1 in Step A also go through Machine 3 in Step B. Also, there are no products in Machine 3 that come from machines other than Machine 1. As the root cause is in Step A - Machine 1, this is problematic, as we cannot know which of the tuples is the root cause. Therefore we have overlap between possible root causes (or tuples that represent locations), with apparently the same influence on product quality, while, in reality, only one of them is the true root cause.

Overlap can occur due to stabilization in the manufacturing process - e.g., in continuous-flow manufacturing - where, for example, as soon as a product finishes processing in one machine, it always has the same machine available in the next step, which became available at that moment. Note that overlap can occur between machines that do not operate in contiguous steps. This stabilization is desirable in terms of productivity and efficiency of the manufacturing process, but it introduces the issue of overlap when analysing the data resulting from such process. This clash between what is desirable for production (that should be prioritised) and what is desirable for diagnosis through data analysis is very relevant, as this means that overlap is a problem that will occur frequently whenever we analyse this type of data for performing diagnosis.

TABLE 3.1: Example of data in a Location-Time type of problem.

Product	**Step A**	**Step B**	**...**	**Step N**	**Problem**
1	*Machine 1*	*Machine 3*	...	Machine N1	*1*
2	Machine 2	Machine 4	...	Machine N2	0
3	Machine 2	Machine 5	...	Machine N1	0
4	*Machine 1*	*Machine 3*	...	Machine N2	*1*
...	...	...	...	...	...

Overlap becomes even more critical when the analysis is performed automatically, specially when using classification algorithms. When training a classifier on a certain dataset, the knowledge structures (e.g., decision trees) are generated by selecting the most representative factors, often disregarding highly correlated factors as they contain what is considered redundant information. In other words, the quality problem can be originated by a factor that is highly correlated to another, and the latter factor is selected at the expense of the factor that originated the problem, which is discarded. The criterion used to determine what is considered redundant or not may lead to hiding a factor that is indeed the root cause, assigning a higher importance to another factor. For example when generating a Decision Tree (DT), the use of information gain criterion for node splitting favors factors with more levels, while the root cause may be a factor with a smaller number of levels. In the example above, Step B - Machine 3 would be selected because Step B can be performed by more machines, while the root cause is located in Step A - Machine 1. Therefore, using a DT with information gain would yield the incorrect root cause, and hide the real one.

It is important to note why it is not better to simply remove the overlapped factors, which is the approach followed by classical methods to remove multicollinearity (e.g., information criterion, eigenvalues of correlation matrix, minimum mean partial correlation). Such approach is also followed when using regularization in machine learning models. However, these popular approaches used to prevent multicollinearity appear in the context of predictive analytics. Their objective is to remove/prevent multicollinearity to avoid misleading predictions. The approach proposed in this study appears in the context of diagnostic analytics, where the objective is not to predict a label value (as in predictive analytics), but to understand which factors led to a certain (undesirable) state (which is reflected on the label). We believe that in the context of performing diagnosis in location data, the traditional approaches (as used in regression and classification), may lead to an incorrect diagnosis because they remove variables with multicollinearity. As correlation does not perfectly reflect the causal structure between variables, eliminating factors simply because of multicollinearity can lead to a wrong diagnosis, as these factors can have a major causal impact on the state we wish to diagnose. The approach we propose allows us to recognise all possible factors with high impact on the analysed state, without them being eliminated due to measures applied for prediction, and without requiring knowledge of the causal structure.

3.3 Case Study

The required background knowledge about the manufacturing system our case study is based on is detailed in this section.

3.3.1 Semiconductor Industry

First, it is important to have a clear picture of how the semiconductor industry is organized. Considering Figure 3.2, the first step is to transform the raw materials in order to create the silicon wafers that are used in semiconductor manufacturing. A wafer refers to a circular slice of silicon where the desired circuits are imprinted. They are called wafers before and after the circuits have been imprinted, but for clarity, in this document the term “silicon wafer” will refer to the wafer before the circuits are imprinted. A silicon ingot is created from silicon crystals (S1). Then, silicon wafers are obtained after slicing a silicon ingot (S2). The chip and its circuits are designed (S3), and masks are created (S4) that enable the circuits to be engraved in the wafers using photo-lithography. In the step S5, the desired design is engraved on the wafers, in what is called front-end production. This is, several dies (or chips)

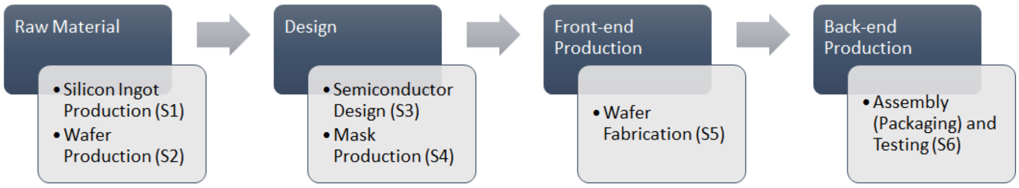


FIGURE 3.2: High level representation of the semiconductor manufacturing industry.

with the same design are engraved in the wafer in a layered process, resulting in a fabricated wafer, similar to the one that can be seen in Figure 3.3.

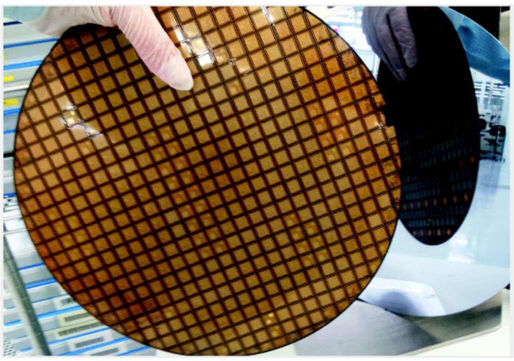


FIGURE 3.3: Fabricated wafer. Each square in the round wafer corresponds to a die (image property of Amkor Portugal).

In the step S6, the chips are tested and packaged, in order to ensure their quality and increase the durability of the chip and its resistance to environmental hazards, respectively. Besides improving the durability of the die, this step also performs the redistribution of contacts from the die to the external structure that supports the die.

Each of the different steps is usually performed by different companies, with some making more than one step, depending on the degree of verticalization. The project that was part of the context of this doctoral work aimed to answer a root cause identification problem in a company dedicated to semiconductor packaging, i.e., S6. As such, we will now describe this step's processes in more detail.

3.3.2 Semiconductor Packaging Process

Traditionally, Outsourced Semiconductor Assembly and Test (OSAT) companies used to first slice and separate the multiple dies in the wafer, put them in a packaging material, and then redistribute the contacts by soldering the Inputs/Outputs (I/O) of the die to the external I/O of the external package.

Nowadays, a more advanced process called Wafer-Level Packaging (WLP) can be used. As the name indicates, this technique focuses on packaging the dies at a wafer level in a way that enables the manufacturing of smaller final products by packaging them on a wafer level, which reduces the outer area of the package. There are two types of WLP: Fan-In and Fan-Out. In Fan-In, the dies are packaged while still in the silicon wafer. This enables to obtain very small chips. However, they have a reduced area in which to put the solder balls that constitute the external I/O. As a compromise solution between size and number of I/O, in Fan-Out WLP the dies are first sliced, and then put in a new, reconstructed epoxy wafer, where the package will have greater superficial area than the original die. This leads to an increased number of I/O when compared to Fan-In WLP, and smaller size when compared to the traditional methods. A cross section of the final product of a Fan-Out WLP can be observed in Figure 3.4. The original chip lies at the core, and is enveloped on top by mold (the fan-out area), and on the bottom by a dielectric, which protects the chip. This dielectric has in its core a redistribution layer that connects the chip in the core with the solder balls that are the interface with the exterior.

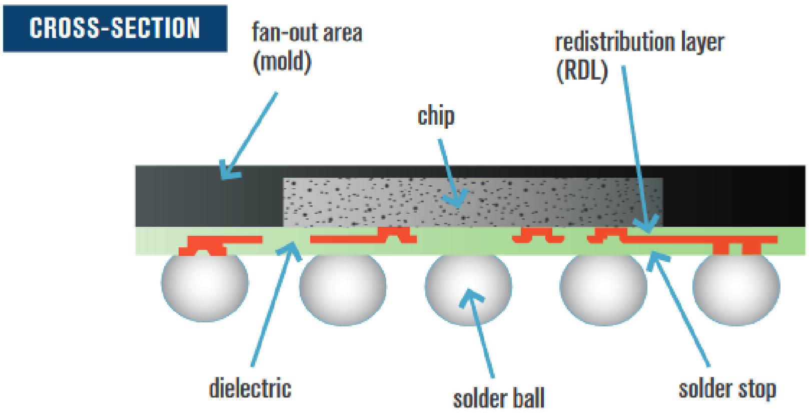


FIGURE 3.4: Schematics of a cross-section of a chip packaged trough Fan-Out WLP (image property of Amkor Portugal).

In order to package and redistribute the connections, the technology used in WLP is very similar to the one used in the front-end step (see Figure 3.2), which is much more complex than the one used traditionally by the OSATs. The WLP packaging process is represented in Figure 3.5

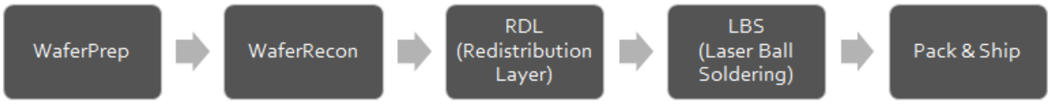


FIGURE 3.5: Wafer-Level Packaging Process.

The stages involved in this process can be described as follows:

WaferPrep is a step that focuses on preparing the fabricated wafer to be packaged. The wafers are laminated and thinned, and afterwards the dies are sliced. This step is exclusive to Fan-Out WLP.

WaferRecon is also exclusive to Fan-Out WLP, and takes the individualized dies from WaferPrep, reconstructing the wafer using epoxy. The dies are picked and placed on a temporary wafer, with their active part (the one with electrical connections) facing down, and then an epoxy mold is put on top, reconstructing the wafer. After that, the temporary wafer is separated, and the epoxy one moves along the production line. The output of this step corresponds to a wafer where the chips are enveloped in the epoxy, similar to the black part in Figure 3.4. It is important to note that all processes are applied on a wafer level, in order to process several dies concurrently. This is one of the aspects that enables the production of semiconductors to scale.

Redistribution Layer (RDL), is the core process of WLP, and it is where the electrical connections of the die are redistributed in order to reach the external I/O in the desired format, as established by the chip design. This step is also the one that is technologically closer to the front-end production processes, as it uses layered photo-lithography in order to build the connections. The end result is a wafer where the dies have the dielectric and a fan-out area (light green and black layers in Figure 3.4). The case study considered focuses on this step.

Laser Ball Soldering (LBS), aims at soldering the balls that are the final I/O of the chip, which connects them with the external components. First, a reflux is applied to facilitate the soldering, and then the balls are attached to the wafer. Finally, the chips are individualized, and the final product (as seen in Figure 3.4) is obtained.

Pack & Ship is a logistic step where the individualized ships are marked using laser (with product type and serial number), and are packed and shipped to the next step of the logistic chain (usually the assembly of several electronic components to make consumer products).

As the considered case study is focused on the RDL step, a more detailed description of the process will be provided.

3.3.3 Redistribution Layer (RDL) Process

As mentioned previously, the objective of the RDL process is to redistribute the electrical contacts from the die to the solder balls that are the external I/O. This redistribution of contacts is achieved through a layered procedure. Figure 3.6 presents a scheme of the RDL layers.

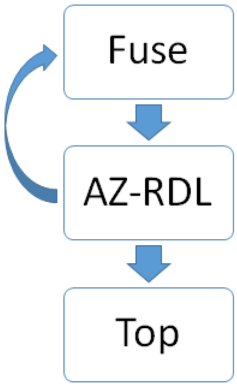


FIGURE 3.6: The different layers of the RDL process.

- The **Fuse** layer is the first dielectric layer, intended to protect the active part of the die. It covers the whole die and the fan-out area, except for the internal I/O, present at the active part of the die.
- The **AZ-RDL** layer (AZ comes from the material used) is the layer where the conductive pathways that redistribute the contacts are created. First, a thin

layer of titanium and copper, known as the seed layer, is deposited on top of the Fuse layer. Then, photo-lithography is used to design the conductive path. After that, the conductive paths are filled with copper, starting from the seed layer. Then, the dielectric used in the photo-lithography and the seed layer that was not filled are stripped away. As such, this layer is also known as the sacrificial layer.

- The **Top** layer, is the final dielectric layer, intended to protect the redistribution pathways. It covers the whole packaged chip, only leaving in the open the areas where the solder balls will be attached. As can be noticed in the diagram in Figure 3.6, not always the AZ-RDL layer proceeds to the Top layer. For example, for more complex designs, several RDL-layers may be required to achieve the necessary redistribution. In such cases, another Fuse layer is necessary on top of the existing pathways in order to protect them, and the AZ-RDL layer process has to be repeated. The Top layer is only put after all redistributions are performed.

After describing what are the different layers involved in the RDL process, it is also important to understand the sub-processes involved in the RDL process. These sub-processes can be divided into three categories: Lithography, Wet, and Dry.

- **Lithography** processes are the core processes, as it is through them that the several dielectric layers (see Figure 3.4) are placed on the chip, and the design is imprinted on the chip.
- **Wet** processes are processes that use liquid chemicals to either add or remove material.
- **Dry** processes also focus on adding or removing material, but use physical processes and gases or plasma to do so. They also use furnaces to remove humidity from the wafer, to strengthen the dielectric layer, and to correct dimensional changes in the wafer (expansions and warpages).

In addition to these, **Metrology** tests are performed in order to control the quality of the products. Two metrology tests are of interest for this work:

- **Process Control Monitoring (PCM)** tests, which are electrical tests that measure the contact resistance of conductive parts, and if there are any electrical current leaks between pathways.
- **Automatic Optic Inspection (AOI)** are tests to compare the surface of the packaged chip with a standard to verify if the product is in conditions of moving forward. The comparison is performed automatically with image recognition algorithms, and the defects are validated by human operators.

Figure 3.7 presents a scheme detailing the sub-processes that are done at each layer, and to what category they belong to.

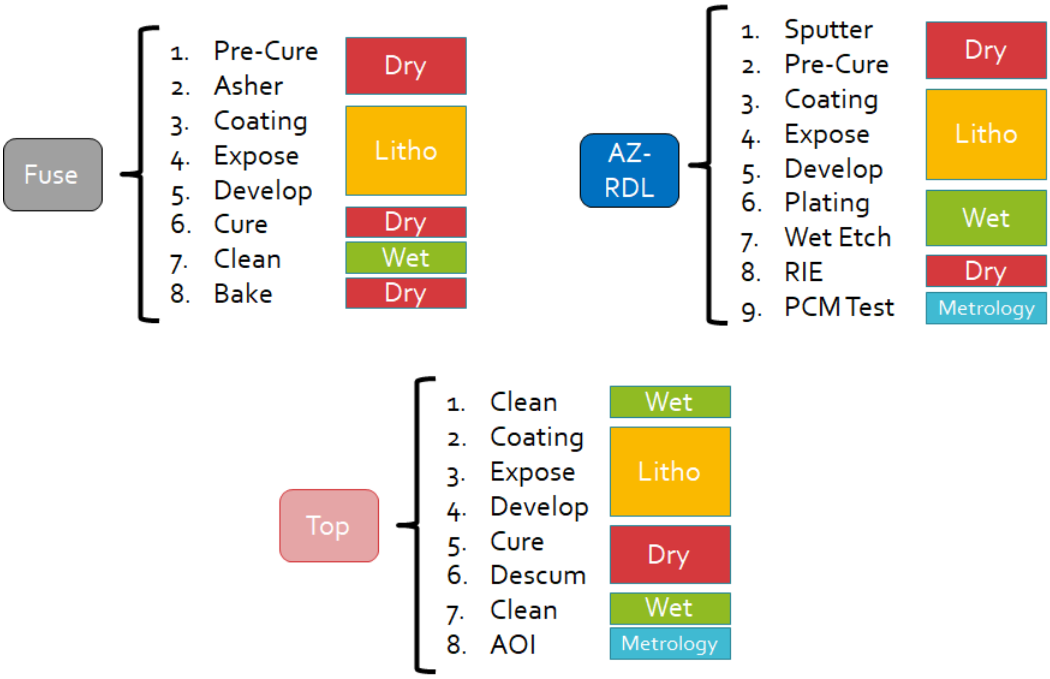


FIGURE 3.7: The different sub-processes of each layer of the RDL process.

Sub-processes in the Dry category are briefly described below:

- **Pre-Cure, Cure** and **Bake** use furnaces in order to either remove humidity from the wafer, strengthen the dielectric layer, or correct dimension changes in the wafer (expansions and warpages). Pre-Cure and Cure are more focused on dimensional correction, while Bake is more focused on stabilizing the dielectric layer.
- **Asher** and **Descum** sub-processes' goal is to slightly thin down the active part of the die, in order to remove impurities.
- **Sputter** is the sub-process in which the seed layer is deposited in the wafer. This seed layer is required in order to add the material that will compose the conductive pathways. This layer is constituted by extremely thin layers of titanium and copper (in this order), and also help connecting the die to pathways.
- **RIE**, or Reactive Ion Etching, is a mix of a wet and dry process, and has similar purposes to Asher and Descum processes. It is indicated to prepare the pathways to connect to the next layer.

Lithography sub-processes can be briefly described as:

- **Coating** is a process in which the wafer is covered in a dielectric material called photo-resist, which reacts when exposed to light.
- **Expose** is a process in which, after the wafer is coated in photo-resist, the wafer

is exposed to light filtered through a mask, which "draws" the desired pathways in the dielectric layer, chemically destabilizing part of the dielectric.

- **Develop** is a process through which the destabilized parts of the dielectric layer are removed, revealing the underneath layer.

Wet sub-processes can be briefly described as:

- **Clean** is a process in which liquid chemicals are used to clean the surface of the layers and remove impurities.
- **Plating** is a process in which copper is grown from the seed layer to create the conductive pathways and redistribute the electrical contacts.
- **Wet Etch** aims at removing the sacrificial layer of the dielectric and the seed layer remaining beneath it.

Although the processes remain the same for different products, these same processes may have different parameters for each process' variables (e.g., pressure, temperature), which are defined in recipes, which result from the desired design.

In this doctoral work the focus is on the RDL process, namely in the locating the root causes of the overkill problem. This issue occurs when the AOI generates too many false detections of defects due to exterior changes in the product, that do not necessarily make the product defective. This affects production by increasing the amount of products that need to be manually inspected, increasing cycle time.

3.4 Stochastic Simulator

The case study described in the previous section was the starting point for the work presented in this document. However, the effort of determining the actual root causes by the company's experts revealed to be too burdensome, and it was not possible to obtain a reliable labeling of the root causes. The data from the case study consists in a dataset similar to the one presented in Table 3.1. The data is, therefore, of the Location-Time type, and describes the steps and machines a product has gone through. The case study dataset has 22 steps, 62 machines, and 6144 products (rows), and reports to a month of production data in the case study's factory.

Therefore, instead of relying merely on observational data, a stochastic simulator was developed, to generate data that emulated the data from the case study. This had the advantage of enabling us to pre-establish root causes to see if the approaches developed can find them.

The developed simulator is a discrete and stochastic simulator, and its structure is presented Figure 3.8. Each simulation process triggers or is triggered by events. Triggers are represented by arrows coming from or pointing to the Event List. The number on each trigger identifies the simulation process that is triggered. Arrows going into the event list mean that a simulation process is scheduled for the future, and an arrow coming out of the event list means that a simulation process is executed. Sometimes several simulation processes can be scheduled or executed at the same time, and this is indicated by the multiplication factor next to the arrow. The objective of the simulator is to generate datasets similar to the one in Table 3.1. The manufacturing process is simulated, and, at the end of the process, the information about each product is stored as a row in a dataset, which has the machine/equipment (and sub-equipment) and the time where/when each product was processed for a certain step, and an indication if the product has problems or not.

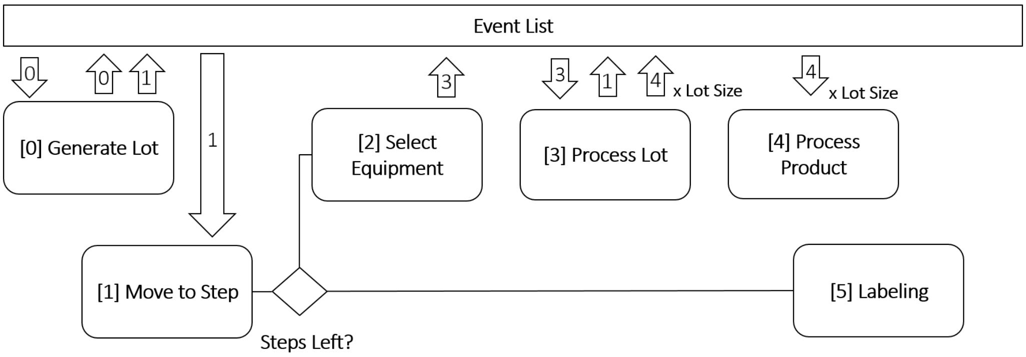


FIGURE 3.8: Diagram of the stochastic simulator structure.

There are three components in the simulator:

- A Clock (not depicted), which controls the flow of time in the simulation. It is basically a counter which is incremented until it reaches a stopping condition (either a time limit or a limit in the number of products generated).
- An Event List, which stores the events that happen in the simulated manufacturing process and triggers their execution when the time is right.
- Simulation Processes, which trigger and are triggered by events. There are six simulation processes: “Generate Lot”, “Move to Step”, “Select Equipment”, “Process Lot”, “Process Product”, and “Labeling”.

The stochastic simulator is initialized with the clock at zero seconds and a “Generate Lot” event, and from there the simulated manufacturing process unfolds. Each lot has a predefined number of steps.

The simulation processes can be described as follows:

0. **Generate Lot** generates a lot of a predefined size, moves it to the first step, and schedules the generation of the next lot according to a negative exponential distribution with a predefined λ_{TBLG}.
1. **Move to Step** checks if there are any steps left for that lot, and, if there are not, sends the lot for labeling. If there are, it starts a "Select Equipment" process.
2. **Select Equipment** selects an available piece of equipment to process the lot and schedules an event to process that lot in that available piece of equipment.
3. **Process Lot** determines the time it will take to process each product, and schedules a number of events of "Process Product" equal to the lot size. It also schedules the move to the next step after all products have been processed.
4. **Process Product** just adds the time and machine when/where that product was processed to the data set.
5. **Labeling** consists of determining whether a product is problematic or not. How this is done is explained in more detail in the last three paragraphs of this section.

The parameters of the simulator are detailed in Table 3.2. The parameters that can be controlled in the stochastic simulator are: Number of Products, Number of Machines, Number of Steps, the Lot Size, the Maximum Number of Sub-Machines, the Time Between Lot Generation (TBLG), the parameters for the Uniform and Gaussian distributions for Processing Times, the range of Setup Time, and the parameters for the Labeling Noise. All values were selected based on an analysis of the data from the case study. All distribution parameters that report to time (λ_{TBLG}, Uniform, Gaussian, Setup Time) are in seconds.

TABLE 3.2: Parameters of the stochastic simulator.

Parameter	**Value**
Num. Products	5004
Num. Machines	63
Num. Steps	22
Lot Size	12
Max. Num. Sub-Machines	6
λ_{TBLG}	400*60
Uniform dist. Range	[5*60, 10*60] - [20*60, 50*60]
Gaussian dist. Range	[5*60, 30*60], [0.5*60, 2*60]
Setup Time	[20, 30]
Normal Noise	Beta(α, β), $\beta \in$ [3,4,5,10,20], $\alpha = 1$
RC Noise	Beta(α, β), $\alpha \in$ [3,4,5,10,20], $\beta = 1$

The parameters Number of Products and Number of Steps define the size of the dataset. The Number of Machines defines the number of levels that exist as a whole on the dataset, which are then distributed through the different steps. The Lot Size defines the number of products that are generated at the same time and are grouped together throughout the process. Each machine has a random number of sub-machines associated with it, and the maximum number is defined by parameter Maximum Number of Sub-Machines. The time between lot generation is defined by a negative exponential distribution, with the parameter being λ_{TBLG}. This value was defined as 400 minutes. The processing time of each step is defined based either on a Uniform or a Gaussian distribution. Which distribution to use in each step is selected randomly when a step is generated. The lower limit of the Uniform distribution varies between five and ten minutes, while the upper limit varies between 20 and 50 minutes. The average of the Gaussian distribution varies between five and 30 minutes, with a standard deviation ranging from half a minute to two minutes. The parameter Setup Time defines the limits of a brief delay when the machines start working, and varies between 20 and 30 seconds. The noise parameter are described in greater detail in the next paragraphs of this section.

The simulation stops when a predefined number of products has been labeled. The labeling process requires a more detailed description, as several characteristics of real operation of a manufacturing process can impact labeling. There is the possibility that noise is introduced in the problem due to human errors in the collection of the data used for labelling purposes. For example, maintenance operators may not register the root cause, or may not do so correctly, or fix a symptom and assume that as root cause. As this possibility could undermine any data-driven attempt to RCA, it was included in the data generator the possibility of controlling the amount of noise present in the labels. When analyzing the noise in the label, it can be divided into two kinds: normal noise, i.e., changes in the label due to unforeseeable circumstances, not related to the root cause, affecting all products the same way; and root cause (RC) noise, which determines how likely a product that went through a RC will be mislabeled.

In the stochastic simulation, the labeling process gives each product a number between 0 and 1, and if this number is above 0.5 it is considered a problematic product. This number is determined based on the two kinds of noise mentioned above: all products receive a *normal noise* component based on a beta distribution with parameters $\alpha = 1$ (fixed) and $\beta >= 2$, and products which go through a root cause have another component based on a beta distribution with parameters $\alpha >= 2$ and $\beta = 1$ (fixed). These distributions and the values of their parameters were selected after analyzing the data provided by the company used as case study. We compared the simulated datasets with the case study dataset, and these are similar in structure. Also, the results in terms of overlap are similar, as can be seen in Sections 5.5.3 and 6.3.2. With this labeling process, it is possible to control the levels of noise present in the dataset. For example, a product that did not pass through a root cause machine when $\beta = 5$ has a 3.125% chance of being a problematic product.

During the exploratory analysis, several datasets were generated with different values for α and β, in the range of $[3, 4, 5, 10, 20]$, which totals of 25 different levels of noise. After the exploratory analysis and comparison with the real case study, the choice was to use three datasets with the levels of noise $(\alpha = 10, \beta = 10)$, $(\alpha = 10, \beta = 20)$, $(\alpha = 5, \beta = 20)$. These three datasets were used in the stochastic simulation experiments described in Chapters 5, 6, and 7, where they are referenced as datasets 1, 2 and 3, respectively. The choice of these datasets is motivated by the fact that they are the most similar to the case study data, and, as validation of the root causes is a manual process, to ensure that the validation is executed in a feasible amount of time.

3.5 Discussion & Conclusions

In this chapter, the central problem of this doctoral work has been described and detailed. In the literature, it is framed within the development of ARCA solutions that use the Location-Time type of data, and within the methodology of associating factors to the problems, without previous knowledge of the root causes for training the models.

The case study that serves as the basis for this work is of a back-end semiconductor manufacturing process, more specifically in finding overkill issues within the RDL process, the most technologically sophisticated part of the back-end semiconductor manufacturing.

Finally, we describe the stochastic simulator developed, which is based on the above mentioned case study, and that is used to generate more data, with controlled labeling, and that aids in making the validation more robust. The datasets provided by the simulator are used in the experiments in the following chapters, in addition to the data about the case study.

Chapter 4

Automatic Root Cause Analysis in Manufacturing: An Overview & Conceptualization

4.1 Introduction

In this chapter, a literature review of Automatic Root Cause Analysis (ARCA) solutions in manufacturing is presented. As mentioned in the previous chapters, Root Cause Analysis (RCA) is a critical process in manufacturing.

In addition to its relevance, RCA is a very complex process, requiring extensive system and execution analysis (Steinhauer et al., 2016). As the volume of data is growing at an unprecedented rate in manufacturing (Choudhary, Harding, and Tiwari, 2009; Industry Today, 2020; PRNewswire, 2020), it is possible to develop more complex models. This increase in data availability and model complexity leads to an increase in complexity of the analysis (Stasko, Görg, and Liu, 2008).

Given RCA's criticality in manufacturing and its complexity, several techniques have been traditionally used to assist the analyst in this process, as mentioned in Section 2.2. Most of these techniques come from the field of quality management and are based on qualitative or semi-quantitative techniques (Chemweno et al., 2016). These techniques were not developed to handle the large amount of data currently being produced in manufacturing, and therefore are not able to guarantee the efficiency of the process. The lack of efficiency of the process leads to an increased cost in human resources, as the analysts need to spend more time looking for root causes. Nonetheless, contrary to human resources, computation resources are becoming less expensive (Rokach and Hutter, 2012). The increased availability in data together with increased availability of computation resources led to researchers and practitioners developing solutions using data mining and machine learning to make to RCA process more efficient, saving the analyst's time by automatizing (at least part of) the process.

Automatic Root Cause Analysis (ARCA) can be defined as the process through which one can find the true cause of a problem through the analysis of data, and at least part of this analysis does not require human intervention.

Although the term ARCA has already been used in papers on communication network analysis (Gomez-Andrades et al., 2016[a]; Gomez-Andrades et al., 2016[b]) and car fleet management (Richter, Aymelek, and Mattfeld, 2017), it has never been used in studies pertaining manufacturing. There has not been a common terminology among the studies on this topic. The only connecting terms among them are "root cause analysis/identification" (Chen, Tseng, and Wang, 2005) or "fault diagnosis", paired with "data mining" (Sim, Choi, and Kim, 2014), or "big data" (Chien, Liu, and Chuang, 2017). Despite the term not being used yet in the context of manufacturing, the characteristics of several manufacturing studies conform with this ARCA's definition, and we believe that this term is the most appropriate for covering this area of study.

A possible cause for this lack of common terminology is that each study is mainly focused on a particular problem of a particular industry, and mostly mentions previous works at that particular industry, and not previous attempts of automatizing RCA in manufacturing in general. As a result, the terms used in each industry are different from each other, despite sharing similar methods and objectives (e.g., "Root Cause Analysis/Identification", more common in the semiconductor industry, and "Fault Diagnosis", used in the chemical industry).

The objective of this chapter is to overview the existing literature on studies in manufacturing that can be considered related to ARCA, encompassing all the terms used in the different applications and industries. Our intention is to develop a common conceptual basis to facilitate the development of new solutions for ARCA in manufacturing.

In this chapter, 31 papers about ARCA in manufacturing published from 2005 to 2021 are reviewed. The search was performed using the query: *("root cause analysis" OR "root cause identification" OR "fault diagnosis") AND ("data mining" OR "machine learning" OR "big data") AND "manufacturing"*. The results of the query were then filtered to select the papers considered relevant to this review.

These papers come from different manufacturing industries. Table 4.1 details the distribution of these papers by industry. The most representative is the semiconductor industry, followed by the chemical industry. The prominence of these industries is in line with the amount of investment in process control in this type of industries, due to the difficulty to control physical processes on a microscopic scale (Relihan, Geraghty, and O'Dwyer, 2007; Asawachatroj and Banjerdpongchai, 2012). This leads to more availability of data, and more pressure to develop efficient solutions to process control and diagnosis.

TABLE 4.1: Table detailing the number of papers per manufacturing industry.

Industry	Number of Papers
Semiconductor	11
Chemical	5
Automotive	2
Electronic	2
Food	2
Machining	2
Power Plants	2
Other	5
Total	*31*

The conceptual framework developed in this chapter to group the existing literature has the following structure:

- **Types of data**
  Location-Time
  Physical
  Log-Action
- **Methodologies**
  Factors ⇒ Problem
  Factors ⇒ RC
  RC ⇒ RC
- **Evaluation Measures**
  Classification Measures
  Adapted Classification Measures
  Ranking Measures
  Expert Validation

When considering an ARCA solution in manufacturing, it can be characterized in three dimensions: (i) the types of data used, (ii) the methodologies applied or developed, and (iii) the evaluation measures employed.

The first dimension of analysis relates to the types of data one can obtain, which influence how we can characterize a root cause. A root cause can be described with increasing levels of complexity, where the first level presents a rough definition of the root cause, and, as we go into more complex levels, the characterization of the root cause increases in completeness.

The second dimension, concerns the methodologies used and/or developed, and focuses on questions like "what were the techniques used?" or "what were the targets/labels used?".

The third and final dimension pertains the evaluation of the solutions, and aims to understand what are the measures used in the literature to define a good/bad ARCA solution.

The remainder of this chapter is divided in three sections corresponding to the three dimensions defined above. In each section it is expounded the contributions of each reviewed study for each of the different parts. Also, a conceptualization of each part is presented, as to establish a basis for comparison. The main gaps in the literature, and the research opportunities that can be seized are then identified. In Section 4.6 some recommendations for practitioners are made on which papers to focus given the characteristics of a problem. To conclude, a summary of the findings is presented, integrating the three parts of the conceptualization, and establishing a conceptual basis for ARCA in manufacturing.

4.2 Types of Data and Root Cause Levels

In this section, the types of data that are used in the literature are reviewed, and it is also discussed how each type of data influences the characterization of the root causes that it is possible to detect. From the literature review, the types of data are divided and identified into three types: (i) Location-Time, (ii) Physical, and (iii) Log-Action. The types of data and their hierarchy are illustrated in Fig. 4.1. At the top we have the type of data with which we are able to define the first, i.e. the simplest level of the root cause (where is the root cause?), followed by the type of data that enables us to define what was the root cause physically (what was the root cause?). At the bottom, we have the type of data that allows us to define what actions lead to the physical change that resulted in a problem (why did the root cause happened?).

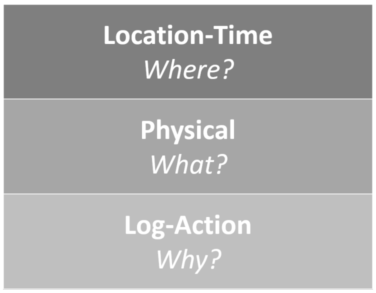


FIGURE 4.1: The types of data/root cause levels that can occur when developing ARCA solutions for manufacturing.

The first type of data mentioned refers to Location-Time data. The option for this new denomination instead of others used in the literature (e.g., machineset, as in

Chen, Tseng, and Wang (2005)) is because it is more general (locations can be, for example, machines, operators, workstations) and it reflects more accurately the characteristics of the root cause and the data. This type of data allows us to define the simplest level of root causes: the location of the root causes. It consists in the product logistics within the factory, where and when was each product transformed or assembled in a certain step of its manufacturing process. With this type of data one cannot precisely define what was the root cause. However, one can define its location, already making the process of finding the root cause more efficient.

In a manufacturing process, the products go through several manufacturing steps, and in each step they are processed in a given machine (as in Figure 3.1). At the end of the manufacturing process, they are monitored for defects, and that information is registered. Whenever a certain step is performed in a product in a certain machine, a time-stamp is registered.

An example of the root causes that is possible to obtain with this type of data is "The root cause occurred in machine 6 doing step 3 at 18h00, 27/04/2020". This gives the location of the root cause (the machine), while also establishing at which moment within the manufacturing process process it occurred (by telling us the step), and at which specific time (through the timestamp). It is possible to consider two "times", one in relation to the manufacturing process, given by the step, and another in relation to time in general, given by the time stamp. The root cause may also take the form of intervals, such as "The root cause occurred in machine 1 doing step 1, between 08h25 and 13:30 of 05/07/2020" (a timestamp interval), or "The root cause occurred in machine 6 doing step 3 and 4, between 12h25 and 13:30 of 06/07/2020" (both a timestamp interval and several steps that can be performed by the same machine).

Six reviewed papers use this type of data (plus Fan, Lin, and Tsai (2016), which uses two types of data).

The Physical type of data consists in knowing the physical factors that can affect a product's quality in each step (e.g., temperature, electrical current, operator). With it, one can establish another level of completeness of root causes: what the root cause is (what physically caused the problem). An example is "The root cause is an increase in temperature above 300ºC", or "The root cause is a decrease in the flow of chemical compound below $0.1m^3/s$". This type of data is used by 23 of the reviewed reviewed papers.

With the remaining type of data, Log-Action, we can go one step further, and not only define what was the physical root cause, but also why it happened. This type of data consists in a log of actions that were executed on elements of the manufacturing process, e.g., maintenance data, and can lead to root causes such as: "The root cause occurred in machine A doing step 3 at 18h00 27/04/2020, because the temperature has risen above 300ºC, as there was no maintenance in that machine for 3 weeks".

Anther example could be “The product presented a dark surface because the flow of chemical compound was insufficient due to the flow valves not being replaced at the time recommended by the equipment manufacturer”. Two reviewed papers use this type of data.

Table 4.2 summarises which type of data was used in each paper.

TABLE 4.2: Classification of the types of data used in each paper, divided by industry.

	Location-Time	**Physical**	**Log-Action**
Semiconductor	Chen, Tseng, and Wang (2005), Hsu and Chien (2007), Rokach and Hutter (2012), Fan, Lin, and Tsai (2016)	Chien and Chuang (2014), Zanon, Susto, and McLoone (2014), Ong, Choo, and Muda (2015), Chien, Liu, and Chuang (2017), Barkia et al. (2013), Hessinger, Chan, and Schafman (2014), Chien, Hsu, and Chen (2013), (Fan, Lin, and Tsai, 2016)	-
Chemical	-	Wang et al. (2017), Rato and Reis (2015), Gins et al. (2015), Chiang et al. (2015), Sun et al. (2021)	-
Automotive	Donauer, Peças, and Azevedo (2015), Ahn et al. (2019)	-	-
Electronic	Sim, Choi, and Kim (2014)	-	Sun, Liu, and Ming (2018)
Food	-	Kitcharoen et al. (2013), Djelloul, Sari, et al. (2018)	-
Machining	-	Du, Lv, and Xi (2012), Saez et al. (2019)	- -
Power Plant	-	Agrawal, Panigrahi, and Subbarao (2016)	Chemweno et al. (2016)
Other	-	Li, Khoo, and Tor (2006), Lee et al. (2013), He et al. (2017), Sabet, Moniri, and Mohebbi (2017), Liu et al. (2018)	-

Starting with the Location-Time data, in the oldest paper studied, i.e., Chen, Tseng, and Wang (2005), the authors define “root cause machineset identification problem” where, products or materials go through a manufacturing process consisting in several steps. In this thesis, the process that generates Location-Time data is defined in detail in Section 3.2.

In some studies (e.g., Chen, Tseng, and Wang (2005), Donauer, Peças, and Azevedo (2015)), the problem is simplified by previously defining a problematic moment to be analysed, i.e., to select a moment in time where there is a problem, and then only consider data from that period of time. This allows for the analysis of the time and location to be done separately, therefore simplifying the analysis.

In addition to Chen, Tseng, and Wang (2005), five other papers had similar data and objective. In the semiconductor industry (like Chen, Tseng, and Wang (2005)), Hsu and Chien (2007) and Rokach and Hutter (2012) also use this type of data to detect the machines or combination of machines that are the root cause(s) of a problem. Sim, Choi, and Kim (2014) uses data on “which machines the lot takes” to find the most likely root causes in a printed circuit board manufacturing process. The two papers centered on the automotive industry that were reviewed also used this type of data (Donauer, Peças, and Azevedo, 2015; Ahn et al., 2019). Ahn et al. (2019) has a particularity, as it presents a method to handle missing data caused by shuffling of products within workstations (groups of machines). Most of the factors used in this type of data are nominal - which machine the product went through in a step. The time, albeit numerical, is usually dropped after defining the problematic moment that needs to be analysed.

The type of data that is most widely used in the reviewed papers (23 of the 31) is data pertaining the physical processes that are executed in the products, and which may influence its quality. While the Location-Time data is mostly nominal, this type of data is mostly numerical (e.g., vibration, temperatures, flows) with a few factors that can be nominal (e.g., operator, recipes, raw-material used). Overall the data is more complex. Some numerical data, like vibrations, requires pre-processing or data fusion (e.g., Wavelet and Fourier transforms) to create new features, to enable the use of data mining or machine learning techniques to analyse the data (Lin, Lucas Jr, and Shmueli, 2013; Saez et al., 2019).

When considering this type of data, a root cause corresponds to the factors that indicate a physical alteration in the product that lead to a problem. An example of a root cause in a setting where the focus is the physical characteristics of the process could be "The problem occurred because the temperature was above 50ºC and the flow was below 5 mm^3/s".

A total of seven papers focused on the semiconductor industry adopt this type of data. Chien and Chuang (2014) refers the use of both nominal and numerical factors in their solutions. Zanon, Susto, and McLoone (2014) works with data from the production of silicon wafers to predict the rate of specific defect called the plasma etch. The data presents high collinearity among factors, and there is an high imbalance between defects and normal products. The study of Ong, Choo, and Muda (2015) works with nominal data about the conditions of processing (type of product, operator). Chien, Liu, and Chuang (2017) also works with nominal data (e.g., tools, recipes, production models) to diagnose the cause of extreme outliers in circuit probing (CP) yield. Barkia et al. (2013) associates quality control parameters to physical factors of each production segment. Hessinger, Chan, and Schafman (2014) relates die defects to yield loss. Chien, Hsu, and Chen (2013) uses 21 process factors collected during a chemical vapor deposition process.

All six papers related to the chemical industry use data that describe the physical properties of the process. All papers use established case-studies and models in the literature such as the Tennessee Eastman Process (Wang et al., 2017; Rato and Reis, 2015; Chiang et al., 2015; Sun et al., 2021), the Macando Well case study (Wang et al., 2017), and the Pensim model (Gins et al., 2015). The physical processes are well defined, where the Normal Operating Conditions (NOC) for each factor are used. These factors consist in values such as flows, temperatures, and pH.

From the remaining papers of other industries, it can be identified a significant diversity of sources of data about the physical processes. Lin, Lucas Jr, and Shmueli (2013) and Saez et al. (2019) use acoustic/vibration data to diagnose faults in additive manufacturing and machining, respectively. He et al. (2017) also uses vibration data, structuring it using product design information, and uses this structured data to diagnose infant mortality in washing machines. Du, Lv, and Xi (2012) uses distance measurement data to identify root causes in a machining operation. Kitcharoen et al. (2013) and Djelloul, Sari, et al. (2018) use numerical data to diagnose problems in food processing. Li, Khoo, and Tor (2006) uses physical factors that are collected continuously for condition-based maintenance in order to diagnose faults.

One particular case in what concerns this type of data is Fan, Lin, and Tsai (2016). This paper, albeit using physical data in the form of 38 numerical factors, uses a two-phase approach to first identify the physical factors that are the root cause of a problem, and then performs a second phase to identify the location-time of the root cause, supported by information of the manufacturing process's structure. In other words, it can be considered a hybrid approach, since it uses data from the second type, but it also considers the structure of the process to detect the location and time of the root cause, in addition to physically characterizing the root cause. This is the reason why in Table 4.2 this paper appears two times: once in the Location-Time column, and another in the Physical column.

In addition to the two types of data mentioned above, we can identify a third type which focuses on maintenance data, or data about actions that could otherwise affect the physical factors. For example, lack of maintenance could lead to deterioration of certain factors, which would lead to a problem.

When considering this type of data, the root causes refer to what actions (or lack of thereof) lead to the occurrence of the problem, instead of just the location or the physical change.

Two of the papers reviewed considered this type of data: Chemweno et al. (2016) and Sun, Liu, and Ming (2018). Although Li, Khoo, and Tor (2006) considers maintenance data, it focuses on condition-based data, which consists of physical factors, and as such it is better framed in the previous type of data. Chemweno et al. (2016) consists in factors such as a description of the specific failure mode, occurrence date of the described failure, and repair actions undertaken to restore the equipment back to the operational state. In Sun, Liu, and Ming (2018), the importance of each machine in the production line is computed from the machine failure frequency, machine failure severity, machine failure detection possibility and the degree of difficulty in machine failure maintenance.

4.3 Methodologies

In this section it is discussed what were the different methodologies used to automatize RCA. To review the methodologies used, the idea of how root causes can be extracted from the data was explored. This question can be answered from two perspectives: 1) which techniques are used to analyse the data, and 2) how do we extract the root cause from the results of the techniques.

To answer the question in terms of the first perspective is simply a matter of categorizing the techniques used according to the existing data mining/ machine learning literature (e.g., classification, clustering). To answer the second perspective, an overview of the literature allows us to divide the solutions in three categories:

- **Factors ⇒ Problem**: techniques are used to associate the different factors with a problem/fault. After this association has been established, we need to analyse the knowledge structure resulting from the techniques' analysis (e.g. rules, regression's coefficients, factor importance) and extract from it the root causes of the problem. The basic idea is that if a technique finds a strong association between some factors and a problem, those factors are likely the root causes.
- **Factors ⇒ RC**: techniques associate factors directly to a root cause. The objective is to identify conditions that correspond to root causes that were identified in the past. In this category, the problem is reduced to a classification, where the root causes are the labels.
- **RC ⇒ RC**: techniques identify relations among root causes that occurred in the past. The idea is to find a sequence of causality, to define which root causes should be prioritized and solved first, and to determine the influence a root cause has in the appearance of other root causes.

The **Factors ⇒ Problem** category only requires a dataset with the problem/fault labels, which are easier to obtain than the root causes. This makes the methodologies in this category easier to develop and apply. This approach can be used to discover new root causes of problems in which the symptoms are abnormal variations in quality KPIs. It can be supported by several techniques, such as classification algorithms or factor ranking. However, it still needs a list of the expected root causes for a more robust validation.

In the **Factors ⇒ RC** category, the task becomes simpler to analyse, as it is only required a classifier, and it can potentially detect root causes faster, since it only needs to consider a specific set of root causes. However, such characteristic makes this type of approach less flexible, requiring updates whenever a new root cause needs to be detected. It also requires a dataset with root causes from the beginning of the development process, which is harder to obtain than a dataset labeled with faults, since root causes take more time to be discovered than faults, and usually the storage of discovered root causes does not follow a standardized procedure. The analysts then need to work with the operators to establish the root causes.

The approaches in the **RC** ⇒ **RC** category focus more on improving the analysis by finding associations or causal paths between the root causes. Given this emphasis on improving the analysis, the approaches in this category can be used in tandem with the other two categories. One can use root causes that were identified either by using the approaches of the **Factors** ⇒ **Problem** category, or identified by the operators and analysts in the factories.

Table 4.3 shows the number of studies within each category described above using each technique.

TABLE 4.3: Summary of the number of papers using each technique, per solution category.

	Factors ⇒ Problem	**Factors ⇒ RC**	**RC ⇒ RC**
Decision Trees	5	0	0
Regression Models	4	2	0
Association/Classification Rules	6	1	2
Neural Networks	0	2	0
Ensemble	2	1	0
SVM	1	2	0
Statistical Test/Index	3	0	0
Bayesian Networks	2	0	1
Causal Maps	2	0	1
Clustering (Factors)	4	1	0
Clustering (Instances)	2	0	0
PCA	4	1	2
Control Charts	4	2	1

4.3.1 Factors ⇒ Problem

The most used techniques in the **Factors** ⇒ **Problem** category are association and classification rules. As, in this category, the root causes emerge from the classifier's structure (e.g., rules, equations), it makes sense that the classifiers with the most easily interpretable structures are used. Rules associate an antecedent to a consequent. The antecedent consists in factor-value pairs (e.g., Factor_1 = 0 and Factor_2 = 1). A consequent can also be a factor-value pair, in the case of Association Rules (AR), or a label (e.g., Problematic/Normal), in the case of classification rules.

When applying AR to RCA, we are interested in finding associations between the factors that indicate a defect or a fault. Chen, Tseng, and Wang (2005) uses AR mining to identify a machine or combinations of machines that are the most likely root causes of a given problem. It also proposes a new interestingness measure to be used with association rule mining, that combines confidence with a measure of continuity between the defective products for a combination of machines. Sim, Choi, and Kim (2014) also uses AR mining to identify the location of root causes, and considers the cumulative effect of upstream machines when evaluating rules. Ong, Choo, and Muda (2015) uses Weighted Association Rule Mining (WARM) for root cause analysis, in order to tackle the issue of imbalanced datasets. As there are much more normal functioning products than defective ones, this can become an issue in ARCA. Sabet, Moniri, and Mohebbi (2017) also tackles the issue of imbalance, using Fuzzy Weighted Association Rules (FWAR). Barkia et al. (2013) uses association rules to find associations between instances clustered by factors and instances clustered by quality results.

Decision Trees (DT) are easily interpretable structures when kept short (Molnar, 2019). As such, it makes sense that they are one of the most used techniques in the **Factors ⇒ Problem** category. Hsu and Chien (2007) uses DT to identify the location of root causes of different defect patterns. Fan, Lin, and Tsai (2016) also does so, but trains the DT to detect the timing of the root causes, and then determines the location of the product at that timing. Chien, Hsu, and Chen (2013) and Kitcharoen et al. (2013) use DT to identify the physical factors that are the root causes of faults. Barkia et al. (2013) uses DT to identify the most discriminant factors in clusters of factors that were deemed the most likely root causes.

Interpreting the estimated coefficients in regression models is another way of identifying likely root causes. The higher the standardized coefficient is, the most likely the associated factor is a root cause of the problem the regression model is trying to predict. Zanon, Susto, and McLoone (2014) defends the use of sparse models, such as lasso logistic regression and relevance vector machine precisely because of the ease of interpretation. Chien, Liu, and Chuang (2017) also uses logistic regression to identify the factors that are more likely to be root causes. Chien and Chuang (2014) proposes a sub-batch regression model, which is estimated with maximum likelihood estimation and the Akaike information criterion. Sim, Choi, and Kim (2014) uses regression in a different way, applying PLS-VIP to select the most important machines to use in the posterior association rule mining.

There are other types of classifiers used in the **Factors ⇒ Problem**. Fan, Lin, and Tsai (2016) applies Support Vector Machine (SVM) and Ada-boost to identify factors with irregular values. Chien and Chuang (2014) selects the most relevant factors using the Random Forest algorithm.

However, not only classifiers or AR are used in the **Factors ⇒ Problem** category. Some studies adapted statistical index/scores in order to identify potential root causes of problems. Donauer, Peças, and Azevedo (2015) proposes the Herfindahl-Hirschman Index (HHI) to identify patterns of concentration of faults in the equipment of the manufacturing process. It then displays the values of the index in a visualization matrix in order to make it easier and faster for practitioners to identify the root causes. It is the only work reviewed that presented a visualization solution. Rato and Reis (2015) uses partial correlations, as they can identify causal connectivity and degree of change among factors, as partial correlation coefficients can eliminate the effects of the controlled factors upon other factors. It also proposes the use of uses sensitivity enhancing transformations on factors to improve the detection and diagnose of faults. More specific to semiconductor industry, Hessinger, Chan, and Schafman (2014) defines a statistical measure called Confidence Factor, which determines which defect patterns in a wafer occurred at random or not, and if these defect patterns have an impact on yield loss.

The concern with causality is revealed in the use of graph-based methods that map the relations among factors. In Agrawal, Panigrahi, and Subbarao (2016), a Bayesian Network (BN) is constructed for each type of fault, in order to diagnose root causes in a coal mill in a thermal power plant. However, the BN's structure and parameters are determined using expert knowledge and not automatically or algorithmically. In Sim, Choi, and Kim (2014), a BN is adopted to identify relations among faults, which can help identify sequences of root causes determined by the association rule mining. Hill-climbing with random restart is used to determine the BN network structure, by optimizing Bayesian Information Criteria (BIC). Another form of graph-based methods are causal maps, which can be used to map relations between factors, and between factors and labels. Chiang et al. (2015) makes use of this technique to identify the root cause and the propagation path of the problems it originated. In Sun et al. (2021), information geometric causal inference is used to investigate the potential root-cause variables and its propagation path, which aid in determining the root cause factor.

In addition to techniques used for associating factors to faults, other support techniques have been used during pre-processing, and are relevant to the discussion. Among them is the use of clustering. Clustering can be used to group two things: factors or instances. When clustering factors, the objective is to reduce the number of factors considered by identifying similar factors. Rokach and Hutter (2012) applies the online clustering algorithm Squeezer (He, Xu, and Deng, 2002) to find combinations of machines that can be root causes together, but not each machine individually. Zanon, Susto, and McLoone (2014) clusters highly correlated factors and chooses a representative of said group, in order to avoid hiding important factors caused by high-correlation among factors. Sun, Liu, and Ming (2018) uses k-means clustering to group alarms by type and time of occurrence. Chien, Liu, and Chuang (2017) groups factor with high correlation using Cramer's V.

When clustering instances, the objective is to find groups of similar products, and then compare the characteristics of the groups that have a higher prevalence of problems with the groups that have behavior characteristic of normal functioning. Barkia et al. (2013) uses k-means to cluster products in two different ways: according to similarities in the factors considered relevant, and quality performance values. The authors then search for associations between these clusters. Chien, Hsu, and Chen (2013) uses Self-Organizing Maps (SOM) to cluster wafers with similar characteristics in semiconductor manufacturing.

Another way of reducing the dimensionality is the use of Principal Component Analysis (PCA). By transforming the factors into several compound principal components, and select only the ones that explain the most variability, one can simplify the analysis and improve the results. Fan, Lin, and Tsai (2016) uses PCA for this purpose, before using SVM and Ada-boost to associate factors to problems. Ong, Choo, and Muda (2015) uses PCA to select the appropriate weights in WARM. Sun et al. (2021) uses a moving window kernel PCA to monitor the nonlinear and dynamic characteristics of complex industrial processes.

To initiate RCA, it is necessary to determine a period to be analysed. For this purpose, a few solutions use control charts to automatically detect the moments that should be analysed by a data mining technique, by identifying when a manufacturing process diverts from its normal operating conditions. It is commonly used in chemical industry, and is in fact used by all the papers from that industry (Rato and Reis (2015), Chiang et al. (2015), and Sun et al. (2021) being the ones in the **Factors** $\Rightarrow$ **Problem** category). Control charts were also used by Kitcharoen et al. (2013) with the same objective.

4.3.2 Factors $\Rightarrow$ RC

Considering now the papers that belong to the **Factors** $\Rightarrow$ **RC** category, in their work the root causes are not extracted from the resulting structures, but are the direct objective of the techniques. As a consequence, the focus lies less on interpretability and more on classification accuracy.

Association rules are also used. Lee et al. (2013) uses AR to identify root causes among defects in garment industry.

In what concerns classifiers with a regression structure, Ahn et al. (2019) employs logistic regression as a classifier between the problematic machines and the root causes. Djelloul, Sari, et al. (2018) employs regression technique to measure the strength of the proposed Neural Network (NN) models.

Neural Networks is a technique that is only used in this category. This is due to its high accuracy, but very low interpretability, resulting from being a "black-box" model. Djelloul, Sari, et al. (2018) uses gradient descent and momentum, with adaptive learning rate and Levenberg-Marquardt NNs to associate production factors to root causes in a milk pasteurization process. Du, Lv, and Xi (2012) uses an ensemble of NN (optimizing the structure using particle swarm optimization and simulated annealing) to associate distance measurements (the factors) to root causes in an engine machining operation.

SVM have similar characteristics to NN in terms of advantages and disadvantages. Gins et al. (2015) uses SVM as a classifier after pre-processing the data, and SVM is also used in Saez et al. (2019) as a classifier, after the process has been partitioned in several GOS - Global Operational States. These states are defined based on physical data and expert knowledge, and help improve fault detection and diagnosis.

In what concerns pre-processing techniques, Liu et al. (2018) clusters factors using fast search and find of density peaks, an unsupervised density-based clustering technique, in order to identify different machine states of an extruder. Gins et al. (2015) employs standard PCA-based fault detection control chart, as it is the standard benchmark. Du, Lv, and Xi (2012) also uses control charts to detect problematic periods that need to be analysed.

4.3.3 RC ⇒ RC

As mentioned above, the papers in the **RC ⇒ RC** category focus on finding related root causes, possibly finding a causal sequence that leads to a better decision on what root causes to prioritise. Chemweno et al. (2016) uses predictive Apriori to find association rules between component failures. It then uses expert knowledge to develop causal maps that relate the association rules, establishing a casual sequence of root causes. He et al. (2017) also uses an association rule mining algorithm to establish relations between root causes. PCA is used to screen factors, and the WARM algorithm is used to associate root causes of infant mortality in washing machne design and manufacturing. Wang et al. (2017) combines semiparametric PCA and BN in a two-phase approach: first, semiparametric PCA is used to find the fault, and second the BN applies deductive and abductive reasoning to assist in RCA and identify fault propagation pathways. One note to Sim, Choi, and Kim (2014), which uses BN to relate faults, and then uses this to sequence association rules. Although there is indeed some focus on establishing connections, this was done with faults/problems instead of root causes, and in order to improve the perception of which faults/problems were more important. For these reasons, the option was to keep it in the **Factors ⇒ Problem** category.

4.3.4 Summary

To summarize the review of this section, Table 4.4 divides the reviewed papers according the ARCA solution category and industry, while Table 4.5 presents an overview of the algorithms used in each paper.

TABLE 4.4: Classification how each paper extracts root causes from the techniques, divided by category and industry.

	Factors ⇒ Problem	Factors ⇒ RC	RC ⇒ RC
Semiconductor	Chen, Tseng, and Wang (2005), Hsu and Chien (2007), Rokach and Hutter (2012), Chien and Chuang (2014), Zanon, Susto, and McLoone (2014), Ong, Choo, and Muda (2015), Fan, Lin, and Tsai (2016), Chien, Liu, and Chuang (2017), Barkia et al. (2013), Chien, Hsu, and Chen (2013), Hessinger, Chan, and Schafman (2014)	-	-
Chemical	Rato and Reis (2015), Chiang et al. (2015), Sun et al. (2021)	Gins et al. (2015)	Wang et al. (2017)
Automotive	Donauer, Peças, and Azevedo (2015)	Ahn et al. (2019)	-
Electronic	Sim, Choi, and Kim (2014), Sun, Liu, and Ming (2018)	-	-
Food	Kitcharoen et al. (2013)	Djelloul, Sari, et al. (2018)	-
Machining	-	Du, Lv, and Xi (2012), Saez et al. (2019)	-
Power Plants	Agrawal, Panigrahi, and Subbarao (2016)	-	Chemweno et al. (2016)
Other	Li, Khoo, and Tor (2006), Sabet, Moniri, and Mohebbi (2017)	Lee et al. (2013), Liu et al. (2018)	He et al. (2017)

In what concerns Table 4.4, what is more evident is that, although in other industries the papers spread out over the different categories of solutions, in the semiconductor industry papers focus only on the **Factors** ⇒ **Problem** category. This may indicate that it is difficult to get datasets with the root causes in this industry, as this is the only category that does not require such datasets. Machining industry also focuses solely on **Factors** ⇒ **RC**. The data used as factors are usually based on vibrations, which consist in numerical factors, and have very high dimensionality and presence of noise. To deal with high dimensionality and noise, the solutions adopt techniques like Neural Networks and SVM. However, these techniques develop structures that are difficult to interpret. This may be an explanation for the focus in this category, as **Factors** ⇒ **Problem** requires the interpretation of the resulting classifiers structures.

Table 4.6 relates the categories presented in this section with the types of data presented in the previous one. Solutions in the **Factors** ⇒ **Problem** category use all types of data. However, almost all studies using Location-Time data present solutions in this category. As the root cause in this type of data consists in tuples of location-time, the same as the factors, it explains the focus in this category of solutions, where the root cause is within the factors. In the **RC** ⇒ **RC** category, the focus is on Physical and Log-Action data. As in this category, the objective is to relate root causes among themselves, it makes sense that the data types used have to define the root cause as completely as possible, therefore the use of Physical and Log-Action data. However, given the low number of papers in this category, some caution when establishing hypothesis is advised. The solutions of the **Factors** ⇒ **RC** focus mostly on Physical data. As the solutions in this category rely directly on the classifier's performance (instead of needing to interpret the knowledge structures), they can use algorithms with more predictive capabilities, but that are not as interpretable (e.g., Neural Networks and SVM). These algorithms can use numerical data more efficiently, and this is a reason for the increased use of Physical data.

TABLE 4.5: Summary of what techniques are used by each paper.

Industry	Paper	Decision Trees	Regression Models	Association Rules	Neural Networks	Ensemble	SVM	Statistical Test/Index	Bayesian Networks	Causal Maps	Clustering (Factors)	Clustering (Instances)	PCA	Control Charts
Semiconductor	Chen, Tseng, and Wang (2005)	-	-	x	-	-	-	-	-	-	-	-	-	-
	Hsu and Chien (2007)	x	-	-	-	-	-	-	-	-	-	-	-	-
	Rokach and Hutter (2012)	-	-	-	-	-	-	-	-	-	x	-	-	-
	Barkia et al. (2013)	x	-	x	-	-	-	-	-	-	-	x	-	-
	Chien, Hsu, and Chen (2013)	x	-	-	-	-	-	-	-	-	-	x	-	-
	Chien and Chuang (2014)	-	x	-	-	x	-	-	-	-	-	-	-	-
	Hessinger, Chan, and Schafman (2014)	-	-	-	-	-	-	x	-	-	-	-	-	-
	Zanon, Susto, and McLoone (2014)	-	x	-	-	-	-	-	x	-	-	-	-	-
	Ong, Choo, and Muda (2015)	-	-	x	-	-	-	-	-	-	-	-	x	-
	Fan, Lin, and Tsai (2016)	x	-	-	-	x	x	-	-	-	-	-	x	-
	Chien, Liu, and Chuang (2017)	-	x	-	-	-	-	-	-	-	x	-	-	-
Chemical	Chiang et al. (2015)	-	-	-	-	-	-	-	-	x	-	-	-	x
	Gins et al. (2015)	-	-	-	-	-	x	-	-	-	-	-	x	x
	Rato and Reis (2015)	-	-	-	-	-	-	x	-	-	-	-	-	x
	Wang et al. (2017)	-	-	-	-	-	-	-	x	-	-	-	x	x
	Sun et al. (2021)	-	-	-	-	-	-	-	-	x	-	-	x	x
Automotive	Donauer, Peças, and Azevedo (2015)	-	-	-	-	-	-	x	-	-	-	-	-	-
	Ahn et al. (2019)	-	x	-	-	-	-	-	-	-	-	-	-	-
Electronic	Sim, Choi, and Kim (2014)	-	x	x	-	-	-	-	x	-	-	-	-	-
	Sun, Liu, and Ming (2018)	-	-	-	-	-	-	-	-	-	x	-	-	-
Food	Kitcharoen et al. (2013)	x	-	-	-	-	-	-	-	-	-	-	-	x
	Djelloul, Sari, et al. (2018)	-	x	-	x	-	-	-	-	-	-	-	-	-
Machining	Du, Lv, and Xi (2012)	-	-	-	x	x	-	-	-	-	-	-	-	x
	Saez et al. (2019)	-	-	-	-	-	x	-	-	-	-	-	-	-
Power Plants	Agrawal, Panigrahi, and Subbarao (2016)	-	-	-	-	-	-	-	x	-	-	-	-	-
	Chemweno et al. (2016)	-	-	x	-	-	-	-	-	x	-	-	-	-
Other	Li, Khoo, and Tor (2006)	-	-	x	-	-	-	-	-	-	-	-	-	-
	Lee et al. (2013)	-	-	x	-	-	-	-	-	-	-	-	-	-
	He et al. (2017)	-	-	x	-	-	-	-	-	-	-	-	x	-
	Sabet, Moniri, and Mohebbi (2017)	-	-	x	-	-	-	-	-	-	-	-	-	-
	Liu et al. (2018)	-	-	-	-	-	-	-	-	-	x	-	-	-

TABLE 4.6: Classification of the papers based on the types of data used and category of solutions.

	Factors ⇒ Problem	**Factors ⇒ RC**	**RC ⇒ RC**
Location-Time	Sim, Choi, and Kim (2014), Rokach and Hutter (2012), Chen, Tseng, and Wang (2005), Donauer, Peças, and Azevedo (2015), Ong, Choo, and Muda (2015), Hsu and Chien (2007), Fan, Lin, and Tsai (2016)	Ahn et al. (2019)	-
Physical	Kitcharoen et al. (2013), Chien and Chuang (2014), Rato and Reis (2015), Chiang et al. (2015), Zanon, Susto, and McLoone (2014), Agrawal, Panigrahi, and Subbarao (2016), Chien, Hsu, and Chen (2013), Chien, Liu, and Chuang (2017), Fan, Lin, and Tsai (2016), Barkia et al. (2013), Hessinger, Chan, and Schafman (2014), Li, Khoo, and Tor (2006), Sabet, Moniri, and Mohebbi (2017), Sun et al. (2021)	Lee et al. (2013), Du, Lv, and Xi (2012), Gins et al. (2015), Djelloul, Sari, et al. (2018), Saez et al. (2019), Liu et al. (2018)	Wang et al. (2017), He et al. (2017)
Log-Action	Sun, Liu, and Ming (2018)	-	Chemweno et al. (2016)

It is also important to highlight some topics usually addressed by the studies analysed. Zanon, Susto, and McLoone (2014) mentions that high-correlation among factors may lead to hiding important factors. Furthermore, this study mentions that one of the challenging aspects of ARCA in the **Factors** ⇒ **RC** category is discovering root causes among factors that are highly correlated. There are a number of reviewed papers that use factor clustering, PCA or other forms of factor processing (such as Sim, Choi, and Kim (2014), refer to Table 4.5 for the full list) to determine the relevant factors based on correlations.

Another topic refers to the periods used to perform RCA. In many papers, the selection of these periods is not mentioned, as it is assumed that the RCA process is done on a previously established dataset, where the period is already defined. However, it is relevant to have a criteria for selecting the periods of analysis in order to automatize the RCA process. Evidence of this lies in several papers that use control charts for problematic period selection (see last column of Table 4.5). There was one paper which did not use control charts, but tackled this issue. Rokach and Hutter (2012) proposed an online methodology for RCA, which automatically detects and determines the locations of faults.

Another topic that is present in the reviewed papers is the introduction of information about the structure of the manufacturing process in the data. Rato and Reis (2015), Chiang et al. (2015), Saez et al. (2019) mention improved results when incorporating information about the manufacturing process structure.

4.4 Evaluation Measures

In this section the papers are reviewed from the perspective of how they evaluate the ARCA solutions. This is a critical perspective, as the evaluation determines the validity of the solutions proposed, and the impact that these solution can have when applied. Based on the reviewed papers, it is possible to distinguish four types of measures used. Either they use traditional classification measures (e.g., accuracy, F1 score), an adaptation of these measures to a RCA context (i.e., instead of evaluating the prediction capabilities of the algorithms, they evaluate the root cause detection), ranking measures, and validation by experts or analysis of case study examples.

The most used type of evaluation present in the reviewed papers is validation through an expert or a case study analysis. In this type of evaluation, the input of an expert is used to validate a subset or a few examples of the case study used. Usually a dataset of a defined period is analysed by the proposed solution, and the results of such analysis are presented to an expert, who then proceeds to validate them. This type of evaluation requires considerable human input, and may be affected by subjectivity. Both these characteristics are undesirable when constructing an ARCA solution. However, this type of evaluation does not require the existence of a dataset with root cause labels, which makes it more accessible, explaining its large adoption.

Also, this type of evaluation may be more adequate for evaluating causal sequences, given their complexity. Therefore, this type of evaluation is more appropriate for solutions of the **RC** ⇒ **RC** category. For a comprehensive list of the reviewed papers that use this type of evaluation, see the rightmost column of Table 4.7.

TABLE 4.7: Classification of the papers by type of evaluation measures.

	Classification Measures	**Adapted Classification Measures**	**Ranking Measures**	**Expert Validation**
Semiconductor	-	Rokach and Hutter (2012), Chien and Chuang (2014), Fan, Lin, and Tsai (2016)	Chen, Tseng, and Wang (2005), Zanon, Susto, and McLoone (2014)	Hsu and Chien (2007), Chien and Chuang (2014), Chien, Liu, and Chuang (2017), Chien, Hsu, and Chen (2013), Ong, Choo, and Muda (2015), Fan, Lin, and Tsai (2016), Barkia et al. (2013), Hessinger, Chan, and Schafman (2014)
Chemical	Gins et al. (2015)	Rato and Reis (2015), Chiang et al. (2015)	-	Wang et al. (2017), Sun et al. (2021)
Automotive	Ahn et al. (2019)	-	-	Donauer, Peças, and Azevedo (2015)
Electronic	Sim, Choi, and Kim (2014)	-	-	Sun, Liu, and Ming (2018)
Food	Kitcharoen et al. (2013), Djelloul, Sari, et al. (2018)	-	-	-
Machining	Saez et al. (2019)	Du, Lv, and Xi (2012)	-	-
Power Plants	-	-	Agrawal, Panigrahi, and Subbarao (2016)	Chemweno et al. (2016)
Other	Liu et al. (2018)	-	-	Li, Khoo, and Tor (2006), Lee et al. (2013), Sabet, Moniri, and Mohebbi (2017), He et al. (2017)

Another performance evaluation strategy is to use classification measures (e.g., accuracy, F1 score) for evaluating the proposed solutions. As many solutions apply classifiers as their core, the adoption of this type of evaluation is understandable. However, their adoption should take into consideration the type of solution to be evaluated. If the solution belongs to the **Factors** ⇒ **RC** category, as the root cause is extracted from the prediction itself, it makes sense to evaluate the solution according to its predictive capabilities. However, when evaluating solutions from the **Factors** ⇒ **Problem** category, this may not be the case, as a root cause is not extracted from the prediction itself, but from the factors used to make the prediction.

Focusing on the papers belonging to the **Factors** ⇒ **RC** that use classification measures for evaluation, accuracy was the most adopted measure. Djelloul, Sari, et al. (2018), Saez et al. (2019), and Liu et al. (2018) use accuracy as the measure of choice. Gins et al. (2015) adopts the average accuracy between six fault classes. The other measure used is the F1 score, which is adopted in Liu et al. (2018) and Ahn et al. (2019).

There are two papers in the **Factors** ⇒ **Problem** category that use classification measures to evaluate their solutions: Kitcharoen et al. (2013) and Sim, Choi, and Kim (2014). Kitcharoen et al. (2013) uses accuracy to evaluate the predictive capabilities of the rules extracted from a DT, and does not evaluate if the root cause of the problem was detected or not. In Sim, Choi, and Kim (2014), the predictive accuracy of the rules is also used, but is coupled with measures on the length of the rules (how simple or easy to interpret is the rule), and a custom measure named cumulative effect, which refers to how much a rule pertaining the beginning of the manufacturing process influence other rules, which focus on later steps of that process.

When developing a solution in the **Factors** ⇒ **Problem** category, the main focus is not in predictive capabilities of the solution, but if the likely root causes that are extracted from the structures correspond to the real root causes, that should have been detected. In order to do this, some papers adapted classification measures to focus on the root cause detection instead of the predictive capabilities of the structures.

Rokach and Hutter (2012) uses adapted versions of accuracy, G-mean and precision, but instead of considering the correct predictions, they consider the correct root cause detections. As the authors consider the issue of detecting combinations of machines, they develop a distance measure to see how far from the true combination of machines is a proposed combination (e.g., if the root cause is combination "A-B-C", and the proposed combination is "A-B", it still is two-thirds correct). Chien and Chuang (2014) uses an adapted version of accuracy. Rato and Reis (2015) uses the "percentage of times that each factor was considered as the faults' root cause", which is an adapted version of accuracy. Chiang et al. (2015) uses an adaption of the misclassification (or Type I) error, which is the complement of accuracy (misclassification error $= 1 -$ accuracy).

Du, Lv, and Xi (2012), albeit belonging to the **Factors** ⇒ **RC** category, uses Correct identification Percentage (CIP) of root causes, making it explicit that the focus is in root cause detection, even when the root cause is directly predicted by the classifier.

The final type of evaluation measures are the ranking measures. When a solution proposes a list of factors ordered by importance or likelihood of being a root cause, it is possible to evaluate the solution by either counting if the true root cause belongs to the first N elements of the list, or the position of the first correct detection of a root cause. This type of measures is adopted in the solutions belonging to the **Factors** ⇒ **Problem** category, as it is in this category that the root causes are factors, and the likelihood of these factors being a root cause can be evaluated. Chen, Tseng, and Wang (2005) evaluates the percentage of true root causes in the top 10 proposed factors. Zanon, Susto, and McLoone (2014) finds commonalities between the root causes detected by the proposed methods by ranking clusters of factors and comparing these rankings. Agrawal, Panigrahi, and Subbarao (2016) considers a correct detection if the actual root cause is in the top two causes diagnosed by the algorithm.

Table 4.8 presents a summary of the evaluation type used by each category of solutions. From Table 4.8, we can see that, for each solution category, the evaluation measures fit the solutions proposed. This is, **Factors** ⇒ **RC** solutions use mostly classification measures, **RC** ⇒ **RC** adopt mostly expert validation. In the **Factors** ⇒ **Problem** category, the evaluation methods are more varied, with an emphasis in expert validation. This emphasis leads to lack of standardized measures, which makes objective comparison between these papers difficult. However, most of the other studies in this category identify the need for objective measures focusing on the root cause detection, either through the use of rankings, or through the adaptation of classification measures.

TABLE 4.8: Classification of the papers by evaluation type used and category of solutions.

	Factors ⇒ Problem	**Factors ⇒ RC**	**RC ⇒ RC**
Classification Measures	Sim, Choi, and Kim (2014), Kitcharoen et al. (2013)	Gins et al. (2015), Ahn et al. (2019), Djelloul, Sari, et al. (2018), Saez et al. (2019), Liu et al. (2018)	-
Adapted Classification Measures	Rokach and Hutter (2012), Chien and Chuang (2014), Rato and Reis (2015), Chiang et al. (2015)	Du, Lv, and Xi (2012)	-
Ranking Measures	Chen, Tseng, and Wang (2005), Zanon, Susto, and McLoone (2014), Agrawal, Panigrahi, and Subbarao (2016)	-	-
Expert Validation	Chien, Hsu, and Chen (2013), Chien, Liu, and Chuang (2017), Hsu and Chien (2007), Donauer, Peças, and Azevedo (2015), Ong, Choo, and Muda (2015), Fan, Lin, and Tsai (2016), Barkia et al. (2013), Hessinger, Chan, and Schafman (2014), Sun, Liu, and Ming (2018), Li, Khoo, and Tor (2006), Sabet, Moniri, and Mohebbi (2017), Sun et al. (2021)	Lee et al. (2013)	Wang et al. (2017), Chemweno et al. (2016), He et al. (2017)

Table 4.9 relates the type of data presented in Section 4.2, and the evaluation types of this section. It is not possible to identify any trend for the Location-Time and Physical types of data. However, all studies using Log-Action data used Expert Validation. As Log-Action data consists in a log of action performed in the manufacturing equipment, it makes sense for solutions using this type of data to be evaluated using expert knowledge, as experts are in a better position to evaluate the effects of those actions. However, given the low number of papers using this type of data, some caution when establishing hypothesis is advised.

4.5 Gaps & Research Opportunities

Taking into account what was discussed in the sections above, it is possible to find some research gaps and opportunities.

First, not much importance is given to the selection of the problematic moments and the corresponding datasets analysed by the solutions. With the exception of the works using control charts and Rokach and Hutter (2012), which proposes an online RCA methodology, no reference to the detection of problematic periods is mentioned. As this can impact the performance of the solutions developed, it is

TABLE 4.9: Classification of the papers by type of data and the evaluation type used.

	Location-Time	Physical	Log-Action
Classification Measures	Sim, Choi, and Kim (2014), Ahn et al. (2019)	Kitcharoen et al. (2013), Gins et al. (2015), Djelloul, Sari, et al. (2018), Saez et al. (2019), Liu et al. (2018)	-
Adapted Classification Measures	Rokach and Hutter (2012)	Chien and Chuang (2014), Rato and Reis (2015), Chiang et al. (2015), Du, Lv, and Xi (2012)	-
Ranking Measures	Chen, Tseng, and Wang (2005)	Zanon, Susto, and McLoone (2014), Agrawal, Panigrahi, and Subbarao (2016)	-
Expert Validation	Hsu and Chien (2007), Donauer, Peças, and Azevedo (2015), Ong, Choo, and Muda (2015), Fan, Lin, and Tsai (2016)	Lee et al. (2013), Wang et al. (2017), He et al. (2017), Chien, Hsu, and Chen (2013), Chien, Liu, and Chuang (2017), Fan, Lin, and Tsai (2016), Barkia et al. (2013), Hessinger, Chan, and Schafman (2014), Li, Khoo, and Tor (2006), Sabet, Moniri, and Mohebbi (2017), Sun et al. (2021)	Chemweno et al. (2016), Sun, Liu, and Ming (2018)

advisable to take this into consideration when developing an automatic solution for RCA.

The dispersion of evaluation measures being adopted indicates that there is no standard established. In this work this is tackled by categorizing the previously used evaluation measures, and suggesting the use of each of them according to the solution's methodological category. It was possible to verify that some papers did not take this into account (see Table 4.8), and this may compromise the validity of the solution and its comparison with other solutions. In fact, none of the papers compare their solutions with other solutions listed here, instead relying on the analysis of particular examples, sensitivity analysis, or comparing with standard (i.e., not specific to RCA) data mining algorithms (e.g., He et al. (2017) compares the proposed methodology with association rules).

Still, concerning the evaluation, it is easily noticeable that most studies use expert validation (see Tables 4.7 and 4.8). This is most likely because, although it is common to have datasets with information on faulty or problematic products, it is less common to have datasets with information regarding the root causes themselves. As RCA is a time consuming process using traditional methods (Chemweno et al., 2016), it makes the collection of data about root causes itself a time consuming process. This represents a challenge for ARCA. One possible solution could be the use of semi-supervised techniques, which focus on partially labeled datasets, and augment the labeled information through iteration.

Another aspect that is mentioned by a few papers is that the inclusion of information about the manufacturing process in the solutions developed has a positive effect on the performance (Rato and Reis, 2015; Chiang et al., 2015; Saez et al., 2019). New ways of including such information could be developed.

In spite of RCA being about finding the cause of problem, few papers mention causation (Chemweno et al., 2016; Sim, Choi, and Kim, 2014; Wang et al., 2017; Rato and Reis, 2015; Chiang et al., 2015; Sun et al., 2021). To establish causation, associations alone are not sufficient, and every causal conclusion must have some causal assumption (Pearl et al., 2009). To establish such assumptions about a manufacturing process, we require information about the structure of the process (i.e., how the data is generated). Inclusion of techniques for finding causal relations, together with the use of structural information about the manufacturing process should constitute the next step for ARCA. This is already done in most papers focused on the chemical industry, most likely due to the well defined physical processes and relations among factors, which facilitates the making of assumptions, and therefore causal analysis.

4.6 Managerial Insights

In order to help practitioners focus their study in the papers most relevant to them, this section provides recommendations on which papers to focus, depending on the data available and the problem they are trying to solve.

The proposed conceptualization divides the literature according to three dimensions:

- Types of data
- Methodologies
- Evaluation Measures

The types of data can be Location-Time data, Physical data, or Log-Action data, and determine what kind of root cause it is possible to discover. The methodologies can be divided according to how the root cause is extracted from the data mining techniques used, and can be divided in **Factors** ⇒ **Problem**, **Factors** ⇒ **RC**, and **RC** ⇒ **RC**. The evaluation measures can be classification measures, adaptations of these to RCA, ranking measures, or expert validation.

If the available data is Location-Time data, the focus will most likely be in detecting the root cause from factors, i.e., in developing a solution in the **Factors** ⇒ **Problem** category (as can be seen in Table 4.6). In this category, we can consider Rokach and Hutter (2012) as the state-of-the-art, as it presents a complete solution, as it considers how to select the time of analysis, and presents validation using adapted classification measures. Another relevant paper is Sim, Choi, and Kim (2014), as it is the only paper with this type of data mentioning the issue of causation, by connecting the variables using Bayesian Networks. Although association is not sufficient to establish causation, it is positive to include the need to identify connections between the problems, and introduce the notion of causality, as it can improve the results.

Donauer, Peças, and Azevedo (2015) presents an unique solution by using visualization to aid in the search for root causes, and may be an appropriate solution for a quick implementation.

In what concerns the use of Physical data (numerical variables, such as temperatures or electrical currents), all three categories of solutions may be adopted. In case there is a well defined process, with the causal relation between the variables well identified, it is desirable to include this information in the analysis. Rato and Reis (2015) presents a solution considering this information in the **Factors** ⇒ **Problem** category, while Saez et al. (2019) does so in the **Factors** ⇒ **RC** category. In the **RC** ⇒ **RC** category, Wang et al. (2017) also considers the issue of causality. For this inclusion of the data generation process and focus on causal structure, these can be considered state-of-the-art in their respective categories with Physical data. Sun et al. (2021) is the most recent paper and, although it does not include information about the manufacturing process, it tries to infer a causal structure between variables directly from the data. Despite not mentioning causality, Fan, Lin, and Tsai (2016) should be mentioned, as it is the only paper that tackles two types of data (Location-Time and Physical) at the same time, and is a relevant study in case both types of data are available.

If the data available pertains the actions taken on the manufacturing equipment (i.e., Log-Action), there are only two studies that may support the definition of the solution, i.e. Chemweno et al. (2016) and Sun, Liu, and Ming (2018). The first one focuses on interconnecting these actions to find the root causes, and the second one uses data fusion and fuzzy clustering techniques to identify the most important machines given the human-machine interactions.

4.7 Conclusions

This chapter presents an overview of the literature on developing solutions for Automatic Root Cause Analysis (ARCA) in manufacturing. It proposes a conceptualization to link the literature, both in terms of standards and nomenclature. ARCA solutions can be considered from three perspectives: the types of data used, the methodologies and techniques used, and how the evaluation of these solutions is performed.

Regarding the types of data, it is possible to identify three different types, which correspond to three different layers of root causes (see Section 4.2 and Figure 4.1 for more details). The Location-Time type of data, which describes when and where (which machines) a product has gone through, allows us to determine where in the manufacturing process is the root cause located. Physical type of data pertains all the factors that describe physical influences on the products (e.g., temperature, flow), and with it we are able to determine what the root cause is, physically. The Log-Action type of data describes the actions that were performed on the structure supporting the manufacturing process (machines and other equipment), and allows us to determine why the physical parameters deteriorated in the first place.

In what concerns the methodologies, the approaches are divided into three categories, according to how the root causes are extracted from the algorithms (see Section 4.3 for more details). In the **Factors** ⇒ **Problem** category, a technique is used to associate the different factors with a problem/fault, and the root cause is extracted from the factors that the classifier considers most relevant to determine whether there is a problem or not. In the **Factors** ⇒ **RC** category, a technique associates factors directly to a root cause. Finally, the solutions in the **RC** ⇒ **RC** category focus on finding relations between root causes, in order to determine the root causes that should be prioritized. Also, a review of the data mining and machine learning techniques used to automatize the RCA process was completed.

For evaluating the ARCA solutions, a diverse group of measures are used in the literature (see Section 4.4 for details). These measures were divided into classification measures (as they are used to evaluate classifiers' performance in data mining literature), adaptation of classification measures to evaluate the performance in root cause detection, ranking methods, and validation through expert analysis. In addition to this categorization, it is noted the importance of selecting the right type of evaluation according the the solutions' methodology category.

Taking everything into consideration, it is possible to say that the literature on ARCA in manufacturing is very diverse, both in terms of data and methods, as in terms of application setting. The broad spectrum of application of ARCA solutions underlines the importance of the development of this kind of solutions for manufacturing. Despite this relevance, the literature does not have a standardized terminology or conceptualization that could serve as a link between these studies. This makes comparisons harder, and could lead to overlaps or oversights that impede the development of ARCA in manufacturing.

This chapter contributes to the literature by, not only reviewing the disperse ARCA in manufacturing literature, but also by proposing a conceptualization from which future studies on the topic can base themselves upon. In addition, gaps and research opportunities are identified, and can be explored to develop improved ARCA solutions. Improved ARCA solutions can make the management of manufacturing operations more efficient, as a quicker diagnose of problems frees managers' time, that can be used to actually finding a solution and putting improvement actions in practice.

Chapter 5

On the Influence of Overlap in Automatic Root Cause Analysis in Manufacturing

5.1 Introduction

In this chapter, overlap is further presented and discussed. As explained in Section 3.2, overlap occurs when trying to find the location of a root cause, using ARCA solutions based on Location-Time data. When all products that go through a certain machine in a manufacturing step are all processed in a given machine in a later manufacturing step, the data generated makes it impossible to determine the influence on quality of these machines that processed all these products. This is especially true when analysed through classifier algorithms, as their focus on predictive analytics clashes with the objective of diagnosis, as explained in Section 2.3. Therefore, overlap can disrupt the use of ARCA solutions, and these must be adapted to be resilient to such phenomenon.

To circumvent this issue, this chapter proposes a way to quantify overlap that can be used to measure its impact. This measure is based on the strength of association between two factors.

Based on the proposed measure, a novel ARCA solution is also proposed. The proposed solution can be divided in two stages:

1. **Problematic Moment Identification**, which aims at reducing the complexity of the analysis by selecting the time window to be analysed, in order to isolate problems.

2. **Factor Ranking Algorithm**, in which are identified all the factors that are potential root causes, even those that are overlapped. This is done using factor ranking algorithms, which avoid the hiding of information that occurs when using classifiers, therefore circumventing the impossibility in distinguishing factors that are overlapped.

While most papers in the literature focus on using classifiers and posterior analysis of their knowledge structure (e.g., decision trees, rules) for the identification of root cases, we argue that such solutions have a risk of having their performance degraded by overlap. Rokach and Hutter (2012) is an exception and does not use classifiers. However, its proposed technique based on product queues can still be affected by overlap. Donauer, Peças, and Azevedo (2015) also proposes a non-classifier approach, based on visualization and concentration measures, but does not identify overlap, nor its pernicious effects on the use of classifiers for ARCA solutions. Lee and Chien (2020) is a recent paper with a broad scope in identifying pitfalls of applying data mining techniques to manufacturing practice. It mentions some pitfalls in the identification of important factors, but does not mention the particular problem of overlap. Detzner and Eigner (2021) also studies feature selection in the context of RCA, but also does not identify the issue of overlap. In this context, this chapter differs from the literature by identifying and conceptualizing an issue we denominated overlap, which disrupts the performance of approaches based on classification algorithms, when analysing Location-Time data to identify root causes.

The contributions of this chapter are the definition, identification, and first measures of overlap, as well as the proposal of a two-stage solution to tackle it. The validation of the proposed methodology is done using both simulated data, and data from a case study in semiconductor manufacturing.

5.2 Overlap Operationalisation

Taking consideration the problem as defined in Section 3.2, one can formally define overlap from the strength of association between two nominal factors. The factors are nominal, since each factor represents a manufacturing step, in which the levels of those factors are the machine used to process those steps. A statistical test for measuring the association between two nominal factors is the Chi-square independence test (see Section 2.3.1), and the strength of association can be measured using Crámer's V (Crámer, 1999), which can be defined as:

$$V = \sqrt{\frac{\chi^2/n}{min(k-1, r-1)}}, \tag{5.1}$$

where χ^2 is the chi-squared statistic (as defined in Section 2.3.1), n the number of products (for example, rows in Table 3.1), k the number of levels in one of the factors (for example, the columns in Table 3.1), and r the number of levels in the other factor. Crámer's V varies between 0 and 1, where 0 means no association, and 1 a complete association between factors. This measure of strength of association has been used before in Chien, Liu, and Chuang (2017) to find variables with collinear effects.

Our initial measure of overlap of a certain dataset is the average of the strength of association between all factors, as defined in Expression 5.2:

$$Overlap_{Avg} = \frac{\sum_i^I \sum_j^J V_{i,j}}{\frac{N^2-N}{2}}, \tag{5.2}$$

where N is the number of factors in the dataset, and $i, j \in$ [Factor Set], $\forall\, i \neq j$. The denominator represents the number of non-repeated interactions between the factors. If all factors are overlapped, i.e., all $V_{i,j} = 1$, then $\sum_i^I \sum_j^J V_{i,j} = \frac{N^2-N}{2}$.

However, this measure is not easily interpretable, as it is not possible to know if the overlap comes from many small associations between factors (which could be noise), or from factors that have a very strong association. Such characteristic is not desirable when comparing datasets of different origins, with different levels of noise. This was particularly evident during the exploratory phase, as $Overlap_{Avg}$ led to values with high levels of variance for datasets that are generated with the same number of overlapped columns.

To reduce the variance of the measure, we propose that, instead of computing the average, one determines the percentage of interactions between factors that have a strength of association above a certain threshold (by default 0.99, as it represents a complete overlap between factors, with some slack for the presence of noise). $Overlap_{Count}$ is defined by Expression 5.3:

$$Overlap_{Count} = \frac{\sum_i^I \sum_j^J C_{i,j}}{\frac{N^2-N}{2}}, \tag{5.3}$$

where:

$$C_{i,j} = \begin{cases} 1, & \text{if } V_{i,j} \geq Th \\ 0, & \text{if } V_{i,j} < Th, \end{cases}$$

and Th is the threshold that defines an interaction that represents overlap (by default 0.99), and $i, j \in$ [Factor Set], $\forall\, i \neq j$.

For example, if we take into consideration the dataset in Table 3.1, the three steps “A”, “B” and “N” have a total of three interactions among them (A-B, A-N, and B-N), and the sum of all Crámer’s V values is 1.707, which gives an average of Overlap$_{Avg}$ = 0.569. For the same dataset, only one of the interactions between factors was overlapped above the threshold (the interaction between “A” and “B”), which gives an Overlap$_{Count}$ of 0.333, or a third of the interactions.

Both measures are presented in the results for comparison, and this is discussed in more detail in Section 5.7. The overlap definition and its measures proposed here are a contribution to the literature, as it is the first time the phenomenon is explicitly recognized, and we propose an easy way to compute and an interpretable measure, based on previously defined concepts (i.e., strength of association between two nominal factors).

5.3 Proposed Solution & Problematic Moment Identification Stage

In this section, we describe the proposed two-stage ARCA solution, and present in more detail the first stage. The second stage will be detailed in Section 5.4.

In order to perform RCA automatically from the data, it is essential that the dataset selected for analysis is chosen in a way that it correctly represents a problem for which we need to find the cause. By isolating problems according to the time they occur, it is possible to analyse the data from a stabilized and homogeneous period, which reduces the complexity of the analysis. Problems change with time, and with them change the causes, which leads to the necessity of framing the different root cause analysis within the time periods when such analysis are valid. As such, it is necessary to do so, in a stage we denominated problematic moment identification.

Figure 5.1 presents a flowchart of the ARCA solution proposed. As data is continuously generated by the production process, it is gathered, pre-processed, and then monitored and analysed by an algorithm that selects the period of time relevant for the analysis, and isolates the correspondent data. This is called the problematic moment identification stage. The isolated data is then analysed by a factor ranking algorithm, which then provides a list of most likely root causes' locations, in the form of step-machine tuples.

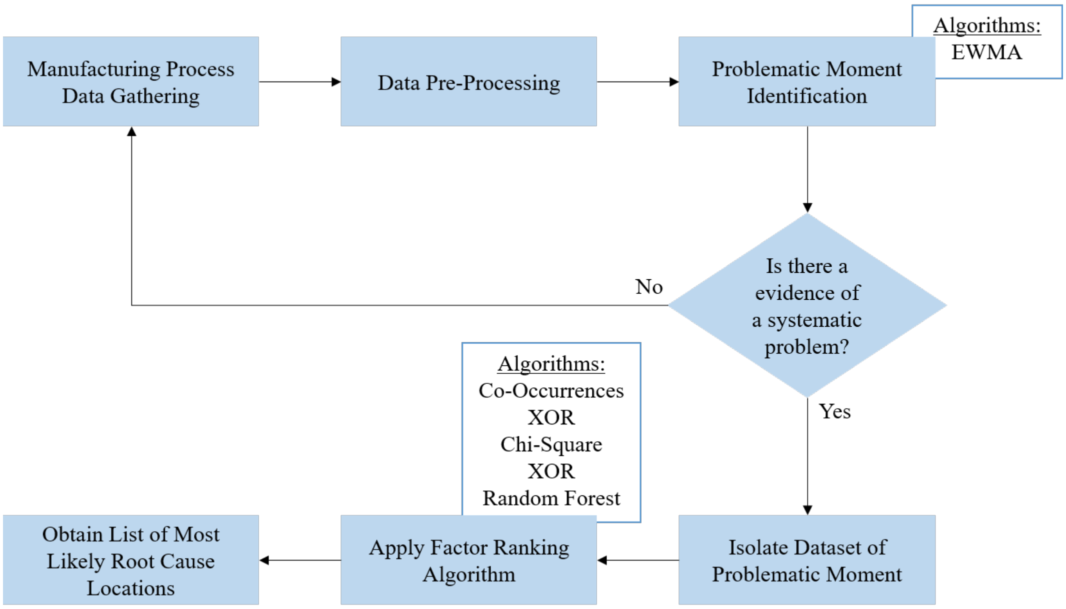

Manufacturing Process Data Gathering
Data Pre-Processing
Problematic Moment Identification
Algorithms:
EWMA
Is there a evidence of a systematic problem?
No
Yes
Algorithms:
Co-Occurrences
XOR
Chi-Square
XOR
Random Forest
Obtain List of Most Likely Root Cause Locations
Apply Factor Ranking Algorithm
Isolate Dataset of Problematic Moment

FIGURE 5.1: Flowchart detailing the proposed ARCA solution.

As mentioned in Section 2.4, the algorithm selected for the problematic moment identification stage was the Exponentially Weighed Moving Averages (EWMA) (Roberts, 1959), a control chart capable of identifying sustained small/medium shifts in a manufacturing process. EWMA focuses on controlling a weighted average of deviations from the expected values, where the most recent lots of products have more influence. This compensates the effects of disturbances and changes in manufacturing processes (Shi and Tsung, 2003). In the proposed solution, EWMA is used to control the proportion of problematic products in a lot. When a sustained shift in this proportion is detected, it means that there is a systematic problem in the manufacturing process that needs to be addressed, and as such the RCA process is triggered. As explained in more detail in Section 2.4, EWMA was selected as the most appropriate algorithm for this stage, after comparing the behavior of three control charts (EWMA, Cumulative Sum - CUSUM - control charts, and Shewhart charts) in data from the case study detailed in Section 3.3.

To estimate the variance to be used in EWMA, the proportion of problematic products per lot is modelled as a beta distribution, estimating the α and β parameters and then computing the variance. The beta distribution is used because it is a flexible distribution that fits into the proportion of problematic products of the real case study. For further details on the EWMA control chart, and the specification of these parameters, please refer to Montgomery (2019).

5.4 Factor Ranking Algorithms

In this section, the focus is on the second stage of the proposed ARCA solution. In order to automatically analyse the data after the problematic moment has been identified, it is necessary to apply an algorithm to extract the most likely root causes. In many ARCA solutions available in the literature, a classifier is used to classify products with problems, and then the knowledge structures (e.g., rules, regression models) are analysed and the factors most used by the structures are selected as most likely root causes. However, as discussed in Section 5.2, classifiers may hide relevant factors from the analysts in the presence of overlap, as they may eliminate from their decision process factors that are highly correlated, and therefore deemed to contain redundant information.

To develop solutions that are able to perform robust to the presence of overlap, it is necessary to consider algorithms that do not hide factors, even if they are highly correlated. As it is not possible to distinguish the root cause among correlated factors (with only Location-Time data), we need to use algorithms that reveal all relevant factors, and do not hide highly correlated ones, as the previous solutions did. This is the case with factor ranking algorithms.

These algorithms enable the detection of the most relevant factors to distinguish between normal and problematic products without hiding information, as the most relevant factors would appear, even if they are correlated.

We considered three different factor ranking algorithms to analyse the data and extract root causes: i) Co-Occurrences (CO), ii) Chi-Square (CS), iii) Random Forest (RF). Although RF is originally a classification algorithm, here we use it to evaluate the importance of factors. These algorithms provide a list of most likely root causes (in the form of step-machine tuples) for a problem, ordering them by importance. Highly correlated tuples appear in close positions in the ranking, with the same importance given to them. This way the analyst can do an analysis having all information required. These algorithms are compared with a Decision Tree (DT) algorithm in the experiments (Sections 5.5 and 5.6). DT is chosen as a benchmark for comparison, as it is used in several papers as a component of ARCA solutions (e.g., Fan, Lin, and Tsai (2016) and Hsu and Chien (2007)), due to it being highly interpretable and easily implemented.

5.4.1 Co-Occurrences

This algorithm is the simplest of the three. The basic idea is to compare the percentage of products that went through a certain step-machine tuple that had problems versus the percentage of products that did not go through a certain step-machine tuple. This algorithm was chosen as it is an easily applicable algorithm with fast implementation to a practical situation, and easily understood by experts.

The Co-Ocurrences (CO) algorithm is detailed in Algorithm 1.

Algorithm 1: Co-Ocurrences algorithm

```
Input: Dataset of problematic moment
Output: Ordered list of most likely root causes
begin
    Output ← [∅];
    To-use-dataset ← OneHotEncoding(Input);
    for Each Step-Machine Tuple do
        Construct Co-Occurrences table;
        Compute importance;
        if importance > 0.5 then
            Output ← [Output, (Step-Machine Tuple, importance)];
        end
    end
    Sort Output;
end
```

The input is the dataset of problematic moment identified in the previous stage. The output is a list with two columns, one which identifies the tuple, and another its importance in terms of co-occurrence with the label. It is initialized as an empty list. One-hot encoding (as explained in Section 2.3.1) is used to enable to group the information in the dataset by step-machine combination.

Then, for each combination of step-machine, a table like Table 5.1 is constructed, which is then used to compare the occurrence of problems between products that have gone through that combination of step-machine, and those who have not. In Table 5.1, each line represents the counts of products that have gone trough ("Yes") or not ("No") a specific combination, and each column whether the products had problems (NP/YP) or not (NN/YN). A table is constructed for each Step-Machine combination tuple.

TABLE 5.1: Table used to compare the occurrence of problems in a Step-Machine combination.

Step-Machine	Problem	Normal
No	NP	NN
Yes	YP	YN

The importance of each combination is computed using Expression 5.4:

$$\text{importance} = \frac{\frac{YP-YN}{YP+YN} - \frac{NP-NN}{NP+NN}}{2} \tag{5.4}$$

The greatest importance is equal to 1 (due to the division by two) and the lowest equal to 0. If the importance of the tuple is above 0.5, the tuple and its importance are stored in the Output list. After all combinations are evaluated, the list is sorted in descending order of importance.

5.4.2 Chi-Square

As overlap is defined based on the strength of association between two nominal factors, which in turn is based on Chi-Square independence test, it makes sense to develop a factor ranking algorithm based on this test. Therefore, this algorithm is the one closest to the proposed measure of overlap.

The Chi-Square (CS) algorithm is detailed in Algorithm 2.

In this case, for each step, a Chi-Square independence test is performed between that factor (a step with the different machine used as levels) and the label (if the product is problematic or not). After that, a post-hoc Fisher test is performed to compare the different levels (machines) in a step. This is done with adjusted p-values using the Bonferroni correction. These methods are explained in greater detail in Section 2.3.1. If a machine has an adjusted p-value below a pre-defined alpha (alpha $= 0.01$) in all comparisons with the other machines in the same step, it is deemed as important

```
Algorithm 2: Chi-Square algorithm
Input: Dataset of problematic moment
Output: Ordered list of most likely root causes
begin
    Output ← [∅];
    for Each Step Factor do
        Perform Chi-Square test between Step Factor and Label;
        for Each Machine in that Step do
            Perform post-hoc Fisher Test with all the other machines in the same
              Step;
            if all p-values < alpha then
                Output ← [Output, (Step-Machine Tuple, importance)];
            end
        end
    end
    Sort Output;
end
```

and added to the list of likely root causes, which is sorted in ascending order, as lower p-values indicate higher importance.

To summarize, the algorithm described above first determines the steps that are correlated to the label, and then uses post-hoc tests to determine the machines that, when products go through them, create the greatest gap between problematic and normal products.

5.4.3 Random Forest

The Random Forest (RF) algorithm (Breiman, 2001), as described in Section 2.3.2, is an ensemble of decision trees usually used as a classifier. It constructs a multitude of decision trees with randomly selected factors, and then the decision of each decision tree is aggregated using a voting system. This way, the decision tree's tendency to overfit is overcome. When used as a classifier, this algorithm is not easily interpretable, and as such is not adequate for ARCA solutions.

However, this algorithm can also be used to measure variable importance, by using a permutation of factors over all trees (see Section 2.3.2 for details), and computing the difference in error. In the context of the proposed solution, we require a factor ranking algorithm that can list the importance of a factor to determine another (namely the importance of a location to identify a product with a problematic label), without hiding factors that have similar degrees of association. In this context, the use of RF as a factor ranking algorithm is adequate, as it is a powerful algorithm able to identify non-linear associations between factors. RF is used, for example, in Chien and Chuang (2014) to select a number of factors that can cover major information of the response variable. Although it can be used to measure variable importance, it requires some cautions. For data including nominal factors with different number of levels, random forests are biased in favor of factors with more levels. To counteract this, we use unbiased random forests (Strobl et al., 2007), and also compute the variable importance regarding the area under the curve (AUC) (Janitza, Strobl, and Boulesteix, 2013), as it is more resilient to label imbalance. As it is common to exist much more normal functioning products than problematic ones, it is preferred to use measures that are more resilient to this imbalance in the label, e.g., AUC.

Taking the above cautions into account, it is possible to use RF to measure variable importance, and present a list of likely root causes, where highly correlated step-machine tuples are presented, without hiding information.

Algorithm 3 details how the RF is used in this context.

Algorithm 3: Random Forest algorithm for factor ranking in ARCA

```
Input: Dataset of problematic moment
Output: Ordered list of most likely root causes
begin
    Output ← [∅];
    To-use-dataset ← OneHotEncoding(Input);
    Trains random forest model on To-use-dataset;
    Extracts variable importance based on AUC from trained model;
    Output ← Extracted variable importance;
    Sort Output;
end
```

After the dataset of the problematic moment is identified, the dataset is one-hot encoded to obtain a model focused on the step-machine tuples. The RF model is trained on the encoded dataset. The variable importance is extracted from the trained model, and these are added to the output, which is then sorted.

5.5 Experimental Setup

To analyse the effect of overlap, illustrate its pernicious effects on data analysis, and to validate the two-stage solution proposed, three experiments were conducted: i) using mockup data (i.e., data from artificially generated and simplified scenarios), ii) using data from a stochastic simulator (as described in Section 3.4), iii) using data from a real case study. In all experiments, EWMA is used for the solution's first stage, while for the second stage the three factor ranking algorithms proposed in Section 5.4 are used. In order to compare the factor ranking algorithms with classifiers, a C4.5 Decision Tree (DT) (Quinlan, 1993) is also used. DT is a classifier that is easily interpretable and is used in previous ARCA works (e.g., Fan, Lin, and Tsai (2016) uses a C4.5 DT, and Chien, Hsu, and Chen (2013) uses a CHAID DT). It serves as a comparison to the factor ranking algorithms. The DT's parameters are tuned to each dataset, using grid-search and 5-fold cross validation. Together, the three factor ranking algorithms and the DT are called RCA algorithms in the remainder of the chapter.

5.5.1 Mockup Data

Mockup data is data that is generated to represent simple scenarios where a certain percentage of factors are overlapped. This type of data is used to assess if the presence of overlap has an impact on the capacity of classifiers to detect root causes. These scenarios assume that the problematic moment already has been identified, so the focus is to validate the hypothesis that overlap leads to a decrease in the performance of classifiers.

A mockup data generator was developed that allows to control the following parameters: the number of steps and products in a dataset, the percentage of overlapped factors, and whether the root cause is one of the overlapped factors. The last parameter was introduced to verify if overlap only decreases performance when it affects the root cause factor, or if it also does so just by being present in the dataset.

The mockup generator creates a dataset by creating first the overlapped factors, ensuring that the levels of each factor are synchronized. If the root cause is in an overlapped factor, it is defined at this moment. Then the factors that are not overlapped are created randomly. If the root cause is not in an overlapped factor, it is defined at this moment. The factors are then shuffled so that the order does not interfere with the analysis. After that, the value of the labels is assigned to each product, depending if the product has passed through the root cause or not. If the product has passed through the root cause, it is considered problematic.

Table 5.2 represents the parameters' values used for the experiments based on mockup data. For each combination of parameters (10 combinations - five possible overlap percentages times two possibilities of RC being overlapped), a total of 25 datasets were generated per combination, adding to a total of 250 datasets, and they were analysed by the four RCA algorithms used (three factor ranking algorithms and a classifier). Each step can be performed by up to four machines (average of 70 machines per dataset).

TABLE 5.2: Parameters' values of the mockup datasets generated.

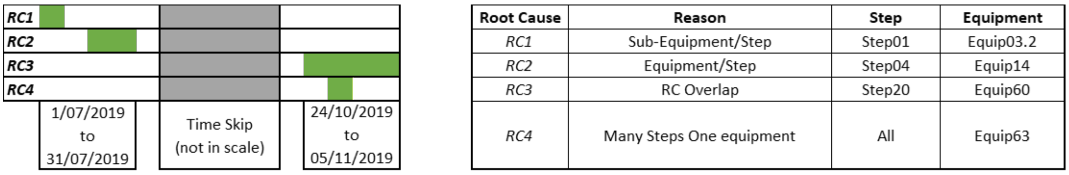
Parameter	Values
Number of Step (Factors)	20
Number of Products (Rows)	100
Percentage of Overlapped Columns	[0.0, 0.1, 0.2, 0.3, 0.4, 0.5]
RC is Overlapped?	[True, False]

5.5.2 Stochastic Simulation Data

The data described in the previous section pertain simple scenarios in which there is control on the amount of overlap and how it manifests in the data. In this section, the aim is to use data as close as possible to the real case, but where it is still possible to define the root causes. This is useful for the validation of the problematic moment identification and RCA algorithms, as it enables us to be certain at which time and machines the problems occurred. As such, the objective with these experiments is to validate if the problematic moment identification algorithms identify the problematic moments correctly, and if the RCA algorithms detect the correct root causes, and in which conditions they fail to do so. In order to achieve this, a stochastic simulator was created, that emulated the available case study in semiconductor manufacturing. This simulator is detailed in Section 3.4.

Three simulations were conducted, with parameters' values similar to the data from the case study, each with: 22 steps; 63 equipment; 5004 products (rows). Each of these simulations' datasets reports to a different scenario in terms of noise, as described in Section 3.4: dataset 1 has an average level of noise, dataset 2 a low level of noise, and dataset 3 a high level of noise. This noise was introduced while labeling the products as problematic or not, with a certain number of products being mislabeled (either they were problematic without passing through the root cause, or were normal despite having passed through a root cause). The higher the noise, the higher the number of mislabeled products. Four root causes were defined in each dataset, as it is depicted in Figure 5.2. As each dataset corresponds to a period of approximately four months, four root causes (one per month) were selected, to obtain a number of problematic products that mirrors the real-case data. It is important to mirror the imbalance in order to verify if the proposed methodology is also robust to it.

The root causes included in the simulated data are depicted in Figure 5.2. On the left side of the figure, we can see the temporal distribution of the different root causes. The first two root causes are located in the beginning of the simulated period, while the last two in the end. This concentration of root causes was done to: i) mirror the imbalance between normal products (the vast majority) and problematic products (a small minority) that exist in real data, and ii) to simulate the simultaneous occurrence of root causes, which may also occur in real datasets. The right side of the figure explains in more detail each root cause, and the reason why it was used. RC1 is a failure in part of an equipment/machine. RC2 is a standard malfunction in a single machine when performing a single function (or step). RC3 is a simultaneous failure which occurs at the same time as RC4. This last root cause is a failure that happens when a machine is malfunctioning in all its functions/steps.



Root Cause	Reason	Step	Equipment
RC1	Sub-Equipment/Step	Step01	Equip03.2
RC2	Equipment/Step	Step04	Equip14
RC3	RC Overlap	Step20	Equip60
RC4	Many Steps One equipment	All	Equip63

FIGURE 5.2: Root Causes defined in the datasets obtained through stochastic simulation.

5.5.3 Real Case Study Data

The last set of experiments use real data from a case study in semiconductor manufacturing. It was the study of this data that originated the need to study overlap. The objective when testing with this data is two-fold: i) to illustrate that overlap is indeed a real problem that occurs when analysing data from a manufacturing process, ii) to validate the proposed solution with real data.

The dataset is composed by 22 steps, 62 machines, and 6144 products (rows). The actual root causes were not identified by the company used as case study, and as such validation of the RCA algorithms is done by comparing the results of all of them, and verify if there is a convergence in the proposed lists of most likely root causes. The problem in this case study is defined as identifying cases of overkill - situations where the Automatic Optic Inspection (AOI) (as described in Section 3.3) generates too many false detections of defects due to exterior changes in the product, that do not necessarily make the product defective. This affects production by increasing the amount of products that need to be manually inspected, increasing cycle time.

5.6 Experimental Results

5.6.1 Mockup Data

In this section the results of the experiments described in Section 5.5.1 are reported. The capacity for root cause detection of the different factor ranking algorithms - Co-Ocurrences (CO), Chi-Square (CS), Random Forest (RF) - are compared with the Decision Trees (DT) classifier algorithm. This is done in two scenarios, one where the root cause is one of the overlapped factors, and another where it is not. The results are presented in Figures 5.3 and 5.4, divided by whether the root cause was one of the overlapped factors or not. In these figures, for each percentage of overlapped columns and method, it is represented the percentage of datasets in which the method is able to correctly detect the root cause.

As the main objective is to test if the presence of overlap impacts the performance of classifiers, we will first focus our analysis on the performance of the DT in both scenarios. When the root cause is one of the overlapped factors (Figure 5.3), it is possible to see that the performance of the DT decreases as the percentage of overlapped factors increases, up to 30%. After that level, the performance of the DT stabilizes at

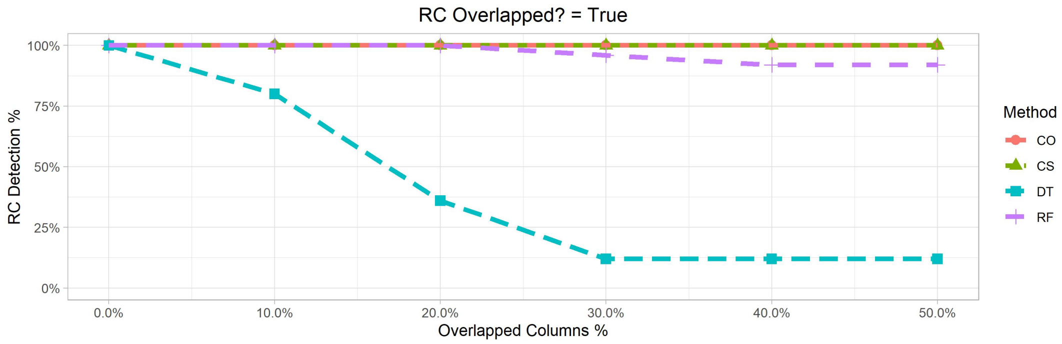


FIGURE 5.3: Results of the Mockup experiments when the root cause is one of the overlapped factors.

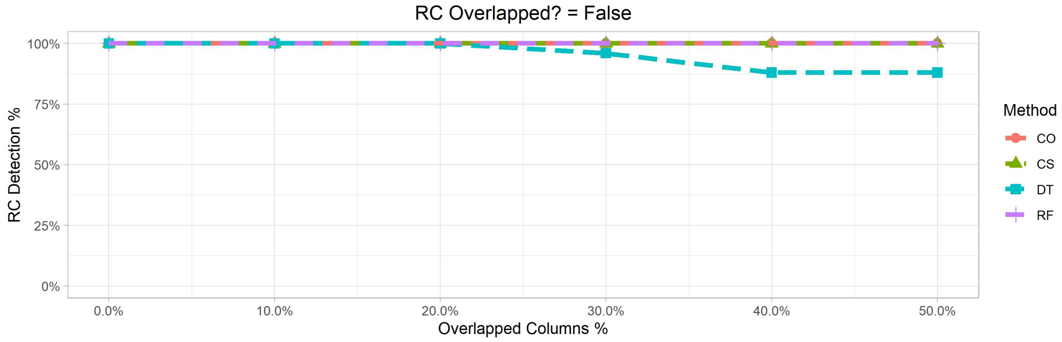


FIGURE 5.4: Results of the Mockup experiments when the root cause is not one of the overlapped factors.

12% of correct detections, which is a poor performance. This is evidence to the hypothesis that if the root cause is one of the overlapped factors, it becomes harder for classifiers to correctly detect the root cause, as explained in more detail in Section 5.2.

However, the performance of DT decreases even when the root cause is not one of the overlapped factors (although in lesser degree than when it is overlapped). This may indicate that overlap can bias classifiers, even if it does not impact the root cause factor itself.

In what concerns the factor ranking methods, it is possible to see that the CO and CS algorithms get perfect performance in these simplified experiments. This seems to suggest that considering all relevant factors instead of filtering out those highly correlated (such as in the case of factor ranking algorithms) makes the methods more resilient to overlap. In the case of RF, the algorithm performs perfectly when the root cause factor is not overlapped, but has a minor decrease in performance when it is. This may be due to RF being an algorithm based on an ensemble of smaller classifiers, and some of these smaller classifiers may be impacted by overlap in case a individual decision tree uses two or more of the overlapped factors.

5.6.2 Stochastic Simulation Data

The objectives of the experiments using data from stochastic simulation are to validate if the problematic moment identification algorithms are able to identify the problematic moments correctly, and if the RCA algorithms detect the correct root causes.

To validate the problematic moment identification algorithm, we compare the problematic moments obtained for the three simulated datasets (as described in Section 5.5.2), as can be seen in Table 5.3, with the information in Figure 5.2. It is possible to verify that all the moments identified fall within the time windows defined in Figure 5.2.

Although there were two periods defined as problematic (July and the period between 24/10 and 05/11), the problematic moment identification algorithm divided each of these periods in two, leading to four problematic moments being identified in most of the simulated datasets. On the second dataset, the moment between 24/10 and 05/11 was divided in three.

TABLE 5.3: List of the problematic moments identified for each of the three stochastic simulation dataset.

Dataset	Moment	$Overlap_{Avg}$	$Overlap_{Count}$	Date Moment Began	Date Moment Ended	Within 1/7/2019 and 31/7/2019	Within 24/10/2019 and 05/11/2019
1	1	65.8%	0.4%	15/07/2019	17/07/2019	✓	
1	2	38.0%	0.0%	17/07/2019	22/07/2019	✓	
1	3	73.0%	8.3%	24/10/2019	25/10/2019		✓
1	4	45.3%	0.0%	25/10/2019	28/10/2019		✓
2	1	70.4%	5.1%	14/07/2019	17/07/2019	✓	
2	2	41.6%	0.0%	17/07/2019	22/07/2019	✓	
2	3	36.4%	0.0%	24/10/2019	29/10/2019		✓
2	4	51.4%	0.0%	29/10/2019	01/11/2019		✓
2	5	48.7%	0.0%	02/11/2019	06/11/2019		✓
3	1	63.2%	0.4%	16/07/2019	18/07/2019	✓	
3	2	41.3%	0.0%	18/07/2019	23/07/2019	✓	
3	3	79.3%	9.1%	25/10/2019	26/10/2019		✓
3	4	52.2%	0.0%	26/10/2019	28/10/2019		✓

There are two major insights that can be obtained from Table 5.3. The first one is that the problematic moment identification algorithm used does identify problematic moments that are within the periods defined in the simulation, therefore validating the problematic moment identification algorithm. The second insight is that there are datasets with a relatively high overlap level, even after problematic moment identification. This is true for dataset 1's moment 3, dataset 2's moment 1, and dataset 3's moment 3.

Please note that, although the percentage may seem small, it corresponds to a large number of factors being overlapped. This seemingly disparity exists because Overlap$_{Count}$ counts the number of overlapped interactions between factors, and not the number of factors that present overlap. Even a seemingly small number of Overlap$_{Count}$ reveals a potential cause for disruption in the analysis, specially in case the root cause is one of the factors involved.

The other objective when analysing the data from stochastic simulation is to validate if the RCA algorithms could indicate the root causes correctly. The root causes are defined as it is depicted in Figure 5.2. Table 5.4 presents the results of the RCA algorithms used, divided by dataset (each moment in a simulation). Each time a root cause is correctly detected, it is presented. In the case of CO, CS, and RF, the ranking of the factor is also presented. In the ranking, ties among positions may happen, which are not as clear as an isolated position. As such, we signal the ties with a "*". For the classifier (DT), in addition to the root causes, the false positives (or incorrect detections) are presented as "FP", because this algorithm does not provide a ranking, and in order to have a better notion of the false detections occurred. Also, empty cells mean that the algorithm was not able to detect any root cause. The $Overlap_{Avg}$ column was suppressed to improve readability.

From the analysis of Table 5.4, it is possible to see that the Co-occurrences (CO) algorithm has a positive performance. The algorithm identifies root causes correctly (i.e., in the first place in the ranking of most likely root causes) in all moments, and only when Overlap$_{Count}$ grows above 5% (in the 3rd, 5th and 12th rows) the root cause either appears in worse positions, or it appears tied with other possible root causes. The only time the root cause is not detected in first place and the overlap is 0%, is in the 9th row, where the root cause is detected in the 6th position.

The Chi-square (CS) method presents a worse performance than the CO algorithm. Although it also detects almost all of the root causes in the first place (with the exception of the 10th, 12th and 13th row), more ties among possible root causes occur (i.e., there are a lot of factors with the same relative importance as the true root cause on the ranking list).

The Random Forest (RF) algorithm has a performance that lies between the other two, as it does not present any ties, but has slightly worse positions for the true root causes, and it is not able to detect the root cause in the dataset with the most overlap (9.1% in the second to last row).

In what concerns the classifier, i.e., the Decision Trees (DT), it has the lowest capacity for identification among all algorithms, and presents a large number of false detections. In the datasets with the highest overlap values (3rd, 5th and 12th rows), the

TABLE 5.4: Table with the results of the RCA algorithms for the Stochastic Simulation data.

Row	Dataset	Moment	Overlap$_{Count}$	CO	CS	RF	DT
1	*1*	*1*	0.4%	**RC2 (1st);** **RC1 (12th)**	RC1 (1st)*; RC2 (1st)*	RC2 (2nd); RC1 (5th)	**RC2**
2	*1*	*2*	0.0%	**RC2 (1st)**	RC2 (1st)*	**RC2 (1st)**	RC2; FP; FP
3	*1*	*3*	8.3%	**RC3 (1st)***	**RC3 (1st)***	RC3 (6th)	FP
4	*1*	*4*	0.0%	**RC3 (1st);** **RC4 (2nd)***	RC3 (1st)*; RC4 (8th)	RC3 (1st)	RC3; FP; FP
5	*2*	*1*	5.1%	RC2 (6th)*	**RC2 (1st)***	RC2 (3rd)	FP
6	*2*	*2*	0.0%	**RC2 (1st)**	**RC2 (1st)**	**RC2 (1st)**	RC2; FP
7	*2*	*3*	0.0%	**RC3 (1st)**	RC3 (1st)*; RC4 (1st)*	**RC3 (1st)**	RC3; FP; FP
8	*2*	*4*	0.0%	**RC3 (1st)**	RC3 (1st)*; RC4 (7th)	**RC3 (1st)**	**RC3**
9	*2*	*5*	0.0%	RC3 (6th)	RC3 (1st)*	**RC3 (1st)**	RC3; FP
10	*3*	*1*	0.4%	**RC2 (1st)**	RC2 (3rd)	RC2 (3rd)	RC2
11	*3*	*2*	0.0%	**RC2 (1st)**	RC2 (1st)*	**RC2 (1st)**	RC2; FP
12	*3*	*3*	9.1%	**RC3 (1st)***	RC3 (5th)*	-	FP
13	*3*	*4*	0.0%	**RC3 (1st)***	-	RC3 (2nd)	RC3, RC4; FP

DT algorithm is not able to correctly detect the root causes, while in the others it has that capacity, even with some false detections.

5.6.3 Real Case Study

As mentioned in Section 5.5.3, the experiments with data from a real case study have the double objective of demonstrating that overlap is indeed a real issue, and to further validate the use of the proposed solution. As it was not possible to obtain the root causes of the problems in the case study dataset, validation is attempted by comparing the results of each algorithm, and checking if there exists a convergence in the results, which would strongly indicate a root cause.

The obtained results, both in terms of problematic moment identification and root causes detected by each algorithm, are summarized in Table 5.5. In this table, only the steps of the step-machine tuples are shown to facilitate reading, and because of a confidentiality agreement.

TABLE 5.5: Table summarizing the problematic moments identified in the Real case study dataset, as well as the common root-causes among the different algorithms proposed.

Moment	Overlap$_{Avg}$	Overlap$_{Count}$	Number of Overlapped Factors	CO	CS	RF	DT
1	27.7%	0.9%	3	COATFUSE	-	COATFUSE	COATFUSE
2	26.2%	1.7%	6	PLATINGRDL	PLATINGRDL	PLATINGRDL	PLATINGRDL
3	34.6%	0.0%	0	EXPOSEFUSE	EXPOSEFUSE	EXPOSEFUSE	EXPOSEFUSE
4	29.8%	2.2%	6	EXPOSEFUSE	EXPOSEFUSE	EXPOSEFUSE	EXPOSEFUSE
5	35.2%	3.5%	10	CUREFUSE	CUREFUSE	CUREFUSE	CUREFUSE
6	27.0%	0.9%	3	COATRDL	COATRDL	COATRDL	COATRDL
7	15.6%	0.4%	2	EXPOSERDL	EXPOSERDL	EXPOSERDL	EXPOSERDL
8	35.3%	3.9%	12	-	AOI1DL2	AOI1DL2	AOI1DL2
9	25.7%	7.8%	13	-	-	-	-
10	29.6%	1.7%	6	-	PLATINGRDL	PLATINGRDL	PLATINGRDL
11	17.6%	0.4%	2	-	AOI1DL2	AOI1DL2	AOI1DL2
12	21.9%	0.4%	2	-	CLEANTOPWLB	CLEANTOPWLB	CLEANTOPWLB
13	24.2%	0.0%	0	AOI1DL2	AOI1DL2	AOI1DL2	AOI1DL2
14	24.2%	3.9%	10	-	AOI1DL2	AOI1DL2	-
15	19.1%	0.4%	2	-	-	-	-
16	22.8%	1.3%	5	-	EXPOSERDL	EXPOSERDL	-

In what concerns the first objective, it is possible to see that the amount of overlap is considerable. After applying the problematic moment identification algorithm, the Overlap$_{Count}$ of this experiment has the same average than the experiments using simulated data (1.8%). The main difference is that, in the case study, overlap is present in more datasets, but the Overlap$_{Count}$ has a lower maximum, resulting in slightly less variance. We can also see that in all but two of the moments identified there were some overlapped factors.

To analyse the convergence of the RCA algorithms, we can see that in most moments there is an overall agreement between algorithms on what the root cause is. Of all the algorithms, the Co-Occurrences (CO) and the Decision Tree (DT) are the ones that have the least convergence and even capacity for root cause detection. However their behavior is different. In the case of the CO algorithm, when it was not able to distinguish between the factors, it did not provide any potential root cause. In the case of the DT algorithm, it always provided several possible root causes, but there were situations where none of the DT's proposed root causes were the same as the ones proposed by other algorithms. The two moments when it was not possible to see any convergence between the algorithms were the 9th and 15th moments. However, these two moments have very different levels of overlap, and the reason for this lack of convergence is not clear.

5.7 Discussion

The experiments described in the previous sections had different objectives, and in this section we discuss their results, how these combine to form coherent conclusions, and discuss further implications in terms of real-world applicability.

Experiments with Mockup data aimed at identifying the effect of overlap on the root cause detection performance on classifiers and factor ranking algorithms. Results presented in Section 5.6.1 prove that overlap can be very detrimental to the performance of classifiers, especially if the root cause is one of the overlapped locations. It is also shown that factor ranking algorithms' performance is resilient to the presence of overlap.

Stochastic Simulation data's experiments had the objective of validating both phases of the proposed solution, i.e., the problematic moment identification and factor ranking algorithms, in a scenario as close as possible to a real scenario. For the first phase, the problematic moments identified were coherent with what was expected, meaning that the EWMA algorithm enables the identification of the problematic moments. In terms of the results of the algorithms, the first conclusion is that the factor ranking algorithms performed much better than the DT classifier. Among the three factor ranking algorithms used, the CO algorithm has the best performance, detecting more root causes, and in higher ranks, and with less ties among locations. The CS algorithm leads to many ties. Both CS and RF failed to detect a root cause in one of the identified moments.

With the Real case study data, the objective was to validate the performance of the proposed solution on real-world data. However, we did not know the true root causes. As such, a indirect evaluation was performed, by comparing the root causes that were proposed as most likely by each algorithm. In these experiments, the CS and RF algorithm performed better than the CO algorithm and the DT classifier.

Considering the results of these three experiments, the first conclusion is that the proposed solution based on factor ranking algorithms performs much better than the solution based on classifiers, as evidenced by the results of experiments with Mockup and Stochastic Simulation data. When comparing the factor ranking algorithms, however, the differences in performance do not enable to determine the most promising ranking algorithm. In the Mockup experiments, only RF had a slight worse performance when the root cause was overlapped, with the other algorithms getting a perfect performance. With the Stochastic Simulation data, CO performed better than the other two. In the Real Case Study data, CS and RF performed better, by having the same possible root causes with a high ranking.

The cross-analysis of all experiments seems to indicate that, although factor ranking algorithms are clearly better than classifiers for root cause detection, there is no conclusive superiority between any of the factor ranking algorithms. However, the first two experiments are obtained in a controlled environment, and their results' level of validity is higher than those from the last experiment with real-world data.

In what concerns the two measures of overlap proposed, and considering the results of the experiments described above, the $Overlap_{Avg}$ presented abnormal levels of variance, and an increase in its values did not always correspond to an increase in the amount of overlapped factors. $Overlap_{Count}$ was more stable in the experiments, with approximately the same variance in the Mockup experiments ($Overlap_{Avg}$ had 0,000252 and $Overlap_{Count}$ 0,00026), 18 times less variance in the Stochastic Simulation experiments (0,02072 vs. 0,00114), and 8 times less variance in the Real case study experiments (0,00358 vs. 0,00042).

Figure 5.5 depicts a comparison between Overlap$_{Avg}$ and Overlap$_{Count}$ in the Real case study experiment. On the vertical axis we have the value of overlap, either from Overlap$_{Avg}$ (in red circles), or Overlap$_{Count}$ (in blue triangles). The horizontal axis represents the number of overlapped steps, which were verified manually on each moment's dataset.

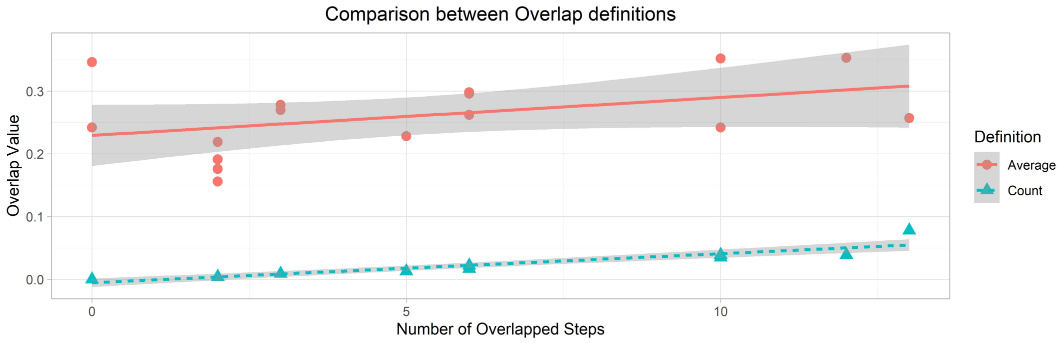


FIGURE 5.5: Comparison between Overlap$_{Avg}$ and Overlap$_{Count}$ behavior with the data from the Real case study experiment.

It is possible to see that not only is the variance greater in the Overlap$_{Avg}$ values, but that for the same number of overlapped steps in the dataset, Overlap$_{Avg}$ values can differ significantly. This behavior is problematic, because this means that Overlap$_{Avg}$ is influenced by aspects other than the desired (e.g., noise), and, as such, Overlap$_{Count}$ should be preferred, as it is more stable and is not as influenced by other aspects.

In addition, $Overlap_{Count}$ has the advantage of being easily interpretable (answering the question "what percentage of interactions are overlapped?"), and allowing for easy comparison between datasets of different origins.

The knowledge of root causes and its impacts on the development and validation of ARCA solutions is worth discussing further. To develop and validate ARCA solutions, aside the data itself, two types of labels are required: i) the labels used for classifying problematic products, and ii) the labels that identify the root causes.

The first type identifies products with problems, and is essential for the training and development of ARCA solutions, as we need to associate the data with a problem, in order to find a root cause for that problem. This type of label is relatively easy to find in manufacturing industry, as results of quality tests are abundant, and sometimes even automatically collected (such as in the semiconductor industry).

The second type of labels, although not necessary for training and development, are essential for a robust validation of the ARCA solutions developed in real data. It is desirable to compare the list of root causes provided by the developed solutions with the real problems, usually identified by experts. However, precisely because it requires human investigation, and due to the inefficient nature of the RCA process, it is often difficult to obtain these. The registers of solved problems by manufacturing companies, if available, can constitute a precious source of information to obtain the second type of labels. However, this aspect is still neglected in many companies, as the focus is to solve the problems impeding day-to-day functioning of the manufacturing operation, and not exactly on collecting information about the problem itself. If this type of labels are not available for real data, an alternative for validation is the use of simulated data, where it is possible to control the root cause of problems.

The proposed solution could be developed only with the first type of labels, but for a more robust validation, the second type of label is also required.

5.8 Conclusions

This chapter presents the issue of overlap, which occurs when performing Root Cause Analysis (RCA) to locate root causes in a manufacturing process (as described in Section 5.2). To the best of our knowledge, this is the first study that identifies and defines this issue. Overlap occurs when all products that go through a certain machine in a manufacturing step are all processed in another given machine in a later manufacturing step. The data generated this way is difficult to diagnose, as the influence those two machines on the quality of the final product is hard to distinguish. Such data is very complex to analyse, especially through predictive analytics algorithms, usually used to develop most ARCA solutions.

In order to develop manufacturing ARCA solutions able to overcome this phenomenon, we propose the use of a two-staged solution: i) Problematic Moment Identification, and ii) the use of factor ranking algorithms. The problematic moment identification stage aims to select data relevant for the analysis, and that frames the problem adequately. The problematic moment identification algorithm used is a EWMA control chart. The idea behind the use of factor ranking algorithms is to avoid hiding highly correlated factors, therefore enabling analysts to have full information on equally probable root causes. Three factor ranking algorithms were proposed (as described in Section 5.4): Co-Occurrences (CO), Chi-Square (CS) and Random Forest (RF).

In order to analyse the effect of overlap, illustrate its pernicious effects on data analysis, and to validate the algorithms presented above to counter the effects of overlap, three experiments are conducted (see Section 5.5 for more details): i) using Mockup data, ii) using Stochastic Simulation data , iii) using data from the Real case study.

With the Mockup data experiments (Section 5.6.1), it is possible to understand that overlap does have an impact on the performance of classifiers, specially when the root cause is one of the overlapped factors. It is also shown that factor ranking algorithms are much more resilient to the presence of overlap. In the experiments using Stochastic Simulation data (Section 5.6.2), we are able to validate both the problematic moment identification algorithm and the factor ranking algorithms using data where the root causes were controlled. By analysing the data directly from the real case study (Section 5.6.3), we are able to illustrate the presence of overlap in real datasets, and validate the algorithms used, albeit indirectly, by verifying the existence of convergence in the proposed root causes by all algorithms. In terms of the factor ranking algorithms proposed, the CO algorithm has better performance in the simulated data, but has worse convergence with the other algorithms with the case study data. The other algorithms (CS and RF) have a more stable performance in the two experiments. There is no conclusive evidence that any of the three algorithms is better than the others. However, it is clear that the proposed solution and any of the factor ranking algorithms have a better performance than when using decision tree classifiers.

This chapter contributes to the literature by identifying a phenomenon which results from the normal (and even desirable) functioning of a manufacturing process, but which impacts the analysis of data. Besides this identification, we define an overlap, propose two ways of measuring it, and a two-staged solution to overcome it, and validate the solution through three different experiments. The proposed solution performs better than solutions using classifiers, in situations where overlap is present.

Chapter 6

Overlap in Automatic Root Cause Analysis in Manufacturing: An Information Theory-based Approach

6.1 Introduction

In the previous chapter, overlap was defined and measured based on the strength of association between two nominal factors, where each factor was a step in a manufacturing process, and the levels of the factors were the machines that could be used in that step.

These definition and measure allow for the identification and quantification of overlap, and the development of ARCA solutions that are able to detect the most likely root causes, even in the presence of overlap. These solutions use rankings, that do not hide possible root causes, as opposed to classification algorithms, where the possibility of hidden root causes exist. However, such measure has two aspects that can be improved: i) it does not consider the direction of the association, and ii) it requires post-hoc testing to determine the effect of overlap between individual machines (more details in Section 5.2). To improve these aspects, a new measure of overlap based on information theory is proposed, namely the use of a variant of mutual information called Positive Mutual Information (PMI), proposed by Brun, Castagnos, and Boyer (2009). Based on this new definition, a new approach to tackle RCA and determine the most likely root causes is introduced.A visualization tool, that makes the task of detecting overlap and the finding root causes easier for practitioners, is also developed.

The remainder of this chapter is structured as follows. In Section 6.2, it is motivated and described the novel proposed definition of overlap, as well as the proposed approach for RCA based on this definition. Sections 6.3.1 and 6.3.2 explain and show (respectively) the results of the experimental procedures taken to validate the proposed methodology. A discussion of these results is presented in Section 6.4. To conclude, a summary of the findings is presented.

6.2 Proposed Methodology

6.2.1 Motivation

The definition of overlap presented in the previous chapter allowed for the quantification of overlap and the development of ARCA solutions that are able to detect the most likely root causes of problems even in a situation where overlap is present. However, such definition can be improved in two aspects.

First, the previous definition tests the strength of the association between factors, but it does not take into consideration whether such association is positive or negative. This means that we are not only considering the association generated by a product that goes through a certain machine in a step and *always* goes through a certain machine in another step (positive association), but also when a product goes through a certain machine in a step and it *never* goes through a certain machine in another step (negative association). Overlap is problematic only in the case of positive association, as it is in that scenario that it becomes impossible to distinguish the tuples. In case of negative association, the tuple where the product goes through can be immediately determined as a root cause (in the situation where the product is problematic and only those tuples are considered as possible root causes).

The second aspect to be improved is that the definition focuses on comparing steps (factors) and not tuples (pairs of step-machine). This could be somehow circumvented by using one-hot-encoding on the factors (which represent steps). However, that is not enough, because the expression would take into consideration the interactions between tuples of the same steps, which would constitute an example of negative association (if a product goes through a machine in a certain step, it does not go through other machines in the same step). This situation conflates with the first shortcoming and prevents the above-mentioned definition of being used to detect overlap between tuples instead of just between factors.

6.2.2 Proposed Measure

In order to improve the measure proposed in the last chapter, an alternative measure is proposed in this chapter, based on concepts from information theory. As mentioned in the previous chapter, overlap is present when the values of two columns in a dataset like the example in Table 3.1 are synchronized (in the example, "Machine 1" with "Machine 3"). As overlap has a pernicious effect on the analysis using DM and ML algorithms, it is better to consider other perspectives to measure overlap.

As such, and to include resilience to noise, overlap is defined and measured from a probabilistic and information theory perspective.

Considering overlap from a probabilistic perspective, an overlap between two tuples is the very high probability of a product going through a certain machine given that we know that it went through another machine in another step. Formally, high overlap between two tuples occurs when:

$$P(Y = y|X = x) \geq Th, \tag{6.1}$$

that is, overlap occurs when the probability of going through machine y in step Y, given that a product goes through machine x in step X, is above a certain threshold (0.9 is proposed to be the default, as it considers cases of very high overlap, while leaving some margin for errors caused by noise).

From an information theory perspective, if we consider each tuple as a random variable, it is possible to say that knowing the value of one of the variables provides a high amount of information about the other variable. This information "shared" by both variables is quantified in Shannon (1948) as Mutual Information (MI) or:

$$I(X;Y) = \sum_{x}\sum_{y} p(x,y) log \frac{p(x,y)}{p(x)p(y)}, \tag{6.2}$$

where $I(X;Y)$ is the MI. MI varies between 0 and 1, where 1 means the highest association between variables. Alternatively, MI can be defined in terms of entropy:

$$I(X;Y) \equiv H(X) - H(X|Y) = H(Y) - H(Y|X), \tag{6.3}$$

where $H(X)$, $H(Y)$ represent the entropy of X and Y (two random variables), respectively. $H(X|Y)$ and $H(Y|X)$ represent conditional entropy, as defined in Section 2.3.2. This definition aligns with some of the aspects considered relevant for measuring overlap, except that it considers both positive and negative association. In order to improve the definition, the use of Positive Mutual Information (PMI) is proposed, a measure defined in Brun, Castagnos, and Boyer (2009), which takes into consideration only the positive (normalized) mutual information between the two binary variables:

$$PMI(x,y) = \frac{2}{H(x)+H(y)} \left(p(x_p,y_p) log \frac{p(x_p,y_p)}{p(x_p)p(y_p)} + p(x_n,y_n) log \frac{p(x_n,y_n)}{p(x_n)p(y_n)} \right), \tag{6.4}$$

where x_p and y_p represent positive values (in this specific case "went through machine" and "problematic product", respectively), and x_n and y_n represent negative values ("did not go through machine" and "normal product", respectively). So, in the case of overlap between two tuples, PMI only takes into consideration the cases where either the product goes through both tuples, or through neither of them. PMI aligns better with the aspects considered relevant for defining overlap than mutual information does. To compute the effect of overlap on a dataset like the one in Table 3.1, first one-hot encoding of the factors representing steps is performed, in order to get binary variables representing the tuples, and then the PMI between all the new tuples is computed. The number of interactions that have a PMI above a certain threshold is then counted, and is divided by the number of interactions with a valid value of PMI. The number of valid interactions is used, because PMI on factors that have no positive association is undefined, and considering the total number of interactions would "dilute" the presence of overlap, diminishing the perception of the issue.

Expression (6.5) defines mathematically the effect of overlap on a dataset, based on the concept of PMI. $C_{i,j}$ is one when the PMI between the factors i and j is greater than the threshold, and zero otherwise. The numerator counts all interactions between i and j (where $i \neq j$), above the threshold.

$$Overlap_{PMI} = \frac{\sum_i^I \sum_j^J C_{i,j}}{N_{VI}} \tag{6.5}$$

where:

$$C_{i,j} = \begin{cases} 1, & \text{if } PMI(i,j) \geq Th \\ 0, & \text{if } PMI(i,j) < Th \end{cases}$$

and $i \neq j$. N_{VI} stands for the number of valid interactions between factors. Please note that, as we only take into consideration positive associations, negative associations are filtered out, and this definition is able to consider both steps and tuples.

Considering the example in Table 3.1, we would perform one-hot encoding, which would yield Table 6.1. The values of the label *Problem* are included, as they are relevant to identify which tuples are more closely associated with a problematic product. Each column represents a tuple, identified by the letter of the step followed by the code of the machine.

With this table, it is possible to compute the PMI values between all tuples, using Expression (6.4), which can be arranged in a matrix, as represented in Table 6.2. Cells with a "-" represent undefined values of PMI.

TABLE 6.1: Example of Table 3.1 transformed with one-hot encoding, before computing the PMI values.

A1	A2	B3	B4	B5	NN1	NN2	Problem 0	Problem 1
1	0	1	0	0	1	0	0	1
0	1	0	1	0	0	1	1	0
0	1	0	0	1	1	0	1	0
1	0	1	0	0	0	1	0	1

TABLE 6.2: Matrix with the PMI values of the interactions between tuples in the example of Tables 3.1 and 6.1.

	A1	A2	B3	B4	B5	NN1	NN2	Problem 0	Problem 1
A1	1	-	1	-	-	0	0	-	1
A2	-	1	-	0.505	0.505	0	0	1	-
B3	1	-	1	-	-	0	0	-	1
B4	-	0.505	-	1	-	-	0.505	0.505	-
B5	-	0.505	-	-	1	0.505	-	0.505	-
NN1	0	0	0	-	0.505	1	-	0	0
NN2	0	0	0	0.505	-	-	1	0	0
Problem 0	-	1	-	0.505	0.505	0	0	1	-
Problem 1	1	-	1	-	-	0	0	-	1

Notice that, in this example and in other situations, the PMI between tuples of the same step does not exist, which is expected because those tuples are mutually exclusive. Finally, it is possible to compute the $Overlap_{PMI}$ of the whole dataset using Expression (6.5). With the default threshold value of 0.9, and for the example discussed, the numerator of the expression is 8. The denominator, i.e., the number of valid interactions, is 40. This yields a value of $Overlap_{PMI} = 0.2$.

6.2.3 Proposed Approach

Given the above definition, a new factor ranking algorithm is proposed to evaluate the overlap between factors, and detect the tuples that are most likely to be root causes.

The proposed factor ranking algorithm is integrated in the approach illustrated in Figure 5.1, at the same stage the other factor ranking algorithms described in Section 5.4 are used. As the manufacturing process is operating, data is extracted from it and is analysed by a problematic moment identification algorithm, which tries to identify parts of data relating to periods where there were problems in the manufacturing process. It then selects those parts of the data, which are analysed by a factor ranking algorithm, that outputs a list of the most likely root causes.

As in the previous chapter, the algorithm selected for the problematic moment identification stage was the Exponentially Weighed Moving Averages (EWMA) (Roberts, 1959). To estimate the variance to be used in EWMA, the proportion of problematic products per lot is modelled as a Beta distribution, estimating the α and β parameters and then computing the variance.

In what concerns the novel factor ranking algorithm proposed, the intuition is straightforward: the goal is to find the tuples that are overlapped with the tuple of the factor representing the final quality (e.g., Label) of a problematic product. In addition, this procedure makes it possible to determine the tuples that are overlapped among themselves and the label, finding clusters of machines that are related to problematic products. After the PMI is computed for all interactions, the tuples presenting a PMI value in relation to "Label High" tuple greater than the threshold are kept. These values are then sorted in decreasing order and presented as a list to the analyst. Algorithm 4 summarises the novel factor ranking algorithm.

Algorithm 4: PMI algorithm

```
Input: Dataset of problematic moment; Threshold
Output: Ordered list of most likely root causes
begin
    Output ← [∅];
    To-use-dataset ← OneHotEncoding(Input);
    Compute PMI Matrix of To-use-dataset;
    Output ← PMI Matrix's column related to problem w/ tuples' names;
    Remove from Output values below Threshold;
    Sort Output;
end
```

Considering the example in Table 6.2, the only tuples that overlap (i.e., have a PMI value above the threshold) with the problematic product tuple ("Problem 1") are the tuples "A1" and "B3" (excluding the self overlap of "Problem 1"). Of these two tuples, we can see that they also overlap each other, and that there exists no other overlap clusters among other tuples. Therefore, the tuples "A1" and "B3" would be presented to the analyst as the most likely root causes.

To ease the detection of root causes by practitioners, and in addition to the list, a visualization is proposed, that depicts tuples as nodes, and the overlap between them as edges between the nodes. Directed edges are shown, to represent the order in which the steps occurred. The nodes are colored according to their steps, e.g., one color is used for Step04, and another for Step06. Figure 6.1 is an example of the visualization of one of the experiments (different from the previous example). The visualization displays all the nodes and overlaps as a grid of nodes, but these can be moved to ease the perception of clusters, and it is possible to zoom in certain areas. In detail, the focus is on the cluster which contains the node "Label High", which indicates a problematic product. We can see a cluster connected with the tuple Label High, which is composed of three Step-Machine tuples. The arrows indicate the order in which each step occurs (e.g., in this example, Step04_Eq occurs before

Step06_Eq and Step20_Eq). Each node is colored according to its step. It is possible to produce a new visualization for each time the overlap threshold is adjusted.

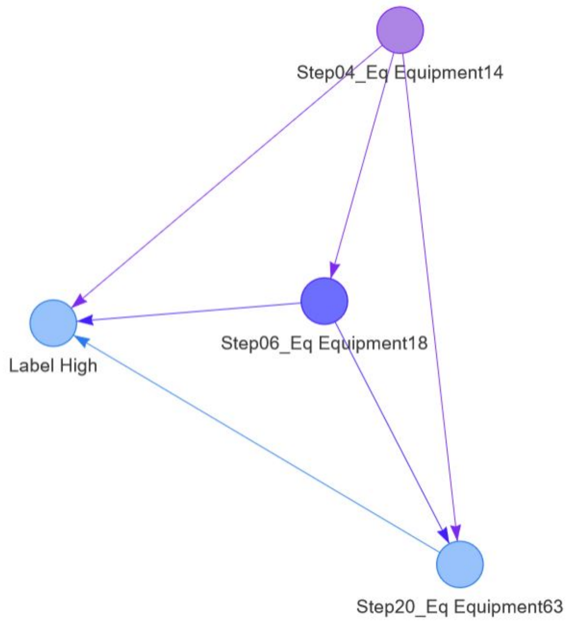


FIGURE 6.1: Detail of an example of visualization of PMI overlap among different tuples in Stochastic Simulation data.

In the example provided by Figure 6.1, the true root cause is "Step04_Eq Equipment14". We can see that it is correctly identified (it appears in the network associated with "Label High"), but that it is overlapped with two other tuples. These three tuples would compose the list given by the algorithm. This exemplifies a situation where overlap is present, and makes it impossible to determine which of the three tuples identified is the root cause. However, the number of possibilities has been greatly reduced, from 91 possible tuples (22 steps with 4 possible machines each, plus 1 step with 3 possible machines), in this experiment, to three, easing the effort required by the analyst to detect the root cause, making the process more efficient.

6.3 Evaluation

6.3.1 Evaluation Setup

To validate the performance of the proposed approach, experiments using several datasets were conducted. These are the same as the ones used in the previous chapter. This makes comparisons between both overlap definitions and proposed approaches easier.

The proposed approach (denominated PMI) is compared with the Co-Ocurrences (CO) algorithm, arguably the factor ranking algorithm with the best performance in the previous chapter, and a Decision Tree (DT) classifier.

6.3.2 Evaluation Results

Mockup

A total of 250 mockup datasets were generated, and the results are presented in Figures 6.2 and 6.3. For each percentage of overlapped columns and method it is represented the percentage of datasets that the method is able to identify a small group of factors that contain the root cause. The capacity for root cause detection of the different factor ranking algorithms (CO and PMI) and the classifier algorithm (DT) is evaluated in two different scenarios, i.e., one where the root cause is one of the overlapped factors, and another where it is not.

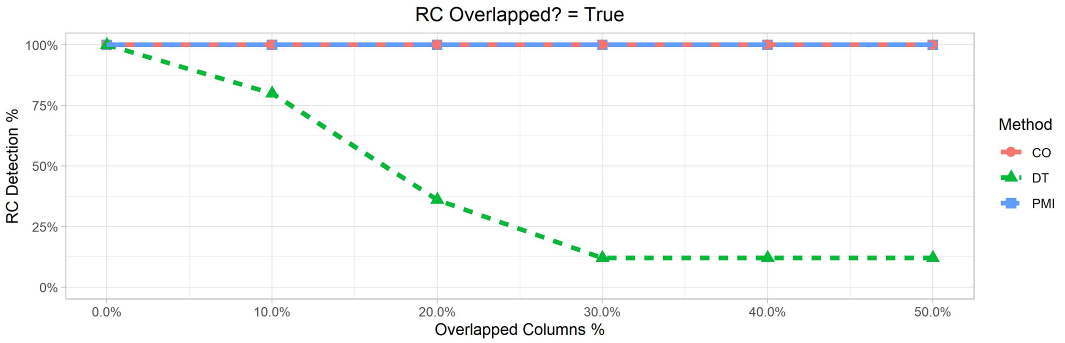


FIGURE 6.2: Results of the Mockup experiments when the root cause is in one of the overlapped factors.

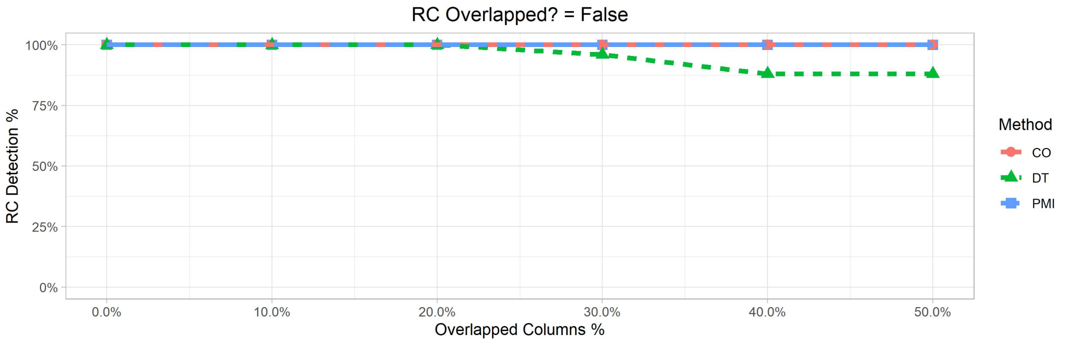


FIGURE 6.3: Results of the Mockup experiments when the root cause is not in one of the overlapped factors.

It is possible to see that, both when the root cause is in one of the overlapped factors and when it is not, the PMI algorithm is able to identify a small group of factors that contain the root cause, in all datasets. This is in line with the best performance from the algorithms proposed in the previous chapter, where, for these simplified scenarios, the CO algorithm achieved the same result (the graphs for CO and PMI overlap each other in the figure).

It is also presented the results of the DT classifier, as its performance declines in the presence of overlap, even more so when the root cause is in one of the overlapped factors.

Stochastic Simulation

This section focuses on evaluating the performance of the proposed solution as a whole. Table 6.3 presents the results of the RCA algorithms used, divided by dataset, and each moment where it is possible to detect a problem. Each column represents the root causes detected, and in brackets the position in the ranking of variables. The code of the root causes is as explained in Figure 5.2. Each time a root cause was correctly detected, it is presented. In the case of CO and PMI, the ranking of the factor is also presented. In the ranking, ties among positions may happen, which are not as clear as an isolated position. As such, the ties are signaled with "*". For the classifier (DT), in addition to the root causes, the false positives (or incorrect detections) are presented as "FP", because: i) this algorithm does not provide a ranking, and ii) in order to have a better notion of the false detections occurred.

TABLE 6.3: Table with the results of the RCA algorithms with Stochastic Simulation data.

Dataset	**Moment**	**Overlap**$_{Count}$	**Overlap**$_{PMI}$	**CO**	**PMI**	**DT**
1	*1*	0.4%	4.20%	RC2 (1st); RC1 (12th)	**RC2 (1st); RC1 (4th)**	RC2
1	*2*	0.0%	0.05%	**RC2 (1st)**	**RC2 (1st)**	RC2; FP; FP
1	*3*	8.3%	16.11%	**RC3 (1st)***	**RC3 (1st)***	FP
1	*4*	0.0%	0.16%	**RC3 (1st); RC4 (2nd)***	RC3 (1st)	RC3; FP;FP
2	*1*	5.1%	14.24%	RC2 (6th)*	**RC2 (2nd)***	FP
2	*2*	0.0%	0.10%	**RC2 (1st)**	**RC2 (1st)**	RC2;FP
2	*3*	0.0%	0.00%	**RC3 (1st)**	**RC3 (1st)**	RC3;FP;FP
2	*4*	0.0%	0.67%	**RC3 (1st)**	**RC3 (1st)**	**RC3**
2	*5*	0.0%	0.22%	RC3 (6th)	**RC3 (1st)**	FP; RC3
3	*1*	0.4%	3.60%	RC2 (1st)	**RC2 (1st); RC1 (10th)**	RC2
3	*2*	0.0%	0.00%	**RC2 (1st)**	**RC2 (1st)**	RC2;FP
3	*3*	9.1%	18.38%	**RC3 (1st)***	**RC3 (1st)***	FP
3	*4*	0.0%	0.61%	**RC3 (1st)***	**RC3 (1st)***	RC3, RC4; FP

First, if we take into consideration the values of Overlap$_{Count}$ and Overlap$_{PMI}$, we can see that Overlap$_{PMI}$ can provide a finer detail of analysis. That is, it is able to detect and make more evident smaller cases of overlap. This behavior is expected, as the new definition of overlap based on PMI considers overlap on the level of the Step-Machine tuples, and not only at the Step factor level, as the previous definition. There are even situation where no overlap was detected by Overlap$_{Count}$ (e.g., dataset 1, moments 2 and 4), and some overlap was detected by Overlap$_{PMI}$.

When considering the performance of the algorithms, it is possible to see that the PMI algorithm has, overall, a better performance than the CO algorithm. It is able to detect what is detected by the CO algorithm, but does so placing the true root causes in better positions (e.g., dataset 1, moment 1; dataset 2, moments 1 and 5).

One note is that, to enable the detection of tuples overlapped with the Label, there was the need to lower the threshold (of Expression 6.5) to 0.8 on the first two datasets, and 0.3 in the last dataset. As these have increasing levels of noise, this seems to indicate that the threshold needs to be adapted to the datasets at hand, and that the noisier the dataset is, the lower the threshold needs to be to detect tuples overlapping with the label. The default threshold value of 0.9 is still used to measure the overlap of the whole dataset (Overlap$_{PMI}$), and it was chosen as to include instances with very high overlap, but with some leeway for noise.

Real Case Study

The experiments with data from a case study have the objective of further validating the use of the proposed algorithm, in real data. As mentioned in Section 5.5.3, due to the lack of information on the real root causes, validation is attempted by comparing the results of each algorithm, and see if there exists a convergence in the results, which would strongly indicate a root cause.

The results are expressed in Table 6.4. Only the steps of the step-machine tuples are shown to facilitate reading, and because of confidentiality agreement.

Comparing Overlap$_{Count}$ and Overlap$_{PMI}$, it can be seen that what happened in the experiments with Simulation data in the previous section also happened in real data: Overlap$_{PMI}$ is able to detect and make more evident smaller cases of overlap (e.g., moments 3, 5 and 13).

In terms of the threshold used for PMI, it is fixed at 0.3, like in the dataset 3 of the Stochastic Simulation experiments. Even with such a low threshold, there are many moments when the PMI algorithm is not able to detect any root cause (four in total). However, such is the case with the CO algorithm as well.

When considering the identified moments with a detection, it is possible to see that there is indeed a convergence towards a common root cause. Even in moments 14 and 16, where, of the three listed, only the PMI algorithm is able to detect a root cause, the root causes detected are the same as the two algorithms presented in the previous chapter, in Table 5.5, but not listed here.

6.4 Discussion

In this section the results of the different experiments, and how these combine to form coherent conclusions are discussed and summarised.

TABLE 6.4: Table summarizing the problematic moments identified in the Real Case Study dataset, as well as the common root-causes among the different algorithms proposed.

Moment	Overlap$_{Count}$	Overlap$_{PMI}$	CO	PMI	DT
1	0.90%	0.96%	COATFUSE	-	COATFUSE
2	1.70%	4.74%	PLATINGRDL	PLATINGRDL	PLATINGRDL
3	0.00%	1.25%	EXPOSEFUSE	EXPOSEFUSE	EXPOSEFUSE
4	2.20%	1.67%	EXPOSEFUSE	EXPOSEFUSE	EXPOSEFUSE
5	3.50%	15.88%	CUREFUSE	CUREFUSE	CUREFUSE
6	0.90%	1.89%	COATRDL	COATRDL	COATRDL
7	0.40%	0.67%	EXPOSERDL	-	EXPOSERDL
8	3.90%	3.12%	-	AOI1DL1	AOI1DL2
9	7.80%	9.16%	-	-	-
10	1.70%	1.50%	-	PLATINGRDL	PLATINGRDL
11	0.40%	0.79%	-	AOI1DL2	AOI1DL2
12	0.40%	1.45%	-	-	CLEANTOPWLB
13	0.00%	0.74%	AOI1DL2	AOI1DL2	AOI1DL2
14	3.90%	9.87%	-	AOI1DL2	-
15	0.40%	0.66%	-	-	-
16	1.30%	1.54%	-	EXPOSERDL	-

In the Mockup experiments, the proposed PMI approach was able to achieve a performance equal to the best solutions based on the previous overlap definition (CO), and a much better performance than the DT classifier.

In the experiments with the Stochastic simulation data, PMI achieved better performance than the other algorithms, as it put the true root causes in the same or higher rankings than the CO approach. Also Overlap$_{PMI}$ provides a finer detail of analysis than Overlap$_{Count}$, as it is able to detect and make more evident smaller cases of overlap.

When applying the different algorithms to the Real Case Study, the proposed approach also achieved better results, as it identified root causes in common with most methods, even those proposed in Chapter 5.

From the results of all experiments, the conclusion is clear that the proposed PMI algorithm is an improvement over the algorithms proposed in the previous chapter, as well as the PMI definition of overlap is an improvement over the previous definition.

6.5 Conclusions

This chapter presents a new measure of overlap. The novel measure of overlap uses information theory concepts, more specifically Positive Mutual Information (PMI). This measure considers two critical aspects, namely whether the associations are positive or negative, and it is appropriate to detect overlap among step-machine tuples. This measure is the basis of a factor ranking algorithm that is used to detect root causes, in an ARCA solution.

To validate this new approach, three experiments are conducted: i) using mockup data, ii) using simulated data that emulates a case study, iii) using real data from the case study itself. It was possible to conclude that the proposed algorithm achieved better performance with data from the Stochastic Simulator, competing with the benchmark algorithms in the other two experiments.

This chapter contributes to the literature by presenting a robust measure of overlap, which allows for a better understanding and analysis of this characteristic of the problem. In addition, a new factor ranking algorithm is presented, with positive results. A visualization was also developed that eases the analysis by practitioners, by depicting the relevant overlaps between tuples in a manipulable graph.

Chapter 7

Understanding Overlap in Automatic Root Cause Analysis in Manufacturing Using Causal Inference

7.1 Introduction

In the previous chapters, two ways of measuring overlap were presented. The first was based on the strength of association between two nominal factors, while the second one tackles shortcomings in the first measure, and proposed an alternative one based on information theory concepts, that surpassed those shortcomings. However, the solutions developed based on these measures merely list the most likely root cause locations, representing overlapped tuples as having the same importance, and not distinguishing which of the overlapped tuples is the true root cause.

In this chapter, the aim is to tackle this issue by using causal inference and *do* calculus to develop an approach that can determine the true root cause from a group of possible root causes. These methods enable us to estimate, in certain conditions, the effect of a product passing through a certain location while disregarding the effect of overlap. This new approach led to the development of a causal inference model, which allowed us to study the effect of overlap in more detail, enabling us to determine the conditions when it is possible to determine the root cause tuple of a given problem, and determining the true root cause, when such conditions are met. For further details on causal inference, please refer to Section 2.5.

This chapter contributes to the literature by understanding the limitations of performing RCA on data with overlap, determining the conditions on which it is possible to determine the exact tuple that causes the problem, and proposing methods to do so. In addition, the impact of noise on the performance of ARCA solutions is studied, based on the proposed model.

The remainder of this chapter is structured as follows. In Section 7.2 the proposed methodology is presented, in the form of a causal model of overlap and interventional probabilities that are used to determine the true root cause. In addition, these are also used to determine the conditions that are required for a solution to be able to detect the true root cause in the presence of overlap. Sections 7.3.1 and 7.3.2 present the setup of the experiments used to validate the proposed approach, and the results of these experiments. A discussion of these results is presented in Section 7.4. To conclude, a summary of the findings is presented. Finally, the conclusion is presented in Section 7.5.

7.2 Methodology

Previous approaches to deal with overlap have as a result a list of the tuples that are the most likely root causes, in which overlapped tuples are represented as having the same influence on quality. The methodology proposed in this section uses the results of previous studies to identify and select relevant and overlapped tuples, and then tries to identify the true root cause from among them. For that, it is required a methodology that is able to estimate the effect that a product passing through a specific tuple has on quality, while disregarding the effect of overlap. As such,the use of causal inference and *do* calculus is proposed, as these allow us to estimate such probabilities, conceptualize overlap, and understand when and how can we detect the true root cause from a set of overlapped tuples.

Of the previous chapters, the approach with better results is the Positive Mutual Information (PMI) algorithm. As such, this is the approach used to select the tuples X_i, before trying to find the true root cause from among them.

The motivation for the use of causal inference to analyse overlap comes from the idea that, if we can "disconnect" a certain tuple from the influence of overlap, we would be able to see which of the tuples is the real root cause, because we would be able to disregard the influence of overlap. Such evokes the notion of intervention (as seen in Section 2.5). The causal effect of a tuple X_i on the label Y is given by Expression (7.1). Considering binary variables (1 if the passes through X_i or the label Y is problematic, and 0 if it does not pass, or the product is normal), this expression means the probability of a product being problematic, if we had "forced" the product to go through that tuple.

$$P(Y = 1|do(X_i = 1)) \tag{7.1}$$

To compute such probability, we require a causal graph. The proposed causal graph is defined in Figure 7.1. An overlap U (unmeasured), makes the product go through all the tuples X_i (from 1 to N) it affects, or through none of them. Each tuple may have an influence on the label Y, where a 1 signifies a problematic product.

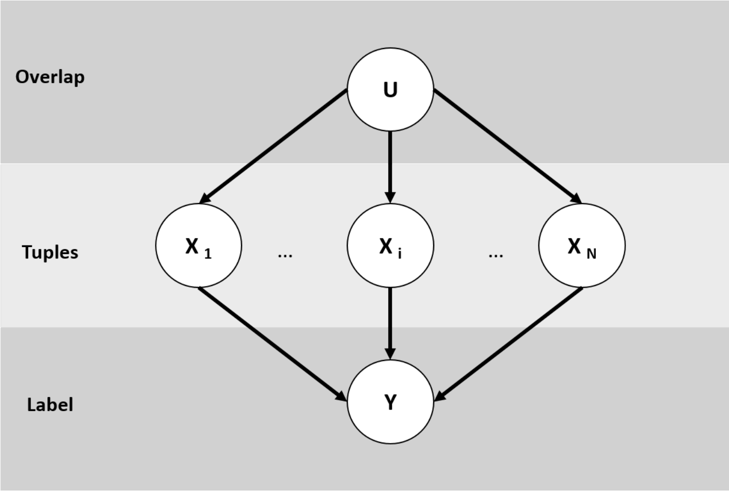


FIGURE 7.1: Causal Model of overlap.

Figure 7.1 represents the causal relations among variables involved in a situation with overlap, and can be considered a graphical definition/conceptualization of overlap. Overlap can be understood as an unmeasured variable U, which represents a local "synchronicity" of the manufacturing process, and this variable defines the value of all the tuples X_i that may influence the label Y. All tuples may have an influence on the label, but we assume that not all of them lead Y to become one (considering all variables are binary, either 0, or not active, and 1, active).

The assumptions encoded in the model are: i) there is a variable U that "synchronizes" whether a product goes through all the selected tuples or not; ii) the fact that a product goes through one tuple is not the cause that makes it go through another tuple (hence no arrow between tuples X_i); iii) the "synchronicity" U does not directly influence whether a product is problematic or not (hence no direct arrow from U to Y); iv) the root cause node is one of the tuples, but does not need to be all of them - the arrows from X_i to Y represent a possibility of tuple X_i being a cause of Y.

Figure 7.1 has associated the functional causal model below:

$$\begin{aligned} &U = \{0,1\} \\ &X_1 = U \\ &... \\ &X_i = U \\ &... \\ &X_N = U \\ &Y = X_i^*, \end{aligned} \tag{7.2}$$

where X_i^* is the root cause tuple, which is unknown, and finding it is the objective of RCA. This functional causal model simply states that U randomly assumes the value of 0 or 1, and, without the interference of noise, all X_i have the same value as U. The label Y is active when the root cause node X_i^* is active. As we do not know which of the nodes is the root cause, the objective is to estimate the causal effect using interventional queries.

Note that, for each group or cluster of overlapped tuples, one would have a different variable U_i, that represents a local "synchronicity pattern" in the manufacturing process. In RCA, we are mainly focused on the cluster which contains the label, because it is the one that can lead us to the root cause of a problem identified in the label. Any tuple overlapped with the label will be overlapped with the other tuples that are also overlapped with the label.

To exemplify with a specific case, consider the example in Figure 3.1. In this example, the overlap exists between the tuples "Step A - Machine 1" and "Step B - Machine 3", and the label "Problem = 1". This translates to the causal model in Figure 7.2.

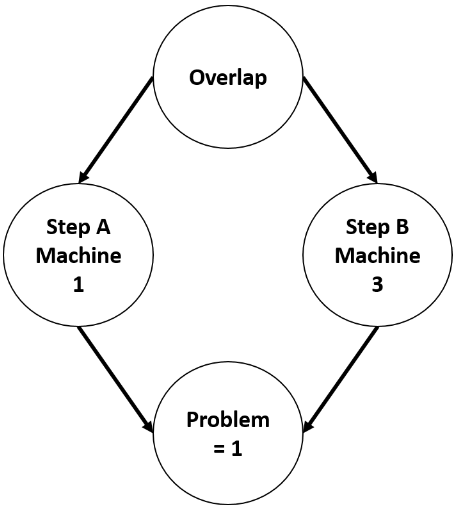


FIGURE 7.2: Overlap model of the example in Figure 3.1.

The corresponding functional causal model would be:

$$\begin{aligned}&\mathit{Overlap} = \{\text{False}, \text{True}\}\\&\text{“}\mathit{Step\ A\text{-}Machine\ 1}\text{”} = \mathit{Overlap}\\&\text{“}\mathit{Step\ B\text{-}Machine\ 3}\text{”} = \mathit{Overlap}\\&\text{“}\mathit{Problem} = \mathit{1}\text{”} = \mathit{Tuple}^*\end{aligned} \tag{7.3}$$

where Tuple^* is one of the two tuples that is the true root cause. In this specific example, as we know that "*Step A Machine 1*" is the true root cause, the expression becomes "*Problem = 1*" = "*Step A-Machine 1*".

It is clear from the model that, when there is no interfere with the influence of overlap in the tuples, it is impossible to distinguish between the tuples. Despite this seemingly trivial conclusion, the model still allows us to obtain knowledge useful to tackle the presence of overlap in ARCA. This model enables the identification of the minimal amount of interference that allows us to compute the causal effect of a tuple X_i on the label Y. That interference can be caused by random events that disrupt the local synchronicity in production, i.e. overlap, which leads to a few products not being affected by overlap. The knowledge of such limit is relevant because the ARCA solutions need to be able to automatically identify the situation they are evaluating, and present results that are adequate to such situations. When a solution identifies a situation where it is possible to do so, it should determine the single true root cause. However, if such is not possible, it should still present a group of most likely root causes, a reduced search space that makes the analysts' task easier. It is necessary to understand if this limit is relative to the number of instances in the dataset (if it is required that a percentage of the instances in the dataset are not overlapped), or if such limit is absolute (if it is required an N number of non-overlapped instances), and if it depends on the number of overlapped factors, or on other variables.

The interference threshold can be identified by taking a closer look on how to compute the interventional query $P(Y|do(X_i))$ (simplified notation: $Y = 1 \Leftrightarrow Y$; $Y = 0 \Leftrightarrow \overline{Y}$). Taking into consideration the adjustment for direct causes mentioned in Section 2.5, the above mentioned query can be computed by the following expression:

$$\begin{aligned} P(Y|do(X_i)) = P(Y|X_i,U)P(U) + P(Y|X_i,\overline{U})P(\overline{U}) = \\ P(U)\frac{P(Y,X_i,U)}{P(X_i,U)} + P(\overline{U})\frac{P(Y,X_i,\overline{U})}{P(X_i,\overline{U})} \end{aligned} \tag{7.4}$$

From Expression (7.4), it is possible to verify that, when considering the model described above without any interference, the denominator $P(X_i,\overline{U})$ is always 0, leading to an impossibility of computing the query $P(Y|do(X_i))$, and consequentially the causal effect. As such, the limit to how "pure" overlap can be, so as that it is possible to compute the causal effect, is $P(X_i,\overline{U}) > 0$. This means that we need to have at least one instance where X_i and $\overline{U}$ occur at the same time, or, in other words, X_i is equal to one, while all other tuples are equal to 0. As we need to compute the causal effects for all X_i and then compare them, the minimal condition for being able to compute the causal effect is that we have at least $N-1$ instances (N the total number of X_i tuples) where one of the tuple's value is one, while the others have a value of zero. As such, it is possible to say that the limit is absolute, and depends on the number of overlapped tuples.

As U is unmeasured, it is not possible to use the adjustment for direct causes, and as such it is necessary to use the Back-Door (BD) criterion. When examining Figure 7.1, it is possible to see that, to block any interference through the BD, we need to control all the other tuples. Therefore, we can compute such a query using Expression (7.5), where z are all possible combinations of $N-1$ binary variables representing all the tuples except X_i.

$$P(Y|do(X_i)) = \sum_{z} P(Y|X_i, z)P(z) \tag{7.5}$$

As an example, to compute the causal effect of tuple X_1 for a model with three tuples, the expression would become:

$$P(Y|do(X_1)) = \sum_{x_2}\sum_{x_3} P(Y|X_1, x_2, x_3)P(x_2, x_3) = P(Y|X_1, X_2, X_3)P(X_2, X_3) +$$
$$P(Y|X_1, \overline{X_2}, X_3)P(\overline{X_2}, X_3) + P(Y|X_1, X_2, \overline{X_3})P(X_2, \overline{X_3}) + P(Y|X_1, \overline{X_2}, \overline{X_3})P(\overline{X_2}, \overline{X_3})$$

After determining the overlap limit, it is necessary to consider the different aspects of causation. As explained in Pearl (1999) and in Section 2.5, the notion of causation can have several types, namely Probability of Necessity (PN - how necessary is for a product to pass through a tuple for the product to become problematic), Probability of Sufficiency (PS - if it is sufficient for a product to pass through a tuple to become problematic), and Probability of Necessity and Sufficiency (PNS - if it is both necessary and sufficient to pass through a tuple for a product to become problematic). However, we cannot use counterfactuals in this context, because we do not know the full functional causal model of overlap. When U is active, all the X_i tuples are active, but we do not know which of the tuples activate the label Y. In other words, we do not know X_i^*, in Expression (7.2), because that is precisely what RCA tries to identify.

To circumvent this limitation, we propose an adaptation using interventional queries with a similar reasoning, like in the following expressions:

$$PS = P(Y|do(X_i)) \tag{7.6}$$

$$PN = P(\overline{Y}|do(\overline{X_i})) \tag{7.7}$$

$$PNS = P(Y, X_i)PS + P(\overline{Y}, \overline{X_i})PN \tag{7.8}$$

These probabilities may be computed after the most relevant tuples have been identified using PMI to compute the overlap of the tuples with the label. This makes the process more efficient, and avoids computing interventional queries on tuples of the same step. These queries are problematic because they become not defined due to division by zero.

7.3 Evaluation

7.3.1 Evaluation Setup

To evaluate the behavior of the model and to test the performance of the methodology proposed in the previous section, a mockup dataset generator was developed. Using this generator, we are able to simulate an overlapped cluster where it is possible to control: i) the number of products and the number of tuples; ii) which tuple is the root cause; iii) the noise/interference that affects the influence of U on the tuples X_i (ϵ_u), iv) the noise that can lead to misclassification (ϵ_l). Figure 7.3 illustrates how both noises, ϵ_u and ϵ_l, are added to the model in Figure 7.1 to test the behavior and sensitivity of the model in different situations.

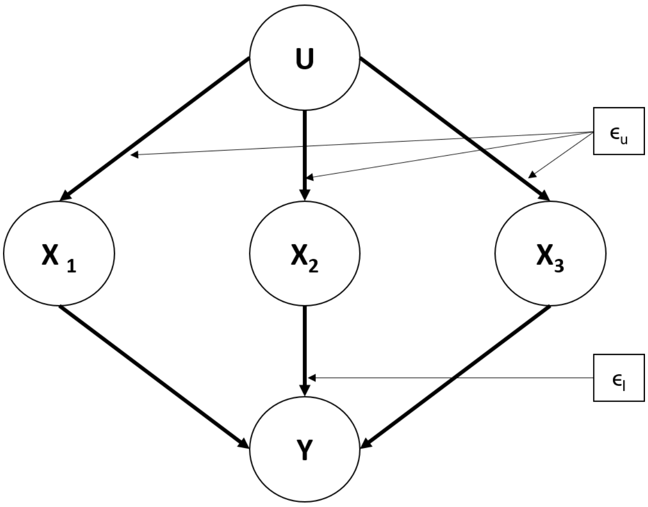


FIGURE 7.3: Causal Model of overlap used to generate mockup datasets.

Figure 7.3 illustrates how both noises, ϵ_u and ϵ_l, are introduced in the system. ϵ_u represents the probability of a product in an overlapped tuple not assuming the same value as U. This is, when $U = 1$, there is a ϵ_u% of products in that tuple that are not equal to one, and vice-versa when $U = 0$. This noise parameter is included to test the limits to the possibility of finding a root cause, defined in Section 7.2. ϵ_u represents the interference in overlap, which is caused by random events that disrupt the local synchronicity in production that is overlap. In the mockup datasets generated, all tuples have the same ϵ_u.

ϵ_l represents the percentage of products that passed through the true root cause tuple X_i^*, but were not faulty. This noise parameter is included to check how resilient is the model to adverse conditions that can impact the performance (e.g., imperfections in labeling process, non-systematic and spurious problems). In the mockup datasets generated, only the true root cause is affected by ϵ_l. The true root cause varies depending on the dataset (it is not always X_2).

For the mockup datasets generated, Table 7.1 summarises the values of the parameters used for the experiments. For each combination of parameters, 25 datasets were generated, and the values of PN, PS, PNS, and PMI were computed. As there are 25 possible combinations of noise repeated 25 times each, a total of 625 datasets were generated.

TABLE 7.1: Parameters' values of the mockup datasets generated.

Parameter	**Value**
Number of Tuples	3
Number of Products	10.000
ϵ_u	$[1e^{-05}, 1e^{-04}, 1e^{-03}, 1e^{-02}, 1e^{-01}]$
ϵ_l	[0.0, 0.1, 0.2, 0.3, 0.4, 0.5]

A small example of a mockup dataset is represented in Table 7.2. This dataset has 5 products, and they can pass through one of the three tuples, represented by columns X_1, X_2, X_3. These tuples are under the influence of overlap U, which makes a product pass through all products or none of them. However, noise or interference may lead to one of the tuples to not be momentarily under the influence of overlap, for example in product 3 and tuple X_1.

TABLE 7.2: Example of a small mockup dataset.

Product	**U**	X_1	X_2	X_3	**Y**
1	0	0	0	0	0
2	1	1	1	1	1
3	1	0	1	1	1
4	0	0	0	0	0
5	1	1	0	1	0

As the mockup datasets have access to the value of U, that value was used to verify if the values of the causal effect obtained through adjustment of direct causes and through the BD criterion were the same. Such was verified and validated. The values of PN, PS and PNS are presented, as well as the PMI values, as benchmark. In addition, each Mockup dataset represents a dataset where the relevant features have been already selected using an approach existing in the literature, for example PMI.

In addition to the Mockup datasets generated, the proposed methodology is also tested on the Stochastic Simulation data and Real Case Study data, used in the previous chapters. Table 7.3 presents a reminder of the metadata of the Simulated and case study datasets, which are different from the Mockup datasets.

TABLE 7.3: Metadata of the Simulated and case study datasets.

Metadata	Simulated	Case Study
Num. Datasets	3	1
Num. Moments	[4;5;4]	16
Num. Steps	22	22
Num. Machines	63	62
Num. Products	5004	6144

These four datasets consist in three simulated datasets and a real one. In the simulated datasets the root causes are clearly identified, but not in the real dataset. Moments in the dataset that contain systematic problems are automatically identified using Exponentially Weighed Moving Average (EWMA) algorithm. In the three simulated datasets, a total of 13 moments were identified (divided among the three dataset as [4,5,4]), and in the case study dataset 16 moments were identified. In each of the moments identified, PMI is used to select the tuples, in order to keep the ones that have the highest overlap with the label. Only the tuples that share the maximum overlap with the label were considered for causal inference analysis. This is a major difference in relation to the Mockup datasets. These last experiments are done not only to test the proposed methodology, but also to identify how "pure" is the overlap in a real world dataset.

7.3.2 Evaluation Results

Mockup

With the objective of testing how sensitive are the proposed model and the approach to noise, a total of 625 datasets were generated, and the PN, PS, and PNS queries are tested in them. The PMI method, explored in a previous work is also tested as a benchmark. The interventional queries and PMI values are computed for all the three tuples, and then these values are checked to see if the true root cause node is the one with the single highest value. If such happens, the root cause is correctly detected.

The results are presented in Tables 7.4 and 7.5, where it is possible to see the variation in ϵ_u and ϵ_l, respectively. The values within each cell represent the percentage of datasets where the root cause was correctly detected when using one of the interventional queries or PMI.

TABLE 7.4: Results of the interventional queries proposed and of the PMI method, when varying the values of ϵ_u.

ϵ_u	PS	PN	PNS	PMI
$1,00E-05$	10%	10%	10%	10%
$1,00E-04$	**66%**	58%	62%	**66%**
$\geq 1,00E-03$	100%	100%	100%	100%

From the analysis of Table 7.4, we can say the noise/interference ϵ_u has a major impact on the performance of all methods. When the interference is 1,00E-03 or above, the influence of overlap no longer constrains the methods and they are able to detect all root causes. As ϵ_u goes below that level, overlap becomes an issue, because the interference threshold (mentioned in Section 7.2) is not surpassed in many datasets, preventing the causal effect from being computed.

Another aspect is that, when the threshold is surpassed, and interventional queries can compute the causal effect, the PMI approach is also able to detect the root cause. This seems to indicate that both methods are valid for the detection of root causes, as long as the interference threshold is surpassed.

TABLE 7.5: Results of the interventional queries proposed and of the PMI method, when varying the values of ϵ_l.

ϵ_l	**PS**	**PN**	**PNS**	**PMI**
$0,1$	**78%**	77%	77%	77%
$0,2$	76%	72%	75%	**78%**
$0,3$	**75%**	74%	74%	74%
$0,4$	74%	72%	74%	**75%**
$0,5$	72%	72%	72%	72%

From the results presented in Table 7.5, it is possible to see that ϵ_l has little to no influence on the root cause detection performance of all methods. It is true that there is a slight decline in performance as the noise is increased, but that could be anticipated, as when the noise surpasses certain levels the label starts losing its meaning. In fact, the surprise lies in how little the effect is when only 50% of the products that go through the root cause tuple are in fact problematic. This indicates the resilience of the models tested to noisy datasets and errors in labeling.

Although not expressed in the tables presented, it is important to reference a certain behavior of the results on a finer level of analysis. It can be concluded from what is said above that ϵ_u interferes with the identifiability of the causal effect. Albeit true that ϵ_l does not seem to interfere with this capacity for identification of the causal effect, the truth is that it affects the magnitude of such causal effect. The greater ϵ_l is, the lower the value of the causal effects computed, both of the true root cause, and of the other tuples. For example, for the datasets with $\epsilon_u = 1,00E - 01$, the average causal effect of the true root cause of all datasets when $\epsilon_l = 0.1$ is 0.90, and when $\epsilon_l = 0.5$, is 0.50. For the non-root cause tuples, the average is 0.45 and 0.25, respectively. This means that the difference between the interventional probabilities of root cause tuples and the others is 0.45 when $\epsilon_l = 0.1$, and 0.25 when $\epsilon_l = 0.5$. As such, the difference between the estimated effect on quality of passing through the root cause tuple and the other tuples is reduced.

This behavior, that here occurs in a controlled environment with mockup datasets, was already observed in the previous chapter, namely at the end of Section 6.3.2, where we already speculated that the need to lower the PMI threshold had to do with noise (defined as randomness in the labeling of the products). There is the belief that the behaviour mentioned in this paragraph is evidence that such interpretation was indeed correct, and that randomness in labeling affects the magnitude of the causal effect one is able to compute. This randomness in labeling may come either from human/machine error, or because other small, random problems occur, aside the main problem caused by the root cause.

In terms of comparing the new approach to the benchmark (PMI), it is possible to see that PS has the best performance in two levels of ϵ_l (0.1 and 0.3), while PMI has the best performance in another two (0.2 and 0.4), with a tie between both approaches (0.5), meaning that there are no significant differences in performance between the two approaches.

Stochastic Simulation & Real Case Study

When analysing the datasets based on the case study, only two outcomes resulted from each of the moments identified: i) a group of possible root causes was identified by the PMI filter, but the interventional queries were not able to distinguish the individual influence of each of the tuples; ii) the PMI filter was able to distinguish a single root cause node. In the Stochastic Simulation datasets, all root causes were correctly detected.

Table 7.6 presents the results on these datasets, based on the outcomes described above. Each dataset is divided in the problematic moments identified, which are then analysed (there were no moments without problems). The three Stochastic Simulation datasets have a total of 13 moments identified, while the Real Case Study dataset has a total of 16 moments identified.

TABLE 7.6: Results from the Simulation and case study datasets.

Dataset	I) Multiple RC Detected	II) Single RC Detected
Stochastic Simulation	38%	62%
Real Case Study	50%	50%

There is confirmation of the result that is obtained in the experiments with the Mockup datasets: when the interference threshold is surpassed, PMI is enough to identify the root cause. The interventional queries reach the same results as the PMI. However, even without an added discriminatory power by the interventional queries, more than half of the moments have a single root cause detected. Even in the moments where multiple root causes are detected, the number of tuples to consider is still greatly reduced.

Additionally, although not presented in Table 7.6, in what concerns the magnitude of the effect in the presence of noise (as mentioned in the previous section), some moments in the case study dataset have a PNS causal effect value of only 0.166, revealing these moments to be very noisy. It is open to discussion if we should even accept such low values of causal effect as indicative of the presence of root causes.

7.4 Discussion

In this section the results of the different experiments, and how these combine to form coherent conclusions are discussed and summarised.

In the Mockup experiments, the proposed approach achieves a performance comparable to the PMI approach proposed in the previous chapter, both when varying ϵ_u and ϵ_l. The results also indicate that ϵ_u affects the identifiability of the true root cause, as hypothesised, while ϵ_l, affects the magnitude of the causal effect, or how low the threshold needs to be to capture the true root cause. These results provide further evidence of the hypothesis proposed at the end of Section 6.3.2.

The results of the experiments with the Stochastic simulation and Real case study data reinforce that, when the conditions are met so that the true root cause can be identified, the PMI approach has the same performance as the one proposed in this chapter.

From the results of all experiments, it is possible to conclude that the performance of the approach proposed in this chapter and the PMI approach proposed in the previous chapter are similar, and these are able to identify the true root cause, as long as the conditions identified in Section 7.2 are met. It was also possible to present further evidence that supports previous hypothesis on the effects of noise on the performance of ARCA solutions.

7.5 Conclusions and Future Research

This chapter presents a different perspective on the issue of overlap in Automatic Root Cause Analyis (ARCA). It attempts to “untangle” the effect of overlap, allowing the analysts to clearly identify the root cause from a group of overlapped variables.

It proposes the use of causal inference to compute the causal effect of a product going through a machine in a certain step (also known as tuple), as causal inference allows us to estimate the effect of a product passing through a certain machine while disregarding the effect of overlap, in certain conditions. To do so, a causal model of overlap is proposed. From this model, it is possible to determine the conditions in which it is possible to determine the true root cause in the presence of overlap. This minimal condition is that the dataset contains at least $N - 1$ instances that do not have a perfect overlap, with N being the total number of overlapped tuples. This allows for the development of ARCA solutions that can automatically understand when is the overlap too restricting to find the single true root cause, and adapt accordingly.

In addition, interventional queries regarding the different aspects of causation - Probability of Necessity (PN), of Sufficiency (PS), and of both (PNS) - are used to compute the causal effect of a product going through a certain tuple. This approach is validated in mockup datasets generated, each with different levels of noise/ interference, both in the influence of overlap and in the labeling process. Results confirm that if overlap is too "pure", it is not possible to identify the causal effects. It is also clear that, when this is possible, the Positive Mutual Information (PMI) approach is also able to detect the root cause correctly. In addition to this, there is evidence that random noise in the labeling does not affect the capacity for the identification of causal effects, but that an increasing level of noise leads to a reduced magnitude of the causal effects of the true root cause.

In addition, the proposed approach is validated in real datasets from a case study in semiconductor manufacturing and stochastically simulated datasets based on the case study, which showed further evidence of the conclusions reached in the mockup experiments: the PMI approach is able to detect the root cause by itself once the theoretical limit has been surpassed, and that noise does decrease the magnitude of the causal effect.

The contributions of this chapter are: i) a causal model of overlap, which further improves our understanding of it, ii) a clear definition of the conditions in which it is possible to locate root causes in the presence of overlap, and iii) it provides further evidence on the effect of noise in locating root causes.

Chapter 8

Conclusions and Future Research

The thesis that now concludes puts forward the issue of overlap, which occurs when performing diagnosis with Location-Time data from a manufacturing process. To develop solutions that are resistant to overlap, three measures are proposed, with an increasing refinement of the aspects that define overlap. The first measures overlap as the strength of association between factors, and is based the chi-square test and Cramér's V. The second one measures overlap as the positive mutual information between two factors. This measure improved the sensitiveness of the measurements. The third measure is based on causal inference, and is based on a causal model of overlap, allowing for the computation of interventional probabilities that can determine the true root cause. These probabilities also allowed for the definition of the limit after which overlap makes the detection of the true root cause impossible. These three measures led to the development of different solutions that are resilient to overlap, allowing for the discovery of root causes, or at the very least enabling the identification of a reduced group of possible root causes. The solutions based on information theory and causal inference had the best results.

This chapter presents an overview of this doctoral work, reviewing its contributions to the literature and practitioners, and summarizing the answers to the research questions made in the introduction. The limitations and directions for future research are also discussed.

8.1 Research Question Conclusions

This thesis focused on proposing efficient and robust ARCA solutions for manufacturing that enable to deal with diverse scenarios. In particular, it focused on the study of solutions when only Location-Time data is available, i.e., logistical data of the manufacturing process. During the research, it became evident the presence of an issue that can prevent the detection of the true root cause, which was denominated overlap. Overlap occurs when all products that go through a certain machine in a manufacturing step are all processed in a given machine in a later manufacturing step. Such situation results in datasets where it is very hard to distinguish the influence of each individual machine on the quality of a product, preventing the detection of the root cause.

Given the general objectives stated above, the specific conclusions reached through this doctoral work are now presented, divided by each research question mentioned in Chapter 1.

> **R.Q. 1**: How can we structure and conceptualize the solutions/approaches on ARCA in manufacturing?

This research question is answered in Chapter 4, which presents an overview and conceptualization of the existing ARCA approaches in manufacturing. In Chapter 4, the approaches are divided according to three perspectives: i) types of data, ii) methodologies and techniques, and iii) evaluation of the solutions.

In terms of types of data, ARCA approaches can work with three types: i) Location-Time; ii) Physical, iii) Log-Action. Location-Time type of data describes at which time and location (which machines) a product has gone through, allowing us to detect where the root cause is located. Physical type of data concerns all factors that describe physical influences on the products, and with this type of data it is possible to determine what the root cause is physically. The Log-Action type of data describes the actions that were performed on machines and other equipment of the manufacturing process, allowing for a more refined level of knowledge about the root cause, as it is possible to know why the physical parameters deteriorated in the first place.

In terms of methodologies, the approaches can be divided into three categories, according to how the root causes are extracted from the algorithms. In the **Factors** ⇒ **Problem** category, a technique is used to associate the different factors with a problem/fault, and the root cause is extracted from the factors that the classifier considers most relevant to determine whether there is a problem or not. In the **Factors** ⇒ **RC** category, a technique associates factors directly to a root cause. Finally, the solutions in the **RC** ⇒ **RC** category focus on finding relations between root causes, in order to determine the root causes that should be prioritized.

For evaluating the ARCA approaches, a diverse group of measures are used in the literature. These measures were divided into classification measures (as they are used to evaluate classifiers' performance in data mining literature), adaptation of classification measures to evaluate the performance in root cause detection, ranking methods, and validation through expert analysis.

> **R.Q. 2**: What is overlap, and how can it be defined and measured?

Overlap proved itself to be a complex phenomenon to characterize and measure. Although it is simple to perceive overlap, how to translate all its aspects into a mathematical definition proved to be complex.

In Chapter 5, a first attempt to mathematically measure overlap was made. It focused on a DM/ML approach, which was based on the notion of strength of association between two nominal factors and is based on Crámer's V. As the root cause's location can be determined from datasets of nominal factors, it is possible to determine overlap between two factors by testing the association between them. It was concluded that, in the presence of overlap, the use of classifiers can lead to the overlook of possible root causes. To avoid this, factor ranking methods should be used.

In Chapter 6, two needs for improvement of the measure presented on the previous chapter were identified: i) it did not take into consideration if the association is positive or negative, and ii) the definition focused on comparing steps and not tuples (Step-Machine pairs). To satisfy these needs, a new measure based on Information Theory concepts was proposed. Positive Mutual Information (PMI) equates to overlap, and represents the amount of information about a factor it is possible to know using another factor, but taking into consideration only the positive associations. This new measure was a successful improvement, presenting a solid mathematical definition. In addition to the information theory-based one, a probabilistic definition was mentioned, namely, that overlap is the very high probability of a product going through a certain machine given that we know that it went through another machine in another step. The approaches developed based on this new measure performed better than the ones based on the Crámer's V measure.

However, the definition and measure of overlap based on PMI, although capturing the relation of overlap between factors/tuples, still did not capture the essence of overlap, i.e., how overlap, as an issue exogenous to the manufacturing process, influences the endogenous variables, i.e. the factors/tuples. This is, that the definition considers overlap as a source of "synchronicity" in the manufacturing process. Chapter 7 focuses on this aspect and uses causal inference to analyse overlap. A causal model of overlap was developed, where overlap is represented as an influence on some variables of the manufacturing process, that makes all products that go through a specific tuple, to go through all other tuples this overlap affects. This causal model is also used to compute interventional queries that can be used to measure the causal effect of a product going through a certain tuple. This final definition of overlap captures all aspects of overlap considered relevant when dealing with this issue.

> **R.Q. 3**: What challenges does overlap bring to the development of ARCA solutions using Location-Time data?

The final research question is answered in different ways by the work presented in different chapters. As a starting point, before considering overlap, Chapter 4 concludes, through an analysis of the existing literature, that, when using only Location-Time data, the finest level of detail possible of detecting the root cause is determining its location. In other words, it is only possible to identify the Step-Machine tuple where the problem occurred. To identify the physical problem, or why there was such a physical alteration, we require other types of data.

In Chapter 5, overlap is for the first time identified as a challenge to the development of ARCA solutions using Location-Time data. Overlap poses an increased difficulty to this kind of solutions, not only because the solutions are restricted to just finding the location of the root cause, but, in the presence of overlap, they cannot find the single true root cause. Instead, they can at most reduce the number of possible locations to the true location and the others which overlap the true location. One particular point of discussion is the use classifiers in ARCA solutions. Overlap is an example of problems that can arise if DM and ML techniques are applied without taking into consideration the original objective of the techniques used and the objective of the problem one is trying to solve. This is originally mentioned in Chapter 2, but most DM and ML techniques were developed with an emphasis on descriptive and predictive analytics and, when applying them to a diagnosis problem, some caution should be exercised. To apply classification techniques to a diagnosis problem where overlap is present means excluding factors that may be the root cause, leading to a wrong conclusion, as explained in more detail in Chapter 5.

To determine exactly what is the overlap limit that prevents finding the true root cause , it was necessary to develop a causal model of overlap in Chapter 7. This model allowed the identification of the limit on the quantity of overlap that can be present in a group of tuples, before it is impossible to distinguish their influence on the final quality of the product. It was possible to establish that the limit is an absolute number of instances, and that the number of non-overlapped instances that are required to be able to detect the single true root cause depends on the number of overlapped factors related to the label.

The overlap limit after which it is not possible to identify the true root cause is clearly defined as: it is possible to identify the true root cause of a problem from a group of overlapped tuples when there is at least one product that passed through that tuple, without passing by the other tuples in the group, for each of the tuples in the group. Meaning that if there are at least $N - 1$ products that differ from the imposition of overlap, it is possible to identify the true root cause from among the group of overlapped tuples.

This limit may seem very permissive, but the experiments done on real datasets show that the overlap existing in these situations can be so “pure” (i.e., there is no difference between the tuples) that, in practice, it is impossible to distinguish the influence of the overlapped tuples.

8.2 Contributions

This thesis makes five main contributions to both the literature and practice of RCA in manufacturing.

First, Chapter 4 presents an overview of the literature on ARCA solutions for manufacturing, and provides a conceptualization that divides the literature and identifies its main aspects and avenues of research. This effort was never done before, and it presents a valuable contribution in standardizing terminology and concepts, which are often presented in distinct ways in different studies, but actually mean the same.

The second main contribution, is the identification of overlap, which is an issue that occurs when analysing data for RCA in manufacturing processes. This is a complex and critical issue, that represents a serious limitation of performing ARCA with Location-Time data, and was never identified before. Overlap is measured with increasing level of detail in the chapters of this thesis, based on different concepts. The first way to measure overlap, based on Crámer's V, is based on the first perception that factors closely related could be considered overlapped. This allows for measuring the overlap between factors, and produces solutions that were robust to overlap. It is also discussed in Chapter 5 if this measure should be based on averages or counting the number of factor interactions that were overlapped. Counting the number of interactions proved to be more stable and more aligned with how overlap is perceived. The second measure, based on information theory, is motivated by a need to improve results even further, and because the realization that overlap as a pernicious issue was only problematic when the positive association between factors was significant. As such, Positive Mutual Information is used, which leads to more refined measurements, and better performance of the approach developed based on this measure. The third way to measure overlap is motivated by the attempt of detecting the true root cause from a list of overlapped locations that are possible root causes. This final measure is based on a graphical definition of overlap developed taking into consideration causal inference principles, and the measure is a probability based on *do* calculus. The contributions of this last measure are mostly theoretical, as results were similar to the ones obtained with PMI, but allowed for the understanding of overlap as an variable exogenous to the manufacturing process, that influences it in a way that is harmful to diagnosis.

Third, Chapters 5, 6, and 7 propose several ARCA solutions that use only Location-Time data, with a **Factors** $\Rightarrow$ **Problem** methodology. These solutions are evaluated in different simulated and real datasets (using ranking evaluation measures), and the results are positive, with the proposed approaches being able to identify the groups of factors/tuples that are the most likely root causes. These solutions could be applied in any manufacturing context where the data type Location-Time type data is available. These solutions are able to identify the presence of overlap, and are robust to it, presenting the true root causes when such is possible, and greatly reducing the number of possible root causes when it is not.

Fourth, not only does this doctoral work identify and measure overlap, in Chapter 7 it clearly defines, from the causal model, the conditions of how "pure" the overlap can be before it becomes impossible to detect the root cause. This was also proved on an empirical basis, and noted that, when the conditions are met, both the PMI approach defined in Chapter 6 and the approach based on causal inference (Chapter 7) are able to correctly identify the true root cause.

Fifth and final contribution, also in Chapter 7, is the empirical conclusions on the effect of noise on the identifiability and magnitude of the causal effect. Interference on the effect of overlap leads to a greater identifiability. Random noise does not interfere with this identifiability. However, it does interfere with the magnitude of the causal effect. The greater the random noise, the less the magnitude of the causal effect.

Overall, this doctoral work is relevant to the literature and practitioners as:

- it structures the existent literature on automatic root cause analysis in manufacturing, conceptualizing it, and providing guidance for future developments of solutions of this kind.
- it identifies an issue, that we named overlap, that has never been discussed before, and that has a great impact when performing automated diagnosis with a specific type of data.
- it proposes automatic solutions that are robust to overlap, and can achieve a correct diagnosis, or at least greatly reducing the number of possibilities to consider.
- it clearly identifies the overlap limit that makes it impossible to detect the true root cause of a problem.
- it studies the effects of noise in the development of ARCA solutions using Location-Time data.

8.3 Managerial Insights

For practitioners, this thesis has three main insights that can be useful when trying to develop an Automatic Root Cause Analysis (ARCA) solution in manufacturing.

First, when developing an ARCA solution, it is important to frame it relation to three dimensions (as detailed in Chapter 4):

- **Type of data:** The type of data establishes what kind of root cause it is possible to determine. If there is only data about the location and time products went through the manufacturing step, it is only possible to determine the location of the root cause. If there is data about the physical attributes of the process (e.g., temperature, voltage), it is possible to determine what physically happened that caused the problem. With maintenance data, it is possible to go beyond this, and define the human action that led to the physical change that caused the problem.

- **Methodology:** It is necessary to select the methodology, and especially how the root cause is extracted from that methodology. For that it is needed to consider if there is access to data labeled with a problem (e.g., abnormal decrease in certain KPI), or if there is also access to information on what the root causes actually were for certain problems. If there is just access to the first type of label, the solution will most likely fit the **Factors** ⇒ **Problem**, where the root cause is extracted from the factors themselves. With the labels about the root causes, it is possible to develop a solution that is framed within the **Factors** ⇒ **RC**
  category, where the technique will determine the root cause as in a traditional classification problem. Furthermore, if the objective is to understand how the root causes influence each other, and determine which should be prioritised, a solution of the category **RC** ⇒ **RC** can be developed to meet that objective.
- **Evaluation Measure:** In order to develop a robust ARCA solution, proper validation must be conducted. As such, it is necessary to select what type of measure is used to evaluate it. In Chapter 4, four types of evaluation measures are identified: 1) classification measures, 2) adaptation of classification measures to evaluate the performance in root cause detection, 3) ranking methods, and 4) validation through expert analysis.

Second, it is necessary to take into consideration the different existing types of analytics (see final discussion of Section 2.3), and understand that there may be discrepancies between the original goal of that technique and the problem it is trying to solve. In the particular problem tackled in this doctoral work, which is framed within diagnostic analytics, most DM and ML techniques need to be used with caution, as they were developed with the objective of being used in descriptive or predictive analytics. Overlap is an example of an issue that can be exacerbated by the incautious application of DM and ML techniques in diagnosis. Techniques that focus on prediction, i.e., classifiers, hide factors that are overlapped and can be the root cause, leading to the wrong diagnosis.

Third, when developing an ARCA solution for manufacturing that uses Location-time data, and is of the category **Factors** ⇒ **Problem**, overlap can be a critical problem, and the solution has to take it into consideration. In this thesis, several solutions that can identify overlap and are robust to it were presented. If there is the need to select one for implementation as quickly as possible, the solution based on information-theory concepts, presented in Chapter 6 is preferred. This solution is one of the solutions with the best performance, and is easier to understand and implement. This solution provides a short list of the most likely root causes, even if they are overlapped, and can even determine the single true root cause, if conditions allow it. To see if the overlap existing in the dataset being analysed allows for the identification of a single true root cause, it is possible to use the conclusions of Chapter 7. A single instance per overlapped factor is required to not be overlapped. If such a condition is met, it is possible to determine the single true root cause.

8.4 Limitations & Future Research

This doctoral work has as limitations:

- The dataset from the real case study does not have the labels that identify the root causes. To counter this limitation, stochastic simulation was used to emulate different scenarios, based on the case study, where it is possible to define the root causes. In addition, the developed approaches were compared for commonalities in the real case study. Nevertheless, this work would be more robust if these labels were obtained, and future work should do so.
- Related to the first point, this doctoral work is based exclusively in a single case study in semiconductor manufacturing. Although all formulations and developments assumed a generic scenario, there is still the possibility that certain idiosyncrasies of the sector might have had an influence on the development of the proposed solutions. As such, it would be significant to further validate what is proposed in this work in other manufacturing sectors.
- Throughout the thesis, and especially in Chapter 7, it is assumed that a single true root cause exists. In reality, this may not be the case, as root causes may be multiple (i.e., several possible root causes for the same problem), and they may even be compounded (i.e., require both root causes to exist in order for the problem to occur). Although the solutions were developed with this assumption, the experiments performed to evaluate their validity considered the possibility of simultaneous root causes. The solutions still showed capacity to detect multiple root causes. However, future work can place more emphasis on the possibility of multiple root causes.

Taking into consideration the above-mentioned limitations, in terms of future research, the first step that could be taken would be to test the developed approaches in more real datasets, from a broader array of manufacturing industries. This would increase the external validity of this thesis. It would also be useful to obtain a real world dataset where the root causes are clearly identified, to enable a more robust validation.

It would also be interesting to see if the conceptualizations and conclusions of this thesis, as well as the proposed methodologies work in other contexts.

Another improvement that could be done to the causal model presented in Chapter 7 is to consider the possibility of more complex functional causal models. The model used considers only a single root cause, with the possibility of several root causes having similar causal effect. However, this could be improved to consider multiple root causes explicitly, joined either by an "AND", an "OR", or an "XOR" (exclusive or) operator.

Bibliography

Agrawal, Rakesh, Tomasz Imielinski, and Arun Swami (1993). “Mining Association Rules between Sets of Items in Large Databases”. In: pp. 207–216.

Agrawal, Rakesh and Ramakrishnan Srikant (1994). “Fast Algorithms for Mining Association Rules in Large Databases”. In: *Proceedings of the 20th International Conference on Very Large Data Bases*. VLDB ’94. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 487–499. ISBN: 1558601538.

Agrawal, Vedika, BK Panigrahi, and PMV Subbarao (2016). “Intelligent decision support system for detection and root cause analysis of faults in coal mills”. In: *IEEE Transactions on Fuzzy Systems* 25.4, pp. 934–944.

Ahn, Gilseung et al. (2019). “A Graphical Model to Diagnose Product Defects with Partially Shuffled Equipment Data”. In: *Processes* 7.12. ISSN: 2227-9717. DOI: 10.3390/pr7120934.

Alcácer, Vitor and Virgilio Cruz-Machado (2019). “Scanning the industry 4.0: A literature review on technologies for manufacturing systems”. In: *Engineering Science and Technology, an International Journal* 22.3, pp. 899–919.

American Society for Quality (2019a). *American Society for Quality*. Accessed on 06/07/2021. URL: http://www.asq.org/.

– (2019b). *Failure Mode and Effects Analysis (FMEA)*. Accessed on 06/07/2021. URL: https://asq.org/quality-resources/fmea.

– (2019c). *Fishbone Diagram*. Accessed on 02/08/2021. URL: https://asq.org/quality-resources/fishbone.

– (2019d). *Five Whys and Five Hows*. Accessed on 02/08/2021. URL: https://asq.org/quality-resources/five-whys.

Asawachatroj, Anukoon and David Banjerdpongchai (July 2012). “Analysis of Advanced Process Control Technology and Economic Assessment Improvement”. In: *Engineering Journal* 16, pp. 1–4. DOI: 10.4186/ej.2012.16.4.1.

Barkia, Hasna et al. (2013). “Semiconductor yield loss’ causes identification: A data mining approach”. In: *Industrial Engineering and Engineering Management (IEEM), 2013 IEEE International Conference on*. IEEE, pp. 843–847.

Berkson, Joseph (1978). “In dispraise of the exact test: Do the marginal totals of the 2X2 table contain relevant information respecting the table proportions?” In: *Journal of Statistical Planning and Inference* 2.1, pp. 27–42. ISSN: 0378-3758. DOI: https://doi.org/10.1016/0378-3758(78)90019-8. URL: https://www.sciencedirect.com/science/article/pii/0378375878900198.

Bland, J Martin and Douglas G Altman (1995). “Multiple significance tests: the Bonferroni method”. In: *BMJ* 310.6973, p. 170. ISSN: 0959-8138. DOI: 10.1136/bmj.310.6973.170. eprint: https://www.bmj.com/content/310/6973/170.full.pdf. URL: https://www.bmj.com/content/310/6973/170.

Breiman, Leo (2001). “Random forests”. In: *Machine learning* 45.1, pp. 5–32.

Brun, Armelle, Sylvain Castagnos, and Anne Boyer (2009). “A positively directed mutual information measure for collaborative filtering”. In: *2nd International Conference on Information Systems and Economic Intelligence - SIIE 2009*. Malek Ghenima (ESCE Université la Manouba - Tunisie) and Sahbi Sidhom (Nancy Université - France), pp. 943–958.

Chemweno, P. et al. (2016). “i-RCAM: Intelligent expert system for root cause analysis in maintenance decision making”. In: *2016 IEEE International Conference on Prognostics and Health Management (ICPHM)*, pp. 1–7. DOI: 10.1109/ICPHM.2016.7542830.

Chen, Wei-Chou, Shian-Shyong Tseng, and Ching-Yao Wang (2005). “A novel manufacturing defect detection method using association rule mining techniques”. In: *Expert systems with applications* 29.4, pp. 807–815.

Chiang, Leo H et al. (2015). “Diagnosis of multiple and unknown faults using the causal map and multivariate statistics”. In: *Journal of Process Control* 28, pp. 27–39.

Chien, C. and S. Chuang (2014). “A Framework for Root Cause Detection of Sub-Batch Processing System for Semiconductor Manufacturing Big Data Analytics”. In: *IEEE Transactions on Semiconductor Manufacturing* 27.4, pp. 475–488. ISSN: 1558-2345. DOI: 10.1109/TSM.2014.2356555.

Chien, Chen-Fu, Tzu yen Hong, and Hong-Zhi Guo (2017). “A Conceptual Framework for “Industry 3.5” to Empower Intelligent Manufacturing and Case Studies”. In: *Procedia Manufacturing* 11. 27th International Conference on Flexible Automation and Intelligent Manufacturing, FAIM2017, 27-30 June 2017, Modena, Italy, pp. 2009 –2017. ISSN: 2351-9789. DOI: https://doi.org/10.1016/j.promfg.2017.07.352.

Chien, Chen-Fu, Chia-Yu Hsu, and Pei-Nong Chen (2013). “Semiconductor fault detection and classification for yield enhancement and manufacturing intelligence”. In: *Flexible Services and Manufacturing Journal* 25.3, pp. 367–388.

Chien, Chen-Fu, Yun-Siang Lin, and Sheng-Kai Lin (2020). “Deep reinforcement learning for selecting demand forecast models to empower Industry 3.5 and an empirical study for a semiconductor component distributor”. In: *International Journal of Production Research* 58.9, pp. 2784–2804. DOI: 10.1080/00207543.2020.1733125.

Chien, Chen-Fu, Chiao-Wen Liu, and Shih-Chung Chuang (2017). “Analysing semiconductor manufacturing big data for root cause detection of excursion for yield enhancement”. In: *International Journal of Production Research* 55.17, pp. 5095–5107. DOI: 10.1080/00207543.2015.1109153.

Choudhary, Alok Kumar, Jenny A Harding, and Manoj Kumar Tiwari (2009). "Data mining in manufacturing: a review based on the kind of knowledge". In: *Journal of Intelligent Manufacturing* 20.5, p. 501.

Crámer, Harald (1999). *Mathematical Methods of Statistics (PMS-9)*. Princeton University Press. ISBN: 9780691005478.

Dempster, A. P., N. M. Laird, and D. B. Rubin (1977). "Maximum Likelihood from Incomplete Data Via the EM Algorithm". In: *Journal of the Royal Statistical Society: Series B (Methodological)* 39.1, pp. 1–22. DOI: https://doi.org/10.1111/j.2517-6161.1977.tb01600.x. eprint: https://rss.onlinelibrary.wiley.com/doi/pdf/10.1111/j.2517-6161.1977.tb01600.x. URL: https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/j.2517-6161.1977.tb01600.x.

Detzner, Alexander and Martin Eigner (2021). "Feature selection methods for root-cause analysis among top-level product attributes". In: *Quality and Reliability Engineering International* 37.1, pp. 335–351. DOI: https://doi.org/10.1002/qre.2738.

Djelloul, Imene, Zaki Sari, et al. (2018). "Fault diagnosis of manufacturing systems using data mining techniques". In: *2018 5th International Conference on Control, Decision and Information Technologies (CoDIT)*. IEEE, pp. 198–203.

Donauer, Michael, Paulo Peças, and Americo Azevedo (2015). "Identifying nonconformity root causes using applied knowledge discovery". In: *Robotics and Computer-Integrated Manufacturing* 36. Sustaining Resilience in Today's Demanding Environments, pp. 84 –92. ISSN: 0736-5845. DOI: 10.1016/j.rcim.2014.12.012.

Du, Shichang, Jun Lv, and Lifeng Xi (2012). "A robust approach for root causes identification in machining processes using hybrid learning algorithm and engineering knowledge". In: *Journal of Intelligent Manufacturing* 23.5, pp. 1833–1847.

Dunn, Olive Jean (1961). "Multiple Comparisons among Means". In: *Journal of the American Statistical Association* 56.293, pp. 52–64. DOI: 10.1080/01621459.1961.10482090. eprint: https://www.tandfonline.com/doi/pdf/10.1080/01621459.1961.10482090. URL: https://www.tandfonline.com/doi/abs/10.1080/01621459.1961.10482090.

Ester, Martin et al. (1996). "A density-based algorithm for discovering clusters in large spatial databases with noise". In: AAAI Press, pp. 226–231.

European Commission (2020). URL: https://ec.europa.eu/eurostat/statistics-explained/index.php/Manufacturing_statistics_-_NACE_Rev._2#Sectoral_analysis.

Fan, Shu-Kai S, Shou-Chih Lin, and Pei-Fang Tsai (2016). "Wafer fault detection and key step identification for semiconductor manufacturing using principal component analysis, AdaBoost and decision tree". In: *Journal of Industrial and Production Engineering* 33.3, pp. 151–168.

Gins, Geert et al. (2015). "Improving classification-based diagnosis of batch processes through data selection and appropriate pretreatment". In: *Journal of Process Control* 26, pp. 90–101.

Gomez-Andrades, Ana et al. (2016[a]). “Automatic Root Cause Analysis Based on Traces forR LTE Self-Organizing Networks”. In: *IEEE Wireless Communications* 23.3, 20–28. ISSN: 1536-1284. DOI: {10.1109/MWC.2016.7498071}.

Gomez-Andrades, Ana et al. (2016[b]). “Automatic Root Cause Analysis for LTE Networks Based on Unsupervised Techniques”. In: *IEEE Transactions on Vehicular Technology* 65.4, 2369–2386. ISSN: 0018-9545. DOI: {10.1109/TVT.2015.2431742}.

Guyon, Isabelle and André Elisseeff (Mar. 2003). “An Introduction to Variable and Feature Selection”. In: *Journal of Machine Learning Research* 3.null, 1157–1182. ISSN: 1532-4435.

Han, Jiawei, Jian Pei, and Yiwen Yin (2000). “Mining Frequent Patterns without Candidate Generation”. In: ACM Press, pp. 1–12.

He, Yihai et al. (2017). “Big data oriented root cause identification approach based on Axiomatic domain mapping and weighted association rule mining for product infant failure”. In: *Computers Industrial Engineering* 109, pp. 253 –265. ISSN: 0360-8352. DOI: https://doi.org/10.1016/j.cie.2017.05.012.

He, Zengyou, Xiaofei Xu, and Shengchun Deng (2002). “Squeezer: An efficient algorithm for clustering categorical data”. In: *Journal of Computer Science and Technology* 17, pp. 611–624.

Hessing, Ted (2020). *Fault Tree Analysis*. Accessed on 06/07/2021. URL: https://sixsigmastudyguide.com/fault-tree-analysis/.

Hessinger, Uwe, Wendy K Chan, and Brett T Schafman (2014). “Data mining for significance in yield-defect correlation analysis”. In: *IEEE Transactions on Semiconductor Manufacturing* 27.3, pp. 347–356.

Hoerl, Arthur E. and Robert W. Kennard (1970). “Ridge Regression: Biased Estimation for Nonorthogonal Problems”. In: *Technometrics* 12.1, pp. 55–67. DOI: 10.1080/00401706.1970.10488634. eprint: https://www.tandfonline.com/doi/pdf/10.1080/00401706.1970.10488634. URL: https://www.tandfonline.com/doi/abs/10.1080/00401706.1970.10488634.

Hsu, Shao-Chung and Chen-Fu Chien (2007). “Hybrid data mining approach for pattern extraction from wafer bin map to improve yield in semiconductor manufacturing”. In: *International Journal of Production Economics* 107.1. Special Section on Building Core-Competence through Operational Excellence, pp. 88 –103. ISSN: 0925-5273. DOI: 10.1016/j.ijpe.2006.05.015.

Industry Today (2020). *Value of Data Increasing Exponentially in Manufacturing*. Accessed on 15/07/2021. URL: https://industrytoday.com/value-of-data-increasing-exponentially-in-manufacturing/.

Janitza, Silke, Carolin Strobl, and Anne-Laure Boulesteix (2013). “An AUC-based permutation variable importance measure for random forests”. In: *BMC bioinformatics* 14.1, p. 119.

Juran (2018). *Guide to Failure Mode and Effect Analysis – FMEA*. Accessed on 06/07/2021. URL: https://www.juran.com/blog/guide-to-failure-mode-and-effect-analysis-fmea/.

Kambilonje, Vitumbiko (2020). *How to Use 5 Common Root Cause Analysis Tools*. Accessed on 23/07/2021. URL: `https://tulip.co/blog/root-cause-analysis-tools/`.

Kitcharoen, Nopparoot et al. (2013). "RapidMiner framework for manufacturing data analysis on the cloud". In: *The 2013 10th International Joint Conference on Computer Science and Software Engineering (JCSSE)*. IEEE, pp. 149–154.

Kohonen, Teuvo (1982). "Self-organized formation of topologically correct feature maps". In: *Biological Cybernetics* 43.1, pp. 59–69. ISSN: 1432-0770. DOI: `10.1007/BF00337288`. URL: `https://doi.org/10.1007/BF00337288`.

Ku, Chien-Chun, Chen-Fu Chien, and Kang-Ting Ma (2020). "Digital transformation to empower smart production for Industry 3.5 and an empirical study for textile dyeing". In: *Computers Industrial Engineering* 142, p. 106297. ISSN: 0360-8352. DOI: `https://doi.org/10.1016/j.cie.2020.106297`.

Lee, Chia-Yen and Chen-Fu Chien (2020). "Pitfalls and protocols of data science in manufacturing practice". In: *Journal of Intelligent Manufacturing*. ISSN: 1572-8145. DOI: `10.1007/s10845-020-01711-w`.

Lee, CKH et al. (2013). "A hybrid OLAP-association rule mining based quality management system for extracting defect patterns in the garment industry". In: *Expert Systems with Applications* 40.7, pp. 2435–2446.

Li, Jing-Rong, Li Pheng Khoo, and Shu Beng Tor (2006). "RMINE: A Rough Set Based Data Mining Prototype for the Reasoning of Incomplete Data in Condition-based Fault Diagnosis". In: *J. Intelligent Manufacturing* 17.1, pp. 163–176. DOI: `10.1007/s10845-005-5519-8`.

Lima, Alexandre et al. (2021). "A sampling-based approach for managing lot release in time constraint tunnels in semiconductor manufacturing". In: *International Journal of Production Research* 59.3, pp. 860–884. DOI: `10.1080/00207543.2020.1711984`.

Lin, Mingfeng, Henry C Lucas Jr, and Galit Shmueli (2013). "Too big to fail: large samples and the p-value problem". In: *Information Systems Research* 24.4, pp. 906–917.

Liu, Jie et al. (2018). "An improved fault diagnosis approach for FDM process with acoustic emission". In: *Journal of Manufacturing Processes* 35, pp. 570–579.

Lloyd, S. (1982). "Least squares quantization in PCM". In: *IEEE Transactions on Information Theory* 28.2, pp. 129–137. DOI: `10.1109/TIT.1982.1056489`.

Maaten, Laurens van der and Geoffrey Hinton (2008). "Visualizing Data using t-SNE". In: *Journal of Machine Learning Research* 9.86, pp. 2579–2605. URL: `http://jmlr.org/papers/v9/vandermaaten08a.html`.

Maydon, Thomas (2017). *The 4 Types of Data Analytics*. Accessed on 24/07/2021. URL: `https://www.kdnuggets.com/2017/07/4-types-data-analytics.html`.

Michigan State University (2019). *4 Types of Data Analytics and How to Apply Them*. Accessed on 24/07/2021. URL: `https://www.michiganstateuniversityonline.`

com/resources/business-analytics/types-of-data-analytics-and-how-to-apply-them/.

Molnar, Christoph (2019). *Interpretable machine learning*. Lulu. com.

Montgomery, Douglas C (2019). *Introduction to statistical quality control*. Eighth. John Wiley & Sons. ISBN: 978-1-119-39930-8.

Ong, Phaik-Ling, Yun-Huoy Choo, and Azah Kamilah Muda (2015). "A Manufacturing failure Root Cause Analysis in Imbalance data set using PCA weighted association rule mining". In: *Jurnal Teknologi* 77.18, pp. 103–111.

Page, E. S. (1954). "Continuous Inspection Schemes". In: *Biometrika* 41.1/2, pp. 100–115. ISSN: 00063444. URL: http://www.jstor.org/stable/2333009.

Pearl, Judea (1999). "Probabilities Of Causation: Three Counterfactual Interpretations And Their Identification". In: *Synthese* 121.1, pp. 93–149. ISSN: 1573-0964. DOI: 10.1023/A:1005233831499.

– (2009). *Causality: Models, Reasoning and Inference*. 2nd. USA: Cambridge University Press. ISBN: 052189560X.

Pearl, Judea et al. (2009). "Causal inference in statistics: An overview". In: *Statistics surveys* 3, pp. 96–146.

PRNewswire (2020). *Big Data Analytics in Manufacturing Industry Set to Exceed $4.5 Billion by 2025 - Condition Monitoring to Register Significant Growth*. Accessed on 15/07/2021. URL: https://www.prnewswire.com/news-releases/big-data-analytics-in-manufacturing-industry-set-to-exceed-4-5-billion-by-2025---condition-monitoring-to-register-significant-growth-301033518.html.

Quinlan, J. Ross (1993). *C4.5: Programs for Machine Learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc. ISBN: 1-55860-238-0.

Radziwill, Nicole (2019). *Root Cause Analysis and the Tools You Need to Drive Continuous Improvement*. https://blog.intelex.com/2019/03/20/root-cause-analysis-rca-central-to-continuous-improvement/, accessed on 2019/11/21.

Rato, Tiago J and Marco S Reis (2015). "On-line process monitoring using local measures of association. Part II: Design issues and fault diagnosis". In: *Chemometrics and Intelligent Laboratory Systems* 142, pp. 265–275.

Relihan, Keelin, Shane Geraghty, and Aidan O'Dwyer (2007). "Some aspects of process control in semiconductor manufacturing". In: *Proceedings of IMC-24; the 24th International Manufacturing Conference*. Waterford Institute of Technology, August, 2007., 1097–1104.

Richter, Felix, Tetiana Aymelek, and Dirk C. Mattfeld (2017). "Automatic Root Cause Analysis by Integrating Heterogeneous Data Sources". In: *Operations Research Proceedings 2015*. Ed. by Doerner, KF and Ljubic, I and Pflug, G and Tragler, G. Operations Research Proceedings. Operations Research Conference (OR), Univ Vienna, Vienna, AUSTRIA, SEP 01-04, 2015. Austrian Operat Res Soc; German Operat Res Soc; Swiss OR Soc, 469–474. ISBN: 978-3-319-42902-1; 978-3-319-42901-4. DOI: {10.1007/978-3-319-42902-1_63}.

Roberts, S. W. (1959). "Control Chart Tests Based on Geometric Moving Averages". In: *Technometrics* 1.3, pp. 239–250. DOI: 10.1080/00401706.1959.10489860.

Rokach, Lior and Dan Hutter (2012). "Automatic discovery of the root causes for quality drift in high dimensionality manufacturing processes". In: *Journal of Intelligent Manufacturing* 23.5, pp. 1915–1930.

Sabet, SAAM, Alireza Moniri, and Farshad Mohebbi (2017). "Root-Cause and Defect Analysis based on a Fuzzy Data Mining Algorithm". In: *International Journal of Advanced Computer Science and Applications* 8.9, pp. 21–28.

Saez, Miguel A et al. (2019). "Context-sensitive modeling and analysis of cyber-physical manufacturing systems for anomaly detection and diagnosis". In: *IEEE Transactions on Automation Science and Engineering*.

Schapire, Robert E. (1999). "A Brief Introduction to Boosting". In: *Proceedings of the 16th International Joint Conference on Artificial Intelligence - Volume 2*. IJCAI'99. Stockholm, Sweden: Morgan Kaufmann Publishers Inc., 1401–1406.

Schubert, Erich et al. (July 2017). "DBSCAN Revisited, Revisited: Why and How You Should (Still) Use DBSCAN". In: *ACM Trans. Database Syst.* 42.3. ISSN: 0362-5915. DOI: 10.1145/3068335. URL: https://doi.org/10.1145/3068335.

Shan, Guogen and Shawn Gerstenberger (2017). "Fisher's exact approach for post hoc analysis of a chi-squared test". In: *PloS one* 12.12, e0188709–e0188709.

Shannon, Claude E (1948). "A mathematical theory of communication". In: *The Bell system technical journal* 27.3, pp. 379–423.

Shi, Dongfeng and Fugee Tsung (2003). "Modelling and diagnosis of feedback-controlled processes using dynamic PCA and neural networks". In: *International Journal of Production Research* 41.2, pp. 365–379.

Sim, Hyunsik, Doowon Choi, and Chang Ouk Kim (2014). "A data mining approach to the causal analysis of product faults in multi-stage PCB manufacturing". In: *International Journal of Precision Engineering and Manufacturing* 15.8, pp. 1563–1573. ISSN: 2005-4602. DOI: 10.1007/s12541-014-0505-8.

SixSigma (2017). *What Are Common Root Cause Analysis (RCA) Tools?* Accessed on 23/07/2021. URL: https://www.6sigma.us/etc/what-are-common-root-cause-analysis-rca-tools/.

Stasko, John, Carsten Görg, and Zhicheng Liu (2008). "Jigsaw: Supporting Investigative Analysis through Interactive Visualization". In: *Information Visualization* 7.2, pp. 118–132. DOI: 10.1057/palgrave.ivs.9500180.

Steinhauer, H. Joe et al. (2016). "Root-cause localization using Restricted Boltzmann Machines". In: *2016 19th International Conference on Information Fusion (FUSION)*, pp. 248–255.

Strobl, Carolin et al. (2007). "Bias in random forest variable importance measures: Illustrations, sources and a solution". In: *BMC bioinformatics* 8.1, p. 25.

Sun, Yanning et al. (2021). “An adaptive fault detection and root-cause analysis scheme for complex industrial processes using moving window KPCA and information geometric causal inference”. In: *Journal of Intelligent Manufacturing*. ISSN: 1572-8145. DOI: 10.1007/s10845-021-01752-9.

Sun, Zhao-Hui, Renjun Liu, and Xinguo Ming (2018). “A Fault Diagnosis and Maintenance Decision System for Production Line Based on Human-Machine Multi-Information Fusion”. In: *Proceedings of the 2018 Artificial Intelligence and Cloud Computing Conference*. AICCC ’18. Tokyo, Japan: Association for Computing Machinery, 151–156. ISBN: 9781450366236. DOI: 10.1145/3299819.3299824.

Tague, Nancy R. (2004). *Seven Basic Quality Tools*. Accessed on 06/07/2021. URL: https://asq.org/quality-resources/seven-basic-quality-tools.

Tenenbaum, Joshua B., Vin de Silva, and John C. Langford (2000). “A Global Geometric Framework for Nonlinear Dimensionality Reduction”. In: *Science* 290.5500, pp. 2319–2323. ISSN: 0036-8075. DOI: 10.1126/science.290.5500.2319. eprint: https://science.sciencemag.org/content/290/5500/2319.full.pdf. URL: https://science.sciencemag.org/content/290/5500/2319.

Tibshirani, Robert (1996). “Regression Shrinkage and Selection via the Lasso”. In: *Journal of the Royal Statistical Society. Series B (Methodological)* 58.1, pp. 267–288. ISSN: 00359246. URL: http://www.jstor.org/stable/2346178.

Wang, Yazhen et al. (2017). “Semiparametric PCA and bayesian network based process fault diagnosis technique”. In: *The Canadian Journal of Chemical Engineering* 95.9, pp. 1800–1816.

Zanon, Mattia, Gian Antonio Susto, and Sean McLoone (2014). “Root Cause Analysis by a Combined Sparse Classification and Monte Carlo Approach”. In: *IFAC Proceedings Volumes* 47.3. 19th IFAC World Congress, pp. 1947 –1952. ISSN: 1474-6670.