Ricardo Alexandre Fernandes da Silva Peres

Master of Science in Electrical and Computer Engineering

An Industrial Data Analysis and Supervision Framework for Predictive Manufacturing Systems

Thesis submitted in partial fulfillment
of the requirements for the degree of

Doctor of Philosophy in
Computer Science and Engineering

Adviser: José António Barata de Oliveira, Associate Professor, NOVA University of Lisbon

Co-adviser: Paulo Jorge Pinto Leitão, Coordinating Professor, Polytechnic Institute of Bragança

Examination Committee

Chairperson: Prof. Luis Manuel Camarinha de Matos
Raporteurs: Prof Luis Domingos Ferreira Ribeiro
Prof. Carlos Baptista Cardeira
Members: Prof. Luis Manuel Camarinha de Matos
Prof. Ricardo Jardim Gonçalves
Prof. José António Barata de Oliveira
Prof. Armando Walter Colombo

FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

September, 2019

An Industrial Data Analysis and Supervision Framework for Predictive Manufacturing Systems

This document was created using the (pdf)LaTeX processor, based in the "novathesis" template[1], developed at the Dep. Informática of FCT-NOVA [2].
[1] https://github.com/joaomlourenco/novathesis [2] http://www.di.fct.unl.pt

To Sara, Mother and Figueiredo because they always understood.

Acknowledgements

In this section I would like to take the opportunity to express my deepest gratitude towards everyone who, in some way, contributed to and supported the development and writing of this thesis.

First and foremost I would like to thank my supervisor, Professor José Barata, for the opportunity to alongside him over the last few years as part of his R&D team, which provided me invaluable professional experiences and lessons that I will always remember.

To my co-supervisor, Professor Paulo Leitão, a big thank you for all the support and availability throughout this entire process, particularly during the writing of this document and of the publications that supported this thesis.

Also, a special thanks to the Thesis Accompanying Committee. Professor Ricardo Gonçalves (FCT-UNL) and Professor Carlos Cardeira (IST) were always ready to provide assistance during the development of this thesis, steering it in the right direction with their crucial input.

I would also like to extend my gratitude to André Rocha, Duarte Alemão, Mafalda Rocha and Pedro Monteiro. Not only our time working together, but also your camaraderie, contributed tremendously to this work and to make the journey more bearable.

My thanks to all my current and past colleagues at UNINOVA-CTS and to all the partners that I had the pleasure of working with over the years in various European projects. All of you contributed with your knowledge, advice and humour. To Michael Gepp, thank you for the thesis fuel!

To my friends, I'm incredibly lucky to have you in my life. Bernardo Moura, Diogo Alves, José Nogueira, João Matos, Liliana Sequeira, Ricardo Ramos, a huge thank you for always being there unconditionally, through thick and thin.

To my Buddy, *"Olá!"*.

Last, but certainly not least, to my family, thank you for all the love and support. José Figueiredo, thank you for all the dedication. Words are not enough and time is cruel. To my mother, Célia Peres, you are tireless and I truly admire you. Thank you for everything, always. To my beautiful Sara Araújo, thank you for all you've done and all you've endured. You guys have always been my pillars, and this would not have been possible without you.

Abstract

Due to the advancements in the Information and Communication Technologies field in the modern interconnected world, the manufacturing industry is becoming a more and more data rich environment, with large volumes of data being generated on a daily basis, thus presenting a new set of opportunities to be explored towards improving the efficiency and quality of production processes.

This can be done through the development of the so called Predictive Manufacturing Systems. These systems aim to improve manufacturing processes through a combination of concepts such as Cyber-Physical Production Systems, Machine Learning and real-time Data Analytics in order to predict future states and events in production. This can be used in a wide array of applications, including predictive maintenance policies, improving quality control through the early detection of faults and defects or optimize energy consumption, to name a few.

Therefore, the research efforts presented in this document focus on the design and development of a generic framework to guide the implementation of predictive manufacturing systems through a set of common requirements and components. This approach aims to enable manufacturers to extract, analyse, interpret and transform their data into actionable knowledge that can be leveraged into a business advantage. To this end a list of goals, functional and non-functional requirements is defined for these systems based on a thorough literature review and empirical knowledge. Subsequently the Intelligent Data Analysis and Real-Time Supervision (IDARTS) framework is proposed, along with a detailed description of each of its main components.

Finally, a pilot implementation is presented for each of this components, followed by the demonstration of the proposed framework in three different scenarios including several use cases in varied real-world industrial areas. In this way the proposed work aims to provide a common foundation for the full realization of Predictive Manufacturing Systems.

Keywords: Predictive Manufacturing Systems, Cyber-Physical Systems, Data Analytics

Resumo

Com os avanços no campo das Tecnologias de Informação e Comunicação no mundo interligado atual, a indústria de manufatura está a tornar-se cada vez mais um ambiente rico em dados, com volumes cada vez maiores a serem gerados diariamente, apresentando assim um enorme potencial para a exploração de novas soluções que visem o aumento da eficiência e da qualidade dos processos de produção.

Isto pode ser conseguido através do desenvolvimento dos chamados Sistemas Preditivos de Manufatura. Estes sistemas têm como objetivo melhorar os processos de manufatura através da combinação de conceitos tais como Sistemas Ciber-Físicos de Produção, *machine learning* e análise de dados em tempo real, de forma a prever estados e eventos futuros da produção. Este conhecimento pode ser usado em diversas aplicações, incluindo políticas de manutenção preditiva, controlo de qualidade através da deteção antecipada de falhas ou defeitos ou otimização energética, por exemplo.

O trabalho de investigação proposto neste documento foca o desenho e desenvolvimento de uma *framework* generica para guiar a implementação de sistemas preditivos de manufatura através de um conjunto de requisitos e componentes comuns. Esta abordagem visa capacitar os fabricantes a extrair, analisar, interpretar e transformar os seus dados em conhecimento sobre o qual consigam atuar transformando-os em valor acrescentado para o seu negócio. Para este efeito é definida uma lista de objetivos, bem como de requisitos funcionais e não-funcionais para estes sistemas baseada numa revisão aprofundada da literatura. Subsequentemente é proposta a *framework* de Análise de Dados e Supervisão em Tempo-Real Inteligente (IDARTS), acompanhada de uma descrição detalhada dos seus componentes.

Para finalizar, uma implementação piloto é apresentada para cada um destes componentes, seguida da demonstração da *framework* proposta em três cenários distintos, incluindo diversos casos de estudo em variadas áreas industriais. Desta forma o trabalho proposta providencia uma base comum para a realização completa de sistemas preditivos de manufatura modernos.

Palavras-chave: Sistemas Preditivos de Manufatura, Sistemas Ciber-Físicos, Análise de Dados

Contents

List of Figures **xv**

List of Tables **xix**

Listings **xxi**

Acronyms **xxiii**

1 Introduction **1**
1.1 Background 1
1.2 Motivation 2
1.3 Relation to Other Research Activities 3
1.4 Document Organization 5

2 Research Problem **7**
2.1 Research Questions and Hypotheses 8
2.2 Research Methodology 9
2.2.1 Design Science and Predictive Manufacturing Systems 9
2.2.2 Design Science Research Methodology 10

3 Literature Review **13**
3.1 Establishing the Baseline 13
3.1.1 Methodology 14
3.1.2 Baseline Analysis Results 16
3.1.3 Refined Survey Results 27
3.2 Industry 4.0 41
3.2.1 Reference Architectures for Industry 4.0 44
3.3 Cyber-Physical Production Systems 49
3.4 Industrial Artificial Intelligence 51
3.5 Predictive Manufacturing Systems (PMS) 53
3.5.1 Existing Applications and Approaches 53
3.5.2 Relation to Current Work 54

4 The IDARTS Framework **57**

4.1 Requirements Engineering . . . 58
4.1.1 IDARTS' Goals and Requirement Sources . . . 59
4.1.2 IDARTS' Elicitation and Analysis of Requirements . . . 61
4.2 Framework Design and Mapping to Industry 4.0 Context . . . 69
4.3 Cyber-Physical Production System Component . . . 72
4.4 Real-time Data Analysis Component . . . 74
4.5 Knowledge Management Component . . . 76
4.5.1 Machine Learning Algorithms . . . 76
4.5.2 Evaluation Metrics . . . 78

5 Implementing IDARTS **81**
5.1 Multiagent System (CPPS Component) . . . 83
5.1.1 Adding and Removing Resources . . . 87
5.1.2 Acquiring and Processing Manufacturing Data . . . 89
5.1.3 Enacting a Predictive Response . . . 95
5.2 Storm Topology (RDA Component) . . . 96
5.3 Offline Data Analysis and Engineering (KM Component) . . . 98

6 Results and Validation **101**
6.1 Laboratory Cluster Deployment . . . 102
6.2 Multi-stage Quality Control Scenario . . . 107
6.2.1 Volkswagen AutoEuropa Use Case . . . 111
6.2.2 Zannini Use Case . . . 113
6.2.3 Electrolux Use Case . . . 115
6.2.4 Assessment of the GO0D MAN Use Case Instantiations . . . 117
6.3 Mitigating Defect Propagation in the Automotive Industry . . . 118
6.3.1 Characterization of the data set . . . 120
6.3.2 Algorithms for Defect Classification . . . 122
6.3.3 Data Analysis Results . . . 124

7 Discussion and Conclusions **129**
7.1 Confirmation of the Hypotheses . . . 130
7.2 Peer-Reviewed Scientific Contributions for Knowledge Transfer . . . 131
7.3 Limitations, Opportunities and Future Outlook . . . 133

Bibliography **135**

List of Figures

1.1 Relation to other research activities . . . 3
1.2 Contributions Related to European Research Projects . . . 5

2.1 Design Science Research Methodology (DSRM) iterative process. Source: (Peffers, Tuunanen, Rothenberger, & Chatterjee, 2007) . . . 10

3.1 Document-Term Matrix (DTM) . . . 15
3.2 Summary report pertaining to the corpus in study from Web of Science . . . 16
3.3 Top 25 origin country distribution of the corpus, extracted from the Web of Science repository . . . 17
3.4 Distribution of publications by domain according to the Web of Science repository 17
3.5 Word Frequency across the Search Space . . . 18
3.6 Bigrams Ranked by Raw Frequency Measure . . . 19
3.7 Trigrams Ranked by Raw Frequency Measure . . . 20
3.8 Bigram Frequency Ranking by PMI score . . . 21
3.9 Trigram Frequency Ranking by PMI score . . . 22
3.10 Co-citation map for predictive manufacturing corpus being studied . . . 23
3.11 Abstract Clustering using a K-Means Model . . . 25
3.12 Research Trends for Topics around PMS . . . 28
3.13 Distribution of the surveyed articles per focus area . . . 40
3.14 Co-citation map for Industry 4.0 . . . 43
3.15 Reference Architecture Model fo Industrie 4.0 (RAMI4.0) . . . 44
3.16 Relevant life cycles for I4.0 components based on IEC 62890. Source: (Plattform Industrie 4.0, 2015) . . . 46
3.17 Industrial Internet Reference Architecture (IIRA). Source: (Industrial Internet Consortium, 2019) . . . 48
3.18 Cyber-Physical Production Systems . . . 49
3.19 Industrial AI ecosystem (Lee, Davari, Singh, & Pandhare, 2018) . . . 52

4.1 The requirements engineering process. . . . 58
4.2 The requirements elicitation and analysis process. Adapted from (Sommerville, 2011) . . . 62

4.3 The proposed conceptual architecture for the Inteligent Data Analysis and Real-Time Supervision (IDARTS) framework targeting intelligent predictive manufacturing systems 70
4.4 IDARTS-RAMI4.0 mapping - Architectural Layers 71
4.5 Framing IDARTS ML model creation and deployment within the three dimensions of RAMI 4.0 72
4.6 CPPS rule-based reasoning flows 73

5.1 Proposal for the implementation of the IDARTS modules (Solid line: Data, Dashed line: Model Updates) (R. S. Peres, Rocha, Leitao, & Barata, 2018) . 82
5.2 CPPS Multi-Agent System Overview (R. S. Peres, Rocha, Leitao, & Barata, 2018) 84
5.3 Interactions between the CPPS' agents during the deployment of new CMAs. 87
5.4 Interactions between the CPPS' agents during the removal of unplugged CMAs. 88
5.5 Visual representation of the transition time timespan. 89
5.6 Visual representation of the action time timespan. 90
5.7 Interactions between the CPPS' agents during the data acquisition process. . 92
5.8 Knowledge inference example regarding a clamp's current state. 93
5.9 Data Processing Algorithm Flow Chart. Represents the process flow for the generation of new knowledge for each agent. 94
5.10 Apache Storm Real-time data processing topology 96
5.11 Storm UI topology monitoring visualization. 98
5.12 IDARTS knowledge management prototype graphical user interface. 99

6.1 IDARTS Testing 4-Node Cluster 102
6.2 Example of the output from the Logistic Regression Model 104
6.3 Apache Storm Topology's Average Capacity and Execution Latency 105
6.4 Shop-floor Data and Model Update Latency 106
6.5 Pluggability Latency 107
6.6 H2020 GO0D MAN adaptation of the IDARTS Framework (GO0D MAN Project (ID:723764), 2017) 108
6.7 The GO0D MAN Architecture for ZDM (GO0D MAN Project (ID:723764), 2018) 109
6.8 Instantiation of the GO0D MAN Predictive Manufacturing Systems (PMS) for VWAE. Adapted from (GO0D MAN Project (ID:723764), 2019a). 112
6.9 Instantiation of the GO0D MAN Architecture for the Zannini use case. Adapted from (GO0D MAN Project (ID:723764), 2019b). 114
6.10 Instantiation of the GO0D MAN Architecture for the Electrolux use case. Adapted from (GO0D MAN Project (ID:723764), 2019c). 116
6.11 Layout for the multi-stage defect propagation scenario. 118

6.12 Possible deployment architecture based on the IDARTS framework (R. S. Peres, Rocha, Leitao, & Barata, 2018). Legend: DI - Data Ingestion; DP - Data Preprocessing; P - Prediction; E- Evaluation; H - Historical database; 1 - Prediction request; 1 (dashed) - Store ground truth; 2 (dashed) - Request mode evaluation; 3 (dashed) - Request updated model. 119
6.13 Matrix of the absolute correlations between the different features based on the Spearman coefficient. 122
6.14 Visualization using Principal Component Analysis (PCA) to reduce the dimensionality of the data set. Data points are colored based on the target variable. 123
6.15 Result from the application of the Yeo-Johnson transform and robust scaler to F12. Raw distribution is presented on the left, transformed values are presented on the right. 123
6.16 Test results from 5-Fold Cross Validation. Results are divided by accuracy, precision, recall, f1 and roc_auc scores. 125
6.17 ROC curves for each of the models compared in this study. ROC is a probability curve and AUC represents a measure of separability between classes. 126
6.18 Confusion matrices for the holdout validation. The tuned XGBoost model achieved perfect recall on cars predicted over the three days after the last sample from the original data set. 127

List of Tables

3.1 Primary search string used for the baseline 14
3.2 Cluster Characterization and Distribution 26
3.3 Systematic Literature Review on the Applications of Predictive Data Analytics in Manufacturing 29
3.4 Key challenges in industrial Cyber-Physical Production System (CPPS). Adapted from: (Leitão, Colombo, & Karnouskos, 2016) 51

4.1 Traceability regarding Non-Functional Requirements 66
4.2 Traceability regarding Functional Requirements 68
4.3 Summary of the requirements defined for Predictive Manufacturing Systems . 69

6.1 Apache Storm Topology Metrics 103
6.2 CPPS Pluggability Test Results 105
6.3 Verification and mapping of the IDARTS requirements in the GO0D MAN implementation. Legend: • - verified; ○ - partially verified; × - not applicable. 110
6.4 Bodyshop Framing Dataset Example 121
6.5 Baseline model results. 124
6.6 Parameters for each model resulting from the tuning through randomized search optimizing for roc_auc. Tuning was performed on 100 iterations with 5-fold cross validation. 125
6.7 Tuned model results. Models are evaluated based on the same metrics used for the baseline models' cross validation. 126
6.8 Verification and mapping of the IDARTS requirements in the pilot implementation. Legend: • - verified; ○ - partially verified; × - not applicable. 127

7.1 Scientific Publications 131

Listings

5.1 DA generic interface. 87
5.2 CMA generic interfaces. 88
5.3 Snippet from data descriptions in markup language. 91
5.4 Snippet from the implementation of IDataOutput. 92
5.5 Generic interface for enacting predictive responses. 95
5.6 Update message to be sent from KM through Kafka. 100

Acronyms

AI	Artificial Intelligence.
ANN	Artificial Neural Network.
AUC	Area Under the Curve.
BDA	Big Data Analytics.
CB	Context Bolt.
CMA	Component Monitoring Agent.
CPPS	Cyber-Physical Production System.
CPS	Cyber-Physical System.
DA	Deployment Agent.
DAB	Data Analysis Bolt.
DS	Design Science.
DSR	Design Science Research.
DSRM	Design Science Research Methodology.
DTM	Document-Term Matrix.
EPS	Evolvable Production System.
ERP	Enterprise Resource Planning.
FIPA	Foundation for Intelligent Physical Agents.
FPR	False Positive Rate.
FR	Functional Requirement.
GNB	Gaussian Naive Bayes.
HMI	Human-Machine Interface.
HMS	Holonic Manufacturing System.

ICT	Information and Communication Technology.
IDARTS	Inteligent Data Analysis and Real-Time Supervision.
IIRA	Industrial Internet Reference Architecture.
IoT	Internet of Things.
IT	Information Technology.
JADE	Java Agent Development framework.
KM	Knowledge Management.
KNN	K-Nearest Neighbours.
KPI	Key Performance Indicator.
MAS	Multiagent System.
ML	Machine Learning.
MMP	Multistage Manufacturing Process.
NFR	Non-Functional Requirement.
NLP	Natural Language Processing.
PCA	Principal Component Analysis.
PdM	Predictive Maintenance.
PHM	Prognostics and Health Management.
PMI	Pointwise Mutual Information.
PMML	Predictive Model Markup Language.
PMS	Predictive Manufacturing Systems.
RAMI4.0	Reference Architectural Model for Industry 4.0.
RDA	Real-time Data Analysis.
RE	Requirements Engineering.
RF	Random Forest.
RMS	Reconfigurable Manufacturing Systems.
ROC	Receiver Operating Characteristics.
RUL	Remaining Useful Life.
SF	Smart Factory.
SIT	Smart Inspection Tool.

SMA	Subsystem Monitoring Agent.
SVM	Support Vector Machine.
TF-IDF	Term Frequency-Inverse Document Frequency.
TPR	True Positive Rate.
VWAE	Volkswagen AutoEuropa.
ZDM	Zero Defect Manufacturing.

CHAPTER 1

Introduction

1.1 Background

Over the past few decades manufacturing has undergone several profound changes, with new market trends obligating manufacturers to steadily shift from the popular mass production concept to highly dynamic and flexible production lines.

Due to the increasing market competitiveness and the growing demand for highly customized products, with many variants and fluctuating demand for each one, companies are required to quickly adapt and adjust to new business opportunities in order to survive. As a direct consequence, manufacturing systems are required to be more and more agile in order for manufacturers to thrive and prosper in such a competitive environment riddled with unpredictable changes, empowering them to rapidly and effectively react to the changing markets driven by the increasing demand for customization.

Traditionally, conventional manufacturing architectures rely entirely on a fully centralized control system, in which a central entity completely governs the decision making process. While hierarchical approaches can potentially introduce optimizations at the control system level at the cost of having massive processing entities, the associated processing time greatly increases as the system's structure and size grow, sacrificing other relevant performance indicators, such as the system's adaptability, responsiveness and agility.

However, more recently the continued increase in the demand for customized products, often to the extreme (a different product for each customer), which are in turn getting more and more complex and varied in regards to their application domain, has translated into shorter changeover times and product life cycles, moving further and further away from the idea of standardized mass production, towards one of mass customization instead (Nagorny, Colombo, & Schmidtmann, 2012). Due to this, the emerging market requirements cannot be met by simply using the conventional manufacturing systems, generally based on hierarchical architectures with centralized decision-making, accompanied by hard-connected, non-interchangeable layouts.

These new business forms, reliant on a desire for a strong collaboration between suppliers and customers, impose further challenges to the shop floor, making older approaches unsuitable for this new reality (Frei, Barata, & Onori, 2007). All of this has culminated in the emergence of several different manufacturing paradigms, in an attempt to meet these new requirements for flexibility, agility and reconfigurability, among which are included Holonic Manufacturing System (HMS), Reconfigurable Manufacturing Systems (RMS) and Evolvable Production System (EPS). These emergent paradigms inspired a considerable amount of work that once associated with the rising of multi-agent technologies bore fruit to varied research efforts, such as the HMS-based ADACOR architecture (Leitão & Restivo, 2006), EPS focused FP6 EUPASS and FP7 IDEAS (Onori, Lohse, Barata, & Hanisch, 2013) and the RMS centric FP7 PRIME (A. D. Rocha, Barata, & Orio, 2015), which contributed significantly by showcasing actual implementations of the theoretical principles advocated by these paradigms.

Simultaneously, the increasing complexity of the manufacturing systems and the need to cope with rapidly changing production environments showcased the importance of employing a predictive manufacturing approach, where an early detection of potential failures using data analytics and leveraging the large volumes of manufacturing data being generated can help preventing unscheduled shutdowns and reducing some production costs. This is particularly interesting due to the recent advances made in Information and Communication Technology (ICT) with novel approaches in the fields of Machine Learning (ML), cloud computing and big data. However, most existing solutions are either still built on the assumption that the layout and requirements of the underlying system will remain the same, or are too application specific, being unable to be generalized to other domains, thus presenting a considerable gap to be further investigated.

1.2 Motivation

The research work described herein is centred on studying the topic of PMS. As such, the focus of this study is to assess how these systems can be employed to provide manufacturers with a business advantage, as well as how they should be implemented in order to support a wide array of applications and manufacturing environments, including legacy systems.

For this purpose, the first step is to establish a baseline regarding current developments and applications of PMS. Combining the availability of online digital research repositories with the advancements in the fields of ML and Natural Language Processing (NLP) in particular,

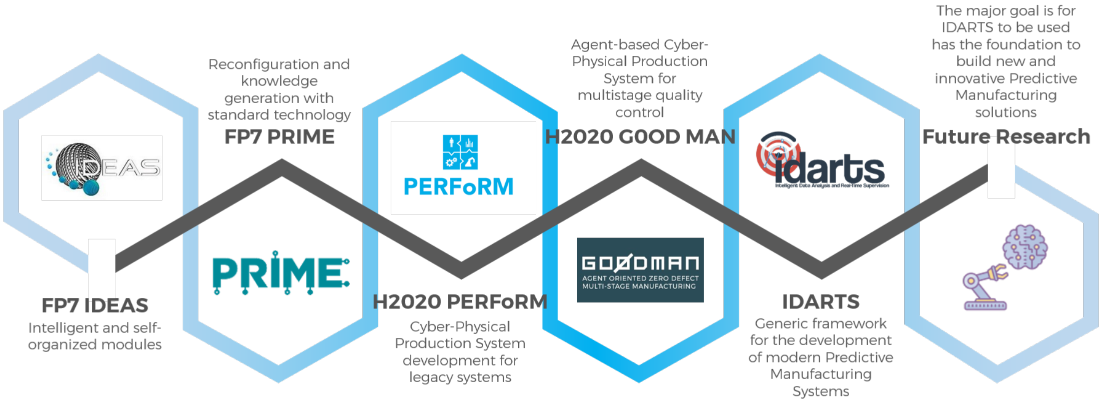

Figure 1.1: Relation to other research activities

this process can be improved enabling the analysis of a larger volume of publications in a fraction of the time it would take to do so manually. On top of this, underlying patterns and trends in the search space can also be identified, which could have gone unnoticed when relying on more traditional methods.

Based on this, a refined survey of the research on PMS performed over the last eight years was conducted, with the intent of identifying the main contributions and more importantly the corresponding implications for future research which should be taken into account in the present work. This facilitates the identification of the main gaps and existing challenges that still need to be resolved. In this case, despite the fact that several approaches have been proposed over the years spanning across a wide domain of application fields, it is clear that there is no concrete generic framework that can be used as the guideline to implement flexible PMS to various scenarios while being fully aligned with the current Industry 4.0 paradigm.

Therefore, this is the main goal of this dissertation, the proposition of the IDARTS framework for the development of flexible and scalable PMS solutions for smart factory environments. This framework aims to serve as the guideline for this implementations, providing researchers, designers and developers with common goals, requirements and components with which to build their PMS solutions.

1.3 Relation to Other Research Activities

The present work stems from an evolution across several past experiences, first and foremost from the author's previous work as part of his dissertation to obtain the degree of Master of Science in Electrical and Computer Engineering, and other research efforts across the years mainly in projects funded by the European Commission from both the FP7 and Horizon 2020 programmes. This evolution is summarized in Figure 1.1.

The FP7 IDEAS project (Onori et al., 2013), which concluded in 2013, laid the groundwork for the implementation of agent-based systems capable of abstracting the shop-floor at varied levels of abstraction, coordinating its execution at different granularities based on a System of Systems approach. This aspect of diving the shop-floor into smaller blocks to reduce its complexity, all the while being able to coordinate the system at both the local and global level is essential to the implementation of IDARTS-based systems.

Still within the same work programme, FP7 PRIME (A. D. Rocha, Barata, Orio, Santos, & Barata, 2015) built upon the aforementioned System of Systems approach to design and implement its multi-agent environments. At its core, PRIME focused two main functionalities, on the fly reconfiguration and the dynamic monitoring and generation of new knowledge based on raw data, both adopting a Plug & Produce approach. The former enables the system to reconfigure the control logic during execution, without the need to directly interfere with the process control. The latter entails the data collection and pre-processing to generate new knowledge based on the raw data, while dynamically adapting to changes on the shop-floor (A. D. Rocha, Peres, Flores, & Barata, 2016). This contributed to the present work by providing the basis for the CPPS data acquisition and pre-processing mechanisms, as well as for a possible way to perform adaptation through self-reconfiguration.

Moving onward to the Horizon 2020 programme, the PERFoRM project is aimed at empowering legacy systems with intelligent capabilities typically associated with the Industry 4.0 paradigm. While it still builds upon the results of previous projects, including for the development of MAS-based CPPS (R. S. Peres, Rocha, & Barata, 2017), a much larger emphasis is put into the integration and deployment challenges of this type of applications (Angione et al., 2017). This contributes to the development of IDARTS in regards not only to the integration of the different modules encompassed in the framework, but also in the integration and deployment of the solution in real production scenarios, particularly those comprising legacy systems.

Finally, the H2020 GO0D MAN project entails the development of an agent-based CPPS to perform quality control on multi-stage production systems, based on the Zero Defect Manufacturing (ZDM) paradigm. The GO0D MAN system will be implemented following the IDARTS framework, thus providing an example of a possible implementation of the framework for quality control, combining smart inspection tools, and agent-based CPPS, complex data analytics through ML and knowledge management. G0OD MAN also contributes through its industrial use cases, which can serve as case studies for the present work. These are focused on a varied array of industrial areas, namely professional appliances, automotive and turned metal components.

To better illustrate IDARTS' contributions to and from these related research activities, a radar chart is provided in Figure 1.2. For this purpose each contribution was given a score

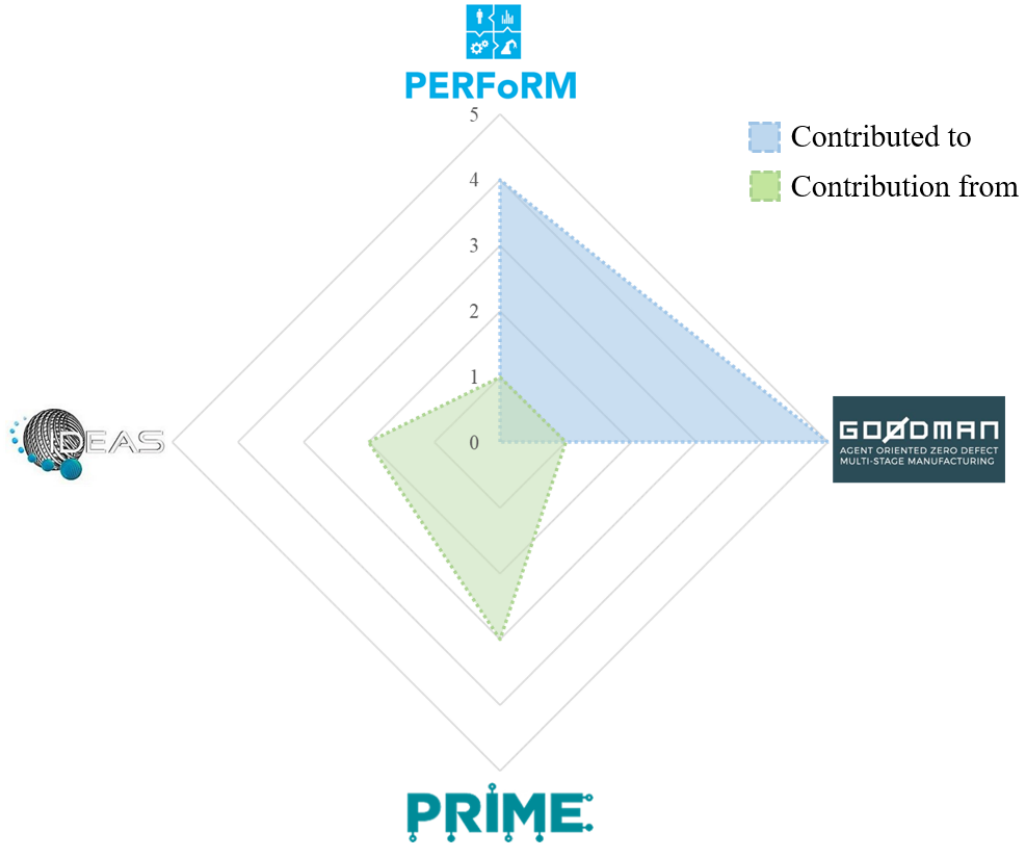

Figure 1.2: Contributions Related to European Research Projects

between zero and five, from "*Low*"to "*Very High*"respectively, representing the degree of significance of the contribution.

In summary, the IDARTS framework builds upon the results from past and current research work in an effort to combine the core elements that constitute PMS into a single, unified and generic framework. The main goal is for this framework to provide the foundation and guidelines for the implementation of PMS, acting as a catalyst for the appearance of new and innovative predictive manufacturing solutions and thus contributing to the further advancement of the field.

1.4 Document Organization

This dissertation is divided into seven chapters, being structured as follows. After this short introductory section, Chapter 2 poses the research questions and the respective hypotheses which will guide the work described ahead, providing a brief rationale for each of them. In addition, it also presents a short discussion about design science, along with an explanation of the adopted design science research methodology.

This is followed by Chapter 3, titled *"Literature Review"*, which aims to first and foremost establish a baseline in regards to the state of the art in the field of predictive manufacturing. These baseline results are analysed and used to conduct a refined survey of the current state of predictive manufacturing applications. The chapter concludes with the introduction of two supporting concepts which are pivotal for the understanding of the present work, more specifically the case of Industry 4.0 and that of PMS.

With the basic concepts already established, Chapter 4 entails two of the main contributions of the present work, starting by introducing the reader to the topic of requirements engineering, which is followed by the specification of the IDARTS goals and requirements, closing the chapter with the full description of the framework's design and its components.

Afterwards, Chapter 5, *"Implementing IDARTS"* proposes a pilot implementation of the IDARTS framework, aimed at serving as an example of a possible way to implement each of the framework's components, their interfaces and functionalities, as well as all the communications necessary to ensure their interoperability.

Being the second to last, Chapter 6 *"Results and Validation"* deals with the demonstration of variations of IDARTS implementations in different scenarios, showcasing the generic nature of the framework's design and mapping each of them to the requirements defined in Chapter 4.

Lastly, Chapter 7 concludes this dissertation by first summarizing the main contributions achieved by the work described herein, followed by the verification of the hypotheses formulated in Chapter 2, closing with a listing of peer-reviewed contributions for knowledge transfer associated with the present research and discussing current limitations, opportunities and the future outlook regarding this topic.

CHAPTER 2

Research Problem

In current literature there can be found a vast number of cases of research around the topic of data-driven support for manufacturing environments. A considerable effort is being put into capitalizing on the growing amount of data being generated in this industrial sector to provide insights and a business-edge to manufacturers, mostly at the level of the cyber world and the array of solutions around artificial intelligence that enable this.

Still, there is a clear lack of a commonly adopted view and set of guidelines towards the development of such predictive systems, particularly when accounting for aspects such as scalability and ease of migration and replication beyond the laboratory experiments or relatively enclosed case studies. Moreover, as it will be later discussed in Chapter 3, in most cases currently available solutions for these data-driven, predictive decision support systems are very monolithic and rigid, which contrasts heavily with the reality of the needs of a modern manufacturing systems in the context of a smart factory. These needs encompass the capability to adapt to changing market requirements in an agile and flexible fashion, along with the capacity to quickly scale capacity to meet floating demands while dealing with the inherent growing complexity of such systems through an easily manageable and robust approach.

Building from this, the current chapter introduces the reader to the research questions that guide the research work documented herein, along the hypotheses formulated as a potential answer to achieve explicit and innovative solutions to this problem. Additionally, the adopted research methodology based on Design Science (DS) is discussed, introducing the different activities involved in the research process described in this document and guiding the reader through the creation of the artefacts that constitute the main contributions of this work to the field of predictive manufacturing.

2.1 Research Questions and Hypotheses

Based on the motivation outlined in Section 1.2, the present chapter defines the research questions which were designed with the purpose of assisting and guiding the proposed research, along with the respective hypotheses formulated for each one. Considering the background presented in the first chapter, the following research question can be raised:

RQ1: *In which way can the core components and principles of a Predictive Manufacturing System be identified, in the context of enabling it to provide a business advantage to manufacturers, while coping with the current market requirements of flexibility and agility?*

Rationale: The goal of this first research question is to guide the literature review and gap identification in regards to PMS, ultimately resulting in a set of principles and core features that should be commonly encompassed in PMS to be incorporated into the framework resulting from the proposed research.

The first research question will be addressed through the following hypothesis:

H1: *If a baseline is established based on the current PMS literature, followed by a refined survey of current applications of PMS in the context of smart factory environments, sufficient information will be acquired to thoroughly identify common and critical requirements and components that should be encompassed in modern PMS.*

From this, a second research question can be raised pertaining to the design of a generic framework encompassing such characteristics:

RQ2: *How can we define a generic framework to guide the full realization of an intelligent, proactive and connected PMS solution for smart factory environments?*

Rationale: The second research question deals with the formalization of the framework encompassing the features that will result directly from the verification of H1.

H2: *If the common ground between existing narrow approaches is studied, it will be possible to formalize a generic PMS framework based on the combination of recent advancements in regards to Cyber-Physical Production Systems, data analytics and data management.*

From this framework implementations of PMS can be generated to be employed in varied application fields.

These will be the main research questions guiding the work described in the coming chapters.

2.2 Research Methodology

The research described herein aims to expand the current body of knowledge on the topic of PMS. The research methodology adopted for the proposed work follows the consensus-based DSRM proposed in (Peffers, Tuunanen, Rothenberger, & Chatterjee, 2007).

The topic of PMS, as an extension of information systems, is an interdisciplinary, applied research field that often borrows and applies theories, concepts and practices from other areas such as computer science and the social sciences to solve problems related to ICT. Despite this, most commonly adopted research paradigms used to conduct, produce and publish research in this field continue to follow traditional descriptive research borrowed from the social and natural sciences. Contrastingly, design is a well accepted research paradigm in engineering fields, as it pertains to the creation of explicitly applicable solutions to problems.

In this context, DS creates and evaluates ICT artefacts aimed at solving organizational problems, involving a rigorous process to design such artefacts, to make research contributions, evaluate the designs and to disseminate the results adequately (Hevner, March, Park, & Ram, 2004). These artefacts can include constructs, models, frameworks, methods and instantiations, or even properties of technical, social or information resources. Essentially, this definition of artefact includes any object designed which adds value or provides some utility in solving a given research problem.

2.2.1 Design Science and Predictive Manufacturing Systems

Sound scientific foundations of design are the cornerstone of our comprehension of the problem and solution domains that enables real-world research impact to be achieved. Design Science Research (DSR) aims to contribute to the existing knowledge base through the creation of innovative artefacts that are directed towards the solving of real-world problems and the improvement of the environment in which they are instantiated. Given this premise, the main outputs from DSR include not only newly designed artefacts, but also a more complete understanding of how these artefacts add value to their respective application contexts (Hevner et al., 2004). Therefore it is extremely important to carry out a search process for a design solution that complies with the problem constraints (such as the requirements) while achieving the desired goals (such as business needs and opportunities).

Given the inherent complexity of manufacturing environments and the agile pace at which they need to adapt to cope with new market trends and paradigms, it is natural to steadily move towards more adaptive solution search approaches based on fast design iterative cycles of building and refining solution artefacts.

Particularly for data science research, in which one can include data-driven PMS, it is important to ground the research efforts in interesting questions that explore the inherent variation in the data to gain competitive insights into the underlying behaviours, translating these to an improvement in decision making through the design of systems that automatically and easily support these processes (Mullarkey, Hevner, Gill, & Dutta, 2019).

2.2.2 Design Science Research Methodology

The DSRM proposed in (Peffers et al., 2007) consists in a six-step nominally sequential iterative process, as represented in Figure 2.1. This methodology builds on prior representative research from the field of design science using a consensus-building approach in order to achieve a commonly accepted framework for carrying research based on design science.

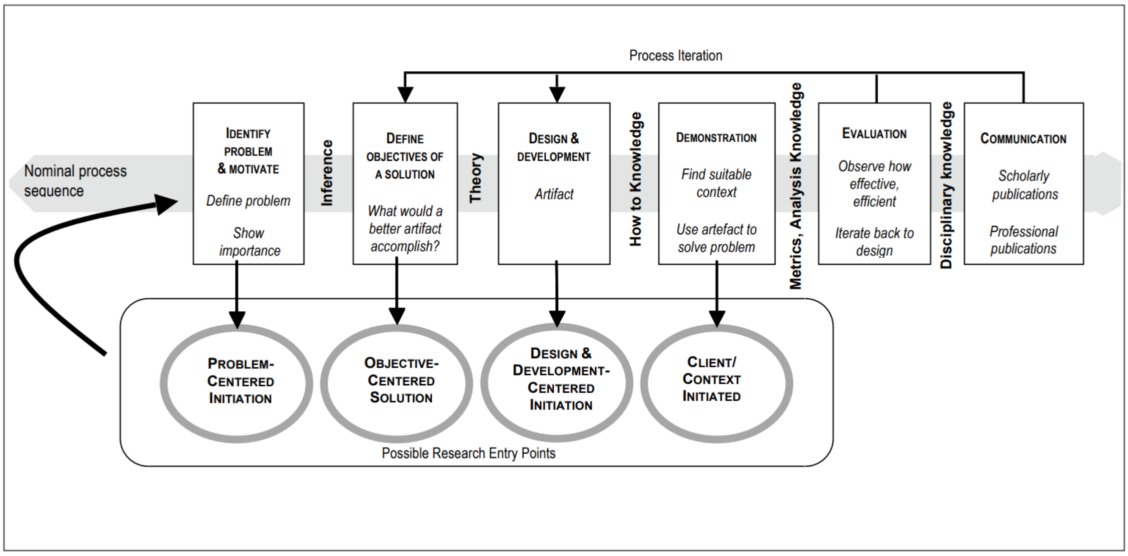

Figure 2.1: DSRM iterative process. Source: (Peffers, Tuunanen, Rothenberger, & Chatterjee, 2007)

According to (Peffers et al., 2007), all of the reference DS research articles presented some component in the initial stages to define a research problem. Hence, the first activity of the consensus-based methodology is *Problem Identification and Motivation.* This activity defines the specific research problem and attempts to justify the value of a solution. Since the problem definition will be used to develop an artefact aimed at providing an effective solution, the authors suggest it may be useful to decompose the problem conceptually to enable the solution to fully capture its complexity. This should accomplish two things, on the one hand it should provide motivation to the researcher and their audience in the pursuit of the solution and acceptance of the results, while on the other hand it should help convey the reasoning associated with the researcher's understanding of the problem.

Typically, to carry out this activity a good knowledge of the state of the problem, as well as of the importance of its solution are required. In the context of the present work, these are encompassed in Chapter 3 with the literature review and discussion of current challenges, gaps and research opportunities concerning PMS.

For the basis of the second activity, some of the reference research incorporated efforts to transforms the problem into objectives or requirements, while others proposed these were implicit as part of the data collection and programming tasks. Based on this, the authors defined the second activity as *Definition of the Objectives for a Solution*, consisting on the rational inference of goals from the problem definition and the knowledge of what is feasible. The goals can be quantitative, referring to conditions in which a given solution would be better than current ones, or qualitative, which can relate to how an innovative artefact can support solutions to problems not previously addressed.

This definition of the objectives is presented in Chapter 4, taking shape as the generals goals and respective non-functional requirements that guide the design of the proposed PMS framework.

A common point across the reference work in different disciplines is the core of DS, which is encapsulated in the third activity designated as *Design and Development*, dealing with the creation of the innovative artefact. As previously stated, these artefacts can be broadly defined constructs, models, methods or instantiations, as well as new properties of technical, social and/or informational resources.

From a conceptual standpoint, a DSR artefact is any designed object in which a research contribution is embedded in the design itself. Therefore, this third activity deals not only with the determination of the artefact's desired functionality and its architecture, but also its actual creation. Based on this definition, this activity can be mapped to Chapters 4 and 5, regarding the sections focusing on the framework's functional requirements, design and pilot implementation.

Regarding the following stages (corresponding to Chapter 6), the solutions analyzed by the authors varied from single demonstrations to prove a given artifact, to more formal evaluations of it, therefore both phases are included in the DSRM. Consequently, the fourth activity deals with the *Demonstration* of the artifact to solve one or more instances of the problem through its use in either simulation, a case study or other appropriate activity. For this effect an effective knowledge of how to use the artifact to solve said problem is required. Following this same logic, the fifth activity relates to *Evaluation*, meaning the observation and measurement of how well the artifact addresses a solution to the problem. This step can be carried out in a multitude of ways depending on the problem at hand and on the artifact's nature itself. A few examples include the comparison between the artifact's functionality and the objectives set out during the second activity, objective quantitative performance measures such as items produced, assessing the stakeholders' satisfaction through surveys, direct feedback or even simulations. It can also comprise quantifiable indicators of system performance such as response time, availability or throughput, including any appropriate empirical evidence or proof.

Once this activity is concluded, researchers can decide based on the evaluation whether or not it is necessary to further improve the effectiveness of the artifact towards the solution to a given problem, meaning that an iteration back to the third activity is required, or move on the the last stage and relay such improvement to future research efforts.

To finalize the DSRM, the *Communication* activity targets the dissemination of both the problem and the artifact, first concerning the importance of the former, and the innovation, rigor and effectiveness of the latter to peers and other relevant audiences such as industry professionals. This step is addressed later in 6 regarding the transfer of knowledge and academic results.

One aspect that should be highlighted and reinforced about the adopted DSRM is that while the process is structured in a nominally sequential order, there is no obligation or expectation that the research should always be conducted sequentially from the first activity to the last. In fact, several points of entry for research are considered, meaning that it may virtually start at practically any step and proceed from there. The proposed sequence can be followed strictly if the research stems from the observation of a problem or from future research suggestions from appropriate literature, but for instance a design-and-development-centered approach would start from the third activity. This could originate from an artifact that has not yet been formally expressed as a solution for an explicit problem domain in which it will be used or further improvements or extensions to its functionality in order to do so if it has already been used to solve a different problem. In such cases it is important to work backward to ensure that rigor is applied to the design process retroactively.

CHAPTER 3

Literature Review

As a starting point to find the answers for the research questions formulated in Section 2.1, a systematic literature review was conducted in order to create a solid foundation to work towards the realization of a PMS framework. Thus, this section provides a review on the applications of predictive data analytics to cross-domain and multidisciplinary areas comprised in the manufacturing sector.

The review was conducted based on the assumption that most current predictive analytics being researched or applied to manufacturing systems employ some form of ML techniques in order to predict future states or outcomes of the system.

To increase the industrial adoption of predictive manufacturing solutions and enable them to have an actual impact in the industry, it is first and foremost necessary to clearly identify its real-life benefits, as well as the existing gaps that must be filled to achieve this. Hence, this process will provide crucial insight into towards answering RQ1.

3.1 Establishing the Baseline

With the plethora of currently existing online research repositories and search engines (e.g. Web of Science, ScienceDirect, Google Scholar) providing access to digital publications, it is possible to take advantage of automated techniques (Yasin, Mohammad Yasin, & Mohammad Yasin, 2011) to search a wider range of the existing literature in a much shorter amount of time and at a much larger depth when compared to more traditional manual methods.

Furthermore, while these online resources allow users to gather a handful of information based on simple word matching queries, they do not take into account the underlying

Table 3.1: Primary search string used for the baseline

Primary Search String
manufactur* *AND* predict* *AND* ("data analy*"OR "machine learning"OR "monitor*")

context of the text. More often than not, in most current languages the particular order in which words or phrases are presented has a major influence in the idea or concept being conveyed. Therefore, analyzing collocations (a sequence of terms that co-occur often, becoming established through repeated context-dependent use) is a particularly important aspect of linguistic analysis.

To this extent, a methodology based on a combination of automated NLP and ML techniques and human screening was employed to analyze a large corpus of digital research publication abstracts in order to assess the current research interest and emerging trends around PMS.

3.1.1 Methodology

To enable the establishment of a baseline of the current research in the field of PMS, a ML/NLP-based methodology consisting in the following steps was used:

- Key Word Search
- Corpus Characterization
- Abstract Grouping

For the first stage, the initial construction of the publication list was performed via a key word search on the Web of Science repository, contemplating publications ranging between 2012 and 2019. For this purpose, the combination of the terms and Boolean operators presented in Table 3.1 was used to identify research articles of interest for the base corpus.

In the preliminary search string three main constraints were imposed. The topic of the publications (as defined by Web of Science), had to encompass some variation of "Predictive Manufacturing" (accomplished through the use of wildcards, including for instance "prediction", "predictive" and "predicting") and at least one term between some variation of "data analytics", "machine learning" and "monitoring".

This resulted in 1449 publications being identified, which would have been considerably challenging to analyze manually, particularly when accounting for emerging trends across

	Document 1	Document 2	Document 3	Document 4	Document 5	Document 6	Document 7	Document 8	Document 9	Document 10	
Term(s) 1	8	1	0	0	2	1	0	3	0	1	
Term(s) 2	0	2	1	3	0	0	0	0	0	0	
Term(s) 3	4	0	5	0	0	0	0	2	0	0	
Term(s) 4	0	0	0	1	0	0	9	0	5	0	
Term(s) 5	0	6	0	0	0	0	1	0	0	0	
Term(s) 6	0	1	0	7	3	0	0	4	0	0	Word Vector
Term(s) 7	0	0	0	0	0	0	0	0	0	5	
Term(s) 8	0	3	0	0	0	0	3	0	0	0	
Term(s) 9	0	0	8	0	0	1	0	0	1	0	
Term(s) 10	0	0	0	0	0	0	2	0	0	0	

Document Vector

Figure 3.1: Document-Term Matrix (DTM)

the search space and the similarities between documents. To tackle this, an automated NLP-based approach was used for the search space characterization, identifying latent structures in the document abstracts. To prepare the corpus for this process, each abstract was first pre-processed and cleaned following a sequence of steps in Python. Initially, the words that have no significance in unstructured text, also called stop words (e.g. “a”, “the” and “in”), were removed in order to optimize the end result. Following this, token N-grams were constructed consisting of one to three words, stemmed through Porter’s stemming algorithm (Willett, 2006). Stemming refers to the process of breaking a word down to its roots, where for instance “challenged”, “challenges” and “challenging” would correspond to the root “challenge” (Yasin et al., 2011). Afterwards, the abstract list was then converted to a (Term Frequency-Inverse Document Frequency (TF-IDF)) matrix.

This is achieved by first counting the word occurrences within each abstract, which are then transformed into a Document-Term Matrix (DTM) (example in 3.1). TF-IDF weighting is then applied, meaning that words that appear more frequently within an abstract but not frequently within the corpus receive a higher weighting, considering that these terms are assumed to have more significance in relation to the characterization of that particular abstract.

Based on this, the cosine similarity (H.Gomaa & A. Fahmy, 2013) can be measured against the TF-IDF matrix, generating a measure of similarity between any two abstracts within the corpus. However, abstract similarity alone is not particularly informative in terms of

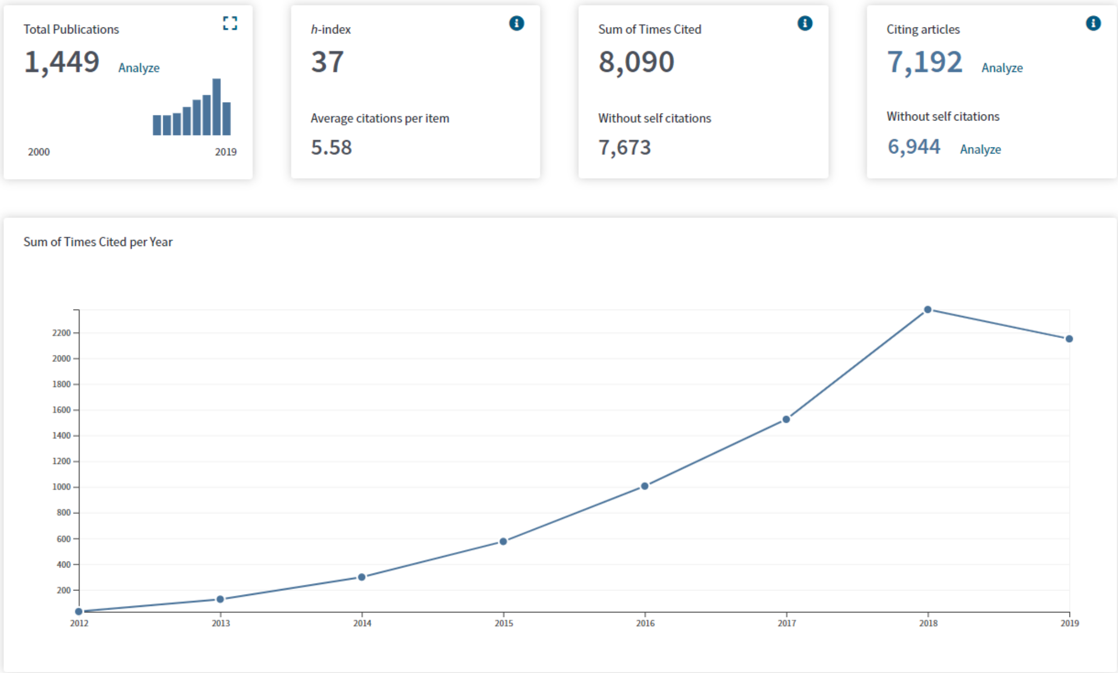

Figure 3.2: Summary report pertaining to the corpus in study from Web of Science

the characterization of the entire search space. It is necessary to identify similar patterns among the corpus and group them accordingly, so that the emerging trends within the search space can be identified. To this end, the final step of the methodology is the abstract grouping, which was performed using a form of K-means clustering. This final step later enables the informed selection of a wider array of publications to analyze in more depth for the literature review.

3.1.2 Baseline Analysis Results

The resulting corpus encompassed 1449 publications indexed to the Web of Science, based on the initial search string shown in Table 3.1. Through the Web of Science repository it is possible to obtain an overview of the citation report for this corpus, provided in 3.2.

From this summary it is possible to observe that there is a steadily growing interest from the scientific research community in the topics encompassed within this corpus, evidenced not only by the increasing number of total publications per year, but also by the similarly increasing number of citations generated per year, with each individual article currently generating 5.58 citations on average.

It is also interesting to take a loot at the origin of these publications, both geographically and in terms of their domain. Regarding the former, Figure 3.3 depicts an overview of the 25 most common origin countries for the publications contemplated in this analysis.

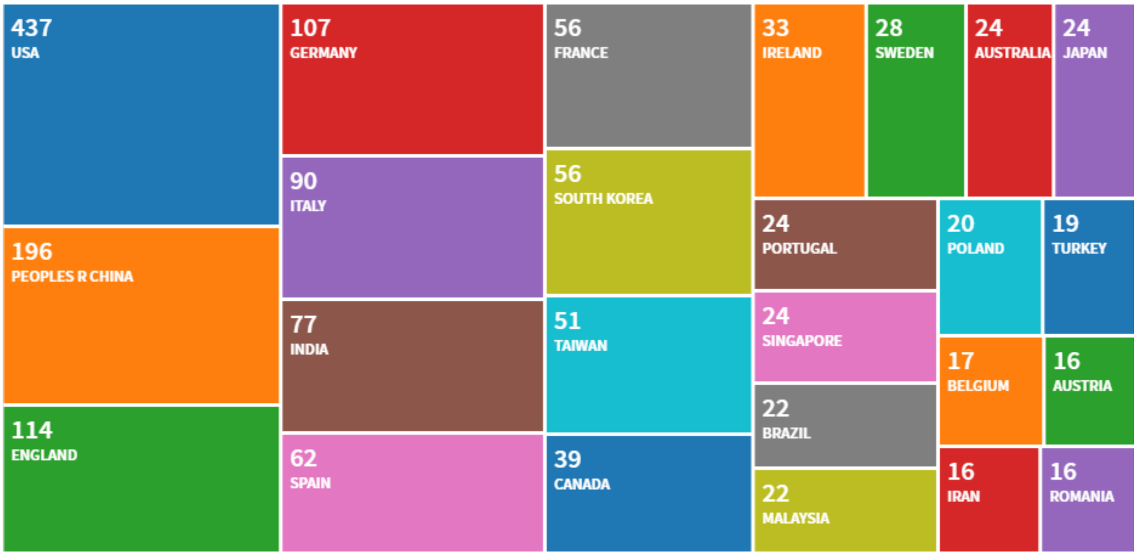

Figure 3.3: Top 25 origin country distribution of the corpus, extracted from the Web of Science repository

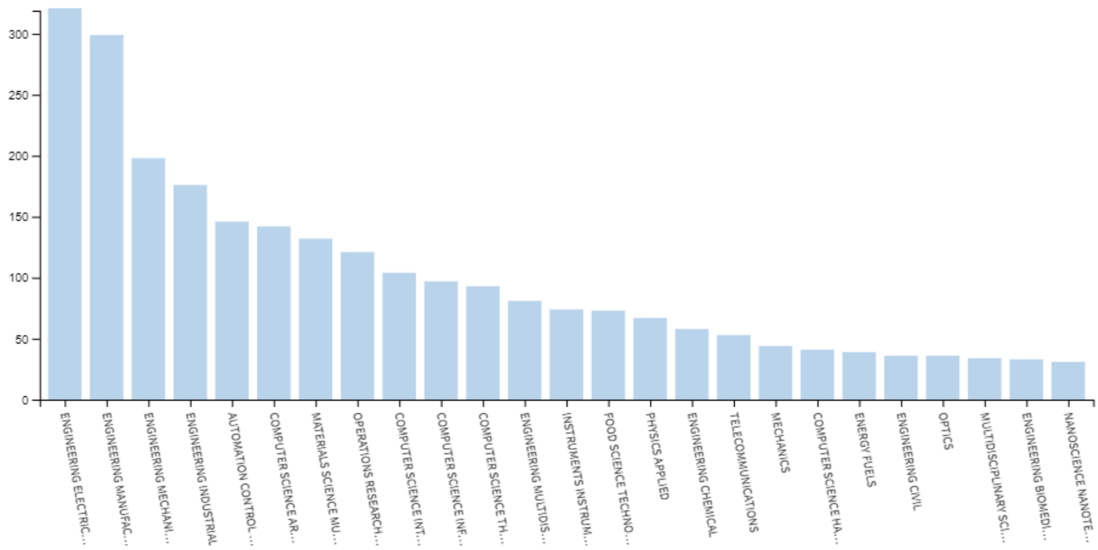

Figure 3.4: Distribution of publications by domain according to the Web of Science repository

Concerning the domain (see Figure 3.4), it can be said that a large portion of the publications stem from the electrical, manufacturing, mechanical and industrial engineering fields, being also heavily tied to computer science and artificial intelligence, along with other multidisciplinary areas.

For a more in-depth analysis of the corpus, it is important to go beyond what is readily made available by the Web of Science repository platform. To get a general idea of the overall scope of the corpus, as well as to make sure that it matches the term based search

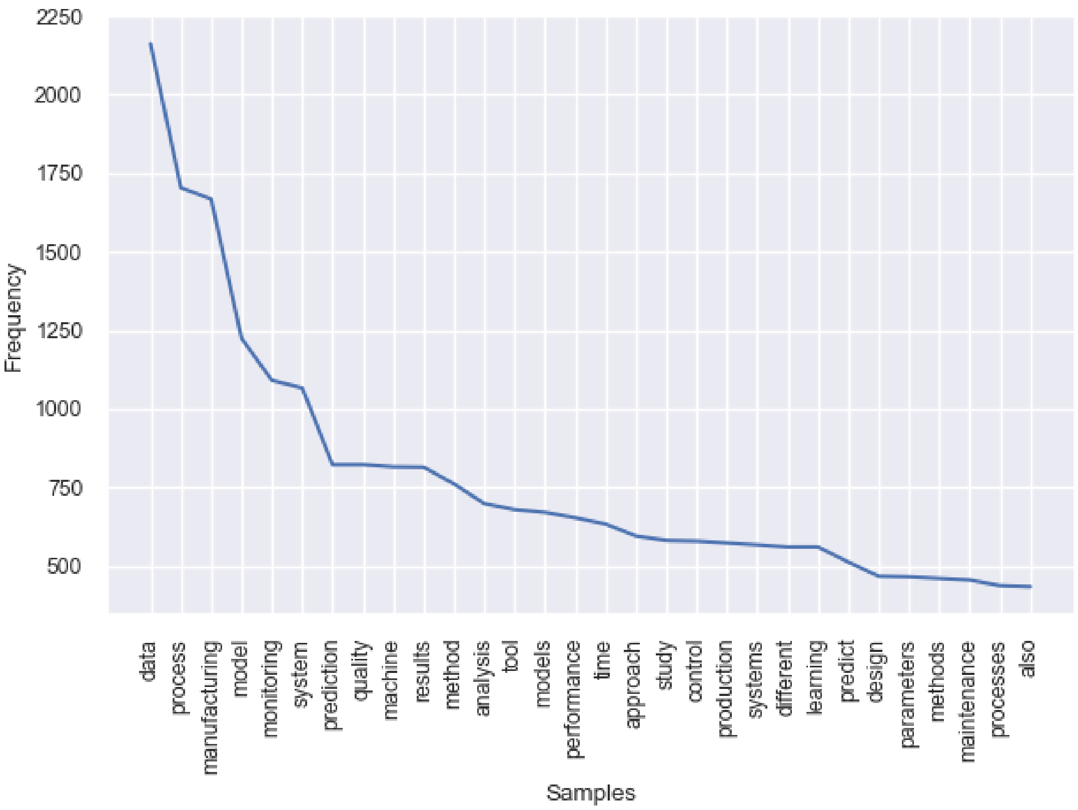

Figure 3.5: Word Frequency across the Search Space

parameters, the frequency of each term was calculated and can be found in Figure 3.5.

As it can be observed, words such as “data”, “manufacturing”, “monitoring” and “prediction” are among the most frequent, indicating that generally the focus of the publications is within the goal. Still, by itself this does not provide much insight other than the broad scope of the corpus. As such, it is also of interest to look at different combinations of N-grams, particularly bigrams and trigrams (e.g. “big data” and “big data analytics”, to get a clearer view of particular topics within the abstracts.

Figure 3.6 shows the frequency of bi-grams ranked by a raw frequency score measure, meaning their occurrence in relation to the document size, so as to not punish bi-grams occurring in shorter abstracts.

Unsurprisingly, “*Machine Learning*” appears as the most frequent bigram, which is to be expected given not only the initial search string, but also the fact that ML models are known to be widely used to make predictions about data or events which are yet to be observed. The list appears to suggest that there’s considerable focus in the research of both existing and new models and algorithms, as implied by the presence of the bigrams “*neural network*”, “*(support) vector machine*”, “*deep learning*” and “*prediction model*”. The high

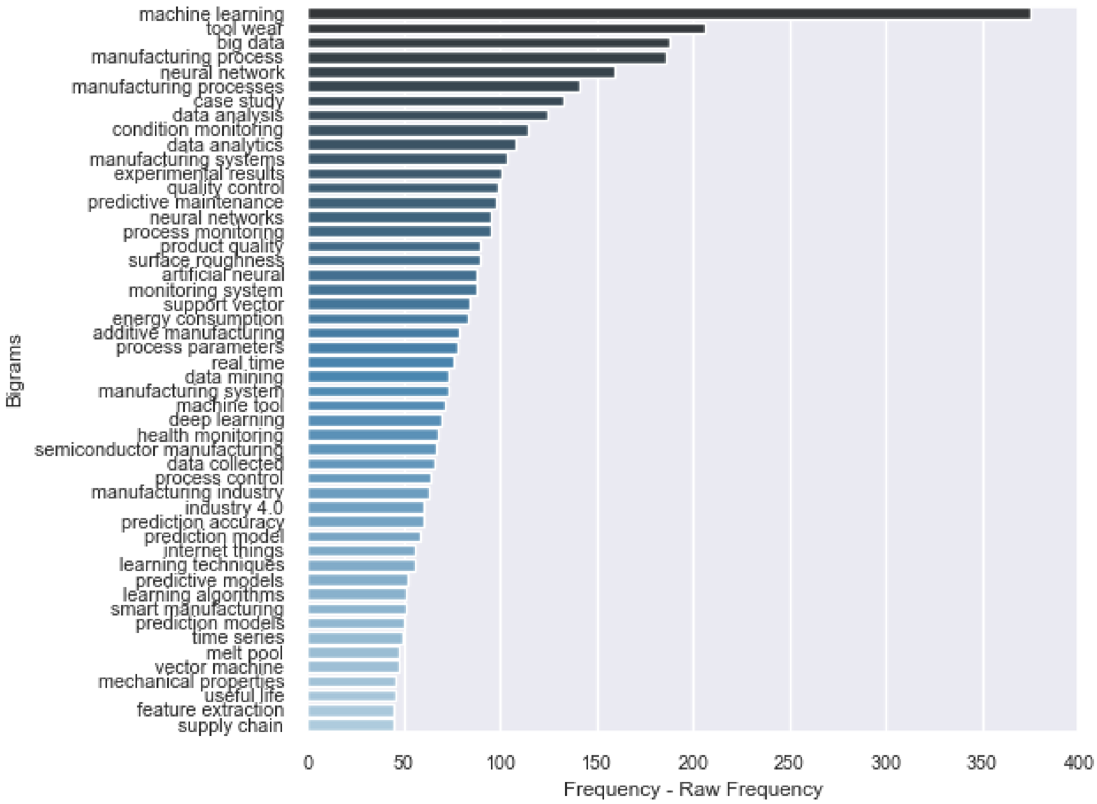

Figure 3.6: Bigrams Ranked by Raw Frequency Measure

frequency of the term "*tool wear*" suggests a large emphasis on the fields of maintenance and monitoring. This assumption is also supported by the presence of several other terms on list including "*condition monitoring*", "*monitoring system*" and "*predictive maintenance*". Other general research venues appear to include "*big data*", "*internet (of) things*", "*quality control*", "*supply chain*" and "*energy management*". Similarly, some conclusions can also be derived from the visualization of the most frequent trigrams found in the corpus. The list of trigrams scored according to a raw frequency measure is presented in Figure 3.7.

From the trigram list one can conclude that there is a clear reinforcement of the ideas resulting from the analysis of the bigrams. Once again, there is a large focus on ML models and algorithms, identified by very frequent variations of "*machine learning algorithms/techniques*", as well as of the terms "*Artificial Neural Network (ANN)*" and "*Support Vector Machine (SVM)*", insinuating a growing popularity of such algorithms in predictive manufacturing research.

It can also be relevant to perform the same process, but ranking the N-grams using the Pointwise Mutual Information (PMI) measure instead (Bouma, 2009). Given a pair of outcomes X and Y, PMI quantifies the discrepancy between the probability of their

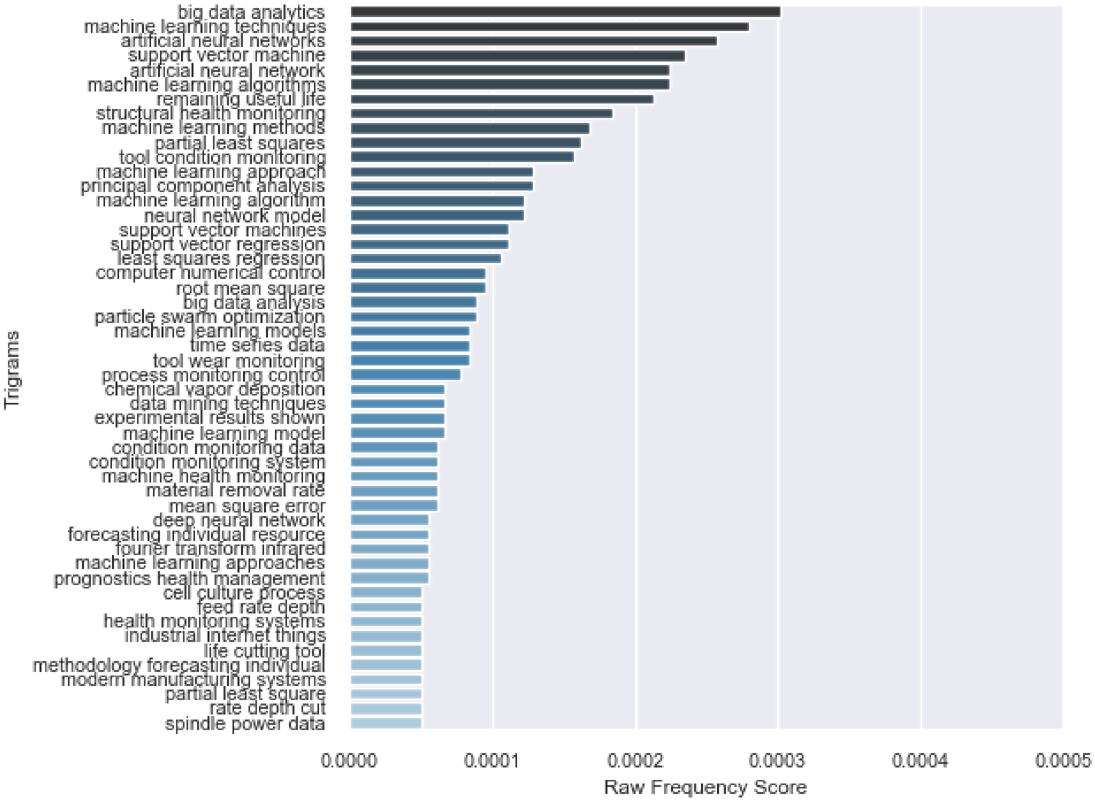

Figure 3.7: Trigrams Ranked by Raw Frequency Measure

coincidence given their joint distribution and their individual distributions. In practical terms, it provides the means to find collocations in a text corpus, meaning a sequence of words or terms that co-occur more often than would be expected by chance (e.g. "*supply chain*"). The corresponding ranking list of bigrams can be found in Figure 3.8.

Although this list appears to be more scattershot, there is still some additional information that can be retrieved from it. Particularly, besides the reaffirmation of "supply chain" and "internet (of) things" as areas of interest, further models are suggested, namely "random forest" and "(partial) least squares", as well as an additional application field ("injection molding"). Following this example, the same process can be done for trigrams, as seen in Figure 3.9. As it can be observed, the most frequent trigrams are aligned with the previous findings, with concepts such as "big data analytics", "prognostics and health management" and "remaining useful life" being considerably frequent in the corpus. In summary, this type of N-gram frequency analysis provides some relevant insights for further consideration during the refinement of the search space. For instance, one can conclude that supply chain management and Predictive Maintenance (PdM) appear to be two of the major application areas for this field. Beyond application areas, one can derive for instance relevant fields such as Big Data Analytics and Prognostics and Health Management (PHM) or even

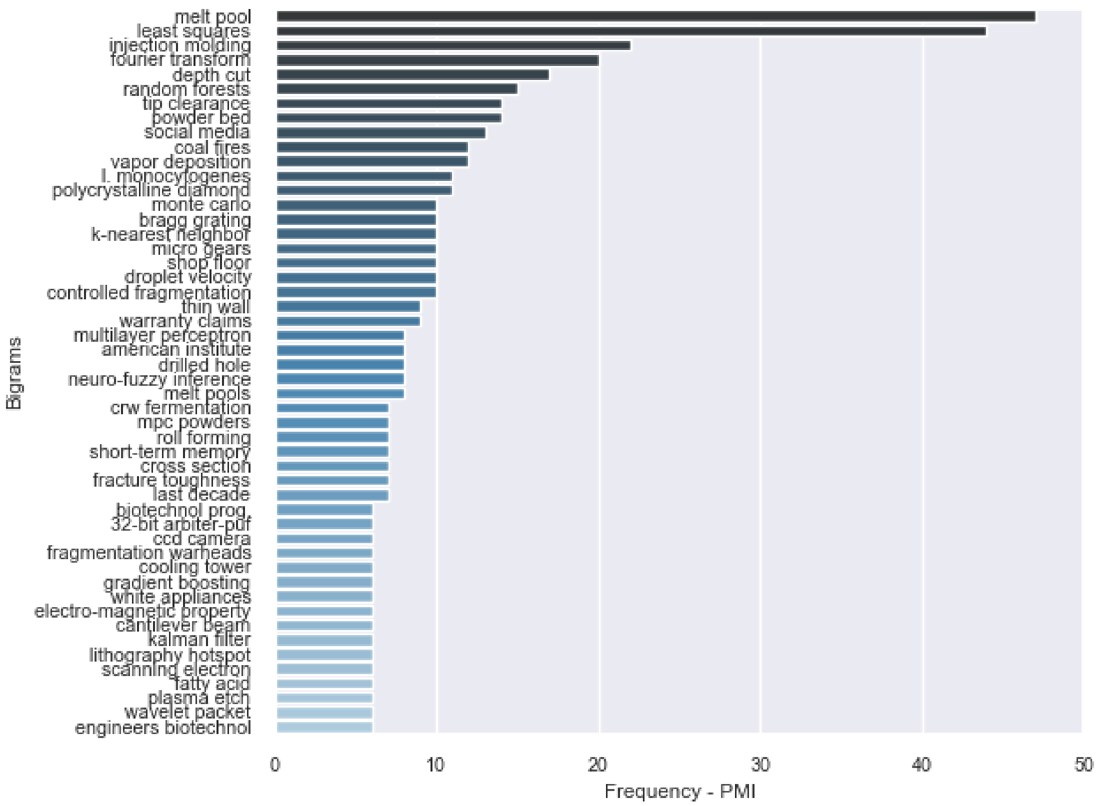

Figure 3.8: Bigram Frequency Ranking by PMI score

frequently used algorithms and models such as Random Forest (RF)s, ANN or SVM.

Given this general idea of common terms and apparent research venues present in the corpus, it is also valuable to look at the relationship between the cited references of each publication, something that can be done for instance through the creation of a co-citation map.

Through such a map, key players and publications in the broad field of predictive manufacturing can be identified, while also providing assistance in regards to the classification of the main emerging trends in the field. Hence, a co-citation map illustrating this can be seen in Figure 3.10, consisting in 100 nodes and 362 edges.

The co-citation map provided in Figure 3.10 was elaborated based on the aforementioned corpus from the Web of Science repository. The records for the eight year period were extracted into a CSV file which was used to generate the network edge list in Python. Afterwards, the data was imported into Gephi (Bastian, Heymann, & Jacomy, 2009) to generate the visualization and manipulate it to identify the most influential nodes.

Finally, community detection is performed using the Louvain method (Blondel, Guillaume,

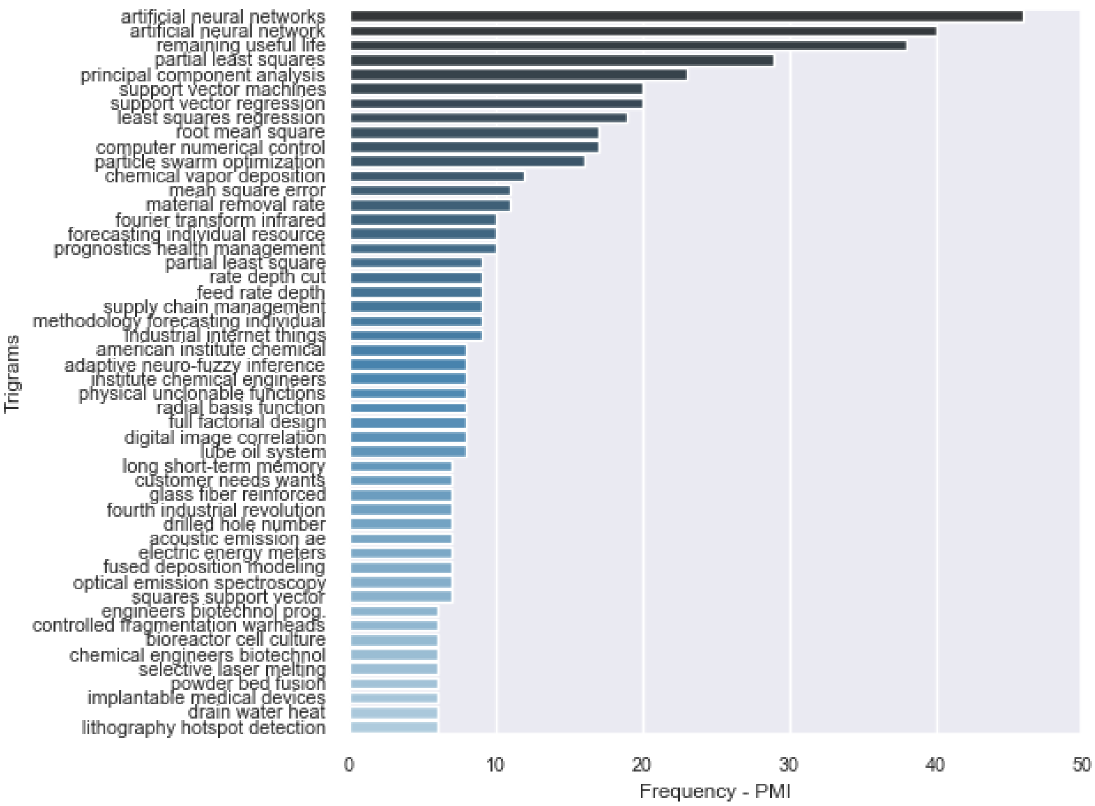

Figure 3.9: Trigram Frequency Ranking by PMI score

Lambiotte, & Lefebvre, 2008), with the resulting communities being attributed a distinct color in the visualization. From this process, seven main communities can be identified, each focused on a particular domain.

The community represented in orange (●) focuses mostly on the topic of *logistics and supply chain management* in the context of big data and predictive analytics. The main representative publications include (Gandomi & Haider, 2015) which presents an overview of concepts and methods related to big data, as well as (Waller & Fawcett, 2013) (Hazen, Boone, Ezell, & Jones-Farmer, 2014) and (G. Wang, Gunasekaran, Ngai, & Papadopoulos, 2016), focusing on the discussion of research opportunities and applications of predictive big data analytics in supply chain management, along with relevant challenges such as the issue of data quality.

The group in dark slate blue (●) is characterized by an emphasis on Cyber-Physical System (CPS) and cloud computing, particularly considering the application for *cloud manufacturing*. From the analysis of the co-citation map it can be said that there are also clear core publications in this community. Regarding CPS, (Lee, Bagheri, & Kao, 2015), typically considered a reference article presenting an architecture for CPS in the Industry

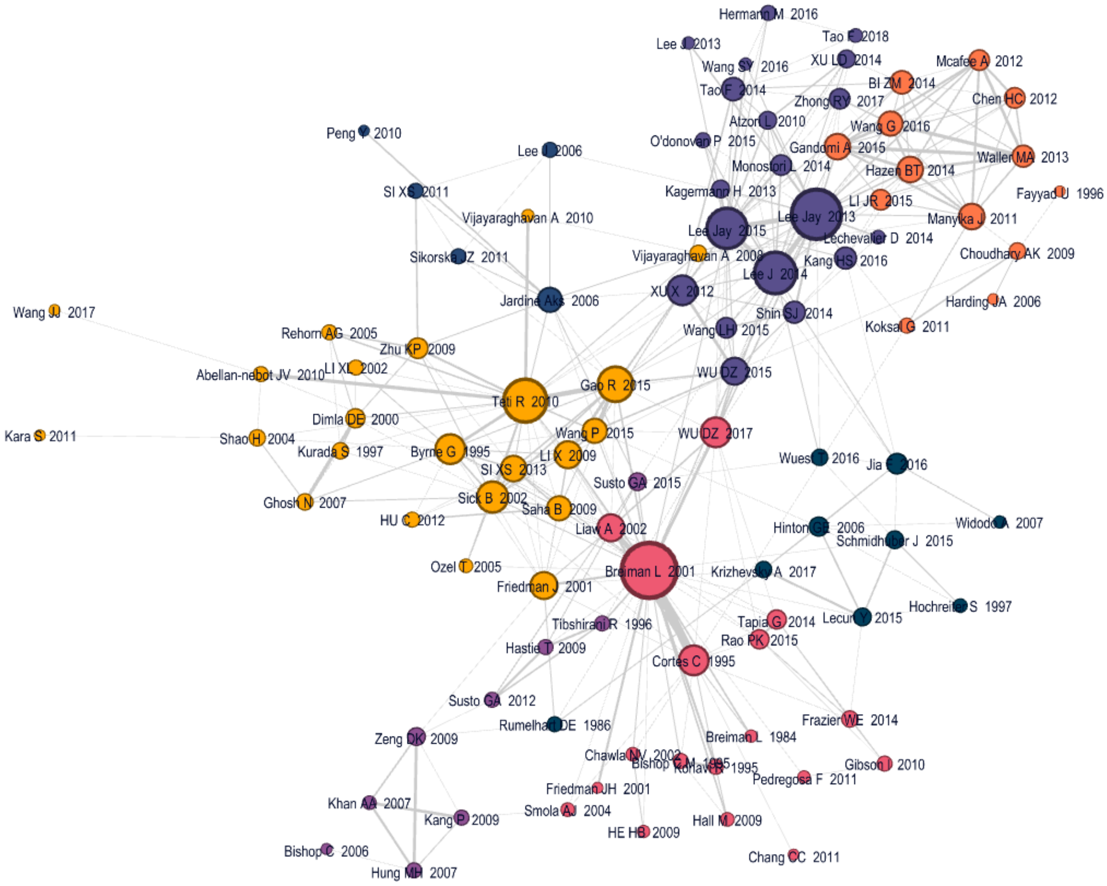

Figure 3.10: Co-citation map for predictive manufacturing corpus being studied

4.0 context, (L. Wang, Törngren, & Onori, 2015) discussing recent advances in CPS for manufacturing and (Lee, Lapira, Bagheri, & Kao, 2013), summarizing recent advances in PMS and relating them with big data and CPS. Finally (Xu, 2012) and (D. Wu, Rosen, Wang, & Schaefer, 2015) focus on the discussion of cloud manufacturing and its role as an innovative paradigm in digital manufacturing.

Condition monitoring and *Remaining Useful Life (RUL) estimation* are the main topics of the community depicted in yellow (●). Some of the more prominent publications include (Si, Wang, Hu, Chen, & Zhou, 2013), (Saha, Goebel, & Christophersen, 2009) and (P. Wang & Gao, 2015), dealing with the estimation of RUL in different applications, along with (Sick, 2002) in which the authors present a literature review of tool wear monitoring using ANN.

Moving to another large community, the group showcased in magenta (●) is broadly related with *ML* in general, its algorithms, techniques and libraries. However, there is a strong relation to the random forests algorithm, as evidenced by the weight and connections around (Breiman, 2001). Relevant publications include a comparative study on ML algorithms for smart manufacturing (D. Wu, Jennings, Terpenny, Gao, & Kumara, 2017), the paper on the synthetic minority over-sampling technique (Chawla, Bowyer, Hall, & Kegelmeyer, 2002) and the publications describing the LIBSVM (Chang & Lin, 2011), WEKA (Hall et al., 2009) and Scikit-Learn (Pedregosa et al., 2012) libraries.

Virtual metrology is another of the topics, pertaining to the community represented in light violet (●). This refers to methods concerning the prediction of wafer properties based on machine parameters and sensor data in the semiconductor manufacturing industry. Through classification and regression techniques, virtual metrology enables manufacturers to avoid performing the physical measurement of the wafer properties, thus contributing to the reduction of costs. Reference publications include (Khan, Moyne, & Tilbury, 2007), (Kang et al., 2009) and (Susto, Pampuri, Schirru, Beghi, & De Nicolao, 2015),dealing with varied approaches to apply virtual metrology in semiconductor manufacturing.

The small community in dark blue (●) relates to *PHM* and *condition-based maintenance*, thus being closely related to the yellow community dealing with RUL estimation. Some of the publications highlighted on the co-citation map include (Jardine, Lin, & Banjevic, 2006) and (Y. Peng, Dong, & Zuo, 2010), both providing literature reviews on machine prognostics towards the implementation of condition-based maintenance.

Lastly, the community represented in dark teal blue (●) is related to the topic of *Neural Networks* and *Deep Learning* in particular for more recent publications, which is natural given the increase in computational power over the last decade, as well as the advent of big data technologies and the massive data volumes available nowadays. The community includes publications by commonly accepted reference authors like Geoffrey Hinton and Yann LeCun (Hinton & Salakhutdinov, 2006) (LeCun, Bengio, & Hinton, 2015), the first discussing the aspect of dimensionality reduction of data with neural networks and the second introducing the concept of deep learning. Additionally, other publications include an overview on deep learning in neural networks (Schmidhuber, 2015) and the application of deep neural networks for fault characteristic mining and intelligent diagnosis with big data (Jia, Lei, Lin, Zhou, & Lu, 2016).

Overall it can be said that this community structure obtained with the Louvain method is aligned with the findings resulting from the application of NLP techniques to the aforementioned corpus, with the main research venues being identified in the lists of bigrams and trigrams presented in Figures 3.6-3.9.

Finally, in accordance with the final step of the methodology, a K-means model was used for the document clustering, as a way to visualize similarities between different publications and further detect emerging research trends in the field of predictive manufacturing. K-Means assumes you know the number of clusters a-priori, so this was determined empirically as well as with the aid of a hierarchical Ward clustering model to suggest the initial number

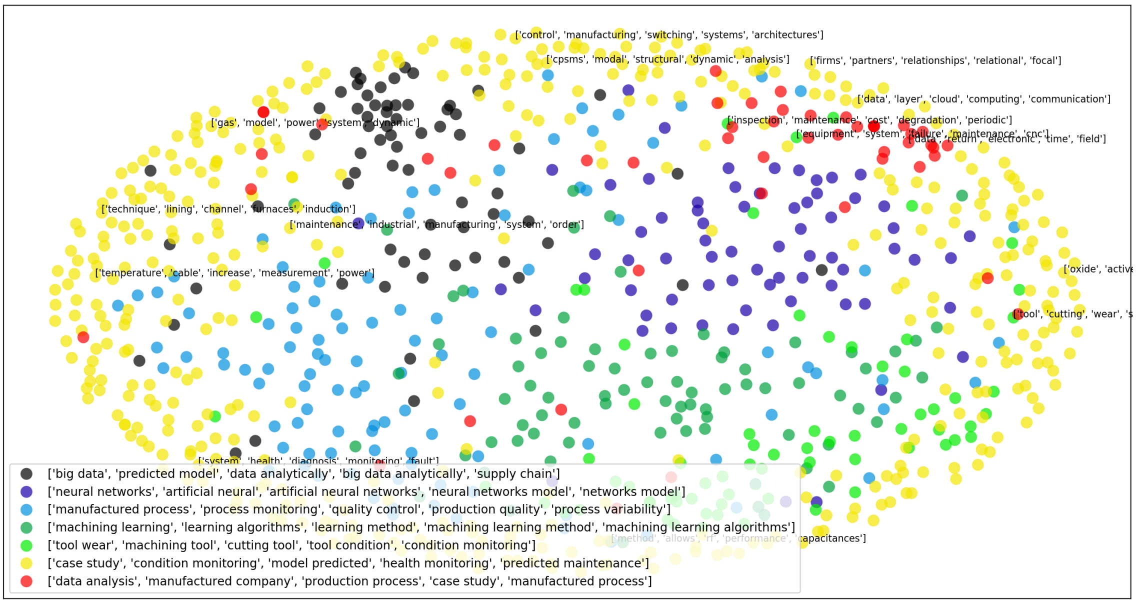

Figure 3.11: Abstract Clustering using a K-Means Model

of clusters to experiment with, until irrelevant or repeated groups started to emerge, having arrived at a division of eight clusters. The results of this process are illustrated in Figure 3.11. Note that for readability, only a small fraction of the abstract labels was kept in the illustration as a reference.

Each cluster's centroid is characterized by its most frequent N-grams, with N ϵ $\{1, 2, 3\}$. In order to reduce the dimensionality of the data and enable a two-dimensional representation of the cosine distance between the different abstracts, multidimensional scaling was applied to the distance matrix, converting it into a two-dimensional array. Therefore, proximity in this space equates to similarity as determined by the multidimensional scaling of the cosine distance between abstracts contained within the TF-IDF matrix. To enable a better understanding of the core concepts upon which each cluster is centered, Table 3.2 summarizes the most relevant N-grams that contribute to characterize each cluster individually in the feature space along with its coverage, as opposed to the illustrations provided previously, which portrayed the overall search space.

Based on the information summarized in Table 3.2, it can be inferred that a large portion of the search space (roughly 50%) is focused on the study of *PdM* and *Condition Monitoring*, two very large topics which dominate the corpus and are considerably related to the remaining topics.

Following this there's *Quality Control*, corresponding to around 12% of the corpus. This suggests a possible connection between predictive manufacturing techniques and quality control challenges that are yet to be solved. An example of this could be taking advantage

Table 3.2: Cluster Characterization and Distribution

Cluster	**Characterization**	**Search Space %**
1	**Predictive Maintenance**	49.55%
	Health Monitoring	
	Condition Monitoring	
	Case study	
	Model predicted	
2	**Quality Control**	12.36%
	Process Variability	
	Process Monitoring	
	Production Quality	
	Manufacturing Process	
3	**Machine Learning**	8.87%
	Machine Learning Algorithms	
	Machine Learning Methods	
	Learning Method	
	Learning Algorithms	
4	**Tool Wear**	7.98%
	Machining Tool	
	Cutting Tool	
	Tool Condition	
	Condition Monitoring	
5	**Neural Networks**	7.87%
	Artificial Neural	
	Artificial Neural Networks	
	Neural Networks Model	
	Networks Model	
6	**Big Data**	7.75%
	Supply Chain	
	Big Data Analytics	
	Data Analytics	
	Predictive Model	
7	**Data Analysis**	5.62%
	Manufacturing Company	
	Production Process	
	Case Study	
	Manufacturing Process	

of the capacity for a PMS to model and understand a system's behaviour on a multi-stage level, to assess and reduce the impact of the defects and their propagation downstream on these production systems.

The following group of four clusters share very similar dimensions (8-9% of the corpus), the first of which being related with the general study and development of ML algorithms and methods to be applied in production environments, comprising roughly 9% of the search space.

Cluster number four pertains to the topic of *Tool Wear* (8%), degradation and condition monitoring, suggesting a particular connection to the context of cutting and machining tools.

With almost the same dimension, the next cluster encompasses the publications dealing with *Neural Networks* and Deep Learning, followed by the last of this group which concerns Big Data Analytics (BDA) in the context of logistics and the supply chain. This is aligned with the current manufacturing setting, where due to the advancements in ICT, Internet of Things (IoT) and Cloud Computing technologies, larger and larger volumes of data are being produced every day. Despite this, only a small fraction of these data is taken advantage of by manufacturers, which presents an opportunity for disruptive technology to emerge from the research efforts surrounding big data.

Lastly, the smallest of the clusters appears to refer to the topic of data analysis in general within the context of manufacturing.

It is of interest to verify that despite resulting from the application of a different method, this document clustering approach corroborates the community analysis discussed in the previous step, with several of the main topics being shared between the two approaches. Regardless, there is still room for improvement particularly in two fronts, first regarding the stemming shown in Figure 3.11, as some of the words have been completed in peculiar ways during their reconstruction. The second point is that of the optimal number of clusters which is admittedly fairly difficult to precise in this type of approach, which could mean that there are for instance smaller clusters that could further divide the large PdM group.

Based on these results it is possible to further refine the initial search query, thus allowing for a smaller, more manageable and more focused number of publications to be manually analyzed for the literature survey.

3.1.3 Refined Survey Results

Combining the results from both of the aforementioned analyses, a group of main topics was identified in order to narrow the focus of the refined search for the literature review.

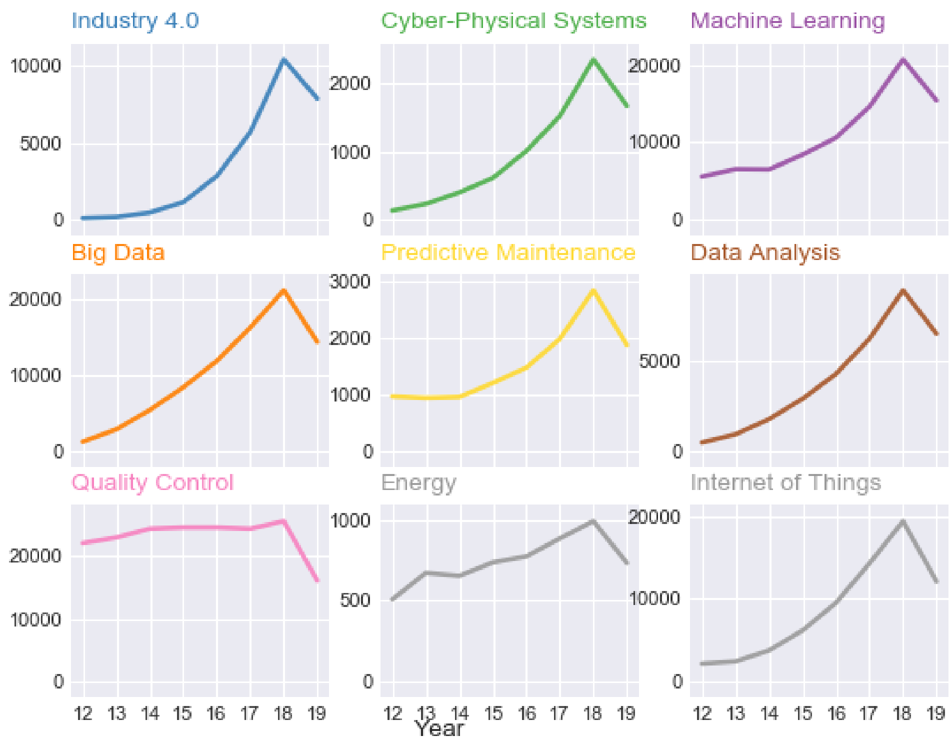

Figure 3.12: Research Trends for Topics around PMS

Once these topics were selected, a script was written in Python to automate the process of crawling Google Scholar and cyclically verify the number of publications registered over the last eight years for each of the topics, with the goal of visualizing their respective current trends. The search string used in this process followed the template "topic *AND* manufacturing" to ensure that the focus is constrained to a manufacturing context, resulting in the trend curves displayed in Figure 3.12.

As it can be observed, overall the main topics present an upward trend in the research occurrence over the last eight years, which suggests a growing interest in the research of new solutions around these topics for manufacturing environments. Based on the baseline established in Section 3.1, a more refined selection of publications was performed from a pool of relevant journals in the field, positioned on the top 25% of the impact factor distribution (Q1) according to the Scimago Journal Ranking (SCImago, 2007), resulting in about 50 hand-picked journal articles on the topics surrounding PMS. The review of these articles is presented in Table 3, where the publications are assessed regarding their application field and implications for future research (in the context of the present work). The application fields are labeled as follows:

- “GEN” – Generic
- “EN” – Energy
- “M” – Maintenance and PHM
- “LOG” – Logistics and Supply Chain Management
- “CON” – Control
- “SCH” – Scheduling
- “QC” – Quality Control
- “ECO” – Costs/Economics

Table 3.3: Systematic Literature Review on the Applications of Predictive Data Analytics in Manufacturing

Summary Analysis	Implications for Future Research
Publication: (Bumblauskas, Gemmill, Igou, & Anzengruber, 2017)	**App.Field:** M
Architecture and conceptual framework for a smart maintenance decision system across the supply chain based on corporate big data, validated through system simulation using databases from ABB Inc.	Highlights the importance of big data as an enabling factor to move from reactive and preventive maintenance policies towards predictive approaches; Provides an example of using an Analytical Hierarchy Process modelling technique for fleet prioritization.
Publication: (Cupek, Ziebinski, Zonenberg, & Drewniak, 2018)	**App.Field:** EN
Presents a methodology and respective architecture for monitoring energy efficiency in discrete production stations, tested in a laboratory test stand using industrial components and performing the transport of elements in a closed circuit using a pneumatic system.	K-means clustering is suggested to discover energy consumption profiles specific for different variants of production.
Publication: (Deng, Guo, Liu, Zhong, & Xu, 2018)	**App.Field:** EN
Propose data cleansing algorithms for energy saving to be applied in wireless sensor networks used by CPPS to achieve low-power, reliable data acquisition.	Reinforces the importance of accurate and reliable data acquisition to enable proper health monitoring, machine learning and other predictive techniques to be successfully applied.
Publication: (Flath & Stein, 2018)	**App.Field:** GEN, QC

Continued on next page

Table 3.3 – continued from previous page

Summary Analysis	**Implications for Future Research**
Developed a data science toolbox for manufacturing prediction tasks to bridge the gap between machine learning research and concrete practical applications, demonstrated in a real-world manufacturing defect prediction case study based on a dataset provided by Bosch. The authors also suggest some guidelines and best practices for modeling, feature engineering and interpretation.	The authors highlight some reasons why machine learning algorithms provide a considerable advantage over traditional statistical methods. Additionally, the article indicates that in order to improve the predictive power of analytics, continuous feature engineering and consolidation, as well as the constant improvement of models are required. This emphasizes the need to combine analytic skills with business knowledge, making industrial big data analytics a cross-disciplinary application of ML.
Publication: (G. Wang et al., 2016)	**App.Field:** LOG
Proposes a maturity framework to assess applications of supply chain analytics within the domain of logistics and supply chain management.	It outlines the latest techniques and methodologies used within the field, as well as current limitations and opportunities for future research in the application of big data analytics for SCA. The surveyed solutions are divided into descriptive, prescriptive and predictive (including time series forecasting and regression analysis), and identify applications sub-fields such as product design and development, network design and demand planning.
Publication: (Gao et al., 2015)	**App.Field:** M
Review of prognosis techniques and envisioned growth due to cloud technology. Proposes an architecture for cloud-enabled prognosis for manufacturing.	Analysis of strengths and weaknesses of different prognostic methods. Architecture highlighting core components of cloud-based prognosis. Summary of potential applications for cloud-based prognosis, with directions for future research.
Publication: (Ge, 2018)	**App.Field:** GEN
A distributed predictive modelling framework is proposed for prediction and diagnosis of key performance indices in plant-wide industrial processes. An example is provided using PCA and Gaussian Process Regression models in a case study on the Tennessee Eastman benchmark process.	The authors emphasize the importance of dividing a plant-wide process into smaller, distributed blocks, each with its own associated model, hence ensuring that data can be efficiently extracted while effectively reducing its dimensionality and thus also reducing the computational burden of the predictive modeling process. The development of multi-level predictive and diagnostic models is also indicated as potentially relevant for future research.

Continued on next page

Table 3.3 – continued from previous page

Summary Analysis	Implications for Future Research
Publication: (Golkarnarenji et al., 2018)	**App.Field:** EN
An intelligent predictive model for energy consumption in thermal stabilization process based on SVR is presented, taking into account production quality and controlling stochastic defects. This was tested in a single tow oxidation pilot oven designed by Despatch industries.	A comparison between SVR and Levenberg- Marquardt algorithm neural network models is provided, with SVR providing superior results for the use case at hand.
Publication: (Guo & Banerjee, 2017)	**App.Field:** GEN
Presents an application of topological data analysis (Mapper algorithm) in the domain of manufacturing using two benchmark data sets.	Results suggest that using only the features showing the most significant causal relationships in this method provides a comparable prediction accuracy to that of approaches using the complete set of features, albeit considerably smaller training times. The authors indicate that there's room to integrate this approach with existing machine learning techniques to increase its robustness and provide a practical method more suitable to the context of high-dimensional, heterogeneous manufacturing data in general.
Publication: (He, Gu, Chen, & Han, 2017)	**App.Field:** M, QC
Outlines an integrated PdM strategy that combines product quality control and mission reliability constraints for single equipment. The authors describe a case study focusing integrated PdM decision-making for a cylinder head manufacturing system. Results suggest that the integrated PdM strategy delivers a better economic performance than periodic preventive maintenance or traditional condition-based maintenance in general.	The evolution of key quality characteristics is used to identify process variables affecting product quality based on an axiomatic design approach. The authors suggest that product quality improvement, production planning and maintenance strategy formulation can be taken into account in future research for integrated production scheduling, along with the inclusion of other costs during optimization including for instance personnel costs.
Publication: (J. Wang et al., 2016)	**App.Field:** CON
The article proposes an online meta-level control framework based on multi-agent technology for large-scale online multitask learning and decision-making. It aims to effectively learn high-dimensional coordination policies and coordinate multi-machine action to achieve manufacturing flexibility. The approach was tested in a testbed at a smart factory of Weichai Power in China.	Multi-agent based CPS are indicated as a way to achieve manufacturing flexibility and scalability, with the proposed approach scaling to at least 200 machines. It is suggested that similar approaches could be extended and used for other applications such as multi-sensor network coordination.

Continued on next page

Table 3.3 – continued from previous page

Summary Analysis	Implications for Future Research
Publication: (Jimenez, Bekrar, Trentesaux, & Leitão, 2016)	**App.Field:** SCH, CON
The authors present a switching mechanism framework aiming at the optimal coupling of predictive scheduling and reactive control in dynamic hybrid control architectures. It was tested using Netlogo to simulate the AIP PRIMECA flexible job shop. The results suggest that the inclusion of a switching mechanism in these types of architectures enhances the agility for the optimality and reactivity of the system.	Extending the monitoring of the execution and system dynamics to deploy real analytics is indicated as one of the main venues for future research efforts in this context. This implies that the monitoring mechanism should not only handle data reporting and trend analysis, but instead be capable of forecasting to predict possible future behavior of the system and thus enable the switching mechanism to better choose the optimal operating mode.
Publication: (J. Wang, Zhang, Duan, & Gao, 2017)	**App.Field:** M
A cloud-based predictive maintenance approach using mobile agents is proposed. The approach was tested in test bed consisting of six induction motors with different failure modes, mimicking a distributed manufacturing system.	SVM classification shows promising results for multidimensional motor defect diagnosis. The usage of mobile agents could be used to enable computational heavy algorithms to be moved to a remote server by having the agents collect the required information from cloud nodes.
Publication: (Kant & Sangwan, 2014)	**App.Field:** EN
Multi-objective predictive model for minimization of power consumption (6.59%) and surface roughness (2.65%) in machining processes. The analysis of variance test concluded that the feed, depth of cut and cutting speed were the most significant parameters.	Example usage of Grey Relational Analysis with Principal Component Analysis for power consumption optimization.
Publication: (Kolar, Vyroubal, & Smolik, 2016)	**App.Field:** EN
Describes an analytical approach to model energy consumption of auxiliary units of a CNC machine with corresponding activity management. An experiment is shown assessing the accuracy of the model on a three-axis horizontal milling machine.	The article highlights the importance of a thorough identification of all the auxiliary units contributing to energy consumption, which greatly affects the resulting accuracy of the model.

Continued on next page

Table 3.3 – continued from previous page

Summary Analysis	Implications for Future Research
Publication: (Lao, Ellis, Durand, & Christofides, 2015)	**App.Field:** M, ECO, CON
Combines a robust moving horizon estimation scheme with an economic model predictive control system with the aim of enabling real-time preventive sensor maintenance and optimal process economic performance with closed loop stability. The design of the proposed approach is illustrated through its application to a chemical process network.	Highlights the importance of intelligent predictive solutions to be capable of coping with changing conditions in the underlying system, such as the case described in the article of a changing number of online sensors.
Publication: (Lee, Wu, et al., 2014)	**App.Field:** M
An extensive review of the PHM field is presented, along with the introduction of a systematic design methodology for converting data to prognostics information. The methodology is illustrated over four industrial use case scenarios.	The authors emphasize the need for a systematic method to develop and deploy a PHM systems, enabling rapid customization and integration of PHM systems for diverse applications. Some of the critical aspects to take into account include the identification of critical components, correct selection of appropriate algorithms for specific applications, as well as visualization for accurate decision-making support. One of the main research venues identified by the authors for future research includes self-maintenance, which refers to a machine's capacity perform regular quality and safety control on itself, detect anomalies and correct them on its own (based for instance on current health assessment of remaining useful life techniques).
Publication: (Lee et al., 2015)	**App.Field:** GEN
Defines a unified five level architecture for the design of Cyber-Physical Systems. This architecture aims to serve as a guideline for the implementation of Cyber-Physical Systems targeting an improvement in product quality as well as in system reliability.	Provides one possible definition of Cyber-Physical Systems based on the proposed architecture. It highlights the role that Cyber-Physical Systems can play towards achieving for instance plug and produce, multi-dimensional smart analytics and self-capabilities such as self-configuration or self-optimization.

Continued on next page

Table 3.3 – continued from previous page

Summary Analysis	Implications for Future Research
Publication: (Leitao et al., 2016)	**App.Field:** GEN
Discusses the role of multi-agent systems in regards to the implementation of industrial cyber-physical systems. A survey of the current state of the industrial application of agent technology presented, along with an outlook on the way agents can contribute to overcoming emerging challenges in Cyber-Physical Systems.	The authors state that multi-agent technology play an important role in enabling Cyber-Physical Systems to achieve flexibility, robustness, adaptation and reconfigurability, despite current TRL levels of deployed solutions being still relatively low. Smart production, smart electric grids and smart logistics are indicated as the main application areas.
Publication: (P. Li et al., 2018)	**App.Field:** M
A systematic methodology for ball screw prognosis is proposed, incorporating fault diagnosis, health assessment and remaining useful life prediction. The methodology is verified in a set of experiments designed for a ball screw test bed.	The practical applications of some machine learning algorithms are discussed, namely SVM for fault diagnosis, Linear Regression for health assessment and Gaussian Process for remaining useful life prediction.
Publication: (Liu, Dong, & Chen, 2018)	**App.Field:** M, SCH
Presents an integrated decision model combining job scheduling, predictive maintenance, prognostic information and resource planning into a complete scheduling and maintenance plan. The model was assessed through long-term wear test experiments conducted in a research laboratory setting.	Genetic Algorithms are suggested as a mature method to solve predictive system maintenance scheduling. It is stated that the proposed model could be improved and extended by addressing multiple machine systems, through the consideration of prognostics and diagnostics information.
Publication: (Loyer, Henriques, Fontul, & Wiseall, 2016)	**App.Field:** ECO
Compares the performance of five machine learning and statistical models on the estimation of manufacturing cost of jet engine components during the early design phase. The research is based on data from five large civil jet engines of one of the top five manufacturers worldwide.	Results indicate that common approaches for statistical cost modelling (such as multiple linear regression and artificial neural networks) tend to perform worse than more recent data mining and machine learning techniques such as gradient boosted trees and support vector regression.

Continued on next page

Table 3.3 – continued from previous page

Summary Analysis	Implications for Future Research
Publication: (Lu & Zhou, 2017)	**App.Field:** M, SCH
An opportunistic preventive maintenance scheduling methodology for serial-parallel multistage manufacturing systems is proposed, with the goal of improving final product quality and system reliability. The methodology is demonstrated in a case study focusing a serial-parallel three-stage machining system.	The proposed methodology highlights the importance of considering the multiple streams of deterioration when addressing the problem of quality improvement in multistage manufacturing systems. One short-coming pointed by the authors is the necessity to extend these types of approaches to consider reconfigurable manufacturing systems. Since the structure of such a system is constantly changing in response to varying product demand, it is crucial to employ a dynamic maintenance policy capable of adapting accordingly.
Publication: (Ma, Kwak, & Kim, 2014)	**App.Field:** LOG
A novel demand modeling technique based on demand trend mining for predictive product life cycle design is proposed. Based on this, a design framework is also presented. The model is illustrated through an example case of smartphone design.	Three different models are used to capture hidden and upcoming trends in demand, namely decision trees for large-scale data, discrete choice analysis for demand modeling and Hyndman's automatic time-series forecasting algorithm for trend analysis.
Publication: (Pandiyan, Caesarendra, Tjahjowidodo, & Tan, 2018)	**App.Field:** M
Describes the development of a predictive classification model for in-process sensing of abrasive belt wear based on SVM and a genetic algorithm. An experiment is conducted using four different conditions of tool states to assess the accuracy of the model.	A genetic algorithm based on a k-nearest neighbors classifier was employed to reduce the feature space considerably. Five models (SVM, kNN, ANN, Naïve Bayes and Bagged Trees) were compared in regards to their capacity to perform multi-classification of belt tool conditions, with quadratic SVM classifiers providing the best results.
Publication: (H. Peng & Van Houtum, 2016)	**App.Field:** M, ECO
Joint optimization model for the determination of production lot-sizing and condition-based maintenance policy. Through the optimization of two decision variables, the model minimizes setup cost per lot, inventory holding, lost sales and predictive/corrective maintenance costs.	The results imply that the determination of production lot size is dependent not only on the tradeoff between setup cost and inventory holding cost, but also the tradeoff between corrective and predictive maintenance costs. Further research is required considering dynamic schedules for production lot-sizing, as well as considering the health status of multiple machines.

Continued on next page

Table 3.3 – continued from previous page

Summary Analysis	Implications for Future Research
Publication: (Qi & Tao, 2018)	**App.Field:** GEN
Applications of big data and digital twin concepts in manufacturing are reviewed, focusing areas such as product design, production planning and predictive maintenance. The authors suggest that both concepts are complementary, and discuss ways in which they can be integrated to promote smart manufacturing.	The digital twin concept is suggested as a way to handle real-time and two-way mappings between physical objects and digital representations, paving the way for cyber-physical integration. As an example, the combination with big data can enable manufacturers to simulate, evaluate and improve production plans or maintenance operations in the virtual environment based in information from big data analysis without negatively impacting the factory.
Publication: (S. Wang, Wan, Zhang, Li, & Zhang, 2016)	**App.Field:** GEN
Smart factory MAS-CPS framework for distributed self-decision making in a self-organized system tested in a simulation environment.	Internet of Things, Big Data and Cloud Computing are identified as key-enablers of a Smart Factory environment; MAS-based CPS is suggested to be used to achieve high flexibility.
Publication: (Sahebjamnia, Tavakkoli-Moghaddam, & Ghorbani, 2016)	**App.Field:** QC
Proposes a fuzzy q-learning multi-agent quality control system for the control of continuous chemical production lines. The system presents self-learning capabilities, gradually forming its knowledge based on the results of the learning process. A real case study is presented focused on a cement industry factory.	The Java Agent Development Framework was used to implement the FIPA-compliant multi-agent system. The combination of the Drools rule engine with a Q-learning algorithm enables the agents to keep a current knowledge base for quality control based on reinforcement learning. The authors point out that future research should focus on generalizing or adapting the solution to expand it beyond the specific application presented in the article.
Publication: (Santos et al., 2017)	**App.Field:** GEN
Presents an architecture and respective proof-of-concept implementation of a big data analytics solution in the context of Industry 4.0. The implementation is tested in a case study based on Bosch Car Multimedia in Braga, Portugal.	The seven-layer architecture for big data analytics presented in the article offers some insights regarding potential key concepts and elements of these kinds of solutions, including real-time data stream brokers and NoSQL technologies to handle real-time data ingestion and storage, as well as tools like Spark and Tableau for the analysis and visualization of the results.

Continued on next page

Table 3.3 – continued from previous page

Summary Analysis	Implications for Future Research
Publication: (Susto, Schirru, Pampuri, McLoone, & Beghi, 2015)	**App.Field:** M
Methodology based on Multiple Classifiers (MC) for predictive maintenance with dynamic decision rules for maintenance management, focusing on integral type faults.	MC-PdM appears to consistently outperform PvM approaches. Furthermore, SVMs seem to offer superior performance compared to k-NN classifiers when implementing MC-PdM.
Publication: (Tao & Zhang, 2017)	**App.Field:** GEN
Explores the concept of digital twin shop-floor, discussing four key components, as well as operating mechanisms and implementation methods.	Identifies the physical shop-floor, virtual shop-floor, shop-floor service system, and shop-floor digital twin data as key components of the digital twin shop-floor concept. Additionally, some of the key enabling technologies mentioned include multi-agent systems, self-capabilities and data fusion.
Publication: (Upasani, Bakshi, Pandhare, & Lad, 2017)	**App.Field:** M, SCH
Describes a multi-agent based distributed approach for preventive maintenance scheduling, targeting identical parallel multi-component machines in a job-shop manufacturing scenario.	Highlights the importance of decentralizing the decision-making process into two levels, local machine level and global enterprise level, enabling the scalability of the solution. The authors also point out that the distributed approach makes it so that the solution can be further improved in future research through real time monitoring of machine health of various machines to incorporate prognostics in machine-level decision, leading to more accurate condition-based maintenance planning.
Publication: (Wan et al., 2017)	**App.Field:** M
Architecture for big data-based active preventive maintenance with both real-time and offline processing. Apache Kafka and Apache Storm are used to achieve distributed message communication and real-time big data processing.	Reinforces the importance of a reliable connection between the cyber and physical parts of the system, with real-time data collection acting as a key enabler of data-driven maintenance. Implies the importance of combining both online and offline processing.

Continued on next page

Table 3.3 – continued from previous page

Summary Analysis	Implications for Future Research
Publication: (D. Wu, Jennings, Terpenny, Gao, & Kumara, 2017)	**App.Field:** M
Introduces a prognostic method based on random forests for tool wear prediction, along with a comparison of its performance with that of feed-forward back propagation ANNs and SVR, based on experimental data collected from 315 milling tests.	The article touches on the subject of data-driven prognostics as opposed to model-driven, due to the fact that in complex manufacturing systems prior knowledge of the system's behavior is not always available. In this case, random forests presented superior accuracy at the expense of longer training times. The authors indicate that future research should focus on parallel implementation of ML algorithms to be applied in large-scale and real-time prognosis.
Publication: (D. Wu, Jennings, Terpenny, Kumara, & Gao, 2017)	**App.Field:** M
Introduces a cloud-based parallel machine learning algorithm, capable of training large-scale predictive models efficiently. A MapReduce-based implementation of parallel random forests on AmazonEC2 cloud is demonstrated using experimental data from milling tests.	Points out that one of the main challenges in data-driven prognostics advents from the need of large volumes of data, which then relates to computational efficiency when dealing with large volumes of real-time sensor data. The results suggest that cloud-based approaches can be used to tackle these challenges, as evidenced by the reduction of training times (about 15 times faster) when compared to previous work.
Publication: (D. Wu, Liu, et al., 2017)	**App.Field:** M
A computational framework for remote real-time sensing, monitoring, and scalable computing for data-driven diagnosis and prognosis is proposed. A prototype has been developed to support real-time, scalable, and plug-and-play data collection for both legacy and modern manufacturing machines, employing "drop-in" sensor nodes for the former.	The work showcased emphasizes the importance of developing flexible software solutions capable of taking advantage of such adaptable and pluggable data collection approaches to create modern intelligent predictive manufacturing systems. The authors reinforce the importance of further research to build predictive models using machine learning and integrate them into these online predictive systems.

Continued on next page

Table 3.3 – continued from previous page

Summary Analysis	Implications for Future Research
Publication: (X. Wang, Wang, & Qi, 2016)	**App.Field:** M
A distributed multi-agent reinforcement learning algorithm is proposed to address the problem of maintenance in a flow line system, focusing on obtaining control-limit maintenance policies for each machine associated to an observed state.	Multi-agent system technology is used, enabling the system to take into account the relation between the local decisions made by each agent and the overall optimization goal to produce the maintenance policies. Results suggest that cost-sharing reinforcement learning approaches might offer superior results than more traditional sequential preventive maintenance or even independent reinforcement learning algorithms.
Publication: (Xia, Jin, Xi, Zhang, & Ni, 2015)	**App.Field:** M, SCH
Proposes a real-time rolling grey forecast for machine health prediction to support dynamic maintenance scheduling.	Highlights the importance of accurate machine health prognosis forecasts for PdM schedules. Suggests through empirical analysis that grey forecasting models can achieve good precision accuracy even with limited data, making them suitable to meet real-time requirements.
Publication: (Yan et al., 2018)	**App.Field:** M
The article presents a concept of device electrocardiogram, as well as an algorithm based on deep learning for the prediction of the remaining useful life of industrial machines.	Deep learning, which has been shown to enable considerable advancements in fields such as computer vision and natural language processing, is suggested to have significant potential for industrial applications. One its main advantages is the capacity to automatically extract features without requiring extensive feature engineering relying on specific production scenarios.
Publication: (Zhang, Dubay, & Charest, 2015)	**App.Field:** QC
PCA model-based predictive control methodology for controlling part quality with the cooling cycle of injection molding processes. Performance was demonstrated in a LabWindows environment.	PCA and regression techniques are used to extract K principal components from measures on cavity pressure and temperature to predict the quality index. Authors suggest that due to the advancements of data mining techniques, the combination of statistical tools with model predictive control can serve to replace the need for offline quality measurements with real-time quality control.

Continued on next page

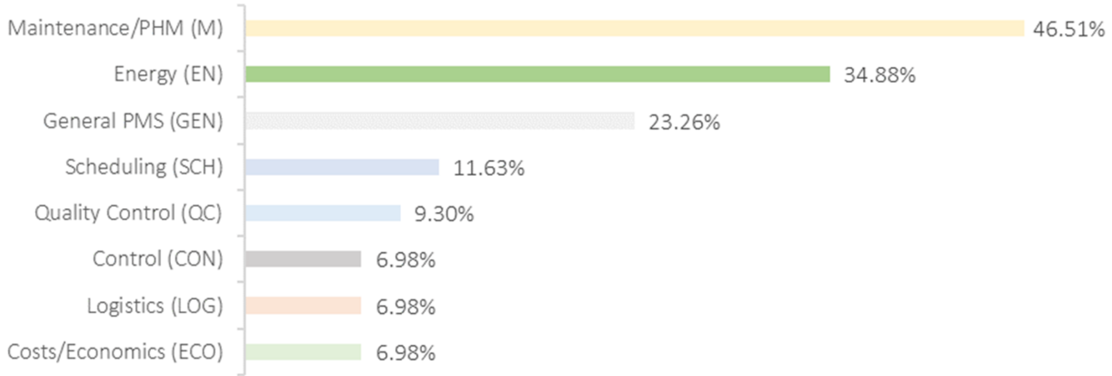

Figure 3.13: Distribution of the surveyed articles per focus area

Table 3.3 – continued from previous page

Summary Analysis	Implications for Future Research
Publication: (Zhong, Xu, Chen, & Huang, 2017)	**App.Field:** LOG
Presents a big data analytics approach for physical internet based logistics data generated by deploying RFID readers, tags and wireless communication networks on production shop floors.	Logistics data presents a considerable challenge in terms of complexity, not only due to the large amount of assets involved, but also the intrinsic dynamic logic of the logistics domain. According to the authors, some possible applications of big data analytics in this domain include visualization of logistic trajectories, evaluate the efficiency of operators and operations, as well as logistics planning and scheduling.
Publication: (Zurita, Delgado, Carino, Ortega, & Clerc, 2016)	**App.Field:** GEN
Introduces a neo-fuzzy neuron method to be applied in industrial time series modeling. The proposed method is validated through the modelling of a critical signal regarding copper refrigeration, based on real data from a Spanish copper rod industrial plant.	The authors discuss argue data-driven solutions require considerable data comprehension and volume, which slows convergence of the solution and tends to mask relation between the inputs and the target, which can turn into a limitation when considering a high amount of inputs such as in the case of industrial time series. Neo-Fuzzy Neuron modelling is then proposed as a method to be explored in this context to tackle such a challenge.

The distribution of the articles presented among the different journals is presented in Figure 3.13, based on the application field classification attributed in Table 3.3.

From Figure 3.13 it is possible to observe that there is a clear focus into the research of maintenance and PHM solutions in predictive manufacturing (46.51%), as well as energy optimization and management (34.88%). In fact, these fields are often related and even complementary, since energy consumption monitoring is often used as an indicator of machine health. The remainder of Section 3 provides further insight into two main concepts which are central to this thesis. The first of which is Industry 4.0, which frames the context in which the present work is being developed, not only from a technological point of view but also from a manufacturing market standpoint. Secondly, the concept of Cyber-Physical Production System is described, as an extension of typical CPS applied to a manufacturing context. Finally, the concept of PMS is thoroughly discussed, being the main focus of the work presented herein.

3.2 Industry 4.0

With the advancements in the ICT field and the exponentially increasing volumes of data being generated every day, a new set of possibilities has been presented to improve the efficiency and the characteristics of production processes. Adding to this is the transformation from a saturated seller's market into a customer-driven one, with its growing demand for highly customized products accompanied by decreasing product lifecycles and smaller lot sizes, pushing companies towards a paradigm shift in order to leverage their data to attain a business advantage in such a competitive and dynamic market (Brettel, Friederichsen, Keller, & Rosenberg, 2014).

As such, the currently ongoing 4th industrial revolution, usually referred to as Industry 4.0 in Europe (Deloitte, 2014) (Gilchrist, 2016) (Kagermann, Helbig, Hellinger, & Wahlster, 2013) and Industrial Internet in the US (Consortium, 2015), aims to introduce and take advantage of the interconnected world along the entire value chain, allowing the sharing and processing of the data available in all of the its actors to generate relevant knowledge and optimize the overall process. The adoption of the Industry 4.0 paradigm encompasses the following three characteristics (Stock & Seliger, 2016):

- "*Horizontal integration across the entire value network*": By integrating the overall value chain it is possible to optimize it beginning with the suppliers, materials, logistics, etc. In this sense, all of the value chain's actors must be connected and coordinated among each other based on their individual requirements, creating a very dynamic ecosystem.

- "*End-to-end engineering across the entire product life-cycle*": The integration and digitalization across all phases of the product's life-cycle is crucial to ensure that data can be collected, stored and processed to generate new knowledge from the product's inception to its end of life. This knowledge can be particularly relevant for the product's improvement, not only regarding its production, but also the for instance its design or material suppliers.

- "*Vertical integration and networked manufacturing systems*": At the shop-floor, the integration among the different components and actors (such as resources, humans and Manufacturing Execution Systems) should be done through a Cyber-Physical System (CPS). This system will allow not only the internal integration and optimization but also a harmonized and smooth integration with the two previously presented functionalities.

With the vertical integration emerges the concept of Smart Factory (SF). According to (Shrouf, Ordieres, & Miragliotta, 2014) these factories must have some characteristics including among others the capacity to deal with mass customization (Kagermann et al., 2013), flexibility (Zuehlke, 2010) and new maintenance strategies (Mobley, 2002), as well as leveraging big data into a business advantage through the generation of new and relevant knowledge and insights (Lee et al., 2013). This plethora of research venues and the growing interest in the fields surrounding Industry 4.0 is evident when looking at the co-citation map related to this topic. This map allows a quick identification of major players/publications regarding Industry 4.0, on top of providing some insight into the different related groups that can be formed throughout the various branches of research that are related to this field. A possible co-citation map illustrating this can be seen in Figure 3.14.

The co-citation map provided in Figure 3.14 was elaborated based on data from the Web of Science repository. Firstly, the full records of all publications between 2010 and 2018 found on the topic of Industry 4.0 were extracted into a CSV file, which was then fed into a Python script to generate the network edge list. Afterwards, the data was imported into Gephi (Bastian et al., 2009) in order to generate the visualization and manipulate it to identify the most influential nodes.

Finally, community detection is performed using the Louvain method (Blondel et al., 2008), with the resulting communities being attributed a distinct color in the visualization.

From a quick glance at Figure 3.14 two main publications stand out, the final technical report from the Industry 4.0 workgroup (Kagermann et al., 2013), which provides the main guidelines for the implementation of Industry 4.0, and a publication from the University of Cincinnati (Lee et al., 2015) focusing on a possible definition for CPS and guidelines for their implementation in this context, which was also included in the survey documented in Section 3.1.3.

Furthermore, five main communities can be identified. Upon closer inspection of each of these communities, their areas of focus appear to be as follows:

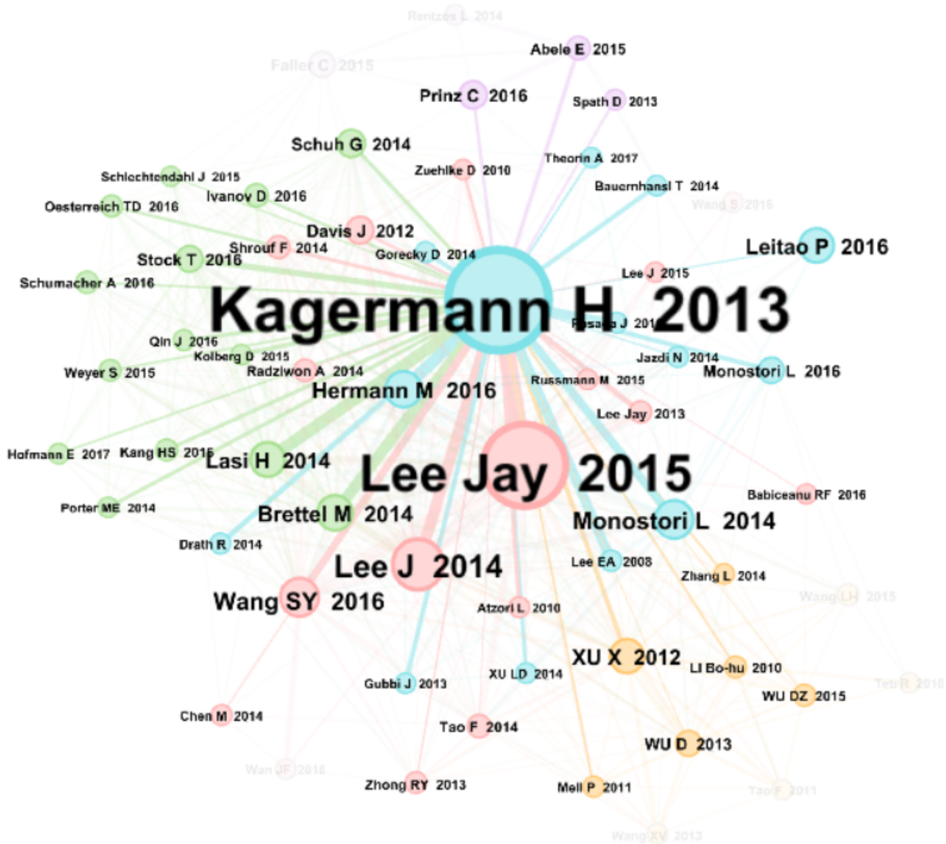

Figure 3.14: Co-citation map for Industry 4.0

and - These communities focus on discussing a mix of design principles and guidelines for either the implementation of Industry 4.0 systems as a whole, or focusing specifically in the aspect of CPS.

- The community represented in yellow presents a clear emphasis on cloud computing, particularly considering the application for cloud manufacturing.

- This group appears to focus mostly on a higher abstraction level, dealing mainly with enterprise management, sustainable manufacturing, and in some cases supply-chain management and planning.

- Lastly, this community's main topic seems to be learning factories, mainly as a way to develop theoretical and practical knowledge in a real production environment in order to pave the way for the adoption of Industry 4.0 solutions.

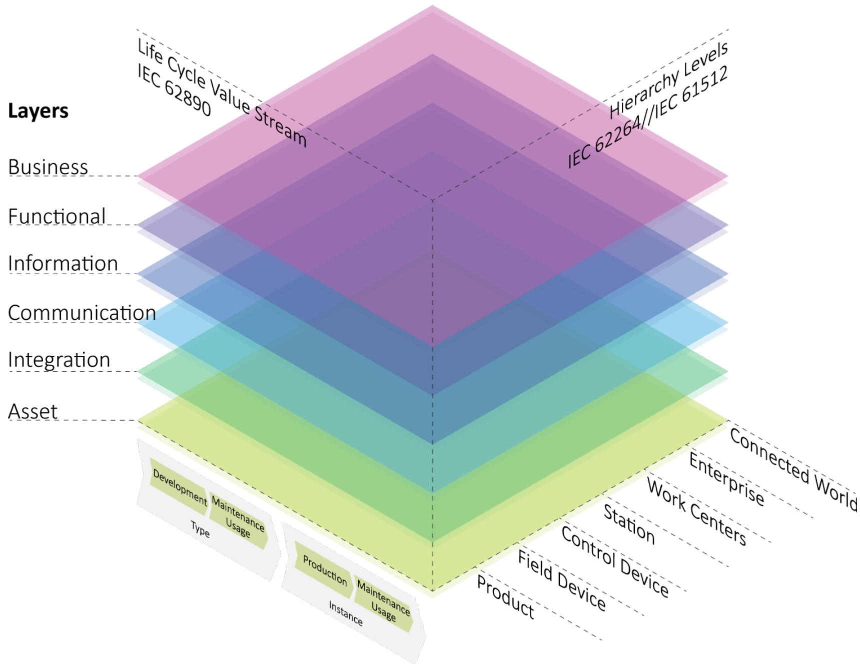

Figure 3.15: Reference Architecture Model fo Industrie 4.0 (RAMI4.0)

3.2.1 Reference Architectures for Industry 4.0

Reference Architecture Model Industrie 4.0 (RAMI4.0)

Reference Architectural Model for Industry 4.0 (RAMI4.0) consists in a three-dimensional construct that encompasses the key aspects of Industry 4.0 as envisioned by the European Plattform Industrie 4.0 network. The core idea behind it is to serve as a reference to harmonize the interests of the different industries involved in the discussions, from process to factory automation, along with each of their different standards, information and communication technologies into a common understanding. Hence, tasks and workflows can be broken down into more manageable parts, with the model serving as a way to identify and assess the fit of existing standards, gaps, possible use cases and relationships among the different subspaces of the reference architecture. A visual representation of this three-dimensional model is provided in Figure 3.15.

To better understand this three-dimensional model it is important to look into each of its axes individually. The *layers* in the vertical axis mimic the ICT approach of splitting up complex projects into groups of smaller, more manageable parts. This includes for instance business goals, functional descriptions, information structure and varied assets. In this way it is possible to describe how development processes, production lines, manufacturing assets and even the products are configured, how they function and how their information can be made available as a virtual representation. In order to maintain a high-level cohesion between the layers with loose connections between them, events should only be exchanged either within each layer or between two adjacent layers.

Starting from the top, the Business layer ensures the integrity of the functions in the value stream, being responsible for mapping the business models and goals to the resulting overall process, including modelling the rules which the system has to follow and orchestrating the services of the Functional layer. Contrary to popular belief, it does not concern concrete systems, such as an Enterprise Resource Planning (ERP) system, as ERP functions would typically reside in the Functional layer. This, in turn, deals with the actual description of the different functions acting as a runtime and modelling environment for the services which support the business processes, along with the applications and technical functionalities. While the actual decision-making logic should be generated inside the Functional layer, part of it can also be executed in the lower layers (e.g. Information or Integration layers) as deemed necessary by a particular use case. The aspects of horizontal integration and remote access are also exclusively within the scope of this layer in order to ensure the integrity of the information and conditions of the technical level. Exceptionally, the Asset and Integration layers may also be accessed temporarily for maintenance purposes as long as this access is not relevant to permanent functional or horizontal integration.

Data persistence and structuring is handled by the Information layer. In it data events can be pre-processed and transformed to generate data as expected by the Functional layer, which is then provisioned via services. In this context, it is also within the Information layer that event-related rules are formally described and then executed in order to process incoming data events. To facilitate this process, the Communication layer provides standardized means for this interaction to occur using an uniform data format in the direction of the Information layer.

In this context of virtualization, every event and system actor in the physical world should have a corresponding representation in the virtual world, which in RAMI4.0's case is facilitated by the Integration layer. This cyber-physical dichotomy is handled in such a way that if something in the real world changes, the corresponding asset's event is appropriately reported to the Integration layer, which can then trigger events signalled to the Information layer through the Communication layer. This integration encompasses elements connected with Information Technology (IT) including for instance RFID readers, sensors and Human-Machine Interface (HMI) components, given that the interaction with humans is also part of this level of abstraction. Lastly, the physical elements of the real world are represented in the Asset layer, which includes not only manufacturing elements such as machines, parts and documents, but also humans themselves who are abstracted in the virtual world via the Integration layer, thus concluding the description of model's vertical dimension.

Another critical aspect to take into account within Industry 4.0 is that of the *life cycle* of products, machines, factories and other elements, as well as the associated *value stream*, represented along the horizontal axis on the left-side of Figure 3.15. IEC 62890 is said to provide the guidelines for consideration of this aspect, particularly with the fundamental

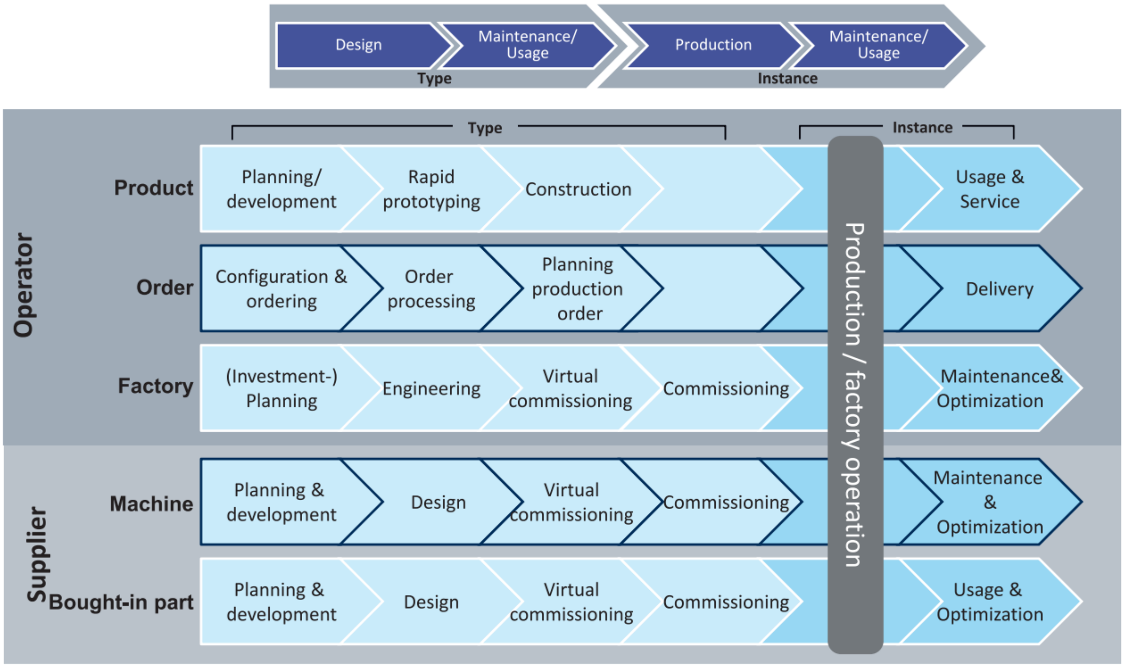

Figure 3.16: Relevant life cycles for I4.0 components based on IEC 62890. Source: (Plattform Industrie 4.0, 2015)

distinction between type and instance which is a central part of this dimension in RAMI4.0.

In this context, a type is created as the product comes into being in its design and development phase. Hence, this entails the placing of design orders, its development and testing up to the production of the first prototype. Once all tests and validation are concluded, the type is release for actual production, with manufactured products representing an uniquely identified (e.g. serial number) instance of that type. Such instances are sold and delivered to customers, from whose perspective they are once again types which become instances once they are installed in a particular system. This cycle of changing from type to instance or from instance to type can be repeated several times, for instance when improvements or updates are made to certain products. Figure 3.16 provides an example of the assignment of types and instances to the life cycle, illustrating its various dimensions of relevance for this context.

It is worthwhile to also reinforce the importance of horizontal integration across the entire value stream. As an example, logistics data can be used in assembly, inventories can be monitored in real-time, the location of parts from suppliers can be known at any point in time and the customer can check the completion status of a product they ordered during production. This presents a huge potential for improvement and innovation, suggesting that the life cycle should be linked together with the value-adding processes it contains, from engineering to suppliers and customers, not from the isolated point of view of a single factory.

Lastly, the *functional hierarchy* within the factories is represented in the right-most horizontal axis. It describes the functional classification within the factory following the IEC 62264 (International Electrotechnical Commission, n.d.-b) and IEC 61512 (International Electrotechnical Commission, n.d.-a) standards. In order to cover varied sectors, ranging from process industry to factory automation, the terms "Enterprise", "Work Unit", "Station"and "Control Device"were adopted. To better reflect the context of Industry 4.0 and the considerations within a machine or system, the "Field Device"level was added below Control Device, representing the functional level of an intelligent field device such as a smart sensor. Additionally, to emphasize the role of the product to be manufactured, an extra level was added at the bottom to permit the homogeneous consideration of the product and its production environment along with their respective interdependencies. Finally, an extension was also made to the upper limit of the hierarchy to accommodate the interactions above the Enterprise level, the "Connected World", since the aforementioned IEC standards only represent the levels within a factory. This enables for instance the description of groups of factories and the collaboration with external engineering partners, component suppliers and customers.

Industrial Internet Reference Architecture (IIRA)

The Industrial Internet Reference Architecture (IIRA) is a standards-based open architecture for industrial IoT systems. It aims to boast a broad industry applicability through an high-level of abstraction and generic nature, promoting interoperability, the mapping of applicable technologies and the guidance of technology and standard development (Industrial Internet Consortium, 2019).

The IIRA combines and abstracts common characteristics, features and patterns from several use cases and is continuously refined and revised as a result of its application in testbeds developed by the Industrial Internet Consortium and the real-world deployment of industrial IoT solutions.

The reference architecture is represented in Figure 3.17 with its different viewpoints, being an architectural framework and methodology for system conceptualization, accentuating important system concerns which may affect the lifecycle process.

The IIRA viewpoints are split into four distinct layers, namely concerning business, usage, functional and implementation aspects. These viewpoints are the foundation which enables a layered analysis of individual sets of industrial IoT concerns, relating them to specific stages of the lifecycle process and within the context of different industrial sectors. These can then be extended or further fleshed out based on specific system requirements.

The higher level *Business Viewpoint* encompasses the identification of the stakeholders and the specification of their business goals regarding the development of an industrial IoT system in their business context. To this end it further details how the system should

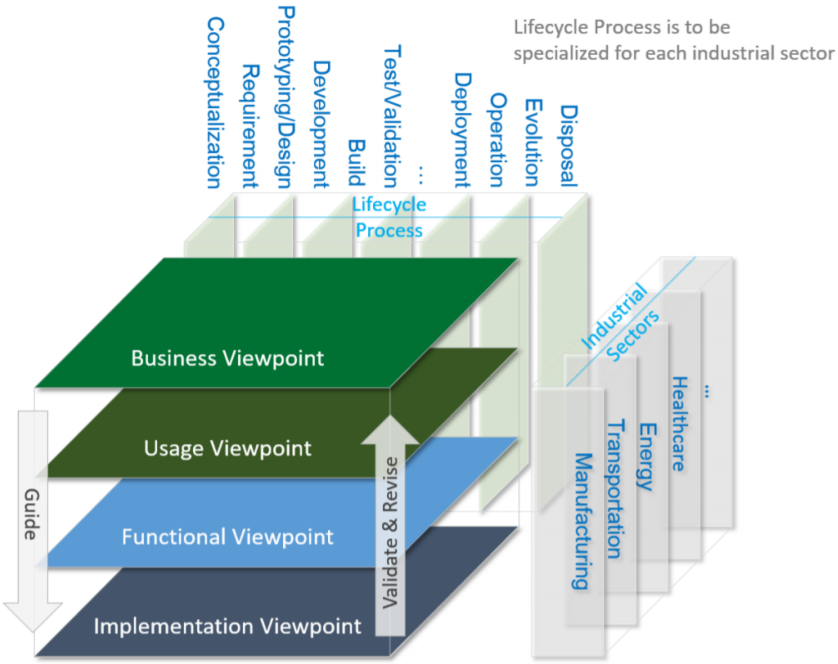

Figure 3.17: Industrial Internet Reference Architecture (IIRA). Source: (Industrial Internet Consortium, 2019)

achieve the aforementioned goals by mapping it to its base system capabilities. As such, this layer is, as the name suggests, mostly business-driven and thus involving business decision-makers, system engineers and managers.

Following this, the *Usage Viewpoint* refers to the expected system usage, hence describing sequences of activities involving the system and its components that are responsible for providing the functionality required to achieve the core system capabilities. Stakeholders at this stage include system engineers and other individuals involved in the system specification.

In this direction, the *Functional Viewpoint* is centred on the actual functional components of the system, their structure and all of the aspects related to interoperability between the system, its components and other external elements in the environment such as interfaces and the specific interactions necessary to support the overall system's activities. Naturally, the main stakeholders in this case are system and component architects, developers and integrators.

At bottom lies the *Implementation Viewpoint*, dealing with the technologies necessary to implement the components from the previous layer, along with their communications and lifecycle procedures. These elements are organized by the activity sequences originating from the Usage Viewpoint and oriented towards supporting the system capabilities defined in the Business Viewpoint. Thus, the involved stakeholders include not only those of the

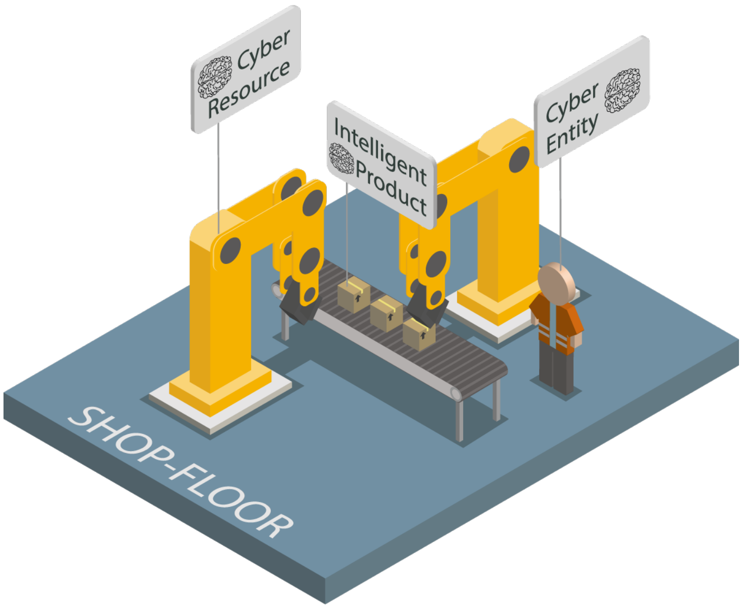

Figure 3.18: Cyber-Physical Production Systems

Functional Viewpoint, but also the system operators themselves.

Summarizing, the reference architecture aims to guide system designers and developers iteratively through the system's conception, design and its implementation, taking into account concerns beyond the design phase of the system and into its full lifecycle.

3.3 Cyber-Physical Production Systems

As previously stated, SFs are implemented and deployed via CPS. The concept of CPPS (Ribeiro, 2017) merges the functionalities and benefits of CPS applied to the industrial context. The main objective in a CPPS is to create an abstraction layer where each of the shop-floor's actors is represented by a cyber entity, as shown in Figure 3.18.

The communication between the heterogeneous components is now made at the cyber level, allowing a smooth and effective integration of all the components and actors avoiding the usual problems related to vendors' specifications and standards. In (Lee et al., 2015) the authors present a comparison between today's factories and the now emerging Industry 4.0 based SFs, implemented through CPPS. With all the resources integrated, sharing information and their behaviors among each other, the shop-floor can adapt and organize in runtime to optimize at different levels (production, maintenance, energy consumption, etc). Moreover, with the advances in the industrial IoT and the increasingly large number of sensors and other data sources available on the shop-floor, the amount of extracted data is growing and the traditional algorithms are no longer able to process these volumes of data. Hence, the big data analysis field is becoming more and more important in several areas as a way to tackle this challenge (Babiceanu & Seker, 2016).

This is often coupled with the usage of ML, allowing manufacturers to obtain insights regarding their factory which would have been otherwise missed. ML can be defined as a system's capacity to improve its performance on a given task or set of tasks over time based on previous results (Patcha & Park, 2007). Therefore, ML algorithms can be used to predict a system's behaviour and/or improve its overall performance, enabling the development of tools capable of analysing data and perceive what are the underlying trends and correlations. Thus, ML-based approaches can be used to predict abnormal events (failures, degradation, energy consumption, etc), generate warnings and advise the system and/or the operator regarding which course of action to take, assisting in diagnosis and maintenance tasks.

It can be said that generally speaking CPPS remain in a relatively infant stage of research and development, with considerable challenges still left to tackle in the coming years including for instance those related with real-time constraints, reliability and security. In this regard, in (Leitão, Colombo, & Karnouskos, 2016) an assessment of the core current challenges for future research on this topic is presented, providing a list of each individual challenge along with its associated difficulty, priority for its resolution and the estimated time required to reach a stable and mature solution. An adaptation of this assessment can be found in Table 3.4.

From the analysis of this table and particularly of the authors' own analysis documented in (Leitão et al., 2016), some very pertinent opportunities can be derived. Firstly, while these areas are all relevant and transversal to the design and development of CPPS, it is important to identify the key areas which directly relate the most with the focus of the present work on PMS, being those of *CPPS Capabilities* and *CPPS Information Systems*.

In terms of the area of capabilities, the authors highlight several issues related to real-time monitoring and control in CPPS as well as the larger and broader systems of CPPS and their optimization, which are regarded as highly challenging. From the perspective of information systems aiming to capitalize on the generated manufacturing data, information and knowledge, several challenges also need to be addressed, mostly in the long-term horizon and with considerable hardships as stressed by the authors, in turn resulting in different priorities for each.

Initially, there is a larger concern with the mid-term goal of transforming CPPS data and information analytics into actionable knowledge, which obviously ties into the integration of artificial intelligence and particularly ML in CPPS. As follow-up actions in the long-term, the cross-domain information integration and knowledge-driven decision making and management are then considered. This also matches and aligns with the emerging trends identified in Section 3.1.

Table 3.4: Key challenges in industrial CPPS. Adapted from: (Leitão, Colombo, & Karnouskos, 2016)

Area	Key Challenges	Difficulty	Priority	Maturity in
CPPS Capabilities	Real-time control of CPPS systems	High	High	4-7 Years
	Real-time systems of CPPS	High	Medium	3–5 years
	Optimization in CPPS and their application	High	Medium	4–7 years
	On-CPS advanced analytics	Medium	High	3–5 years
	Modularization and servification of CPPS	Low	High	3–5 years
	Energy efficient CPS	Medium	Medium	3–5 years
CPPS Management	Lifecycle management of CPPS	Medium	Medium	5–8 years
	Management of (very) large scale CPPS and system of CPS	High	High	5–8 years
	Security and trust management for heterogeneous CPS	High	High	5–8 years
CPPS Engineering	Safe programming and validation of systems of CPPS	High	High	5–10+ years
	Resilient risk-mitigating CPPS	High	High	5–10+ years
	Methods and tools for CPPS lifecycle support	High	High	3–7 years
	New operating systems and programming languages for CPPS and systems of CPPS	Medium	Low	3–6 years
	Simulation of CPPS and of systems of CPPS	Medium	High	3–6 years
CPPS Infrastructures	Interoperable CPPS services	Medium	High	2–5 years
	Migration solutions to emerging CPPS infrastructures	Medium	High	3–6 years
	Integration of heterogeneous/mobile hardware and software technologies in CPPS	Low	Medium	2–4 years
	Provision of ubiquitous CPPS services	Medium	Medium	3–5 years
	Economic impact of CPPS Infrastructure	High	High	3–6 years
CPPS Ecosystems	Autonomous and self-* CPS	High	Medium	7–10+ years
	Emergent behaviour of CPPS	High	Medium	7–10+ years
	CPPS with humans in the loop	High	High	2–5 years
	Collaborative CPPS	Medium	Medium	5–8 years
CPPS Information Systems	Artificial intelligence in CPPS	High	High	7–10+ years
	Cross-domain large-scale information integration to CPPS infrastructures	Medium	Low	6–9 years
	Transformation of CPPS data and information analytics to actionable knowledge	High	High	4–8 years
	Knowledge-driven decision making/management	High	Medium	6–10+ years

3.4 Industrial Artificial Intelligence

The discipline of Artificial Intelligence (AI) is typically connected to the study of cognitive phenomena in machines, ultimately implementing aspects akin to human capacity or even surpassing it in areas such as ML, image processing and NLP.

Industrial AI in particular aims to bridge the gap between academic research efforts in AI and the industry, mostly focusing on the development, validation and deployment of varied ML algorithms in industrial applications to improve value creation, productivity and reduce costs through additional insights into production.

In (Lee, Davari, Singh, & Pandhare, 2018) the authors characterize Industrial AI using five key elements, namely analytics, big data, cloud or cyber technology, domain knowledge and evidence. While analytics can be seen as the core of Industrial AI, it needs the support of the remaining elements to create value. Big data and cloud/cyber technologies are both presented as crucial elements, providing both the source of the information and platform for Industrial AI. Complementary, domain knowledge is essential for understanding the problem and focusing the application of Industrial AI to solve it, despite also being one of the most overlooked aspects in this domain. It provides the necessary system comprehension so that the right data can be collected with adequate quality and ensures that the physical meanings of the studied parameters, their associations to the physical characteristics of a system or process and their variations can be taken into account in the appropriate context. Lastly, evidence relates to the data patterns and labels gathered over time, allowing models to be improved, become more accurate and robust as time passes.

An overview of the current Industrial AI ecosystem, encompassing its needs, challenges, enabling technologies and main methodologies can be found in Figure 3.19:

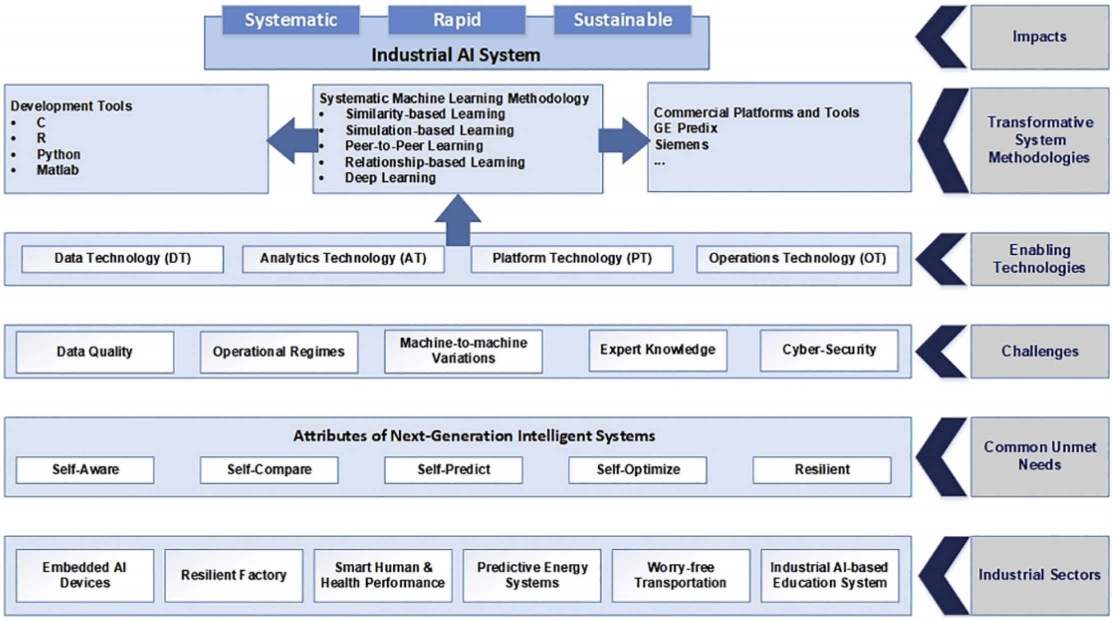

Figure 3.19: Industrial AI ecosystem (Lee, Davari, Singh, & Pandhare, 2018)

Focusing particularly on the main current challenges faced in this domain, three can be highlighted as being highly prioritized in current research and industry efforts, namely machine-to-machine interactions, data quality and cyber-security.

Machine-to-machine interactions impact AI algorithms in the sense that these are susceptible to small variations in their data stemming from variations from machine to machine, or even in the same one over time. Furthermore, it is important to ensure that AI solutions do not interfere or conflict with the operation of other systems, being as minimally invasive as possible.

This also relates to the second point of data quality, as learning from improper or inaccurate data sets can severely impact the performance of the AI models and produce flawed results.

Last but not least, the recent increase in connectivity with the advent of cloud and IoT-based solutions makes smart manufacturing systems vulnerable to cyber attacks. As an emerging issue, it is generally considered that the industry is yet to be fully prepared to face these security threats (Tuptuk & Hailes, 2018).

3.5 Predictive Manufacturing Systems (PMS)

Predictive Manufacturing Systems can be defined as a paradigm whose goal is to empower machines and systems with self-aware capabilities (Lee et al., 2013), enabling them to for instance estimate their own condition, detect the presence of defects or anomalies, forecast future production events or even perform self-maintenance.

In the present section, a brief overview of existing approaches and applications is provided based on the results from Section 3.1, framing afterwards the relation to the current work in the form of existing gaps and research opportunities.

3.5.1 Existing Applications and Approaches

As discussed in Section 3.1, over the last few years, significant efforts have been put into the research of the various facets of predictive manufacturing in Industry 4.0. In (Lee, Bagheri, & Kao, 2014) the authors overview the recent advances and trends regarding CPPS and big data analysis, identifying self-predictiveness and self-awareness as key characteristics to gain insight into Industry 4.0 factories. Also, the authors mention that several sources of information in current prognostics methods remain untapped, such as peer-to-peer evaluation and historical life-cycle information from identical assets. Insightful discussions and guidelines regarding solutions for PMS can also be found in the current literature (Canito, Fernandes, & Pra, 2017) (Siafara, Kholerdi, Bratukhin, Taherinejad, & Wendt, 2017), with some common denominators including the employment of CPPS for virtualization, ML models for data analysis (e.g. early fault detection, quality control), decentralization and self-adjustment. However, the discussions are often on the conceptual or architectural level, with a lack of deployable implementations or results.

Regardless, the growing importance of PMS in the current information age is evident from its multitude of research venues. In (Lechevalier, Narayanan, & Rachuri, 2015), the authors survey several articles pertaining to the applications of PMS and propose to group them into four main application fields, namely system control (J. Li & Shi, 2007), quality control (Penya, Bringas, & Zabala, 2008), fault diagnosis (Chan & McNaught, 2008), and predictive maintenance (K. Wang, 2016). Other recent examples include an architecture for predictive maintenance as a service based on the cloud computing paradigm (Terrissa, Meraghni, Bouzidi, & Zerhouni, 2016), the prediction of power consumption levels in machining processes through BDA (Shin, Woo, & Rachuri, 2014) and a distributed multi-agent oriented framework for failure prediction from real-time sensor data (R. S. Peres, Rocha, & Barata, 2017).

Some frameworks for the application of predictive analytics in manufacturing environments have also been proposed. In (K. Wang, 2016), a CPPS-based framework for intelligent predictive maintenance is presented, using mostly the processing and feature extraction of real-time signal information as enablers of fault diagnosis and prognosis. (Ge, 2018) presents a distributed framework for the prediction and diagnosis of key performance indices in plant-wide industrial processes. One key aspect is the division of the entire process into several smaller blocks, later enabling data to be extracted more efficiently whilst greatly reducing its dimensionality. A fog computing-based framework for data-driven machine health and process monitoring in CPPS is introduced in (D. Wu, Jennings, Terpenny, Gao, & Kumara, 2017), highlighting the importance of scalable, high-performance approaches and the usage of cloud-based ML algorithms for predictive analytics. (Zhong et al., 2017) introduce a BDA framework for radio-frequency identification-enabled shop-floor logistics, which presents a considerable challenge in terms of complexity not only due to the large amount of assets involved, but also the intrinsic dynamic logic of the logistics domain.

3.5.2 Relation to Current Work

The main challenges, and coincidentally the main gaps, in PMS research stem from the fact that most of the work found in current literature is based on a set of assumptions regarding system behaviour that do not always hold true in real manufacturing scenarios.

One such a case relates to the necessity of having in depth knowledge of the way the system behaves beforehand, akin to the application of model-driven approaches. In reality, a-priori knowledge regarding system behaviour is not always available, which makes it often difficult to apply model-based predictive solutions. In this sense data-driven approaches can tackle this problem, but require large volumes of data which in turn demand higher computation power. Consequently, distributed and cloud-based implementations can assist in overcoming this pitfall, providing the necessary computational efficiency and making the system scalable.

However, besides this knowledge-centric gap, real systems often require changes and adaptations during execution in response to a quickly varying market demand. Current solutions are mostly based on the assumption that these systems remain the same and are incapable of providing the required degree of flexibility, thus further research is needed to support on-line adaptation and reconfigurability. One concept of particular interest in this regard is the capacity to implement the plug & produce paradigm, not only on a hardware level (e.g. plugging and unplugging physical assets during execution with minimal downtime) but also on a software level (e.g. change/adapt deployed ML models or data acquisition mechanisms in runtime in a manner that is transparent to the underlying system), hence implying a modular approach for the design of intelligent predictive manufacturing solutions.

Furthermore, despite the fact that some architectures and frameworks can be found in the literature, they are commonly application specific, being overly focused on a specific problem. There is no reference for a generic implementation capable of tackling a broad array of challenges in predictive manufacturing whilst requiring minimal changes to the overall solution.

Overall, in the context of Industry 4.0 systems, new solutions should be flexible to cope with topological (i.e. Plug-and-Produce) or requirement changes on the shop-floor, as well as scalable and high-performing in order to deal with the growing volumes of data. Moreover, it can be said that there is still a clear need to further combine real-time streams of data from the shop-floor with historical data at both the resource and system levels, as well as closing the loop to autonomously act on the results of the predictive analytics (i.e. self-maintenance). These solutions should also be highly adaptable, being capable of changing even after deployment by learning from newly generated knowledge (Lee, Kao, & Yang, 2014), adapting at both the analysis and action fronts. This implies a continuous adaptation and dynamic improvement of their self-adjustment mechanisms during execution, avoiding unnecessary downtime for redeployment and additional programming effort on the deployed system. Finally, the generalization of the solutions should also be taken into account, so that they can be easily migrated and applied to wide array of manufacturing scenarios and domains.

CHAPTER 4

The IDARTS Framework

The present chapter entails the full description of the proposed IDARTS framework for the development of generic PMS, which aims to answer the research questions put forward in Chapter 2.

To this end, first and foremost an overview of the field of Requirements Engineering (RE) is provided, along with some supporting concepts to ease the reader into the specification of the requirements for IDARTS and its PMS. Afterwards, the general design of the framework is presented, followed by a mapping of this design onto the RAMI4.0 reference model. Finally a more detailed description of each of IDARTS modules is provided along with their relation to the requirements that were previously established.

In summary, the aimed contribution of the present chapter is twofold, consisting more specifically of:

- A comprehensive list of the goals and both functional and non-functional requirements that should drive the design and implementation of generic and intelligent PMS.
- The proposition of a predictive manufacturing framework, to be used as the foundation for the implementation of systems of this nature through the thorough formalization of its fundamental constituting principles and modules, as well as of the communications, data flow and necessary interfaces between them;

Therefore, based on these two main contributions, the goal for the coming chapters is not only to achieve the implementation of an intelligent PMS (to be later addressed in Chapter 5) which can be applied to different scenarios, such as quality control or predictive

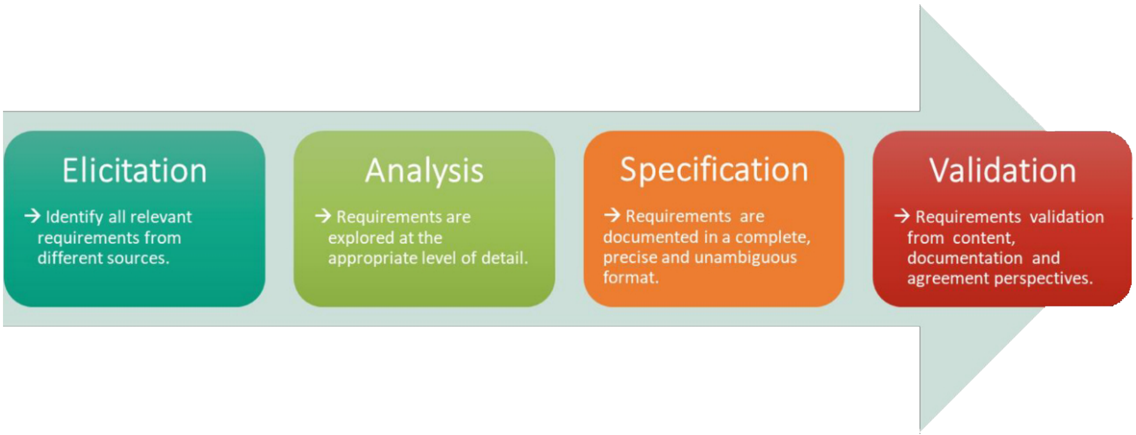

Figure 4.1: The requirements engineering process.

maintenance, but also to provide a formalized framework that can be used by others as the guideline for the implementation of such systems, as well as contributing to the state of the art and research developments in the field of predictive manufacturing.

4.1 Requirements Engineering

As a discipline, RE plays a pivotal role in understanding, designing and developing CPPS systems and, by extension, modern PMS as well. Borrowing one possible definition from software engineering, it can be seen as an iterative, cooperative and incremental process of discovery, analysis, specification and validation of the functionalities and constraints related to the operation and development of such systems (Sommerville, 2011), which is further illustrated in Figure 4.1.

Moreover, it is also crucial to provide a clear definition of requirement in the context of the present work, as the term is often used loosely in the literature with different meanings and degrees of detail attached to it. Within this context, the definition provided in the (ISO/IEC/IEEE, 2017) standard has been adopted, as formalized in Definition 4.1.

Definition 4.1 (***Requirement***)**.** *1. Statement that translates or expresses a need and its associated constraints and conditions. 2. Condition or capability that must be met or possessed by a system, system component, product, or service to satisfy an agreement, standard, specification, or other formally imposed documents 3. Provision that contains criteria to be fulfilled 4. A condition or capability that must be present in a product, service, or result to satisfy a contract or other formally imposed specification (ISO/IEC/IEEE, 2017).*

Going in more depth, requirements can be further categorized into Functional Requirement (FR) and Non-Functional Requirement (NFR). FRs deal with the services and functions that a given system should perform, as referenced in Definition 4.2.

Definition 4.2 (***Functional Requirement***)**.** *1. Statement that identifies what results a product or process shall produce 2. Requirement that specifies a function that a system or system component shall perform (ISO/IEC/IEEE, 2017).*

Contrastingly, NFRs are constraints on the functions offered by the system, which can include time constraints, constraints on the development process or those imposed by standards (Sommerville, 2011), as reflected in Definition 4.3. These often apply to the system as a whole, rather than on individual components or functionalities.

Definition 4.3 (***Non-Functional Requirement***)**.** *1. Software requirement that describes not what the software will do but how the software will do it (ISO/IEC/IEEE, 2017).*

Nevertheless, one cannot distinguish between the different categories of requirements as easily as these definitions seem to imply. While a given requirement concerned for instance with real-time constraints may appear to be a mere non-functional requirement, once developed in further detail it may generate others which are clearly functional. An example could be the need to include real-time data acquisition functionalities in the system. This illustrates that requirements are not necessarily independent and often generate or constrain other requirements, thus not only specifying the services or features that are required, but also the necessary functionality to ensure their proper delivery.

4.1.1 IDARTS' Goals and Requirement Sources

Before the requirements for the system and its respective processes can be analyzed, it is necessary to create the foundation that will support this activity. This consists in clearly defining the context and the boundaries of the system to be developed, defining goals and identifying requirement sources such as stakeholders, systems in operation (i.e. similar or legacy) and documentation. Once these activities are completed one can move onto the elicitation process and the analysis of the requirements that follows it (Rupp & die SOPHISTen, 2014). With that being said, in this case it is hardly possible to record all requirement sources during the initial goal identification without a subsequent analysis of the existing systems in which an IDARTS implementation would be deployed, meaning that a continuous update of this list is to be expected.

In RE it is common for stakeholders to confuse the concepts of requirements and goals. Goals are high-level objectives of a business, organization or system, while requirements pertain to how a given goal should be achieved by a proposed system. In addition, goals should adhere to certain quality criteria to be considered within this scope. More concretely, a goal should be testable in order to enable its verification, feasible and unambiguous. Moreover, it should be solution independent, so as to not exclude possible optimal solutions, as well as include restrictive constraints to narrow down the focus of potential solutions to the problem at hand. With this in mind, a structured form of documenting the goals and their respective constraints was used based on a simple template. This encompasses the goal statement, followed by foreseen stakeholders with respect to the aforementioned goal and its constraints. The full list can be found below:

- ***Goal-01*** – *Improve the utilization of latent data in the system to generate new information.*
  - Stakeholders: Management, maintenance and quality control personnel;
  - Constraints: Should accommodate existing data (e.g. unutilized stored data) as well as new data sources (i.e. addition of sensors or smart devices).
- ***Goal-02*** – *Predict future states of the system.*
  - Stakeholders: Management, maintenance and quality control personnel;
  - Constraints: Should respond in real-time as new data becomes available.

  To ensure the constraint is unambiguous, the standard definition for real-time is provided in Definition 4.4.

  Definition 4.4 (***Real-time***). *1. Problem, system, or application that is concurrent and has timing constraints whereby incoming events must be processed within a given timeframe 2. Pertaining to a system or mode of operation in which computation is performed during the actual time that an external process occurs, in order that the computation results can be used to control, monitor, or respond in a timely manner to the external process (ISO/IEC/IEEE, 2017).*

  In this sense, it is expected that the processing of incoming data and the generation of predictions should be performed in a manner that complies with the time constraints of the particular scenario in which the solution is deployed.
- ***Goal-03*** – *Adapt to changes/disturbances in the production environment.*
  - Stakeholders: Developers;
  - Constraints: Should not require additional programming effort or significant downtime.
- ***Goal-04*** – *Achieve interoperability between the system's data sources and components.*
  - Stakeholders: Developers; Systems in operation;
  - Constraints: Should contemplate the integration of legacy devices.
- ***Goal-05*** – *Applicable to different production environments.*

  - Stakeholders: Developers; Systems in operation;
  - Constraints: Should not depend on underlying communication technologies/protocols and require minimal migration effort.

- ***Goal-06*** – *Act based on predictive data to improve key performance indicators of the manufacturing system.*
  - Stakeholders: Developers; Systems in operation; Maintenance personnel;
  - Constraints: Should be flexible to either trigger for instance warnings and alarms proactively or directly act on the system when applicable; Should not negatively impact said key performance indicators.

Finally, besides the aforementioned stakeholders, other requirement sources were taken into consideration for the elicitation and specification of IDARTS's requirements. Firstly, one of the considered sources consisted in informal conversations and meetings during the developments of the FP7 PRIME (A. D. Rocha et al., 2016), H2020 PERFoRM (R. S. Peres, Rocha, & Barata, 2017) and H2020 GO0D MAN (R. S. Peres, Barata, Leitao, & Garcia, 2019) projects. As presented in Chapter 1, these projects encompassed the development of CPPS with varying degrees of focus regarding the application of predictive solutions and with different purposes regarding their application (reconfiguration, monitoring, forecasting, quality control).

These occasions proved to be a valuable opportunities to discuss the development of PMS not only with other researchers, but also with people participating directly in the day-to-day activities of their respective industrial end-users, including assembly line operators, maintenance technicians and statistical process control engineers. This provides further insight into the actual industrial needs pertaining to PMS beyond those experienced from a research or development point of view.

On top of these discussions, two additional requirement sources were taken into account, namely existing documentation for each of these projects regarding their respective RE process, along with first-hand experience with the varied systems in operation considered in each of this endeavours. The latter provides further insight into the inner-workings of current systems and the opportunities for further extensions and improvements in regards to the application of PMS.

4.1.2 IDARTS' Elicitation and Analysis of Requirements

With the foundation laid out in Section 4.1.1, the requirements elicitation and analysis process can be carried out according to the process model shown in Figure 4.2.

The process activities are as follows. *Requirements Discovery* refers to the process of interacting with stakeholders and other requirement sources in order to discover their

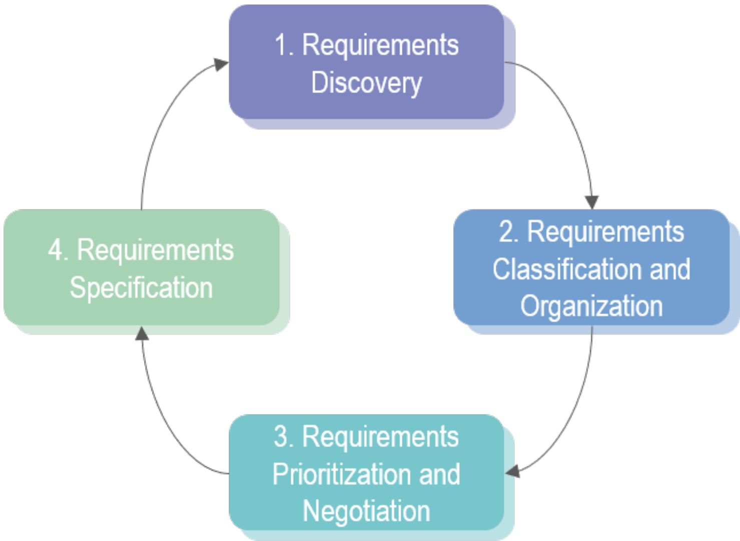

Figure 4.2: The requirements elicitation and analysis process. Adapted from (Sommerville, 2011)

requirements. *Classification and Organization* takes the unstructured collection of requirements, groups those that are related and organizes them into clusters. *Requirements Prioritization and Negotiation* aims to prioritize requirements and solve common conflicts stemming from the involvement of multiple stakeholders through negotiation. Finally the *Requirements Specification* deals with the generation of formal or informal requirements documents which serve as input for the following round of this iterative process.

There are several auxiliary methods that can be used in this elicitation process, centred either around individuals (e.g. interviews, surveys), groups of people (e.g. group dynamics) or artefacts (e.g. system archaeology, reuse). Among these, artefact-based techniques are particularly well-suited in situations where large and complex legacy systems need to be taken into account, as it is often the case in the manufacturing industry. Frequently possible stakeholders who possess deep domain knowledge are not easily available or have already left the organization, meaning that this knowledge must be extracted from the system itself, its documentation or from past experiences embedded in other successful systems.

The employment of such techniques makes it possible to understand, to an arbitrary level of detail, not only how an existing system behaves, but also which parts can be improved upon. Moreover, it can also contribute to an increase in productivity and shortened development times, as engineers aren't required to reinvent the wheel every time, since some requirements have already been elicited and quality tested resulting in less effort being required for revisions and corrections. However, artefact-based techniques (and reuse in particular) are not risk-free, as it is often difficult to find the right candidates for reuse, the quality of old requirements might not be as high as it ought to be and if done without proper quality control, problems can be carried over from previous systems.

To mitigate this, the elicitation and analysis process described in this section was performed using an adaptation of the rule-oriented reuse technique described in (Rupp & die SOPHISTen, 2014), focusing both functional and non-functional requirements. These technique aims to filter possible reuse candidates originating from the requirement sources listed in Section 4.1.1 according to a given set of characteristics, meaning that candidates are more well-suited:

- ...the higher their specification level is;
- ...the more independently they are embedded into a process;
- ...the more abstract and technology independent they are.

As a general rule of thumb, a candidate that does not conform to a particular rule is not automatically deemed unfit, yet the more characteristics it fails to meet the higher the risk and effort required to reuse it. The list of requirements resulting from this process can be found below.

IDARTS Non-Functional Requirements

From the application of the aforementioned process, a list of NFR was derived for the design and implementation of generic PMS. As previously mentioned, these are requirements related not with the functionalities and services offered by the system, but instead with how they should be carried out. To promote the unambiguity and traceability of these requirements, definitions from the (ISO/IEC/IEEE, 2017) standard are provided wherever applicable and each requirement is traced back to its related goals defined in Section 4.1.1.

- **NFR-01** (Adaptability) – *The PMS should be adaptable to cope with changes in the underlying system's hardware, functionality or needs;*

  NFR-01 asserts that the system should not be rigid in order to deal with changes in the manufacturing system, being directly related to Goals 01 and 03. As an example, this entails that if the operation of a physical resource changes and it starts producing new data values, or new resources are dynamically added to the system, the PMS should be able to autonomously encompass them in its data collection and storage cycle. Another example in a different context could be the need to adjust the thresholds concerning the normal operational conditions of a given resource, in which case the PMS should cope with such an adaptation with minimal effort. NFR-01 is thus related with the concepts of adaptability and adaptation data, as specified in Definitions 4.5 and 4.6.

Definition 4.5 (***Adaptability***)**.** *Degree to which a product or system can effectively and efficiently be adapted for different or evolving hardware, software or other operational or usage environments (ISO/IEC/IEEE, 2017).*

Definition 4.6 (***Adaptation Data***)**.** *Data used to adapt a program to a given installation site or to given conditions in its operational environment. (ISO/IEC/IEEE, 2017).*

- **NFR-02** (Interoperability) – *The PMS' components should be interoperable and easily integrated by means of common interfaces and data representations;*

  PMS can easily become large and complex systems of systems, with different components spanning various stakeholders. To facilitate the integration, deployment, operation and management of such systems, it is important that proper common interfaces and data representations are specified and adopted by each of the system's actors. Within this context, the following definition of interoperability is considered:

  Definition 4.7 (***Interoperability***)**.** *Degree to which two or more systems, products or components can exchange information and use the information that has been exchanged. (ISO/IEC/IEEE, 2017).*

- **NFR-03** (Non-invasiveness) – *The PMS shall interact with the manufacturing system without negatively impacting its performance and/or creating permanent dependencies between each other;*

  Adapting the definition of a minimally invasive CPS from (A. Rocha, 2018), a PMS should provide intelligent predictive capabilities to the manufacturing system without requiring profound changes in the native system or negatively impacting its operation. PMS should be designed and implemented in a decoupled way so as to not create permanent dependencies on the manufacturing system.

- **NFR-04** (Reliability) – *The PMS should ensure its reliability with respect to data delivery, process execution and work completeness;*

  Reliability can be defined as:

  Definition 4.8 (***Reliability***)**.** *Degree to which a system, product or component performs specified functions under specified conditions for a specified period of time. (ISO/IEC/IEEE, 2017).*

  In this context the PMS should provide stakeholders with some reliability mechanisms to ensure it operates as expected for as long as possible with minimal external intervention.

- **NFR-05** (Modularity) – *The PMS should be modular in its design to enable the composition of its functionality through the combination of different independent modules;*

  NFR-05 exacerbates the importance of ensuring that each of the functionalities offered by the PMS (later detailed in the list of FR) is encapsulated in its own independent module. This makes it so modules can be effortlessly swapped, added or removed in response to the stakeholders' needs and expectations for the system. Making the modules decoupled from one another, interacting via common communication interfaces result in changes in a single element having little to no impact on the remaining modules, as formalized in Definition 4.9.

  Definition 4.9 (***Modularity***)**.** *Degree to which a system or computer program is composed of discrete components such that a change to one component has minimal impact on other components (ISO/IEC/IEEE, 2017).*

- **NFR-06** (Real-time) – *The PMS should react to new data-driven events in a manner that complies with the system's time constraints;*

  Relating directly to Goal-02 and Definition 4.4, the PMS shall abide by the time constraints imposed by the native manufacturing system and its stakeholders' needs.

- **NFR-07** (Predictability) – *The PMS' predictions and forecasts should be describable using qualitative or quantitative parameters;*

  NFR-07 enforces a degree of accountability to the PMS, as it should be possible to clearly describe the system's predictive behaviour to stakeholders using metrics such as accuracy, recall or precision.

- **NFR-08** (Scalability) – *The PMS should be scalable to handle a large variety of system configuration sizes;*

  Scalability is a key aspect of modern CPS-based solutions, due to the growing complexity of manufacturing systems. Due to this, the PMS should be capable of scaling up or down to mirror the needs of the underlying manufacturing system, be it from a topology standpoint or from that of the system's performance and throughput. These include for instance the number of assets to be virtualized in the shop-floor, as well as the volume, velocity and variety of data to be extracted and analysed. This reinforces a critical point, which is the need to connect, harmonize and transform heterogeneous data received from different sources (Gandomi & Haider, 2015) which will be further fleshed out in the FR specification.

- **NFR-09** (Technology Independence) – *The PMS should be independent from the communication technologies employed by the underlying manufacturing system;*

  In order to increase its applicability to a wide array of potential native systems (in alignment with Goal-05), as well as to facilitate the industrial deployment integration of IDARTS-based PMS, it is crucial that such solutions are designed and implemented without relying on specific communication technological standards and protocols to exchange information with the native system.

- **NFR-10** (Usability) – *The PMS should be easily understood and/or used by its intended users;*

  Typically conveying the results from the application of such a solution to the spectrum of different stakeholders across the manufacturing value chain can be challenging, particularly since it involves dealing with a different people from a multitude of social, cultural and most of all technological backgrounds. As such, is it important that the PMS should be easily interacted with by its users, with key information being conveyed in a manner that doesn't require deep knowledge of the predictive system's inner workings.

An additional NFR could be related to the concept of flexibility as it is described in Definition 4.10, meaning that due to the generic nature of its design, IDARTS could potentially be used outside the scope of manufacturing environments, including for instance the application fields of precision agriculture or smart grids. However, this can be seen as an excitement feature from a RE point of view and as such is outside the scope of this work.

Definition 4.10 (***Flexibility***)**.** *Ease with which a system or component can be modified for use in applications or environments other than those for which it was specifically designed (ISO/IEC/IEEE, 2017).*

Additionally, each of these NFRs can be traced back to the goals initially defined in 4.1.1, as verified in Table 4.2. All of the NFRs are associated with at least one of the goals, verifying their traceability while ensuring they are in fact needed, two key characteristics used for validation in RE.

Table 4.1: Traceability regarding Non-Functional Requirements

	G-01	**G-02**	**G-03**	**G-04**	**G-05**	**G-06**
NFR-01			●			●
NFR-02	●			●		●
NFR-03					●	
NFR-04	●					
NFR-05			●		●	
NFR-06		●				
NFR-07		●				
NFR-08			●		●	
NFR-09	●			●	●	
NFR-10	●				●	

IDARTS Functional Requirements

Complementary to the NFR specification, a list of FRs for the IDARTS framework is also proposed. As previously defined, these entail the specification of the actual functionalities and services to be provided to stakeholders by the PMS. This list is presented with two different levels of grouped requirements, pertaining to the granularity of the specification.

- **FR-01** – *The PMS should collect and store data generated by the manufacturing system;*
  - **FR-01-01** – *The PMS module responsible for data collection should transform raw data into a common representation format shared among the PMS' remaining modules;*
- **FR-02** – *The PMS should pre-process data generated by the manufacturing system on the edge-computing level;*
  - **FR-02-01** – *The PMS module responsible for data collection should be capable of generating new higher-level information about the manufacturing system;*
- **FR-03** – *The PMS should predict future states of the manufacturing system based on incoming data;*
- **FR-04** – *The PMS should enact a predictive response regarding the newly generated knowledge based on reasoning rules;*
  - **FR-04-01** – *The PMS should not control the manufacturing system directly at the field-level;*
  - **FR-04-02** – *The PMS should affect the manufacturing system directly through reconfiguration when applicable/desired;*
  - **FR-04-03** – *The PMS shall interface with the users to notify of potential improvements/problems and inform of any direct actions taken;*

  This response can be either a self-adaptive one, where the PMS adapts or reconfigures the manufacturing system directly, or a more passive one, where the PMS generates an alarm or new ticket for the maintenance team.
- **FR-05** – *The PMS should adapt to changes and/or disturbances in the manufacturing system without requiring additional programming effort;*
  - **FR-05-01** – *The PMS module responsible for data collection should adapt and reflect changes in the data sources of the manufacturing system to an arbitrary level of granularity;*

- **FR-05-02** – *The PMS module responsible for predictive analytics should provide the means for its internal mechanisms to be modified/updated in runtime without incurring in downtimes exceeding the manufacturing systems time constraints;*

Similarly, each of these FRs can also be traced back to the goals initially defined in 4.1.1, as verified in Table 4.2.

Table 4.2: Traceability regarding Functional Requirements

	G-01	**G-02**	**G-03**	**G-04**	**G-05**	**G-06**
FR-01	●	●				
FR-01-01				●	●	
FR-02	●	●	●	●		
FR-02-01	●					
FR-03		●				●
FR-04						●
FR-04-01						●
FR-04-02						●
FR-04-03						●
FR-05			●			
FR-05-01			●			
FR-05-02			●			

One caveat regarding the functionalities encapsulated by the aforementioned FRs is that due to the modular nature of the PMS related with NFR (Modularity), it is possible that some of these requirements are fulfilled by legacy systems and therefore might not require an instantiation of all the different PMS components. A practical example would be the case in which a manufacturer already collects and stores all the field-level data in a real-time database and thus is not interested in replacing or duplicating such functionality. This aspect, allied with NFR (Invasiveness), facilitates the integration and deployment in existing manufacturing systems while promoting the industrial adoption of such a solution.

In addition to this, these requirements should be seen as an initial high-level directive for the design and implementation of PMS. Further refinement at finer degrees of granularity should be carried out on a use case basis depending on the particular needs and desires of its stakeholders.

To finalize, Table 4.3 presents a summary of the requirements defined in this section as a way to facilitate their comprehension and future reference in regards to the coming chapters of this document.

Table 4.3: Summary of the requirements defined for Predictive Manufacturing Systems

Type	Number	Short Description
Non-Functional	1	Adaptability
	2	Interoperability
	3	Non-Invasiveness
	4	Reliability
	5	Modularity
	6	Real-time
	7	Predictability
	8	Scalability
	9	Tech. Independence
	10	Usability
Functional	01	Data Collection
	01-01	Common Representation
	02	Pre-processing
	02-01	Knowledge Generation
	03	Prediction
	04	Predictive Response
	04-01	No Direct Control
	04-02	Reconfiguration
	04-03	User Interface
	05	Adaptation
	05-01	New Data Sources
	05-02	New Updated Mechanisms

4.2 Framework Design and Mapping to Industry 4.0 Context

IDARTS targets not only the acquisition of data at different granularity levels, but also the realization of context-aware data analysis and evaluation based on both real-time and historical data. This analysis outputs predictive data, which in this context can be defined as probable future values or states forecast based on models representing a given process, with prediction referring to the act of making statements about data that is yet to be observed (Kanjanatarakul, Denœux, & Sriboonchitta, 2016). Predictive data can then be used to trigger the system's self-adjustment mechanisms (e.g. reconfiguration) or alert operators in the shop-floor, thus assisting in returning a deviating or unstable manufacturing system to normal operation conditions, before critical breakdown events occur. To this end, IDARTS' foundations lay on top of three core principles, namely:

- The Integration of the Physical and Software Elements – Through the application of a CPPS, IDARTS' real-time computation module should be capable of extracting data from the shop-floor and reason on it in order to assess possible deviations, acting accordingly. This should assist in preventing the propagation of anomalies downstream and returning the system to its normal operation conditions, either via

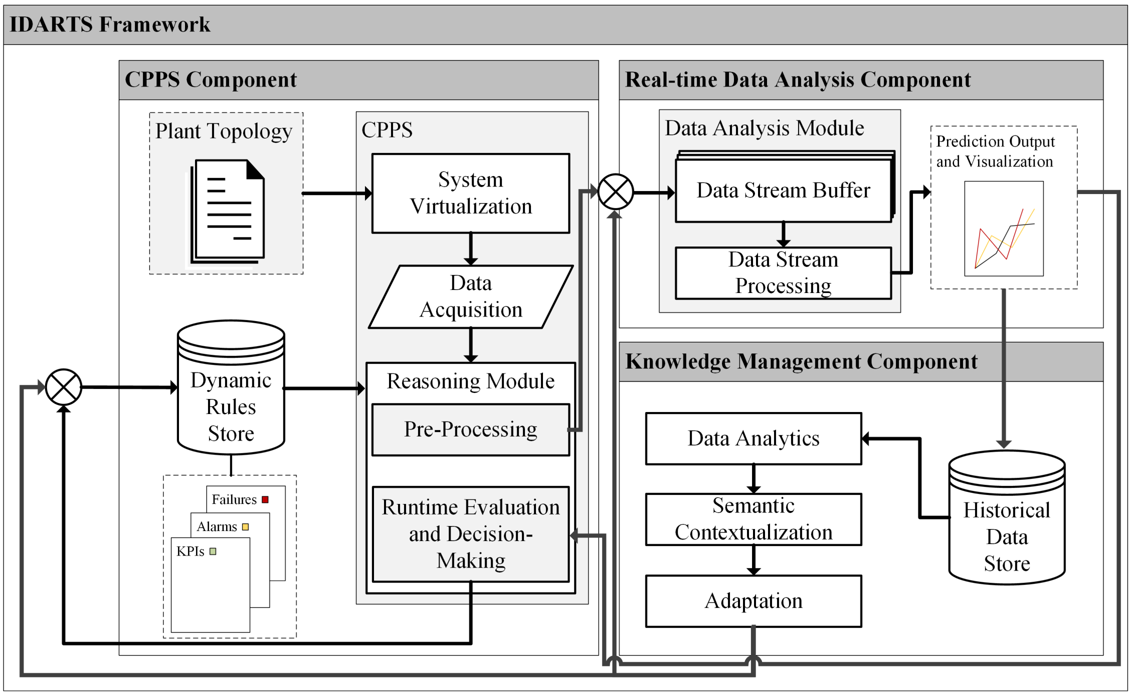

Figure 4.3: The proposed conceptual architecture for the IDARTS framework targeting intelligent predictive manufacturing systems

self-adjustment mechanisms or alerts to trigger human intervention.

- Seamless Data Exchange between Heterogeneous Components – The employment of a common data representation and exchange format in crucial to ensure the interoperability of the heterogeneous components comprising an IDARTS-based platform.

- Knowledge Management and Data Analytics – Despite the exponential growth in the volume and velocity at which data is generated in manufacturing environments (e.g. embedded sensors), a large portion of it remains untapped. IDARTS' approach aims to translate this data into a business advantage by employing advanced data analysis and knowledge management methods on semantically enriched data acquired by the CPPS. The generated knowledge can be then used to improve the CPPS' reasoning system and the real-time analysis, hence further mitigating the occurrence of breakdown events during production. The approach encompasses the combination of real-time and historical data throughout the entire production, allowing the adaptation of the analysis and monitoring algorithms after deployment for a truly adaptable and flexible approach to predictive manufacturing.

An overview of the IDARTS framework can be seen in Figure 4.3. This framework can be used as the basis for the implementation of an intelligent and adaptive PMS.

As it can be observed, the framework encompasses three core modules, namely a CPPS

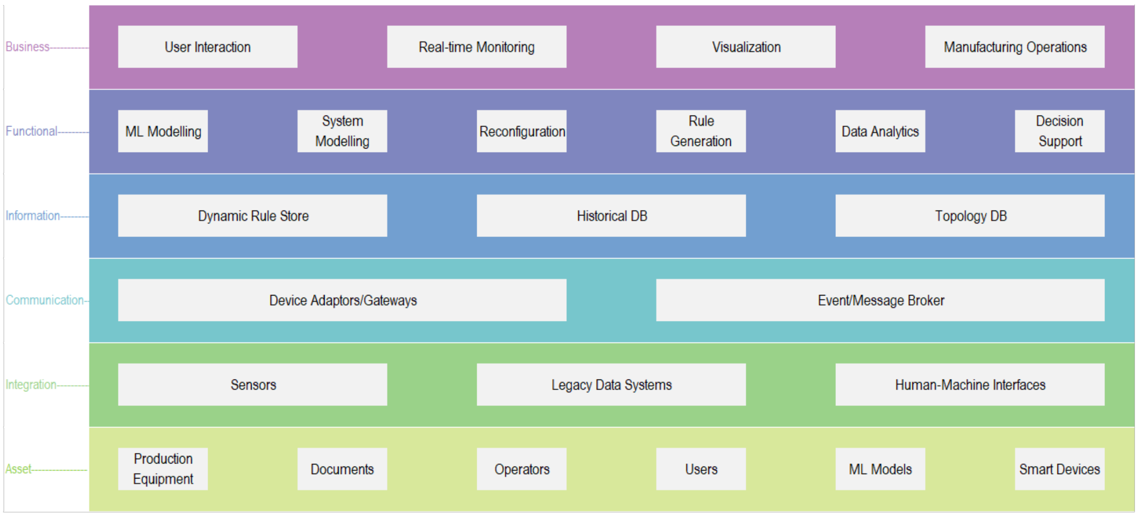

Figure 4.4: IDARTS-RAMI4.0 mapping - Architectural Layers

responsible for the system virtualization, data acquisition and decision-making support, a runtime data analysis component and a knowledge management module, which handles the higher-level data analytics processes based on historical data and provides necessary adjustments to the CPPS' reasoning module.

Furthermore, the IDARTS framework is fully aligned with RAMI4.0 and can be mapped on its representation. An interesting exercise for these types of mappings is to further break down the different dimensions into more in-depth table views of each layer. This is however a fairly challenging exercise, considering that in current literature several interpretations of each dimensions and their representations can be found, in addition to the fact the Industry 4.0 paradigm and consequently RAMI4.0 are still evolving and maturing. This will certainly become clearer over time, as more detailed reports and further work is put into the development of each dimension and their respective standards. As an example, at the time of writing IEC 62890 – Life-cycle management for systems and products used in industrial-process measurement, control and automation has been under development since 2013 and is not available to the public until its release, predicted to be in early 2020.

Within IDARTS, each machine learning model also has its own life cycle, from its conception to deployment and upkeep, meaning that it is important to consider the bottom-up flow of information triggered by shop-floor events, but also the opposite originating from the business side. To provide a better understanding of this aspect, Figure 4.5 illustrates the evolution of a model across the different dimensions of the reference architecture.

On the business level ML engineers handle the interactions to build concrete models oriented towards the business goals (e.g. reduce scrap rate via early defect prediction). This is facilitated by the capabilities offered by the functional layer, such as data analytics and ML modelling, as well as data stored in the information layer. During this design

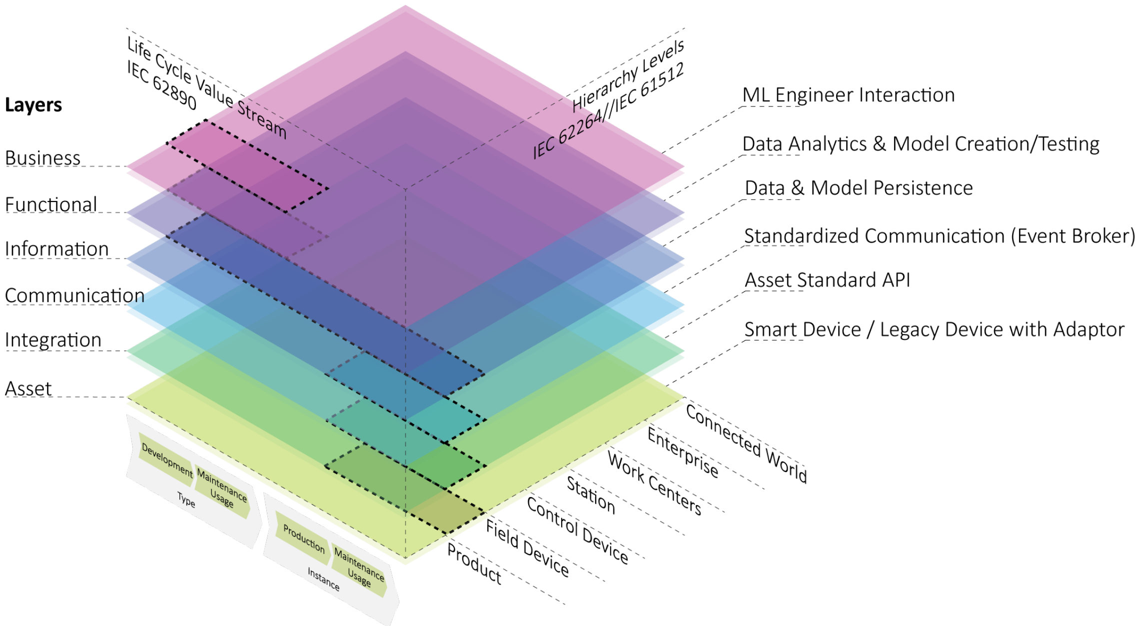

Figure 4.5: Framing IDARTS ML model creation and deployment within the three dimensions of RAMI 4.0

cycle the model can be considered a type as it is being developed, tested and validated. Afterwards, it can be persisted in the information layer and deployed to a corresponding smart device in the shop-floor, at which point it becomes an instance. Assuming the model is running at the edge level, this deployment entails an event is triggered at the information layer, which in turn can be passed through the communication layer via an event/message broker using a common representation format, finally through its virtual interface in the integration layer and finally reaching its destination, effectively becoming an asset itself.

4.3 Cyber-Physical Production System Component

The CPPS Component is composed of three main elements, namely the CPPS itself, the Plant Topology information and the Dynamic Rules Store. As the name suggests, the CPPS acts as the core element at the shop-floor level, playing the role of the centrepiece that glues the different constituents of the framework together, integrating both production and quality control processes.

The Plant Topology data should be an integral part of the system's data model, representing its existing resources, their organizational structure and other relevant information such as connection interfaces and existing data sources. Through it, the CPPS can be instantiated in a way that is capable of virtualizing each of the system's elements and initiate the data acquisition process. This System Virtualization creates a logical one-to-one relationship between each element of the shop-floor and its cyber representation, enabling a non-invasive application of the framework's capabilities. It also represents a way to break down the

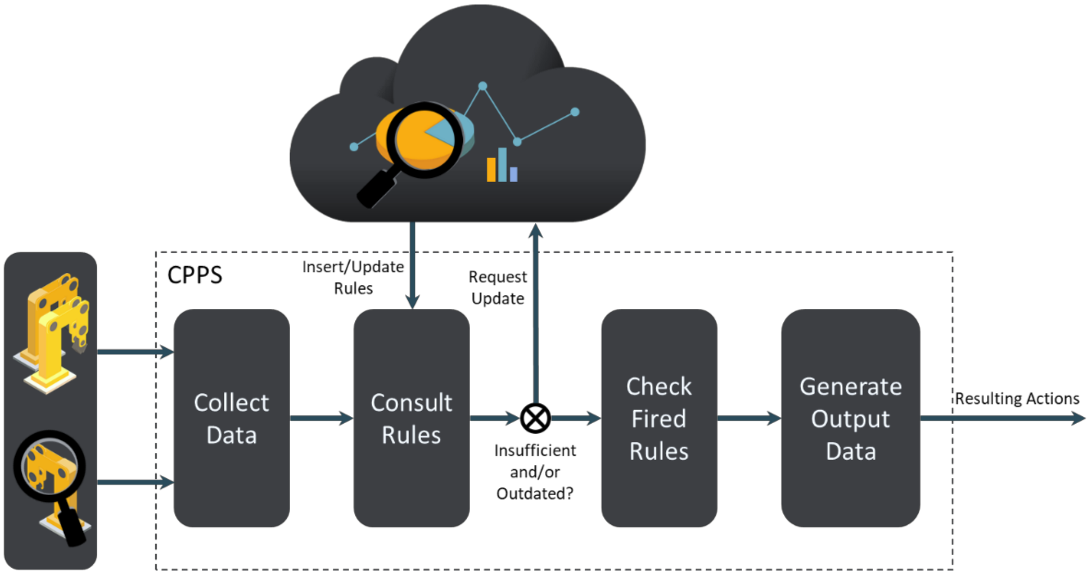

Figure 4.6: CPPS rule-based reasoning flows

system into smaller, more manageable building blocks and thus reducing the complexity of the problem.

This data acquisition process is responsible for feeding new incoming data from the shop-floor not only to the reasoning module, but also to the Real-time Data Analysis (RDA) component and to an historical data store to be used by the Knowledge Management layer. To do so, this system needs to be both flexible and adaptable in order to deal with unforeseen disturbances at the shop-floor level in a robust and efficient manner.

Also, the communication with the shop-floor needs to be specified in a generic way, thus allowing the consideration of different requirements from potentially heterogeneous use cases. For instance, a specific case might present time constraints in the order of weeks or days, while another might require data to be collected and analysed in near real-time, allowing only the consideration of relatively small delays in the communication and processing and therefore requiring different approaches.

Finally, the CPPS is also responsible for the local processing of the collected data. This is done in two stages, the first of which deals with the pre-processing of raw shop-floor data and generation of more complex knowledge. The other refers to being capable of reasoning and following through with rule-based decision-making processes, providing an earlier identification of faults, potential deviations or other critical events. The basis of this behaviour is depicted in Figure 4.6, in which the arrows indicate the general data flow throughout the process.

These rules are contained in a Dynamic Rule Store, and should be modelled using the system's common data representation format. The store can be updated dynamically during runtime by the Knowledge Management layer, if either as a result of the data analysis performed on the historical data it is found that certain changes are required to improve overall quality control, or if the CPPS requests an update from Knowledge Management due to having insufficient or outdated rules.

4.4 Real-time Data Analysis Component

Concerning the runtime domain, the RDA Component encompasses the elements necessary to perform the analysis of relevant production-related data during the system's execution.

The first of these consists in the Data Stream Buffer, which should act as a robust data queue capable of handling high volumes of data while ensuring its reliable delivery. Through it streams of data collected by the CPPS can be then passed on to the Data Stream Processing. This, in turn, is responsible for the actual data analysis, focusing on the early detection of deviating patterns and trends that might lead to breakdown or failure events on the shop-floor. Hence, due to this capacity for predictive analysis in runtime, the RDA component acts a key-enabler of condition- based maintenance, allowing manufacturers to schedule maintenance operations before a failure actually occurs, thus diminishing the direct impact on production.

The output of this module should then be visually represented in order to facilitate its comprehension by human operators, as well as being passed back to the CPPS so that its runtime decision-making component can trigger a self-adjustment response or suggest possible maintenance actions that might be required to return the system to its normal operation conditions.

A set of guidelines for the design and implementation of real-time stream analysis solutions is provided in (Stonebraker, Çetintemel, & Zdonik, 2005), which were adapted to fit the scope of the IDARTS framework. Even so, these should be considered on a use case basis, depending on each specific context.

Guideline 1: Moving Data

In order to achieve low latency (as low as required by the use case), IDARTS implementations should be capable of performing message processing without having costly storage operations in the critical processing path. In most cases it is not necessary to perform such time-intensive operations before message processing takes places, instead messages should be processed "in-stream".

Another potential issue is that of passive systems, since these need to be ordered before initiating the processing, thus incurring in additional overhead and latency. As such processing implementations should be active and avoid this pitfall by incorporating data/event-driven processing capabilities.

Guideline 2: Handling Stream Imperfections

Another important aspect for real-time stream processing is the built-in capacity to deal with stream imperfections in a resilient manner.

This imperfections can include delays, missing messages or out-of-order data. For this reason, it is important to include for instance some form of message time out in any real-time application that includes blocking operations.

Guideline 3: Generating Predictable Outcomes

IDARTS implementations should ensure that data processing occurs in a predictable and reliable manner so that its results can be deterministic and repeatable, in compliance with NFR-04.

This is also important from a a fault tolerance and recovery standpoint, since in the event of system failure and recovery it is crucial that reprocessing the same input data should yield the same outcome regardless of its time of execution.

Guideline 4: Integrating Historical and Streaming Data

The fourth guideline deals with the system's capacity to efficiently store, access and modify historical information, as well as combine it with live streaming data when applicable.

To ensure a seamless integration, the system should adopt a common representation format when dealing with either type of data.

Guideline 5: Data Safety and Availability

To avoid disruptions in real-time processing, particularly when dealing with mission-critical information, it is important to ensure that the necessary fail-safe mechanisms are put in place to guarantee that applications are up and available, and that the integrity of the data is always maintained regardless of any possible failures.

Guideline 6: Partitioning and Scaling Applications

With the current availability of low-cost computing clusters, it is becoming increasingly important to enable the scaling of real-time processing applications through distributed operation over multiple processors or machines (if deemed necessary as the volume of input data or the complexity of the its processing increases), without requiring developers to write low-level code. Ideally, load-balancing should be handled automatically in a transparent manner based on demand, without requiring additional programming effort by the developers or system users.

4.5 Knowledge Management Component

Contrastingly, the Knowledge Management Component operates outside of the constraints imposed by the real-time execution and monitoring of the production system. It consists in combination of a Historical Data Store and three processing modules, namely Data Analytics, Semantic Contextualization and Adaptation.

Each of these modules is responsible for a different step of the knowledge management pipeline. While the Data Analytics component refers to the actual data analysis process, Semantic Contextualization deals with capturing domain-expert knowledge and enriching the results with meaningful, easily understandable context. This is extremely important because it assists not only human operators, but also the CPPS in interpreting the analysis results. Lastly, the Adaptation component handles the management and refinement of the decision-making rules and runtime analysis processes.

While the analysis performed at runtime focuses solely on the constantly incoming streams of raw data, the one done at the higher-level additionally takes into account historical data, not only raw but also the more complex knowledge generated by the CPPS. This makes it possible to generate new knowledge from correlations and patterns that might be harder or impossible to discover in the RDA alone. This can then be used to update the rules that govern the CPPS' reasoning mechanisms or models used in the RDA, either periodically or on request, therefore improving the overall quality of the manufacturing processes.

In the remainder of this section a review of different types of models that can be employed in this component is provided, followed by a brief description of relevant evaluation metrics for the assessment of the performance of the various models.

4.5.1 Machine Learning Algorithms

In this section different ML algorithms are discussed, serving as a reference for possible models to be used in the realization of the Knowledge Management component. When relevant, parallels to Python's Scikit-Learn library specifications are drawn (Pedregosa et al., 2012), as it is later used for the implementation detailed in Chapter 5.

Gaussian Naive Bayes

The Gaussian Naive Bayes (GNB) method is a supervised learning algorithm based on the Bayes' theorem with the naive assumption of conditional independence between the various pairs of features given the value of the target variable. The *GaussianNB* class from scikit-learn implements GNB for classification, with the likelihood of the features assumed to be Gaussian:

$$P(x_i \mid y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} exp\Big(- \frac{(x_i - \mu_y)^2}{2\sigma_y^2} \Big) \tag{4.1}$$

where the parameters σ_y and μ_y are estimated using maximum likelihood.

Some practical applications of GNB include text prediction, document classification and spam filtering. It requires a relatively small amount of training data to estimate the necessary parameters, can be quite fast in comparison to more complex methods and is easy to implement, being often used as a baseline (Rennie, Shih, Teevan, & Karger, 2003). However, while its naive assumptions can make such efficiency possible, they can also adversely affect the quality of the results in several real world applications, such as the use case at hand, in which the feature pairs are unlikely to be independent.

K-Nearest Neighbours

K-Nearest Neighbours (KNN) is a type of instance-based learning algorithm, meaning it does not construct a general internal model, but instead stores instances of the training data with computation being deferred until classification. Over the years it has seen several applications in both statistical estimation and pattern recognition including for instance the classification of heart disease to provide a decision-support system for clinicians (Deekshatulu, Chandra, et al., 2013). Conceptually, such an approach can be carried over to the use case at hand, as we are effectively attempting to identify a condition in the cars, and furthermore, being one of the simplest machine learning algorithms for classification it is at least a good candidate to serve as a baseline.

For KNN, the input consists in the k closest training examples in the feature space, with the output being a class membership attributed by a simple majority vote of the nearest neighbours based on some distance metric such as the Euclidean distance.

XGBoost

XGBoost (Chen & Guestrin, 2016) stands for eXtreme Gradient Boosting and is an optimized implementation of gradient boosted trees, designed to be highly efficient and flexible. It is a non-linear algorithm which typically works well with numerical features and requires relatively less feature engineering and hyperparameter tuning to yield good results.

Generally, such methods can be prone to overfitting, as they constantly involve fitting a model on the gradient. To mitigate this, one can optimize for the number of trees until the out of sample error starts increasing once more.

XGBoost models are frequently used to solve Kaggle challenges across several domains, with real world applications including for instance the identification of complex relationships between variables for rare failure prediction in manufacturing processes (Hebert, 2016).

Random Forest

In the context of classification problems, RF is an ensemble learning method that operates by constructing several decision trees at training time and outputting the class that is the mode of the classes of the individual trees. While a single decision tree can easily run into overfitting problems, being also sensitive to small variations in the data, due to their nature RFs are more robust to such challenges.

Support Vector Machine

The SVM algorithm constructs hyperplanes in infinite-dimensional spaces to achieve classification. Intuitively, a good separation is achieved by the hyperplane that has the largest distance to the nearest training-data point of any class (so-called functional margin), since in general the larger the margin the lower the generalization error of the classifier.

While this is a fairly formal approach to the classification problem, one disadvantage mentioned in the scikit-learn documentation for the *SVC* implementation is that fit time complexity is more than quadratic with the number of samples, making it hard to scale for data sets with more than a couple of 10000 samples. While this is not the case for this particular case study, it is something to keep in mind when comparing to other approaches.

4.5.2 Evaluation Metrics

Accuracy

Accuracy can be used as a statistical measure of how well a binary classifier identifies or excludes a condition. It is the proportion of true results among all the observed cases. The formula for quantifying binary accuracy is:

$$Accuracy = \frac{tp + tn}{tp + fp + fp + fn}. \tag{4.2}$$

where *tp*, *tn*, *fp* and *fn* refer to true positives, true negatives, false positives and false negatives, respectively.

However, while high accuracy is typically regarded as a good indicator of performance, accuracy alone can be very misleading, particularly for imbalanced cases. Also as a metric for comparison the same holds true, as two models can yield the same accuracy results while performing differently with respect to the types of correct or incorrect predictions they provide.

Recall

To assist with the aforementioned challenge, one other metric that can be calculated is recall. Recall represents the proportion of true positives that was identified correctly, thus being a suitable metric to use for model selection when there is a high cost associated with false negatives. It can be calculated as follows:

$$Recall = \frac{tp}{tp + fn}. \tag{4.3}$$

Precision

To complement this, precision is then the proportion of the values identified as positives that was actually correct. As such, it is an adequate measure to use when the cost associated with false positives is high, being calculated as indicated in 4.4.

$$Precision = \frac{tp}{tp + fp}. \tag{4.4}$$

F1 Score

For cases in which a balance between precision and recall is preferable, and particularly when there is an uneven class distribution, the F1 score is often used as the evaluation metric. It is the harmonic average of the precision and recall, with 1 and 0 being its best and worst values, respectively, as given by the formula:

$$F_1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}. \tag{4.5}$$

Area Under the Receiver Operating Characteristics

Area Under the Curve (AUC) - Receiver Operating Characteristics (ROC) curve is a performance measurement for classification problem at various thresholds settings. ROC is a probability curve and AUC represents degree or measure of separability. It provides an indication of how well a model is capable of distinguishing between classes. More specifically for this case study, the higher the AUC, the better the model is at predicting cars that are OK as OK, and cars that are NOK as NOK.

The ROC curve is plotted with True Positive Rate (TPR) against the False Positive Rate (FPR) where TPR is on y-axis and FPR is on the x-axis.

Chapter 5

Implementing IDARTS

In the present chapter a possible realization of the IDARTS framework is presented, aiming to serve as an illustrative example for future implementations of PMS using this framework as the foundation, encompassing each of the modules defined in the previous chapter. Nevertheless, it is worth noting that different implementations are possible, so long as the goals and requirements (both functional and non-functional) for each of the modules are respected.

As mentioned in Section 4.3, the CPPS needs to be capable of extracting data from the shop-floor during execution in a flexible and robust manner. To this end, a Multiagent System (MAS) can be implemented using the Java Agent Development framework (JADE) (Bellifemine, Caire, & Greenwood, 2007) based on the monitoring approach developed in (A. D. Rocha, Barata, & Orio, 2015), previously validated in an automotive industry cell under the FP7 PRIME project (A. D. Rocha et al., 2016). JADE provides a robust infrastructure which supports the agents' core behavioural logic and communication, as well a wide array of auxiliary tools to further facilitate the development process.

The MAS-based CPPS abstracts both components (e.g. robots, conveyors, sensors) and subsystems (e.g. cells, workstations), taking care of the acquisition of their respective data as well as its pre-processing, preparing it to be further analysed by the other framework modules. The adoption of MAS technology confers additional flexibility and robustness to the CPPS, allowing it to quickly adapt to changes in the shop-floor as imposed by the framework's requirements. The MAS' support for pluggability, combined with the framework's modular design, will allow for the components it abstracts to be plugged/unplugged in runtime without requiring additional reprogramming effort in the overall system. The implementation of the RDA module should be split into two components, namely a data message queue based on Apache Kafka, and a stream processing network developed in Apache Storm. This development approach builds on the work and guidelines laid out in (R. S. Peres, Rocha, & Barata, 2017) with the instantiation of the Kafka broker and the implementation of the Storm topology. Hence, the following implementation structure is proposed as depicted in Figure 5.1.

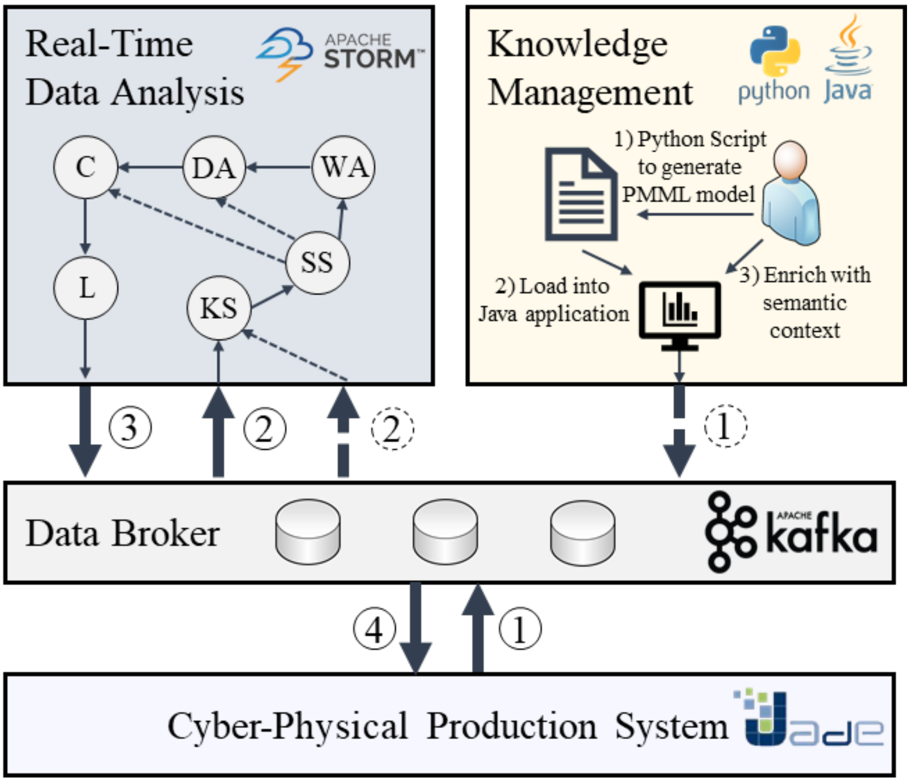

Figure 5.1: Proposal for the implementation of the IDARTS modules (Solid line: Data, Dashed line: Model Updates) (R. S. Peres, Rocha, Leitao, & Barata, 2018)

Upon collecting or generating new data, the CPPS will publish it to Kafka (1), which acts as a reliable, highly-scalable, high-throughput real time data broker. These characteristics are compliant with NFR-04, NFR-06 and NFR-08 previously defined in Section 4.1.2. This broker handles all the communications between the distributed modules of the IDARTS framework, enabling the seamless exchange of data between them in a common format, as imposed by NFR-02. Once data has been published to a Kafka topic, it can be easily consumed by the Storm topology via a Kafka Spout (2). In Apache Storm, spouts represent data stream sources, typically reading tuples from external sources and emitting them into the topology, in this case triggering the processing of shop-floor data. Once this process is concluded, the result is once more published to Kafka (3) and made available to be ultimately consumed by the CPPS (4), enabling the emergence of a predictive response (i.e. through self-adaptation or human interaction).

Finally, the Knowledge Management tool focuses on empowering the RDA with additional flexibility, by providing the means to adapt the way data analysis is performed in run-time, without requiring any stoppages or re-redeployment of the running system.

In brief, a generic pilot implementation based on the glsIDARTS framework is proposed, focused on the aspects of data analysis and real-time supervision of manufacturing systems. Being aligned with the Industry 4.0 vision, it aims to take advantage of the ongoing data explosion, presenting a scalable and flexible solution for predictive manufacturing, being as minimally invasive as possible. Its efficacy is however dependent on the availability, volume and quality of the data from the underlying production system. The pilot implementation is thus the main contribution of this chapter, supported in the following key characteristics:

- Capacity to support a plug-and-produce paradigm in the context of predictive manufacturing through dynamic system virtualization via a MAS-based CPPS, coping with changes and disturbances at the shop-floor.
- Integration of real-time and historical data at both the component and system levels, enabling the adaptation of the real-time analysis and rule-based supervision algorithms after deployment;
- Support for context-aware self-adaptation coupled with human-machine interaction, allowing the system to either adjust its operation parameters through self-reconfiguration, or suggest corrective actions to an operator in order to return to normal production conditions and product quality.

The coming sections will further detail the implementation aspects concerning each of the aforementioned IDARTS components.

5.1 Multiagent System (CPPS Component)

In this section an overview of the agent-based CPPS architecture and implementation is presented, along with a full description of its agent types and respective interactions. The proposed architecture is composed by three generic types of agents. Each type is responsible for abstracting different parts of the manufacturing system. The main goal of the proposed architecture is to perform virtualization and supervision at different granularity levels in any kind of system configuration. In Figure 5.2 a global overview of the entire multiagent environment is presented with all possible deployed types of agents and external connections, such as connections to collect or to send data to external environments.

As it is possible to understand from this representation, the proposed reference architecture is composed by three different generic agents, namely the Component Monitoring Agent, the Subsystem Monitoring Agent and the Deployment Agent:

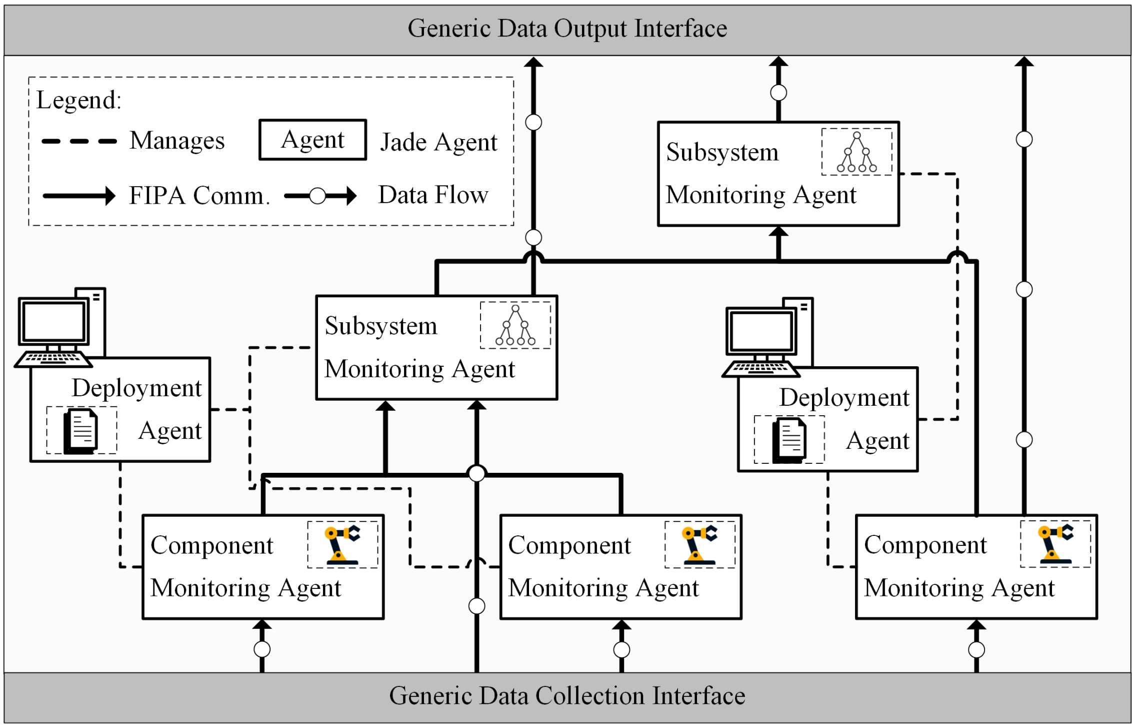

Figure 5.2: CPPS Multi-Agent System Overview (R. S. Peres, Rocha, Leitao, & Barata, 2018)

- **Component Monitoring Agent (CMA)** – The CMA represents the lowest level entity in the architecture. Each CMA abstracts a physical resource such as a robot, conveyor belt, tool, etc, summarizing a cyber-physical device which composes part of the production line. This agent periodically collects raw data from the physical device using the Hardware Communication Library to interface the respective physical resource. This library is specific for each resource and makes the raw data available for the CMA. After the collection of the raw data, the CMA can pre-processes the data to retrieve important information from the component's performance, such as transition times and action times. To allow this, each CMA contains a list of rules that describe possible events that can be retrieved from the raw data provided by the component. These rules are loaded in the beginning of the agent's execution, using the Event Description Library to access the repository (XML file, database, etc) and respective rules. Additionally, each CMA is also responsible for enacting a local predictive data-driven response based on the analysis from the other modules corresponding to its physical resource using the two-way communication from the Hardware Communication Library;

- **Subsystem Monitoring Agent (SMA)** – When a set of resources work together to perform higher level capabilities in the line it is crucial to have higher level virtualization running at the same level as well, having in fact the same granularity as the aggregation of capabilities performed by this set of components. Hence, the SMA is responsible for abstracting a certain set of cyber-physical components which can cooperate in order to offer higher-level capabilities. In sum each SMA abstracts a subsystem in the line (station, workgroup, etc). The SMA receives raw data from a computational device, such as a computer or a PLC containing data referent to the abstracted subsystem, and receives pre-processed data from the CMA which subscribed this SMA. All the cyber-physical devices that can cooperate must subscribe the same SMA. Similarly to the CMA, the SMA has inherent rules used to pre-process the data received from the computational device and from CMAs. This type of agent implements the Data Acquisition Library and the Event Description Library as in the CMA's case to allow the same capabilities in the SMA. In this case the raw data received from the Data Acquisition Library belongs to the subsystem and not to a cyber-physical component (Ex. Cycle time of the station, emergency stop, etc);

- **Deployment Agent (DA)** - The final agent class is the Deployment Agent, which is the centrepiece of the plug and produce functionality which promotes seamless adaptation and scalability to the native system's capacity needs. The DA instantiates a Hardware Detection Library (HDL), through which it cyclically checks the system's topology in order to launch/remove CMAs and SMAs accordingly. More specifically, once a any physical resource is plugged or unplugged within the system, the DA manages the corresponding virtual representation in the multi-agent platform as needed, either by launching or killing the associated agents.

This categorization of the agents builds upon a first iteration (R. S. Peres, Rocha, Coelho, & Barata, 2017) in which an additional type of agent, the Output Coordinator Agent, was included. This agent type was responsible for handling all the output of information from the MAS to external modules such as data analytics. In the current implementation, this functionality was embedded into each agent abstracting manufacturing resources, being thus responsible for the entire data flow regarding their respective individual resource. As such, even if a particular agent fails to operate properly unexpectedly, only the data flow pertaining to its particular resource is affected, instead of potentially impacting the whole system.

It is also important to note that these libraries mentioned for each of the agents are implementations of the generic common interfaces adopted by the framework. This approach makes it so all other implementation details are decoupled from the native system's logic, meaning that only the particular instantiation of the interfaces needs to be adopted for each use case as a parametrization.

Moreover, in this way the proposed architecture can take advantage of recent IT developments, such as cloud computing as a service with large amounts of computational capacity, to reduce the computational load required on the field-level to perform these predictive tasks. Data acquisition, pre-processing and adaptation can be performed at the edge, while more intensive computational tasks can be relayed to the cloud. This approach has two big effects:

- Improvement of system performance – By reducing the supervision load inside the controllers, the control system has more hardware capacity available. With this the control system improves its performance and consequently the performance of the entire system;
- Increasing the possibilities of predictive supervision – Using an external entity, such as a remote server or a data centre with a huge computational capacity, the amount of data that can be processed and the number of variables and combinations that can be modelled and analysed is increased, resulting in a better and more accurate supervision.

The existence of a historical database is imperative because industries constantly need to consult historical data to understand unexpected behaviours and the system's performance. This data can be used either by the external entity or by other entities such as a Human Machine Interface. The usage of an external remote entity, running in powerful machines with the capacity to store any amount of data provided by the system, allows it the capacity to model and retrieve more information from raw and pre-processed data. During execution the external entity is capable of continuously computing all received data, modelling the system's behaviour and triggering events that can result in important alerts, avoiding line stoppages and allowing the operators to perform preventive and predictive maintenance.

After this introduction to the MAS-based CPPS component, it is appropriate to take a closer look into the main interactions between the agents, necessary to carry out their responsibilities within the larger PMS. More specifically, these include the addition and removal of resources, the acquisition and processing of manufacturing data and finally the execution of a predictive response, as detailed in the following sub-sections.

It should be noted that all JADE agents are compliant with the Foundation for Intelligent Physical Agents (FIPA) specification (Bellifemine, Poggi, & Rimassa, 1999), so all the communications among the agents were implemented according to the standards defined by FIPA. The communications between the agents are typically based on two protocols: the FIPA Request Protocol (FIPA, 2002) and the Contract Net Protocol (FIPA, 2000). The FIPA Request Protocol is used to make point-to-point communication between two agents requesting to perform a specific task, while in the Contract Net Protocol case, a negotiation is performed in order to understand which responder agent is the most appropriate to perform the required task, after which the task is requested by the initiator and performed by the responder agent. In the current implementation, since there is no need for negotiation in the agents' interactions only the former was used.

5.1.1 Adding and Removing Resources

The addition and removal of physical resources to/from the manufacturing environment (and by extension of its corresponding agents) its closely tied with the PMS' capacity to adapt and scale up or down to reflect the needs of the underlying manufacturing system.

In this sense, the DA carries out most of the key functionality by calling a library implementing the *IHardwareDetection* interface, shown in the snippet from Listing 5.1:

Listing 5.1: DA generic interface.

```
public interface IHardwareDetection{
    ArrayList<String> getPluggedResources();}
```

This library's method is called periodically on a cyclic behaviour which enables the DA to get a list of currently plugged resources and compare it the one it maintains internally. If an addition is detected, the interaction detailed in the sequence diagram from Figure 5.3 is started.

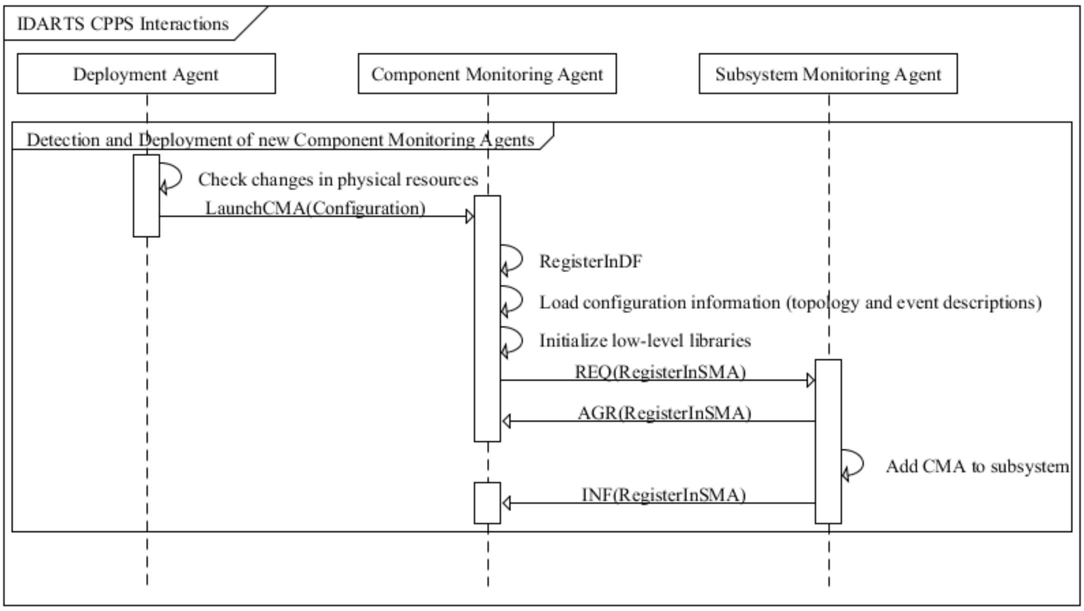

Figure 5.3: Interactions between the CPPS' agents during the deployment of new CMAs.

Once a CMA is launched with proper parametrization provided by the DA, it immediately registers itself in JADE's Directory Facilitator, which acts similarly to an yellow pages service, followed by loading its necessary configurations and instantiating and initializing the respective libraries, namely regarding the *IDataCollection*, *IDataDescription* and *IDataOutput* interfaces as shown in the code snippet from Listing 5.2.

Listing 5.2: CMA generic interfaces.

```
public interface IDataCollection {
    public void initializeHWConnection(String compName,
        ArrayList<MonitoredSystemData> dataList, Agent agent);
    public void closeHWConnection();
    public MonitoredSystemValue readHardwareValue(String idTag);
}
public interface IDataDescription {
    public MonitoringDataDescription getDataDescription(String entity);
}
public interface IDataOutput {
    public void initialize(String resourceID, String endpointAddress);
    public void sendOutput(MonitoredSystemValue v);
}
```

Lastly, if deemed necessary in its topology information, it communicates with the SMA responsible for abstracting the particular subsystem in which the resource was plugged into to register itself onto that same subsystem's list of resources.

Mirroring this process, if the DA detects that a resource has been unplugged when computing the differences between the resource lists, it initiates the process of removing the corresponding agent from the MAS platform. The set of interactions associated with this process is shown in the sequence diagram from Figure 5.4.

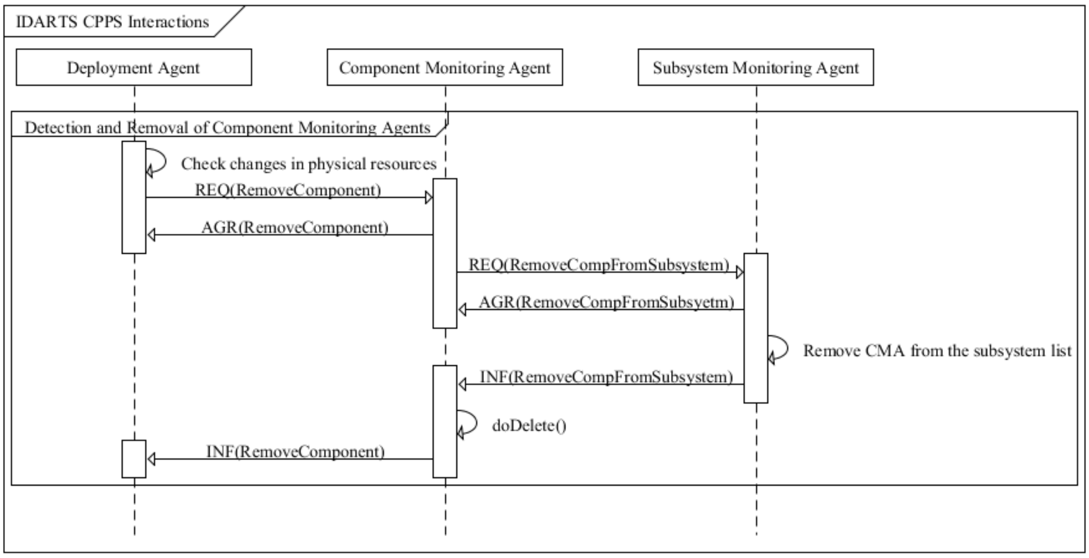

Figure 5.4: Interactions between the CPPS' agents during the removal of unplugged CMAs.

In this case, once the removal is detected, the DA requests the corresponding CMA to remove itself from the platform. In turn, the CMA needs to ensure that if he is part of a subsystem, he should then request the parent SMA to remove it from the subsystem's list of resources. The inform message signalling the successful conclusion of this operation is only sent back to the DA once the subsystem's information is also update to ensure that consistency is kept within the platform.

5.1.2 Acquiring and Processing Manufacturing Data

This section presents the proposed solution for the problem of data extraction. Data extraction is usually viewed as a problem mainly concerning system integration, consisting on the retrieval of available data provided by a system's data sources (e.g. PLCs and other devices that can provide and store data related to the system's execution). This process usually leads to the acquisition of raw data which requires further processing in order for it to be useful in regards to the description of the system or its analysis. For instance, the bits stored in a controller's memory which describe a state indicated by sensors and actuators can be an example of such a case. In some systems the controllers might contain values that can be used for monitoring and to describe the system in a way that is understandable for the system's operators, such as times or counters, but the usage of this data might not be easy due to the difficulty associated to collecting this data from an industrial machine and display it in a common user friendly visual interface.

A common case of raw data that can be processed to retrieve useful values is data related to transitions. Transition times describe the time that resources need to change from a certain state to another. Figure 5.5 presents an example of a Transition time extraction in which the transition is performed by a clamp, where the first state represents the immediate state before starting the transition, the second one the transition itself and the last one the state immediately after the transition is performed when the sensors indicate that the clamp is on the final position. The Transition time, in this case, reflects the time needed by a clamp to close, meaning the time passed since that clamp starts the movement from the initial state until it achieves the final state.

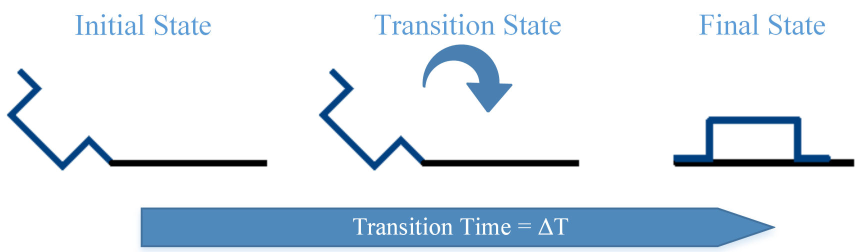

Figure 5.5: Visual representation of the transition time timespan.

Transition times are values that can be very interesting to monitor, since generally an increase in this value probably indicates that maintenance is required for the corresponding resource. Monitoring trends and deviations in such values makes it possible to recognize abnormal behavior and avoid stoppages on the line due to potential problems with its physical resources. Another example that can be extracted by processing raw data is the Action time. Action time represents the delay verified since the input is given until the resource starts the execution of a given action. This value indicates the responsiveness of each component, as illustrated in Figure 5.6, pertaining to the extraction of the action time for the previous example.

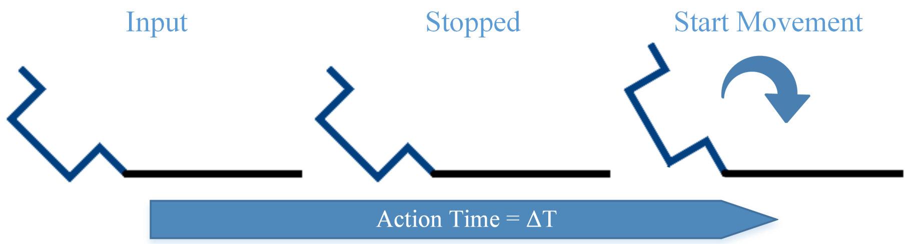

Figure 5.6: Visual representation of the action time timespan.

In this case, the extraction of the Action time is based on inputs and outputs that describe each state. The initial state is defined as the moment when the input is triggered and the clamp is in the home position, and the final state is the moment when the clamp starts the movement leaving the home position. In essence, the Action time is the time since the input is triggered until the real response of the resource.

To summarize, this data acquisition and pre-processing step is thus a key enabling factor for the operation of a PMS. It is necessary that the agents abstracting the physical resources are capable of discerning the data that is ready to be used and that which needs some form of pre-processing in order to be useful for the other system actors to understand the system's behavior and model, improve and visualize it.

Event Description

Each agent possesses its own Knowledge Base (KB) capable of storing the rules and descriptions that define their associated monitoring data. For this purpose, the EDL should contain methods to allow the CMAs and HLCMAs to access an external data source (e.g. a DB or an XML file) in order for them to learn information regarding the data they will be monitoring, more specifically which values they should extract, how often they should be extracted (polling rate) and all the rules concerning the conditions that define possible events that need to be computed by the agents themselves (pre-processing).

In order for the agents to recognize all the possible events and values related to their abstracted component or subsystem, it is necessary for each of them to have a knowledge base capable of storing the rules and descriptions that define their associated monitoring data. Therefore, each CMA and SMA loads a library which implements *IDataDescription* (see 5.2) to retrieve said rules from an external source, such as an XML file or a database, so long as it is not immutable. This should provide the agents with all information regarding the data they will be monitoring, more specifically which values they should extract, how often they should be extracted (polling rate) and all the rules concerning the conditions that define possible events that need to be computed by the agents themselves (pre-processing). Basically, an event description contains a list of all states verified during the execution of the event, as illustrated in the snippet from Listing 5.3.

Listing 5.3: Snippet from data descriptions in markup language.

```
<Component id="Line.UR5_1" name="UR5_1">
  <Data>
    <State id="cycle" type="xs:boolean" source="opc.workgroup1"
      sourceParam="UR5_1_ParamCycle"/>
    <State id="moving" type="xs:boolean" source="opc.workgroup1"
      sourceParam="UR5_1_ParamMoving"/>
    <Timespan id="gripperForceSensor" source="opc.workgroup1"
      sourceParam="RG2_leftForceSensor"/>
    <Timespan id="cycle_time" source="agent">
      <TimespanMapping>
        <StartConditions>
          <StateValue id="cycle" value="true"/>
        </StartConditions>
        <EndConditions>
          <StateValue id="cycle" value="false"/>
        </EndConditions>
      </TimespanMapping>
    </Timespan>
  </Data>
</Component>
```

In this case, by reading this configuration the CMA/SMA knows that for its resource there are two states, *cycle* and *moving*, and two numerical values to be extracted from an OPC server. In this case the latter pertain to timespan measurements, with one originating directly from the controller (indicated by `"source=opc.workgroup1"`) and the other having to be computed by the agent based on the *cycle* state (indicated by `"source=agent"`).

Runtime Execution

The main MAS execution cycle can be divided in three main phases as described in the sequence diagram from Figure 5.7. These phases encompass the collection of raw data, its pre-processing to generate new information and finally the relaying of the aforementioned data to the remaining system's actors through the message broker, with these being in this case being Kafka and the Storm data-stream analysis topology.

During the execution each CMA and SMA collects data from the hardware through its data acquisition library. These agents contain a circular buffer where the recent states

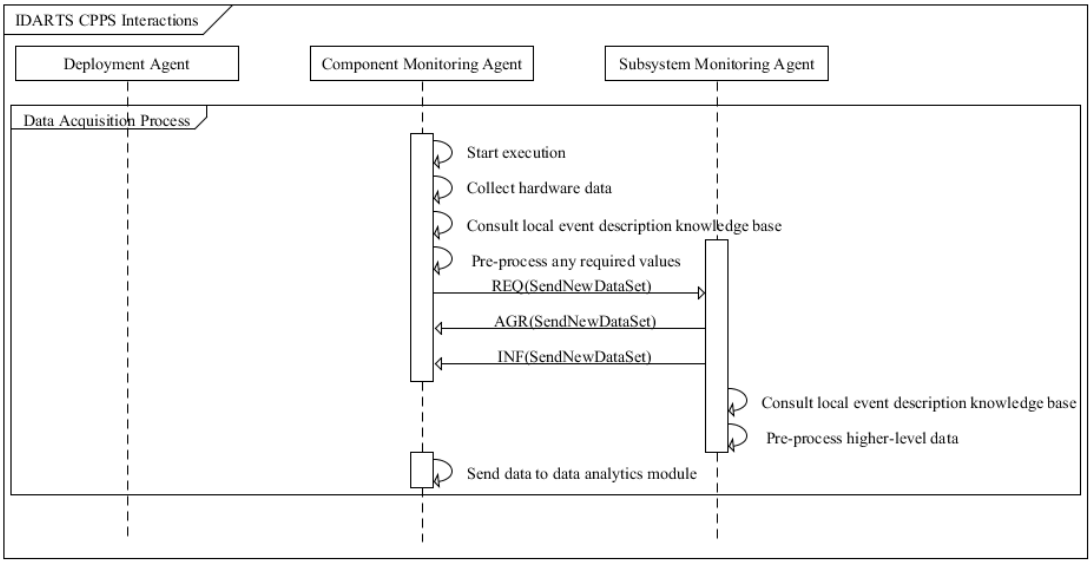

Figure 5.7: Interactions between the CPPS' agents during the data acquisition process.

are stored and where it is possible to consult the recent past states in order to extract performed events according to the existent event description rules. The states and events collected by each of these agents is then relayed to the external entities according to the agents' data output library, which in this case consists in a Kafka producer. This producer emits newly collected or generated data to a Kafka topic, which can then be consumed by the Storm topology. A snippet from the implementation of IDataOutput illustrating this can be found in Listing 5.4.

Listing 5.4: Snippet from the implementation of IDataOutput.

```
@Override
public void sendOutput(MonitoredSystemValue v) {
  producer.send(new ProducerRecord<>(DATA_TOPIC,
    v.getSourceID(), v.getValue() + ";"
    + v.getSourceTimeStamp() + ""));
  if((elapsedTimeSinceMetrics<0) ||
    (System.currentTimeMillis()-elapsedTimeSinceMetrics)>10_000) {
    getProducerMetrics();
  }
}
```

The collected and pre-processed data is also sent to the agent located on the layer immediately above the current agent's layer, basically meaning that the data is also sent to a higher level node of the tree allowing the higher level agents to use the data provided by the lower level ones. Consequently with this kind of approach different insights can be obtained which would not have been available otherwise using the data provided by the lower layers. This process can be seen as a form of data and information fusion, as defined

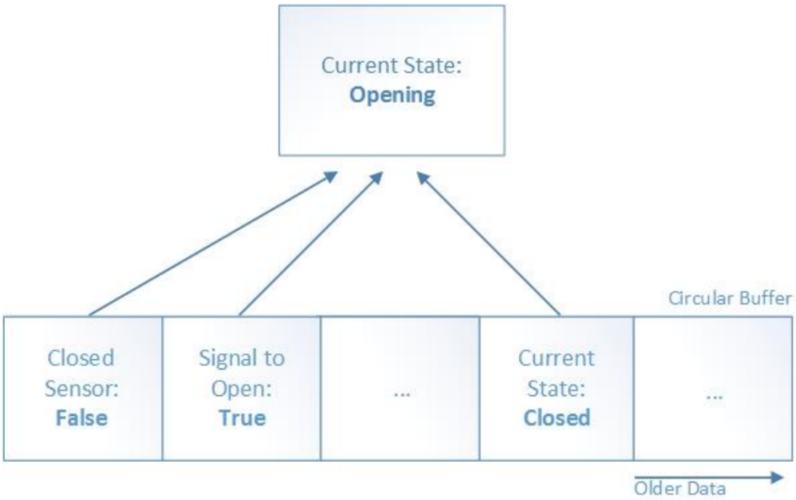

Figure 5.8: Knowledge inference example regarding a clamp's current state.

in (Khaleghi, Khamis, Karray, & Razavi, 2013), which pertains to the transformation of data from different sources and even different points in time into a representation that can support human or automated decision making effectively.

This data fusion is performed by the data processing algorithm implemented by each agent as a *OneShotBehaviour* (a type of agent behaviour that gets executed only once, in this instance every time a new data point is collected). This process is the cornerstone that enables the proposed architecture to provide more useful data to the external processing entities, permitting them to perform the analysis of relevant trends and tendencies in the extracted data's values.

The inference process consists initially in computing the raw data values according to a certain rule set stored in each agent's respective knowledge base, as shown in Figure 5.8.

However, as more complex states are inferred they can also be used to compute new data values, as described above using a clamp's current state as an example. The general workings of the full algorithm are portrayed in Figure 5.9.

As previously stated, during the CMA's and SMA's initialization process all the information regarding what kind of events can be computed and the rules that define them is loaded onto each agent's knowledge base. The agent then waits until new monitored data is received, regardless of it coming from the agent's own data extraction or its children's, starting the main processing cycle upon its arrival.

The processing cycle consists in a series of procedures that for each possible value to be computed, allow the CMA/SMA to decide whether the collected data present in its circular

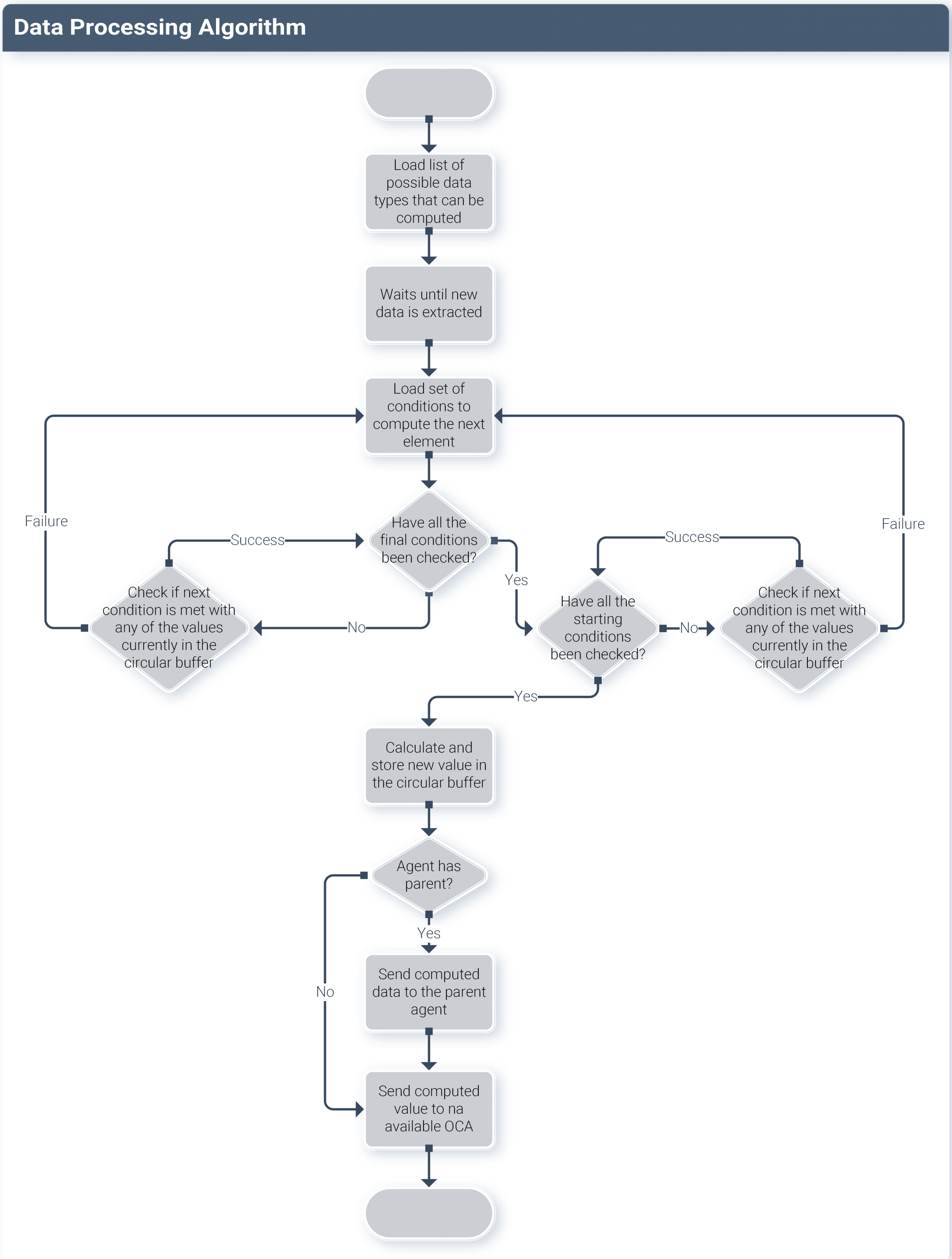

Figure 5.9: Data Processing Algorithm Flow Chart. Represents the process flow for the generation of new knowledge for each agent.

buffer in that given moment is enough to compute said value, according to the conditions defined in the monitored data description contained in its knowledge base.

While the processing of some data types, for instance states, is done in a fairly straightforward manner by simply checking if all conditions that define the referenced data type are met, the same does not apply to timespans.

In the former's case, only a single set of conditions must be met in order for a given state to be computed. Therefore all that is required is for the agent to iterate over the entire set of conditions, checking if for each and every one of them there is a value stored in the circular buffer that satisfies it. If all conditions are met, the new inferred state itself is then stored in the circular buffer and propagated to the upper layers of the monitoring tree.

However, in the latter's case, two different sets of conditions must be met in order for a timespan to be calculated, namely those that define the beginning and the end of the relevant period of time. For this reason, the agent starts by checking if all the ending conditions are met, similarly to how a new state is processed. If they are, it stores the latest timestamp among the values that verify that set of conditions, moving on to repeat the process for the starting conditions. However, if at either stage the conditions are not verified the computed value is discarded and the agent starts the cycle anew for the next possible computed value.

If all the conditions are satisfied then the CMA/SMA computes the timespan by calculating the difference between the timestamps associated with the starting and ending conditions sets, storing it in its circular buffer and sending it up the monitoring tree.

5.1.3 Enacting a Predictive Response

The implementation of the CPPS' adaptation behavior is heavily supported by the automated interpretation of the analysis' results which is performed at the later stages of the stream-analysis topology (further detailed in Section 4.4). The basic idea behind it is that the monitoring agent periodically polls Kafka through a consumer to check if there are newly published analysis results. As it will be seen later on during the description of the Storm topology, these results include the corresponding action that should be carried out by the agent through the implementation of the *IDataResponse* interface from Listing 5.5.

Listing 5.5: Generic interface for enacting predictive responses.

```
public interface IDataResponse {
  public void initializeConnection(String resource, Agent agent);
  public void closeConnection();
  public boolean enactResponse(String address, String value, String action);
}
```

On top of these, the action the agent should carry out through the generic *enactResponse* method depends on the use case being considered. If the goal is to enact a self-adaptive response, then the method should be implemented in a way that allows it to directly affect parameters or configurations for the physical resources. Otherwise, it should interact with the system's users through some human-machine interface to provide warnings, alerts or to issue for instance maintenance requests directly.

5.2 Storm Topology (RDA Component)

The topology's actual processing is performed by its bolts, each containing its own specific logic. The proposed topology is multi-layered, with each bolt performing a specific task on the incoming tuples, as it can be seen in the pipeline represented in Figure 5.10.

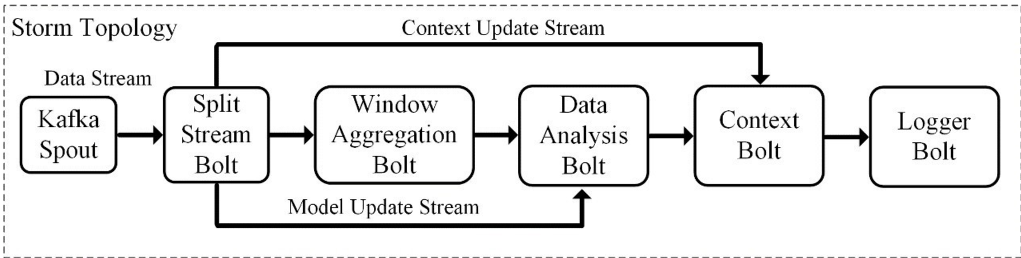

Figure 5.10: Apache Storm Real-time data processing topology

First and foremost, tuples are filtered by the Split Stream bolt according to their respective topic of origin, as they can be related to either data or updates to the running system. Tuples are then aggregated in window slots of a given size and emitted to the Data Analysis Bolt (DAB), enabling a sliding window approach.

The DAB is responsible for the actual processing task, using ML models to generate predictions for the shop-floor resources based on the incoming real-time data (e.g. likelihood of failure within a certain number of cycles). These models can vary from resource to resource, and can be provided in Predictive Model Markup Language (PMML) (Guazzelli, Zeller, Lin, & Williams, 2009) by either their respective agent during deployment, or by the Knowledge Management tool as an update in runtime, being then stored in memory by each of the bolt's workers. The basic procedure involved in the execution step of each DAB is detailed in Algorithm 1.

Once the processing is concluded, the results are relayed to the Context Bolt (CB), which enriches them with their significance in the physical world so that the CPPS can later on interpret them and act accordingly. Depending on the adopted approach, this can mean for instance triggering self-reconfiguration, scheduling predictive maintenance or simply alerting an operator by providing said context via a human-machine interface. This procedure is shown in Algorithm 2.

Algorithm 1 Data Analysis Bolt basic execution algorithm

```
procedure EXECUTE(Tuple tuple)
    if sourceStream = UPDATE_PMML_STREAM_ID then
        Update the corresponding model
        Emit to log stream the update status
    else if sourceStream = DATA_STREAM_ID then
        Get resource name from tuple
        Find reference to corresponding model
        if model ≠ null and tuple.numOfInputs ≥ model.numOfInputs then
            Feed input to the model
            Emit result to default stream
        end if
    end if
    Ack the tuple
end procedure
```

Algorithm 2 Context Bolt basic execution algorithm

```
procedure EXECUTE(Tuple tuple)
    if sourceStream = UPDATE_CONTEXT_STREAM_ID then
        Deserialize model context from the tuple
        Update internal context map
        Emit to log stream the update status
    else if sourceStream = DATA_STREAM_ID then
        Load the respective context from the context map
        Enrich output with the corresponding semantic context
        Emit result
    end if
    Ack the tuple
end procedure
```

Finally, the enriched results are sent to the Logger Bolt to be stored. These can be simply logged to the local disk, or ultimately stored in a database like MongoDB or Cassandra, being then available to be used by the Knowledge Management tool to further analyse and improve the runtime process.

Another advantage of adopting Apache Storm for the implementation of the real-time data analysis component is its support for monitoring and maintenance of the topology itself. Storm provides a user interface that enables developers to monitor its execution for bottlenecks, execution speeds and data rates, as it can be observed in Figure 5.11.

The color of nodes indicates whether a bolt is exceeding cluster capacity, with red denoting a data bottleneck and green indicating components operating within capacity. Data flow is also represented by the connections between components, with thicker lines pertaining to larger data flows. From the example above, taken during the PMS execution for the tests which will be later presented in Section 6.1, it can be seen that the largest flow of

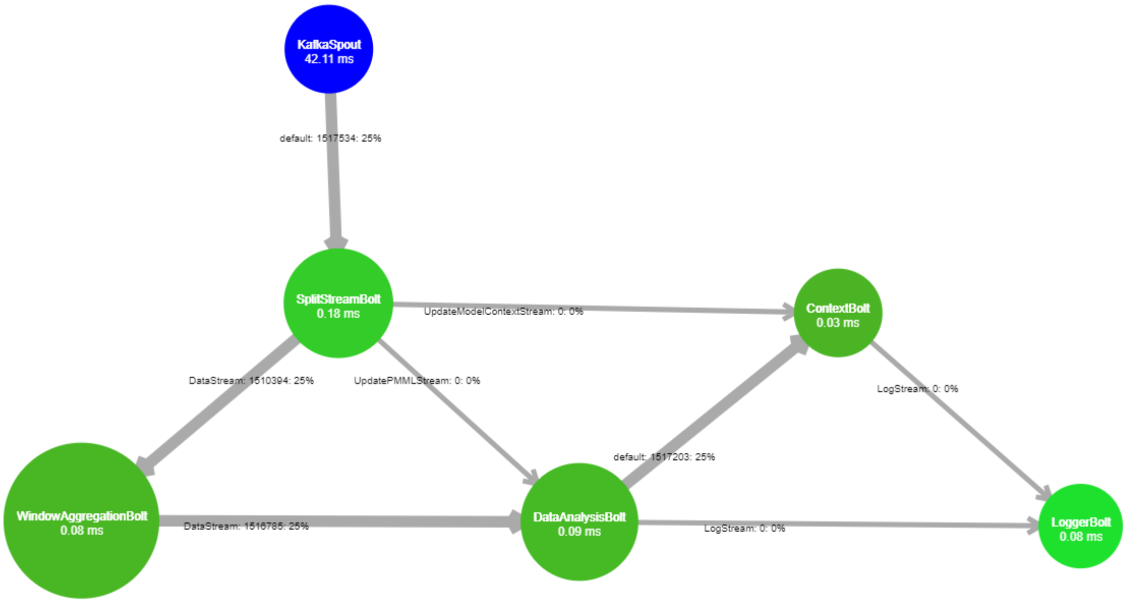

Figure 5.11: Storm UI topology monitoring visualization.

data corresponds to the regular data stream, while the model update stream presents much lower traffic since the updates are considerably less frequently than the rate at which new data is generated. Also, the capacity of the network is not being exceeded with no considerable bottlenecks as shown by the green colouring of the nodes, with the blue one being used to represent the entry point of the topology, in this case the Kafka spout.

5.3 Offline Data Analysis and Engineering (KM Component)

Firstly, ML models can be implemented for instance in Python using the Scikit-Learn library (Pedregosa et al., 2012). The models are created offline, trained using historical data and then serialized into PMML. This constitutes the application-specific stage of the solution, however, the usage of PMML ensures that developers are not restrained to using Python-based models, as long as they can be serialized to the adopted format. Generally speaking, this process follows the steps detailed in Algorithm 3.

Algorithm 3 Knowledge Management general model creation and serialization algorithm

```
1: procedure TRAIN_MODEL
2:     Load data set
3:     Split data into training and testing sets
4:     Pre-process both sets equally        ▷ Imputting missing values, normalization, etc
5:     Train machine learning model
6:     Test machine learning model
7:     Create PMML pipeline
8:     Serialize pipeline to PMML file
9: end procedure
```

This streamlined process representation can be seen as an over-simplification, as often it can deviate at certain stages depending on the task as hand. More specifically, there can be for instance additional steps for feature selection, cross-validation or hyper-parameter tuning to name a few examples. Moreover there can be additional steps regarding data exploration, prototyping and baseline modelling, but the goal here is to provide the reader with a general view of the typical process.

Afterwards, the resulting model description is then loaded by a Java-based application, which also allows the user to input additional model information such as the model ID, the resource or resources it is related to, and its context. This context can be used to capture domain knowledge and enriching model outputs with their meaning in the physical world, allowing the CPPS to autonomously interpret the results of the RDA later on. A prototypical implementation of this application is shown in Figure 5.12.

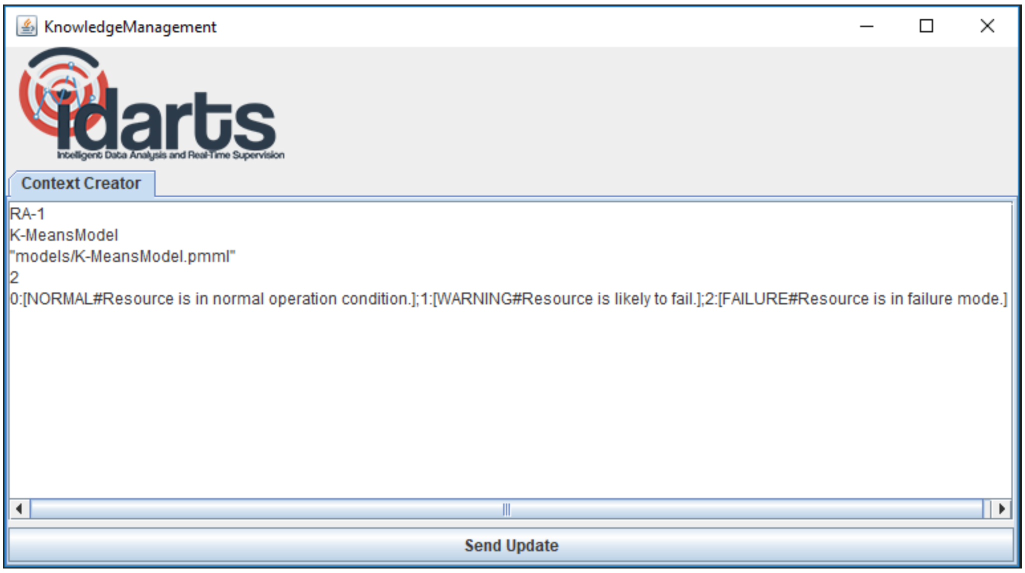

Figure 5.12: IDARTS knowledge management prototype graphical user interface.

In this example, through the application a series of parameters are specified regarding the context for the model with id "K-MeansModel", namely that it pertains to resource "RA1", the relative path to its PMML file instance and that it accepts two input parameters. Last but not least, the context that allows the MAS-based CPPS to enact the predictive response is also provided, in this case meaning that if the output from this classification is 0 it means the resource is in $NORMAL$ operating condition and no action is needed, 1 should trigger a $WARNING$ and 2 signals that the resource has failed and required immediate action, each with the respective action/message associated to it.

All this information is then serialized and pushed to a dedicated Kafka topic to be consumed by the running Storm topology, with a format similar to what is displayed in Listing 5.6. One key difference is that the path to the PMML model is replaced by the markup code detailing the actual model to be used, which is then parsed within the Storm topology.

Listing 5.6: Update message to be sent from KM through Kafka.

```
RA-1#K-MeansModel#"models/K-MeansModel.pmml"#2#K-MeansModel##{0=[NORMAL#
Resource is in normal operation condition.],1=[WARNING#Resource is likely
to fail.],2=[FAILURE#Resource is in failure mode.]}
```

Once there, the Split Stream bolt takes care of dividing the message into the updates to be processed by the DAB and the Context Bolt, thus effectively adapting the runtime system without requiring any additional effort or considerable downtime.

CHAPTER 6

Results and Validation

Chapter 6 focuses on the description of the application of IDARTS-based implementations of PMS to different use cases and experiments. To this extent, the results from the application of these implementations to three distinct scenarios are discussed with respect to the goals and requirements defined in Chapter 4.

The first scenario relates to a laboratory experiment for machine failure prediction focusing on the verification of the core IDARTS requirements. This was conducted in a set-up in which the execution of the MAS-based CPPS, the communication broker and stream processing were distributed in a local cluster of four machines.

Following this, a quality control scenario is presented as part of the H2020 GO0D MAN project. This scenario targeted a different implementation of a PMS targeting the ZDM paradigm in three industrial use cases having adopted as a basis the IDARTS framework (GO0D MAN Project (ID:723764), 2017). Each of the use cases is representative of a different type of production, with highly-customised products within professional appliances, serial production in the automotive assembly and batch production in turned and machined metal components.

Lastly, a variation of the aforementioned automotive use case is presented, putting a higher emphasis on the data-driven predictive modelling and showcasing how easily the IDARTS modules can be swapped depending on the needs of the use case at hand using an adaptation of the implementation proposed in Chapter 5.

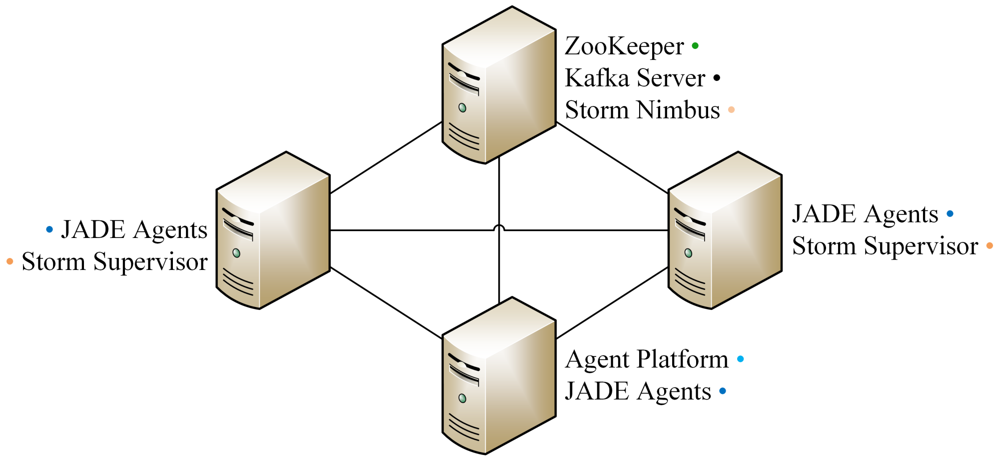

Figure 6.1: IDARTS Testing 4-Node Cluster

6.1 Laboratory Cluster Deployment

This section details the steps taken during the testing and validation of the initial implementation of the IDARTS framework proposed in Chapter 5. The tests were conducted with the goal of verifying the NFRs identified in the specification of the framework, with a larger emphasis on the aspects of scalability, adaptability reliability and real-time constraints. The testing environment consisted in the four-node cluster shown in Fig. 6, consisting on four machines running Core i7-4770 processors with 12GB of memory each.

One machine was dedicated to running Zookeeper (a dependency of both Kafka and Storm), the Kafka server and Storm Nimbus, which is responsible for assigning tasks and monitoring other nodes. Two others act as Storm supervisor nodes. These are the nodes that host and govern several worker processes to complete the tasks previously assigned by the Nimbus node. Finally, the remaining machines were allocated to the hosting of the agents running in each test (evenly divided between each of the three machines).

Regarding the CPPS, the agents were instantiated with a dummy hardware communication interface, which for each agent simulated the generation of data from its respective emulated resource. This was achieved by developing a small Java graphical application which enabled the user to plug and unplug virtual resources that, once plugged, generated a new random value between 100 and 200 every 100 milliseconds. Additionally, every time a new value was generated the resource had a three percent chance to enter failure mode, which forced it to generate data of increasingly larger values in small increments for 30 cycles.

For each resource a CMA was deployed, collecting all its data and publishing it to a Kafka topic. This topic was consumed by the Kafka Spout in the Storm topology in order to enable the processing of the emulated shop-floor data. Once this process was concluded, the result was once more published to Kafka and made available to be consumed by the

Table 6.1: Apache Storm Topology Metrics

#Agents	Bolt	Capacity	Execution Latency (ms)
75	Split Stream	0.026	0.145
	Window Agg.	0.034	0.062
	Data Analysis	0.032	0.059
	Context	0.016	0.023
150	Split Stream	0.034	0.146
	Window Agg.	0.055	0.077
	Data Analysis	0.048	0.060
	Context	0.048	0.035
300	Split Stream	0.100	0.224
	Window Agg.	0.207	0.123
	Data Analysis	0.189	0.119
	Context	0.181	0.067

CPPS one more.

For the scalability tests, three different deployment configurations were used, each doubling the number of resources/agents deployed in the previous one and running for 30 minutes. A summary of the data collected during these runs can be seen in Table 6.1.

The metrics observed in Table 6.1 are capacity and execution latency, extracted using Storm's UI daemon. The former is referent to the processing capacity of the bolts deployed in the Storm topology. The closer the value is to "1.0", the closer the bolt is to be running as fast as it can. This is useful to verify if the parallelism of the topology needs to be adjusted. The latter refers to how long each tuple takes on average (in milliseconds) to be executed by the respective bolt. The Logger Bolt was excluded because during testing it was only quickly logging the model updates, thus its metrics were very close to zero and not relevant for comparison. An additional metric that was extracted was the complete topology latency, meaning the time it takes each tuple, on average, to be fully processed and acknowledged by the entire topology. Its values were 11.452ms, 13.441ms and 16.222ms, for 75, 150 and 300 agents respectively.

On the other hand, the second data flow, represented by the dashed line, pertains to the Knowledge Management updates used to demonstrate the flexibility of the initial implementation regarding the analysis process. The data generated by the virtual resources was used to train two different ML models beforehand, more specifically K-Means Clustering and Logistic Regression classifiers. An example of the output from the latter can be seen in Fig. 6.2.

As it can be observed, the model outputs the probability of failure within a given number of cycles. For testing purposes, the training data was labelled based on a 5-cycle period,

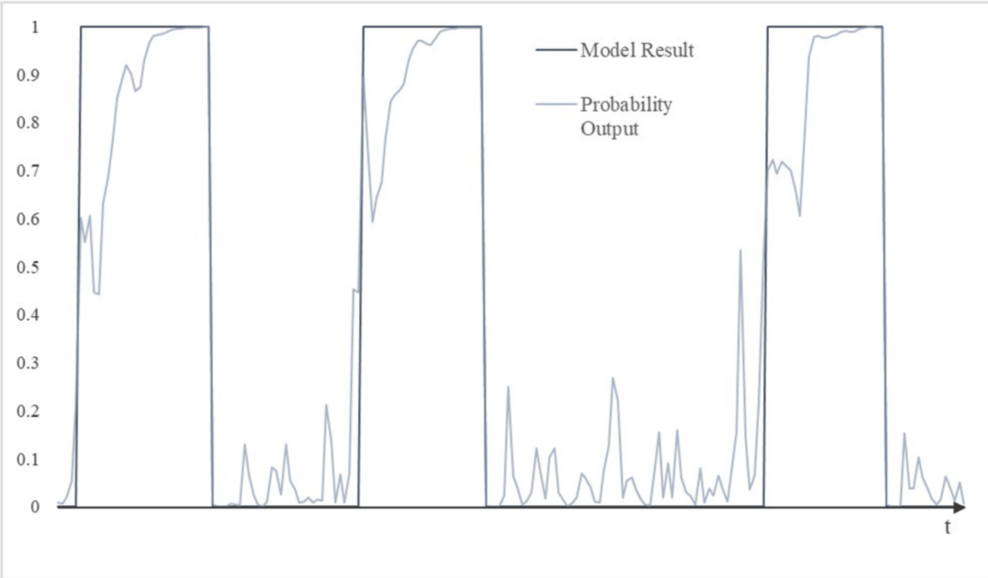

Figure 6.2: Example of the output from the Logistic Regression Model

and it was considered that a probability above 60% would result in a model classification of “1” as a representation of impending resource failure.

Upon deployment, every agent is initialized being associated with the aforementioned model, passing this information to the Storm topology via Kafka. During runtime, this configuration was then changed by the Knowledge Management tool, individually shifting certain resources to the clustering model instead. This process is initiated once the user has finished introducing all the information pertaining to the new model in the Java application. The serialized update is published to a Kafka topic, being then consumed by the Kafka Spout and emitted into the Storm topology. Internally, it is then split into the model and context updates, and sent to the respective bolts. The latency associated with these updates was measured over 100 iterations, consisting in the average timespan in milliseconds between the user publishing the update to Kafka, and each bolt completely updating its internal execution process accordingly.

Finally, the pluggability (associated with both scalability and adaptability) was tested using the deployment tool mentioned in the beginning of this section. For this purpose, timestamps were extracted from two specific moments for both the plugging and unplugging of resources/agents. Once when pressing the button to launch/remove a new resource, mimicking the detection of a new/removed shop-floor asset by the Deployment Agent, and then again when the respective agent is deployed and ready to start publishing data, or when it is removed from the agent platform. The results are summarized in Table 6.2, representing the average latency measured when unplugging and plugging randomly chosen agents 100 times during execution, one at a time, with arbitrary intervals in between each

Table 6.2: CPPS Pluggability Test Results

#Agents	Plug Latency (ms)	Unplug Latency (ms)
75	248.889	3.556
150	252.000	5.050
300	249.560	11.300

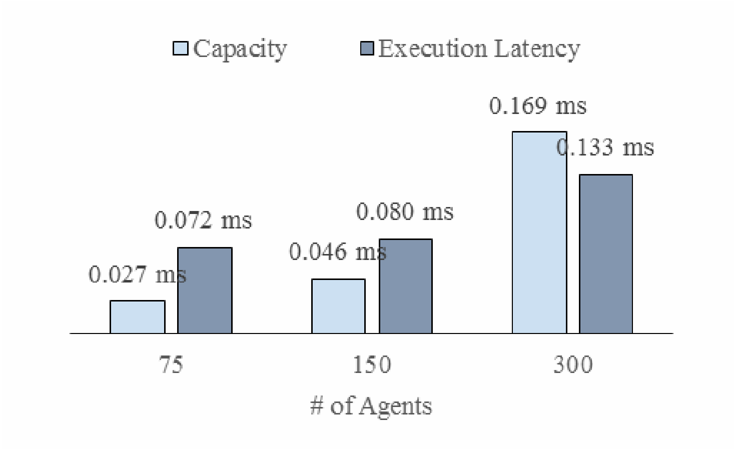

Figure 6.3: Apache Storm Topology's Average Capacity and Execution Latency

action and with a varying number of active agents for each run.

Regarding the first set of tests pertaining to the scalability of the solution, the implementation was deployed on a four-node cluster and tested with varying data throughput rates. This rate started at around 750, then 1500, and finally 3000 data points per second, corresponding to 75, 150 and 300 virtual resources/agents, respectively. The comparison of the results for the RDA module can be seen in Fig. 6.3.

As it can be observed, even with 300 concurrent resources/agents each tuple was spending slightly less than 0.2 millisecond at most per bolt. The average capacity of each bolt increases significantly for the last set of the test, but is still considerably less than "1.0" meaning that the bolts are not being forced to run faster than they can, which would result in unwanted queueing and delays. Regardless, once the capacity reaches higher values as the system scales further, this issue can be tackled by increasing the parallelism of the topology.

Regarding the knowledge management flow, the latency associated with each update to the running system was also measured under different throughput rates. The goal was to verify the impact that increasing volumes of data could have in the system's capacity to adapt the RDA process.

The results for the average update latency can be visualized in Figure 6.4, along with a comparison with the overall complete latency for the tuples pertaining to the shop-floor data stream.

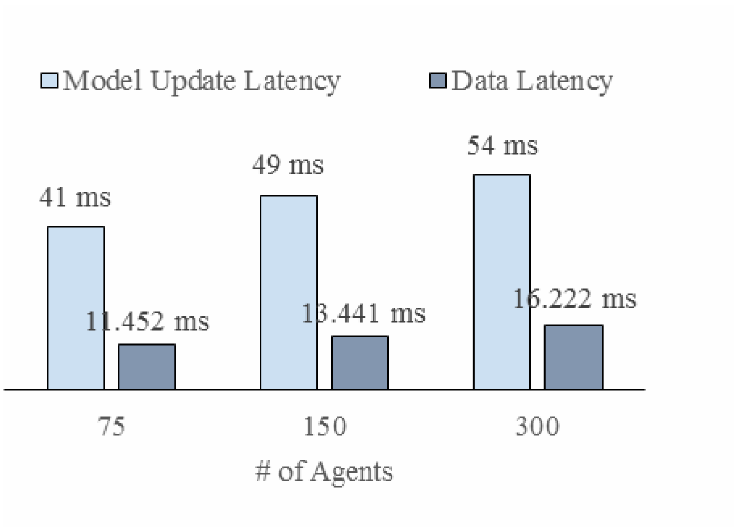

Figure 6.4: Shop-floor Data and Model Update Latency

The latency associated with the knowledge management updates is shown to increase as the system scales, albeit very slightly. These results show that not only is the Knowledge Management tool capable of adapting the RDA during execution, but also that the system's scale has very little impact in the performance of this updates. As far as the complete latency for the data stream is concerned, the overall impact of the system's scale is also relatively small in comparison. This is considering that while the scale doubles for each run, the latency only increases by a factor of around 1.174 and 1.217 between each of them.

This can be attributed not only to the increase in the processing time, but also in the increased network load and respective associated delays. Lastly, for the pluggability tests the latency was measured between the moment a new/removed resource is detected and the instant the agent is fully initialized/removed from the platform. The comparison of the results is illustrated in Fig. 6.5.

It is interesting to note the difference between the results for unplugging and plugging events. At first glance, it might seem that the results suggest the agent platform simply takes a lot longer to handle the deployment of new agents. However, this difference lies mainly in the monitoring agent's ramp-up time, since the measurement is taken only after it has fully completed its setup process, which includes initializing its data collection and output communication libraries.

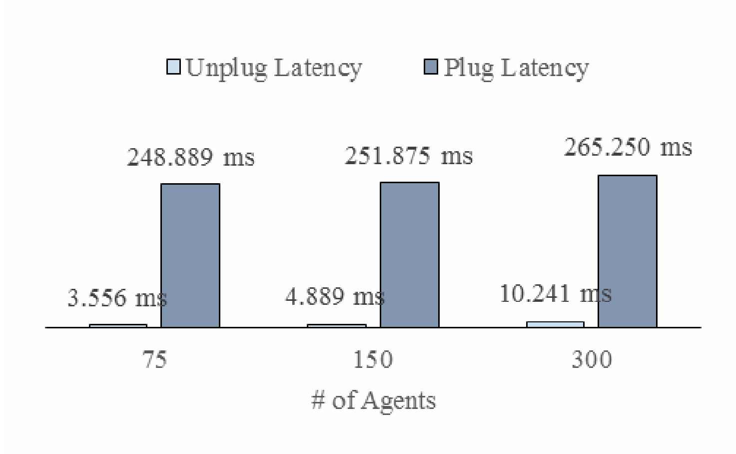

Figure 6.5: Pluggability Latency

6.2 Multi-stage Quality Control Scenario

The first large scale scenario for the application of the IDARTS framework is focused on the aspect of multi-stage quality control (see Definition 6.1), having been encompassed within the scope of the H2020 GO0D MAN Project. This scenario is an example of an application of IDARTS and its principles in three different real-world use cases, in spite of following a different route in regards to the way IDARTS was implemented.

GO0D MAN targets the development of a cyber environment capable of integrating both process and quality control in multi-stage production to achieve zero defect manufacturing. To do so, an agent-based CPPS is proposed to integrate the shopfloor components with higher-level functionalities, such as services to provide big data analytics and knowledge management.

Definition 6.1 (***Quality Control***)**.** *1. Set of activities designed to evaluate the quality of developed or manufactured products 2. Monitoring service performance or product quality, recording results, and recommending necessary changes (ISO/IEC/IEEE, 2017).*

In order to provide such an intelligent platform capable of targeting Zero-Defect Manufacturing even in existing legacy systems, GO0D MAN focuses on the combination of the latest ICT advancements encompassed in the concept of Industry 4.0, including both Edge and Cloud Computing, the Internet of Things and Big Data Analytics.

To this end, the IDARTS framework was adopted with the purpose of guiding both the architecture design and implementation stages of the GO0D MAN ZDM approach throughout the project's life cycle. Due to the lack of hard real-time constraints within the project's requirements, there was no need to adopt a real-time data analysis component. Due to the modular nature of IDARTS' components, the framework could be employed

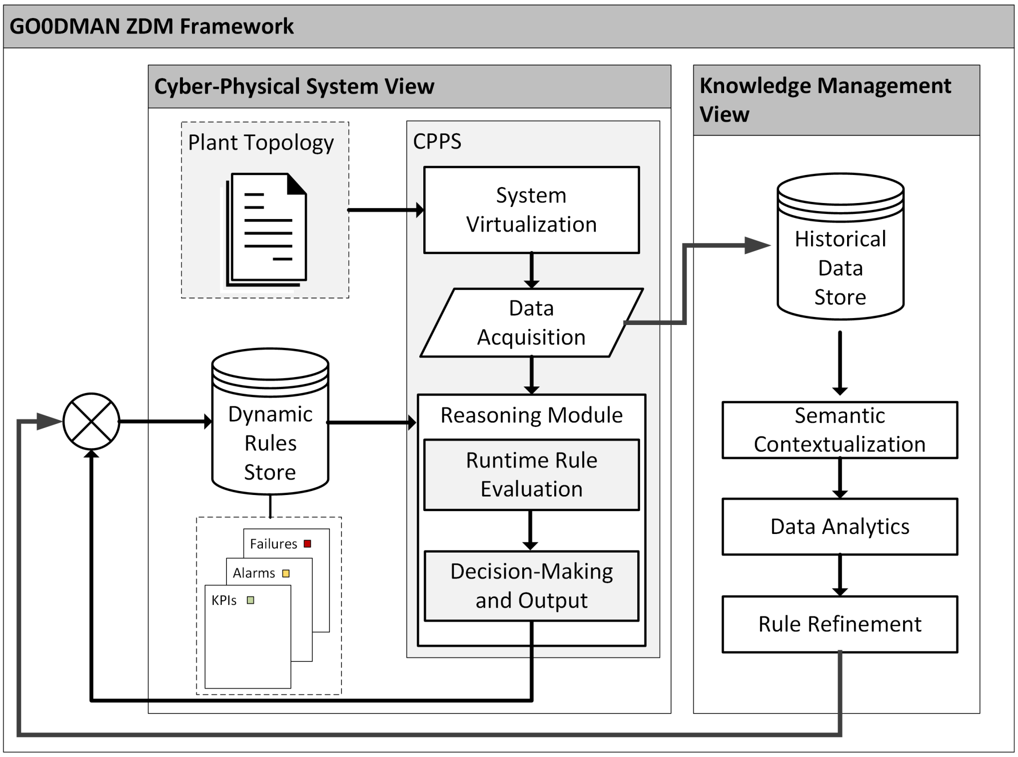

Figure 6.6: H2020 GO0D MAN adaptation of the IDARTS Framework (GO0D MAN Project (ID:723764), 2017)

simply using its remaining components without requiring any additional change. An overview of the complete adaptation of the IDARTS framework and the interactions between its elements can be seen in Figure 6.6.

Within the scope of the project, the key contributions that the framework provided were:

- The Integration of the Physical and Software Elements – Through the application of a CPPS, GO0D MAN's real-time computation layer should be capable of extracting data from the shop-floor and reason on it in order to assess possible deviations and act accordingly, thus preventing the propagation of defects downstream in a multi-stage manufacturing environment.

- Seamless Data Exchange between Heterogeneous Components – The employment of a common data representation and exchange format in crucial to ensure the interoperability of the heterogeneous components comprising the GO0D MAN platform.

- Knowledge Management and Data Analytics – Despite the exponential growth in the volume and velocity at which data is generated in manufacturing environments (e.g. embedded sensors), a large portion of it remains untapped. GO0D MAN's approach aims to translate this data into a business advantage by employing advanced data

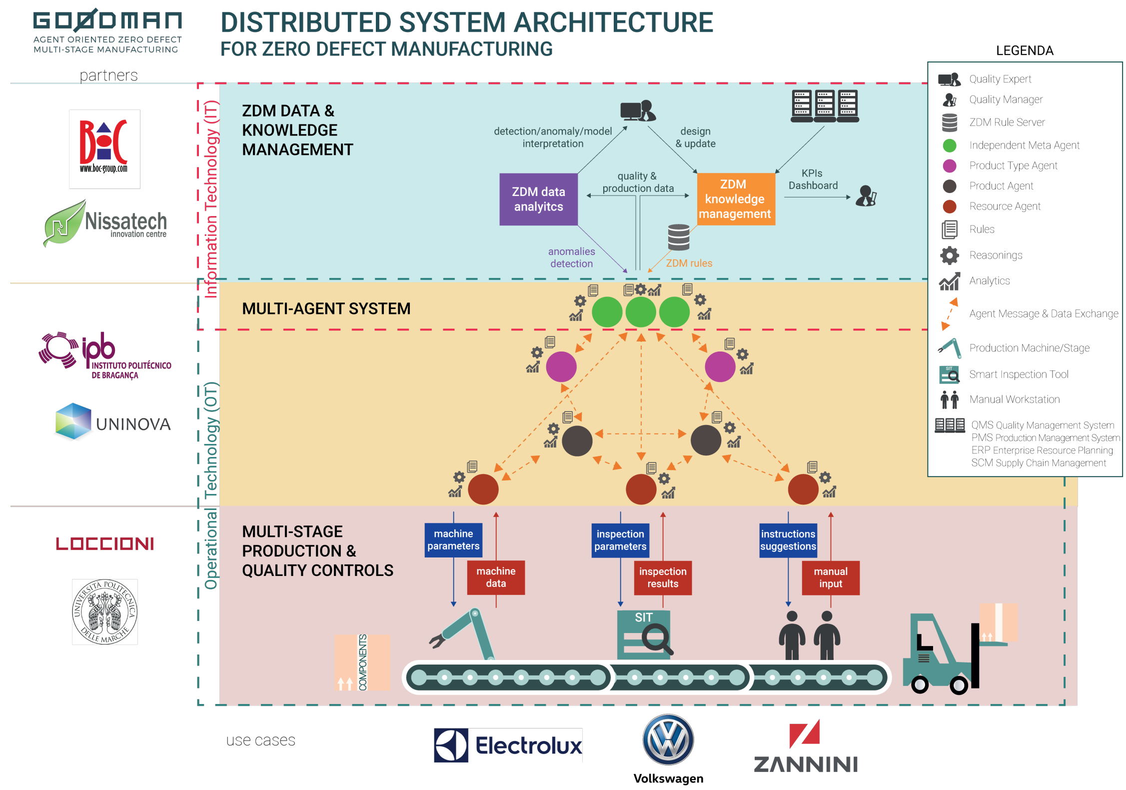

Figure 6.7: The GO0D MAN Architecture for ZDM (GO0D MAN Project (ID:723764), 2018)

analysis and knowledge management methods on semantically enriched data acquired by the CPS. Generated knowledge can be then used to improve the CPPS' reasoning system, hence further mitigating the occurrence and propagation of defects during production.

From this, the GO0D MAN distributed system architecture was designed, comprising three main levels of action. At the field level, smart inspection tools are deployed within the multistage production environment to collect additional data and send it to the cyber-level in a common data representation format adopted by all of the PMS' actors (R. Peres, Rocha, Matos, & Barata, 2018). At the edge-computing level a MAS-based CPPS is employed to serve as an intelligent "virtual glue"that connects the different system elements. This MAS abstracts the multistage production system and the smart inspection tools, collecting event data and process them according to a set of reasoning rules. Lastly, the MAS also passes these data to the cloud data and knowledge management component, which is responsible for the actual data analysis, the capturing of domain expert knowledge and the refinement of the reasoning rules used by the MAS. These layers are reflected in the architecture representation provided in Figure 6.7.

To verify this implementation's fulfillment of the requirements specified in Section 4.1.2, each of these requirements was mapped to the GO0D MAN's architecture layers, as illustrated in Table 6.3.

Table 6.3: Verification and mapping of the IDARTS requirements in the GO0D MAN implementation. Legend: ● - verified; ○ - partially verified; × - not applicable.

	QCs	MAS	KM		QCs	MAS	KM
NFR-01	●	○	○	FR-01	●	●	
NFR-02	●	●	●	FR-01-01	●	●	
NFR-03	●	●	●	FR-02	●	●	
NFR-04				FR-02-01		●	
NFR-05	●	●	●	FR-03			●
NFR-06	●	●	●	FR-04		●	●
NFR-07	×	×	●	FR-04-01	●	●	●
NFR-08	●	●	●	FR-04-02		●	
NFR-09	●	●	●	FR-04-03			●
NFR-10	●	●	○	FR-05		○	○
				FR-05-01		○	
				FR-05-02			○

From the observation of Table 6.3, it can be concluded that while most of the requirements are completely fulfilled, there are still some gaps to fill which suggest some requirements (from both the functional and non-functional sides of the table) are harder to be fulfilled when migrating these types of solutions from a laboratory or prototype level implementation, such as the one discussed in Section 6.1. Particularly, as far as non-functional requirements are concerned, this relates mostly to the aspects of adaptability (NFR-01), reliability (NFR-04) and usability (NFR-010).

Regarding adaptability, from a data analytics standpoint it can be considerably challenging to ensure that data-driven models are always capable of coping with any changes at the shop-floor level, especially if we're imposing that no additional programming or modeling effort should be required. One way to partially tackle this, which was in fact adopted in this implementation, is to schedule periodic updates to the models deployed to production, thus ensuring they remain relevant and operational as time passes. This does not, however, guarantee that the data analytics approach is robust when faced with the addition or removal of several different data sources, as even if it remains operational it certainly isn't guaranteed that it is utilizing the data to its full potential without reworking the analysis and modeling process.

Now concerning the topic of reliability, this is also considerably difficult to guarantee while also respecting the requirement of non-invasiveness, as typically the necessary infrastructure simply isn't in place and ready to be used. This can be tackled by adopting technologies with built-in reliability mechanisms (such as Apache Kafka) to handle the transport of data, but these are often still fairly complex and require considerable effort to maintain in a production environment.

Last but not least, the subject of usability is critical when discussing the industrial adoption of these solutions. They need to be easy to use and also able to convey information in a way which is readily understood by people coming from a wide array of educational and technical backgrounds. While parts of the solution can easily act as a black box while carrying out their respective duties (e.g. the MAS' data acquisition), the Knowledge Management layer in particular needs to convey the results from the data analytics module in a way that aids the production managers and the operators, instead of adding more complexity to their already complex processes.

6.2.1 Volkswagen AutoEuropa Use Case

The Volkswagen AutoEuropa (VWAE) use case consists is an automotive production plant for the Volkswagen group, responsible for manufacturing several models, such as the Alhambra, Scirocco, Sharan and T-Roc, with the latter being the focus of the use case at hand.

The VWAE production line comprises four main areas, more specifically the Press Shop, the Body Shop, the Paint Shop and Final Assembly. For the purposes of this use case, only two of these areas will be considered, namely the Body Shop, responsible for the manufacturing of the vehicle's body in white, and the Final Assembly area, responsible for assembling the final components onto the painted vehicle, including for instance the rear and tailgate lights. The Final Assembly also includes quality inspection concerning the assembled parts and if necessary the correction of the assembly operations, either through human operators fitting the car along the line, or ultimately through repairs at a repair station.

The assembly of the tailgate is of particular interest for the use case, since this process is completely manually performed by an operator. Another point of interest is the assembly and the alignment of both tailgate and rear lights performed in the Assembly area. In this area, an inspection of the product is executed in terms of its gap and flush measurements, which is also made manually by an operator.

Based on these points, the Body Shop and Final Assembly areas are considered as particularly suitable candidates for improvement via the application of a PMS solution focusing on controlling the characteristics that affect the gap and flush measurements of the product. To this end, an instantiation of a generic MAS-based CPPS was deployed as illustrated in Figure 6.8.

As it can be seen, there are four main sources of data from different production stages and areas, namely two in the Body Shop and two in the final Assembly area. Regarding

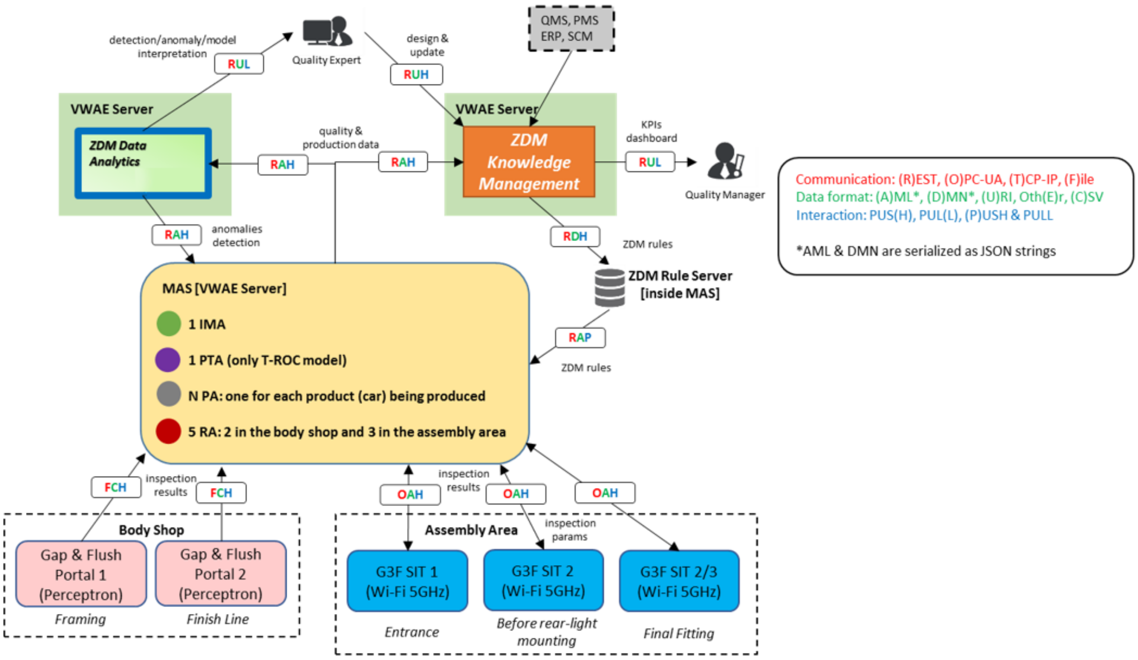

Figure 6.8: Instantiation of the GO0D MAN PMS for VWAE. Adapted from (GO0D MAN Project (ID:723764), 2019a).

the former, these are generated by a Perceptron inspection stations which have been integrated as legacy systems, abstracted by the MAS-based CPPS. The Assembly are is covered by the deployment of hand-held Smart Inspection Tool (SIT) which can be used by the line operators to quickly collect measurement data, being then relayed to and aggregated by the MAS. In turn, the MAS sends these data to the data analytics and Knowledge Management (KM) components, whose responsibilities relate to finding possible correlations and predictive modeling, as well as visualization and capturing domain-expert knowledge, respectively. Combined, these can then be used to adapt the rules used by the MAS to enable the early detection of anomalies in the use case's multi-stage production environment.

For this use case, the following agents were be deployed:

- 1 Product Type Agent representing the car model being produced in the factory plant (note that the product variants are reflected in parameterized process plans). This PTA manages the process plan of this car model and is responsible for launching Product Agents according to the production orders coming from the Manufacturing Execution System; thus, for each car that starts the production, one Product Agent is launched.

- As many Product Agents as the number of cars being produced simultaneously in the factory plant. Each Product Agent is responsible for managing the production execution of the car along the line, namely, collecting and storing the production data, and monitoring the evolution of the car's production.

- 5 Resource Agents, two associated each inspection station from the Body Shop and the remaining three associated to SITs in the Assembly inspection area.

- 1 Independent Meta Agent that is responsible for several tasks, such as the aggregation of the collected data, triggering of the big data analysis aiming to obtain new knowledge and propagation of the new rules generated by the KM to their respective agents.

The data collected by these agents is sent to the data analytics component via a REST API, following a common data representation implemented using AutomationML as described in (R. Peres et al., 2018), which is used for all communications with the MAS. With this, clustering models are used to find outliers in the data set using the historical data collected over time, updating the deployed models on a daily basis using a scheduled job.

The results from the previous step are then passed onto the KM component which provides the functionality regarding the visualization of the analysis results, enabling quality control experts to interpret the data and enrich it with their domain knowledge through a web-based interface, generating new rules or adapting existing ones.

Lastly, the rule updates are then stored in a rule knowledge base using once more a REST API, which also automatically handles their transformation onto the common format accepted by the MAS-based CPPS. These are then passed back to the MAS through the Independent Meta Agent, who is then responsible for distributing them onto the appropriate agents responsible for abstracting the rule's respective entity, be it a product type or a resource.

6.2.2 Zannini Use Case

The Zannini use case consists in a multi-stage batch production process of high precision turned components for automotive applications, particularly focusing the production of sleeves for hydraulic electro-valves used in car engines.

The production process is characterized by a high volume of parts (reaching several hundred thousand of work-pieces per year) which are managed in batches of a pre-determined size (e.g. one thousand pieces per batch), passing through several production stages with a lead time that can vary between a minimum of one week up to a few weeks. Each individual production stage involves a different machine with a particular takt time, with the quality of the final product depending upon the accumulated quality results of each individual operation. More concretely, the stages contemplated in this use case involve turning, deburring, honing and final check operations.

These characteristics imply that that the individual produced metal parts are not uniquely identified, so quality inspection results from different stages can be compared only on a batch level and not on a single product level. Another interesting aspect of this use case that attests the implementation's ease in terms of its migration is the fact that all the elements instantiated for this use case were first tested in Zannini's facilities in Italy and later on replicated and deployed for production in Poland. The instantiation of the PMS for this use case is described in Figure 6.9, with two SITs deployed at the field level, the MAS-based CPPS running in a local server and the data analytics and KM running at the cloud level.

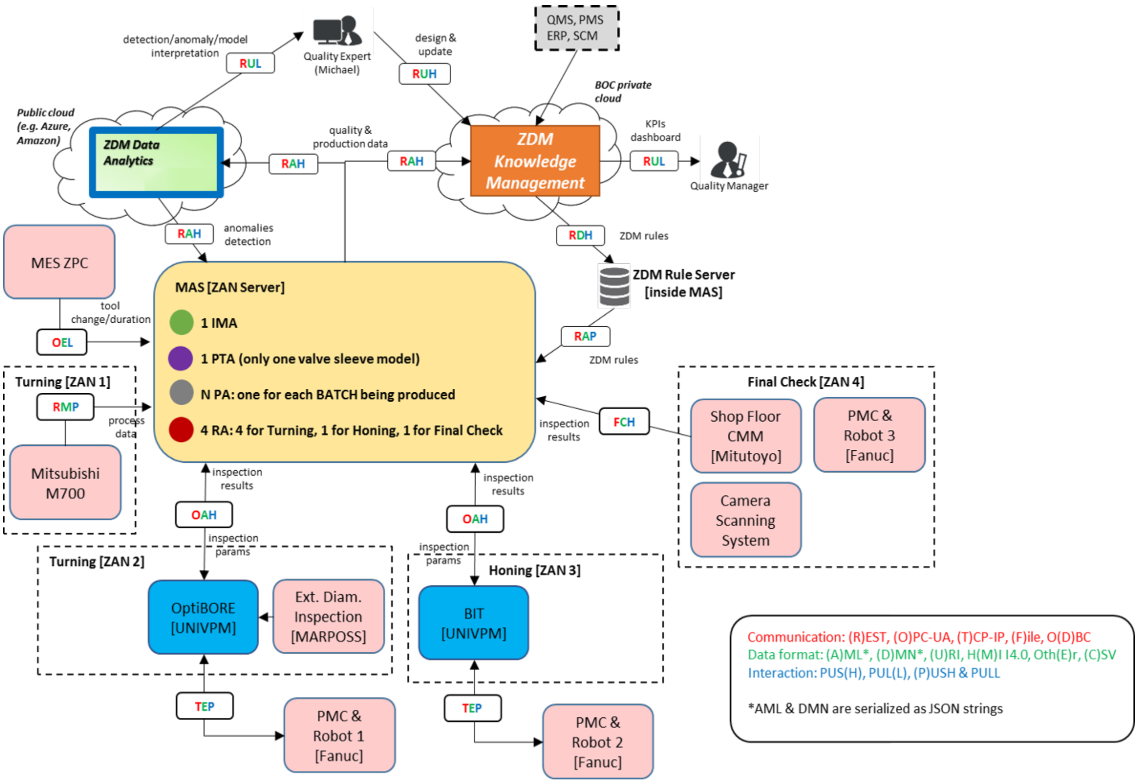

Figure 6.9: Instantiation of the GO0D MAN Architecture for the Zannini use case. Adapted from (GO0D MAN Project (ID:723764), 2019b).

From the observation of Figure 6.9 it can be concluded that there are four main sources of data related to different stages of the production, namely two stations associated with turning, one with honing and one pertaining to the final quality check. Based on this, the MAS was deployed as follows (each agent's responsibilities were omitted as they are the same as described in Section 6.2.1):

- 1 Product Type Agent representing the valve sleeve model;

- 1 Product Agent for each batch being produced, since there is no product traceability at the individual level;
- 1 Independent Meta Agent;
- 6 Resource Agents, four associated with turning (related with axis current, spindle load and speed in ZAN1, geometrical characteristics such as internal and external diameters and length in ZAN2), one for the honing (associated with the detection of burrs after the honing process) and one for the final dimensional check.

All of the different data is extracted and aggregated by the MAS-based CPPS, which similarly to the previous use case is responsible for passing it along to the data analytics and KM components.

For this use case the dataset corresponds to the process parameters per batch, consisting of around 1000 products each. For the first stage 22 parameters were analysed, 3 parameters for the second stage (being more specifically the overall length, the internal diameter and the external diameter) and 2 parameters for the fourth stage. The data analytics approach that was employed is unsupervised, thus not relying on the involvement of domain experts.

To this end, statistical quality control techniques were applied to detect possible anomalies in past data via outlier detection. Since these techniques rely only on the available data, it is possible to detect outliers without specific knowledge of the specification and tolerance values for the use case.

From the outlier threshold obtained via this method, it is possible to then use this knowledge to monitor deviating parameters online and avoid the propagation of defects downstream. These values can also be later matched to the use case's own specifications and tolerances to further filter anomalies from simple outliers.

Regarding the KM component, the deployment was made to a private server which enables the visualization of the data analytics results as well as the calculation of different Key Performance Indicator (KPI)s such as the capability index. Other functionalities include the generation of the rules related to the thresholds resulting from the data analytics and a process view to describe the corresponding task flow.

6.2.3 Electrolux Use Case

The last of the three GO0D MAN use cases is part of the Vallenoncello plant from Electrolux Professional in Italy, a producer of large equipment for professional kitchens. Regarding this use case, the assembly line producing a particular line of ovens was chosen as the focus, being organized in a lean manufacturing concept characterized by multistage single piece flow, low volume with mixed models and intensive operator labor.

In general terms, this line encompasses several pre-assembly stages for varied components, which are then provided and assembled in the main assembly line. At the end of this part a final functional and quality test is carried out which checks every single feature of the oven being produced. The instantiation of the PMS enables the inclusion of preliminary on-line quality checks between the different stages supporting the reduction of the number of ovens which are not right on first pass, which serves as an indicator of the overall capability of the whole process. This instantiation can be found in Figure 6.10.

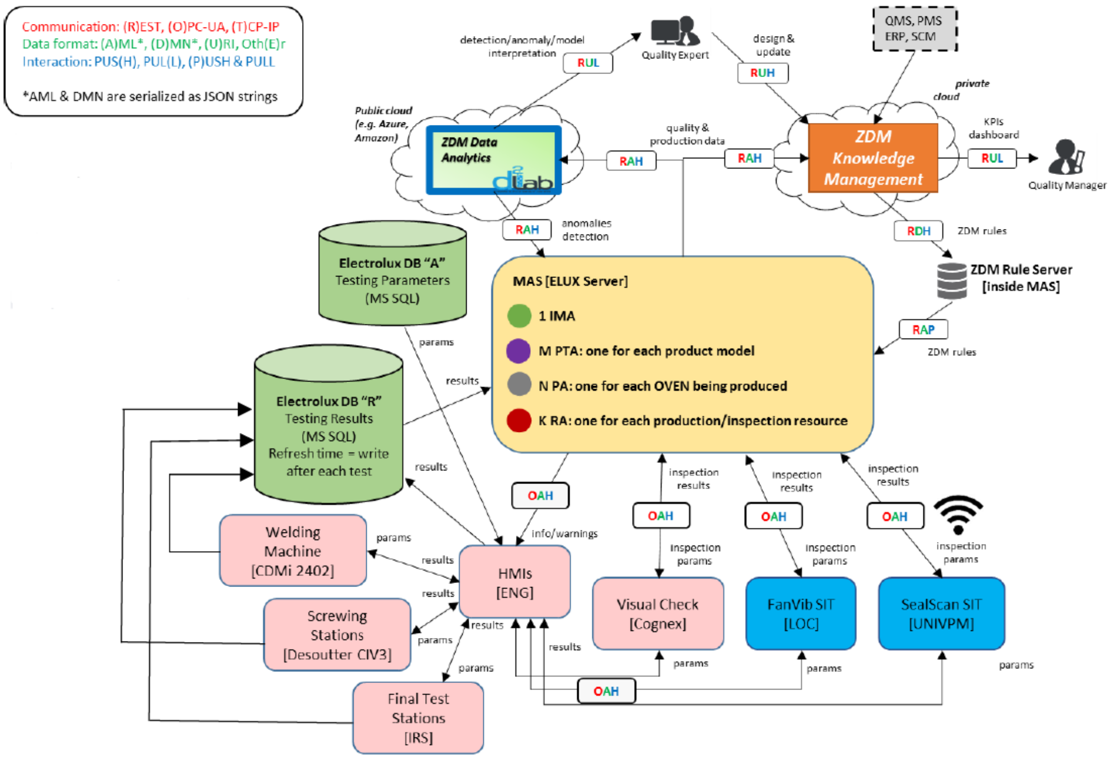

Figure 6.10: Instantiation of the GO0D MAN Architecture for the Electrolux use case. Adapted from (GO0D MAN Project (ID:723764), 2019c).

As it can be observed, in this use case there are several heterogeneous sources of data, considerably higher than those of the instantiations contemplated in Sections 6.2.1 and 6.2.2. These include various production stages within the multi-stage line, such as welding, screwing, assembly visual checks, motor-fan vibration tests, gap and flush measurements on the oven doors, steam and hot air leakage test and the final quality check.

With the exception of the motor-fan vibration, door seal leakage and the gap and flush tests which are performed by SITs, the remaining data sources entail the integration of several legacy systems in order to enable the whole PMS functionality. Akin to the remaining use cases, all these data is collected and aggregated by the MAS-based CPPS running at a local server, which then relays it to the remaining components on the cloud-level. The agents were instantiated for this line as follows (please refer to Section 6.2.1 regarding each agent's responsibilities):

- 1 Product Type Agent for each oven model;
- 1 Product Agent for each individual oven being produced;
- 1 Independent Meta Agent;
- 8 Resource Agents, one for each of the production and inspection resources.

The data analytics approach for this use case follows a similar approach to that of Zannini. The dataset contains measurements from the final inspection stage, corresponding to 1395 products, each uniquely identified by a bar code and a part number.

For this stage 101 have been analysed, first in terms of their internal correlations and then in regards to the detection of outliers in the multivariate space. Due to a high number of missing values in several of the parameters, 46 of them had to be excluded from the study.

Once again the data analytics approach is unsupervised, following the application of statistical quality control techniques to possible detect anomalies in past data.

Similarly to the previous use case, from the outlier threshold obtained via this method once can monitor deviating parameters online and avoid the propagation of defects downstream.

As a step further, it is also important to figure out the root cause of these outliers. As a first approach to this, the frequency with which each parameter was outside of a given statistical threshold can serve as an indicator for which parameters should be stabilized in order to improve the stability of the entire process.

6.2.4 Assessment of the GO0D MAN Use Case Instantiations

As far as the GO0D MAN use case instantiations are concerned, the aspects pertaining to data collection and pre-processing are assured by the CPPS comprising the MAS, the SIT and the legacy systems.

Furthermore, the adaptation rules originating from the KM and employed by the MAS are also successfully deployed with the assistance of the rule server which serves as a mediator between the two components. As a consequence, the instantiations show successful results in terms of the solutions capacity to identify outliers and contribute to the mitigation of defect propagation downstream.

However, the short-comings identified in Section 6.2 regarding the verification of the requirements are also evident, in addition to the lack of truly predictive analytics in the context of the multistage analysis. While at this point in time these instantiations show promising results at the single stage scope, there is still considerable work to be done to realize the full predictive potential of the proposed architecture in the context of a multistage system.

Furthermore, while the adoption of mainly unsupervised approaches grants an additional degree of independence to developers (as there is no explicit dependence on domain expert knowledge), as discussed in Section 3.4 the inclusion of domain experts in Industrial AI approaches can contribute to the interpretation of parameters, processes and in realizing the full potential of such solutions.

Given these points, Section 6.3 proposes an additional scenario as an extension of the VWAE use case, independent from the GO0D MAN project and focused on the predictive aspect of the solution through a supervised approach.

6.3 Mitigating Defect Propagation in the Automotive Industry

The VWAE use case is particularly interesting from a data analysis standpoint due to the complexity of its multi-stage process. This section presents an extended scenario contemplating the VWAE automotive bodyshop, focusing concretely on the assembly of the tailgate onto the car's frame, as well as the corresponding pre-alignment tasks carried out by human operators. This application appears as the result of a direct contact VWAE, with its analysis and implementation being independent from the GO0D MAN project described in Section 6.2. The overview of this scenario can be observed in Figure 6.11.

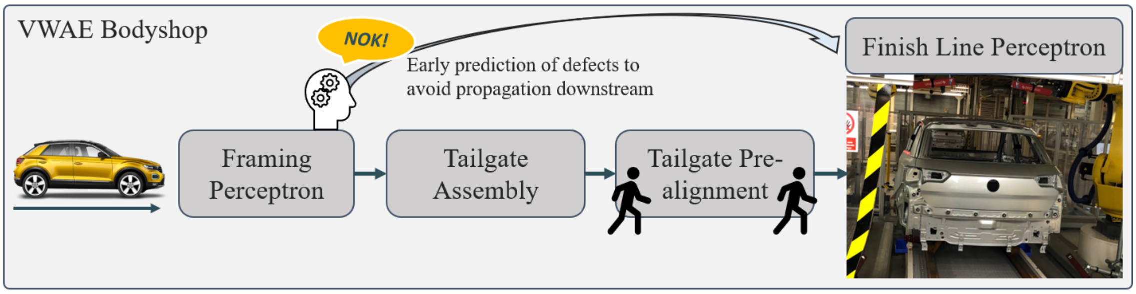

Figure 6.11: Layout for the multi-stage defect propagation scenario.

The core challenge of this scenario is the identification of correlations between the measurements taken by the Perceptron system in the framing stage, with the gap and flush quality control performed at the finish line Perceptron after the tailgate is assembled. In the former, the relative X, Y and Z positions of the car's frame are compared to those of the design, with deviations being recorded by the Perceptron system into a local database. The latter takes place after the pre-alignment of the recently assembled tailgate, which is performed by human operators, and as mentioned takes gap and flush measurements and also stores them into a local database.

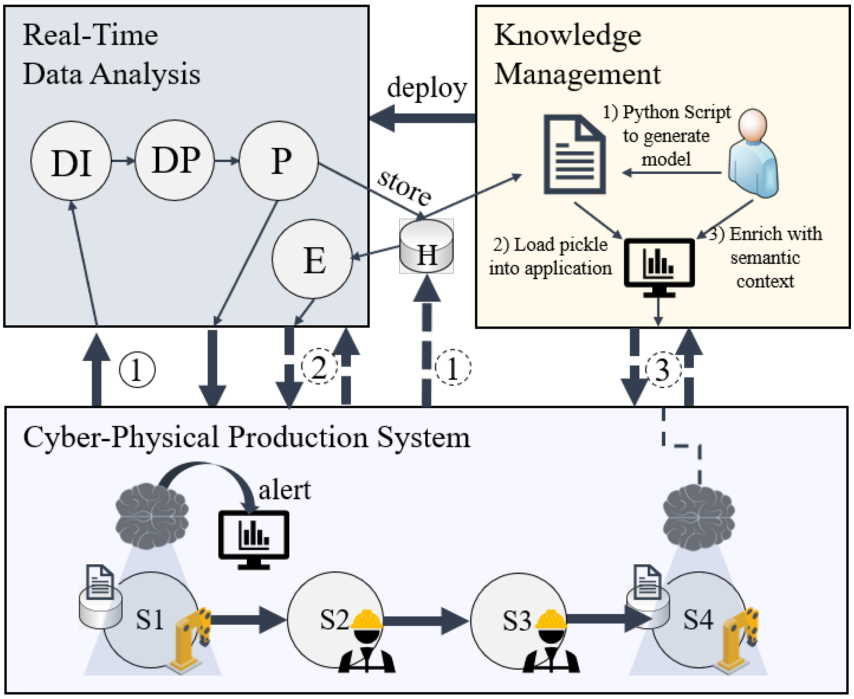

Figure 6.12: Possible deployment architecture based on the IDARTS framework (R. S. Peres, Rocha, Leitao, & Barata, 2018). Legend: DI - Data Ingestion; DP - Data Preprocessing; P - Prediction; E- Evaluation; H - Historical database; 1 - Prediction request; 1 (dashed) - Store ground truth; 2 (dashed) - Request mode evaluation; 3 (dashed) - Request updated model.

The inclusion of the human in the loop performing corrective alignment actions (fitting) adds a considerable amount of variation to the process, making it fairly to detect direct correlations between the numerical values collected in the framing stage and those of the finish line. In turn, small deviations in the frame alignment are propagated downstream through all the stages in between and can have a considerable impact in the gap and flush measurements resulting from the assembly of the tailgate.

Hence, the main goal here is to develop a solution that is capable of predicting whether or not a given car will be within the quality tolerance values at the finish line based on the frame measurements taken considerably earlier in the framing Perceptron station. For this purpose the PMS was instantiated as shown in Figure 6.12, based on an adaptation of the implementation described in Chapter 5.

The core difference here stems for the use case's time constraints, as real-time in this case is within the minute range as opposed to very tight time constraints in the order of a few milliseconds, as one would encounter in field-level control applications. Therefore an implementation of the real-time data analysis using Apache Storm could introduce an additional layer of unjustified complexity, having then been replaced by a Flask server which exposes the deployed model's functionality through a REST API.

For such an instantiation, the MAS can be used to implement the CPPS that abstracts the Multistage Manufacturing Process (MMP) with one agent associated to the framing stage (S1) and another to the final one (S4). While the framing agent can request quality predictions from the server hosting the deployed classifier and alert operators as defects are identified, the other can check for the ground truth associated with the measurements taken at the end of the line. These values can be stored in a historical database, with the agent either periodically requesting a re-evaluation of the model to trigger a re-fit if the performance goes below a given threshold, or having the model be periodically updated using the static model as a starting point, for models that support such a functionality. This can be more efficient than the first approach, as it reuses the existing state instead of discarding it, only updating it on the most recent historical data.

The usage of a MAS also enables the system to adapt to other changes in run-time, including for instance the addition or removal of elements from the line during production without requiring additional programming effort or downtime. This means that for instance handheld smart inspection tools can be added in to provide additional measurements for the stages in between with the system being able to automatically enact a self-organized response and accommodate such devices and new data into the existing solution.

To provide the reader with a different perspective from that of the previous sections, the remaining subsections of this chapter will bear a greater emphasis on the aspect of the data analytics and predictive modeling which empowers the data-driven side of the PMS.

6.3.1 Characterization of the data set

The data set encompassed a total of 18148 unique cars with 29 dimensional features from the framing inspection station. Considering that the raw data generated from the Perceptron stations was originally unlabelled, in order to turn this into a supervised learning problem the first step was to label the data from the Finish Line station.

To achieve this, interviews with domain experts from the bodyshop were conducted in order to extract the tolerances for each of features from this station. Based on this, each sample was attributed a binary label based on whether or not any feature was outside of the tolerance boundaries, more specifically each car sample is labelled as 'OK' or 'NOK' according to a domain expert's assessment based on the gap and flush measurements at the last station, with 11331 and 6545 samples belonging to each class, respectively. Considering that the order of cars can change between the two stages, samples from each station could not be matched simply by the timestamps, having instead been aligned using the cars' serial numbers. Finally, with this the framing dataset was labelled based on the ground truth regarding the presence of defects later on in the finish line station (typically within a time window of 90 minutes). An example excerpt of this dataset (with anonymized features) can be found in Table 6.4.

Table 6.4: Bodyshop Framing Dataset Example

F1	**F2**	**...**	**F45**	**DEFECT**
0.841178	0.482368	...	0.576703	0
0.822833	0.276967	...	0.335001	0
0.799453	0.357920	...	0.551354	1

Out of these 29 features, 10 present over 85% entries of missing values, resulting in only 19 features being used in the analysis.

Furthermore, to address the class imbalance, random under-sampling was performed on the data, generating a balanced data set with 12012 samples. Considering that the chosen sampling technique might result in some information loss, synthetic minority over-sampling was also tested as an alternative (after the train-test split) but provided significantly worse results.

Finally, 119 observations in the balanced data set still presented missing values, either due to the car still being on its way along the line between the two inspection stations, or due to some measuring or communication disturbance. Since these were relatively rare occurrences the samples with one or more missing values were discarded, although in future work it might be interesting to study the impact of different imputation techniques instead.

The absolute linear correlation matrix for the resulting data set can be found in Figure 6.13. As it can be observed, most features present low linear correlation coefficients with the target, suggesting that if there is in fact a relationship between the features and the car's quality downstream, non-linear classifiers might be more adequate for the case at hand. Also, there is some evidence of multicollinearity, with cases of high correlation between some of the features, which is to be expected given that the data set pertains to several dimensional characteristics of the car which are expected to be correlated.

To facilitate the visualization of the data set, PCA was applied to reduce the feature space to only three dimensions in order to make it possible to visualize it in 3D space. The resulting plot is shown in Figure 6.14.

Roughly 81% of total variance is captured in the first three components resulting from the application of PCA, however the results seem to suggest that based on the features available in the data set a reasonable class separation can be achieved. The following section provides an overview of the different algorithms employed to this effect in this study, as well as of the corresponding methods and implementation.

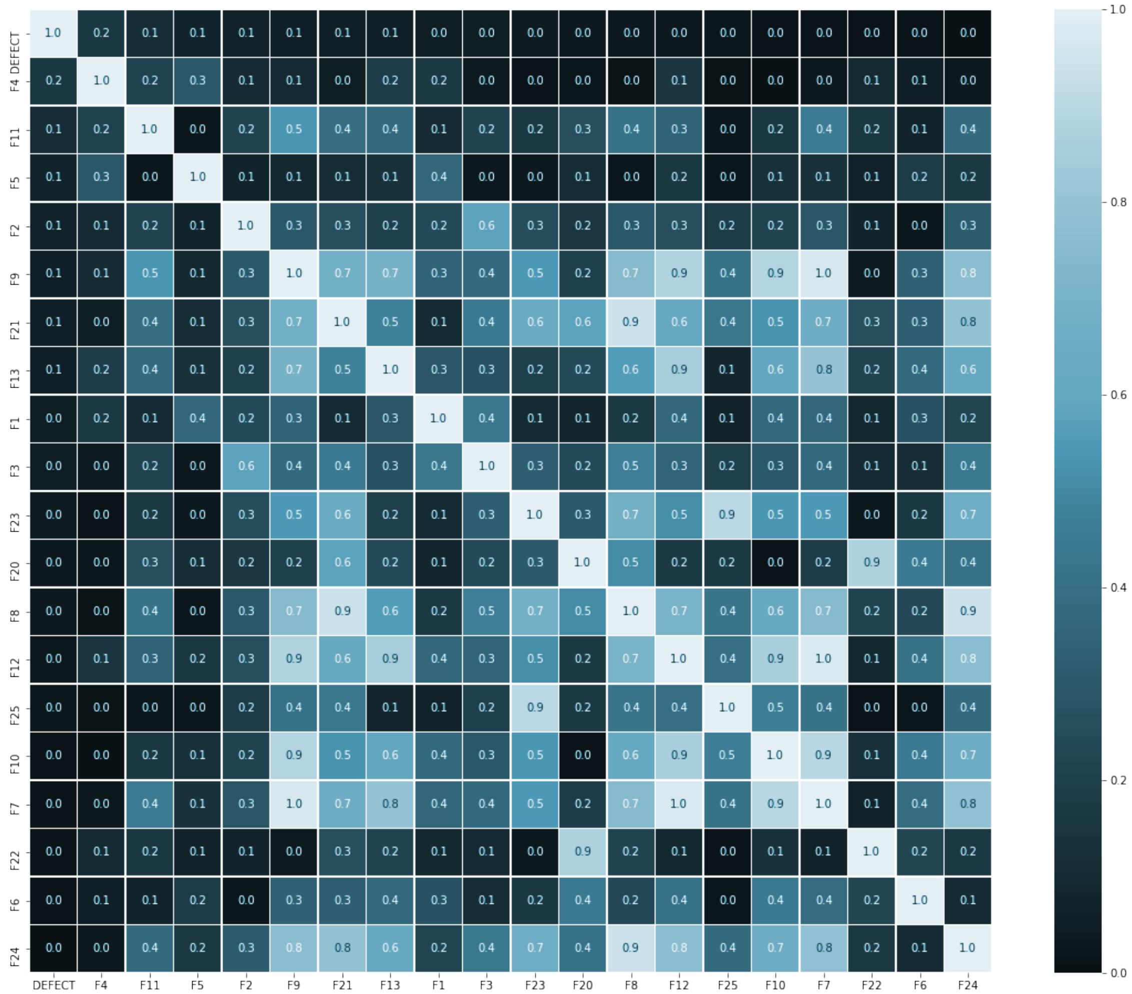

Figure 6.13: Matrix of the absolute correlations between the different features based on the Spearman coefficient.

6.3.2 Algorithms for Defect Classification

The implementation of the models contemplated in this study followed a fairly straightforward methodology, using Python 3.6 and the scikit-learn module (Pedregosa et al., 2012) for all models except XGBoost (Chen & Guestrin, 2016).

Firstly, the data set was split into train and test sets. Afterwards, given that several features present skewed distributions with both positive and negative values, Yeo-Johnson transform was applied to reduce the shift followed by standardization to center and scale each feature individually using the *RobustScaler* from scikit-learn's preprocessing module. This was used instead of a standard scaler due to it being more robust to outliers in the data. Both the power transform and scaler were fitted only to the train set to avoid test set contamination. An example of the result from this preprocessing step for F12 is illustrated in Figure 6.15.

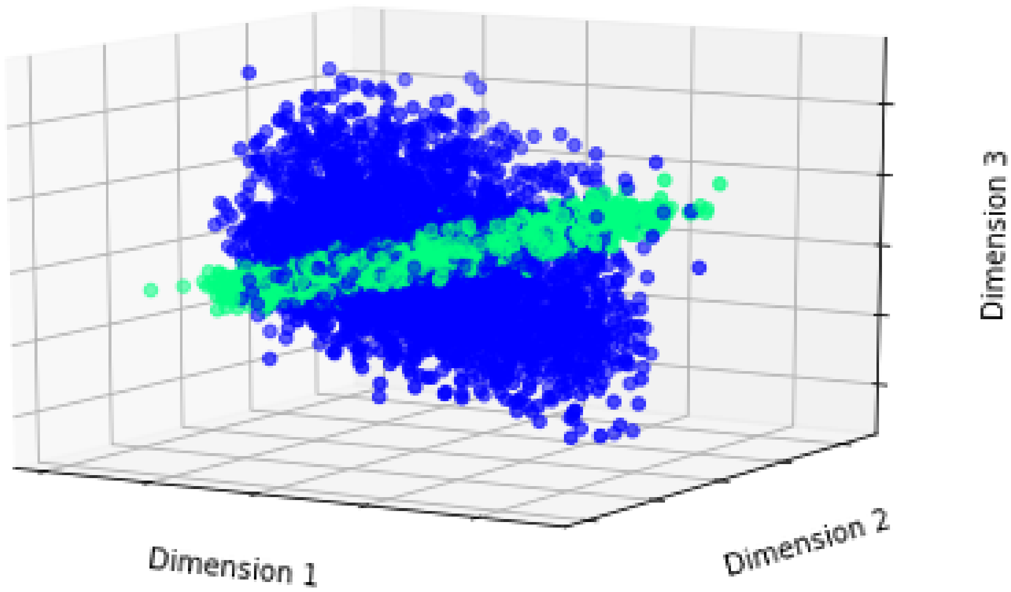

Figure 6.14: Visualization using PCA to reduce the dimensionality of the data set. Data points are colored based on the target variable.

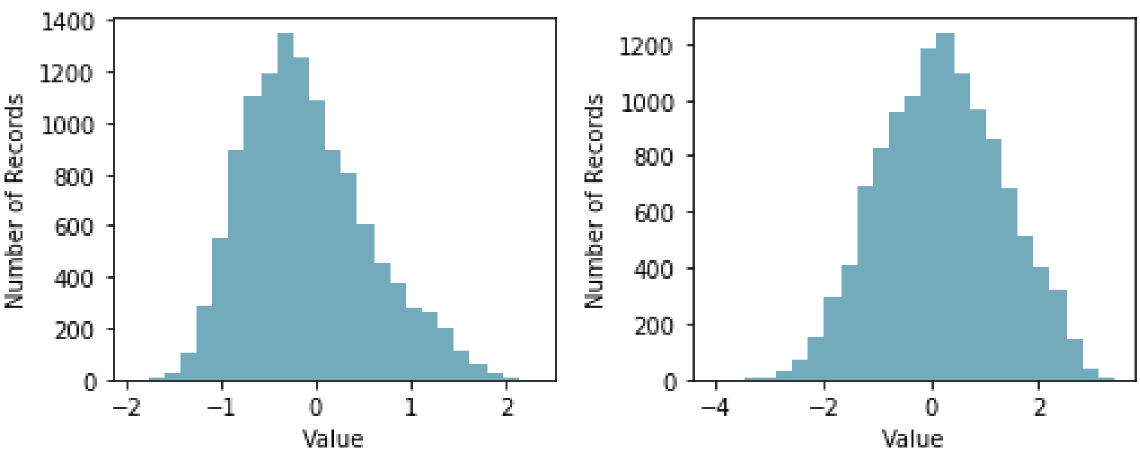

Figure 6.15: Result from the application of the Yeo-Johnson transform and robust scaler to F12. Raw distribution is presented on the left, transformed values are presented on the right.

Table 6.5: Baseline model results.

Model	Accuracy	Training (s)	Prediction (s)
Random Forest	87.873	0.148	0.006
SVC (RBF Kernel)	85.325	2.076	0.470
XGBoost	84.331	0.301	0.007
K-Nearest Neighbours	77.962	0.008	0.371
Logistic Regression	57.936	0.073	0.001
Naive Bayes	56.178	0.004	0.001

After this preprocessing step, several models were trained to establish a baseline with 5-fold cross validation being performed on the top scoring models. Following this, the best models from this step were selected for hyperparameter tuning and finally tested on a separate holdout set, consisting of cars collected over the three days after the last sample from the original data set.

6.3.3 Data Analysis Results

At first, several models were implemented without any hyperparameter tuning to create a baseline. The model training was performed on a machine with an Intel Core i7-9700K, 2x8GB 4000MHz DDR4 memory and an NVIDIA GeForce RTX 2070. The results are summarized in Table 6.5.

As hypothesized during the exploratory data analysis from Section 6.3.1, the two linear models, Logistic Regression and Naive Bayes, performed significantly worse than the worst performing non-linear classifier, suggesting a stronger non-linear relationship between the features and the target. Based on this, 5-fold cross validation was performed on the four non-linear classifiers in order to obtain a more realistic measure of accuracy and avoid overfitting on the training data. The results from the cross validation step can be found in Figure 6.16.

From the observation of Figure 6.16, it can be said that the baseline RF model performed better on average across all metrics except for recall, for which XGBoost was considerably superior. The XGBoost and SVM models performed slightly worse, with KNN performing considerably worse overall. One particularity to take into account in this MMP is the possibility of the feature distributions to change over time. This can happen for several reasons and is further discussed in Section 7.3, but to tackle this challenge, an approach could be to monitor the accuracy of the deployed model and retrain it if it drops below a certain threshold. This means that more computationally expensive models that take longer to train and perform cross validation on, like SVM, might not be adequate for such a scenario.

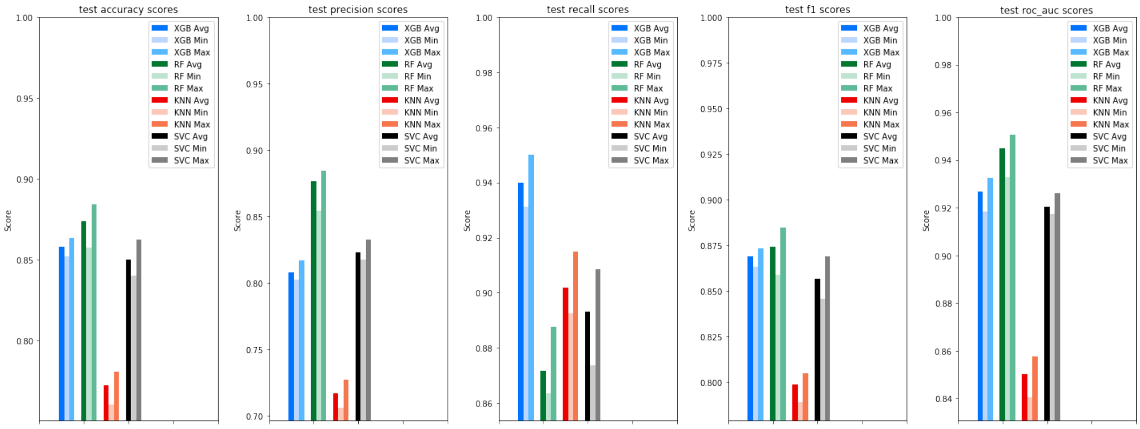

Figure 6.16: Test results from 5-Fold Cross Validation. Results are divided by accuracy, precision, recall, f1 and roc_auc scores.

Table 6.6: Parameters for each model resulting from the tuning through randomized search optimizing for roc_auc. Tuning was performed on 100 iterations with 5-fold cross validation.

Model	**Parameters**
XGBoost	colsample_bytree: 0.970, gamma: 6.079, learning_rate: 0.202, max_depth: 11, min_child_weight: 11.507, n_estimators: 59, reg_alpha: 0.232, subsample: 0.962
Random Forest	n_estimators: 500, min_samples_split: 2, min_samples_leaf: 1, max_features: auto, max_depth: 50
SVC	C=10, gamma=0.01, kernel='rbf'

Based on this, hyperparameter tuning through randomized search was performed on the three best models, which were then compared on the test set based on the same evaluation metrics used for cross validation. The tuned parameters can be found in 6.6, where any omitted parameters are assumed to take the default values from their respective implementations.

The results are summarized in Table 6.7. Additionally, the corresponding ROC curves can be found in Figure 6.17, in which the dashed diagonal line defines the reference point for which the models have no capacity to distinguish between classes.

The results are extremely close, especially for the two ensemble models, with XGBoost being superior in three out of the five evaluation metrics, if only by a slight margin when compared to the values scored by the Random Forest model. The SVC model appears to not have generalized as well as the others as evidenced by its lower capacity to separate the target classes in the ROC curve, albeit with marginal differences and while still yielding fairly improved results over those of its baseline counterpart.

Table 6.7: Tuned model results. Models are evaluated based on the same metrics used for the baseline models' cross validation.

Model	Accuracy	Recall	Precision	F1	ROC AUC
XGBoost	0.928	0.979	0.888	0.931	0.972
Random Forest	0.925	0.981	0.883	0.929	0.977
SVC	0.914	0.977	0.868	0.919	0.969

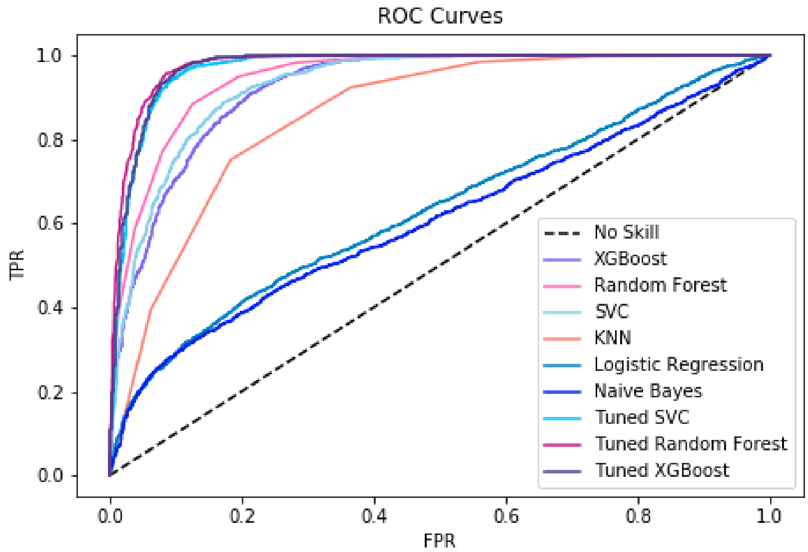

Figure 6.17: ROC curves for each of the models compared in this study. ROC is a probability curve and AUC represents a measure of separability between classes.

Finally, each model was tested on a new holdout data set, originating from measurements taken from 1000 cars over the three days following the last entry of the original data set (8 samples were discarded due to missing values). The resulting confusion matrices are depicted in Figure 6.18. The results suggest that the models were capable of generalizing well, being able to accurately predict the occurrence of defects in real car samples outside of the original data and thus provide important support in the earlier identification of deviations in the assembly line.

Assuming a 95% confidence level, for both the XGBoost and RF models the classification error is given as:

$$44/992 \pm 1.96 \cdot \sqrt{\frac{0.0443 \cdot (1-0.0443)}{992}} \rightarrow 0.0443 \pm 0.0128 \tag{6.1}$$

Based on Equation 6.1, there is a 95% likelihood that the confidence interval [0.0315, 0.0571],

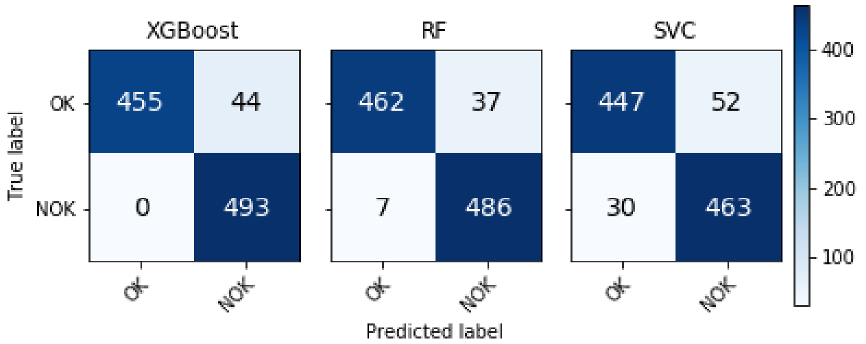

Figure 6.18: Confusion matrices for the holdout validation. The tuned XGBoost model achieved perfect recall on cars predicted over the three days after the last sample from the original data set.

or between 3.15% and 5.71%, covers the true classification error of the model.

Once more, to verify this implementation's fulfilment of the requirements specified in Section 4.1.2, each of these requirements was mapped to the implemented components shown in Figure 6.12, as illustrated in Table 6.8.

Table 6.8: Verification and mapping of the IDARTS requirements in the pilot implementation. Legend: ● - verified; ○ - partially verified; × - not applicable.

	CPPS	RDA	KM		CPPS	RDA	KM
NFR-01	●	●	×	FR-01	●	×	×
NFR-02	●	●	●	FR-01-01	●	×	×
NFR-03	●	●	●	FR-02	●	×	×
NFR-04	○	○	○	FR-02-01	●	×	×
NFR-05	●	●	●	FR-03	×	●	×
NFR-06	●	●	●	FR-04	●	●	×
NFR-07	×	●	●	FR-04-01	●	●	●
NFR-08	●	●	●	FR-04-02	●	×	×
NFR-09	●	●	●	FR-04-03	×	×	●
NFR-10	●	●	○	FR-05	●	●	×
				FR-05-01	●	●	×
				FR-05-02	×	●	●

It can be seen that generally all requirements are satisfyingly fulfilled, however, some common challenges can be drawn from the mappings of the different demonstrations, especially concerning difficulty associated with NFR-04 which deals with reliability.

In this last implementation some reliability mechanisms can be integrated into the different communication actors (i.e. repeating failed prediction/model update requests in case of failure until a time-out), meaning the MAS behaviours, the Flask application serving the models for real-time predictive analytics and the KM application. However, this doesn't necessarily ensure the service is reliably available, as much as it increases the likelihood of it being robust in case of sporadic temporary communication errors for instance.

As a counterpoint, the Kafka based implementation for the data exchange part of the system can easily fulfil this requirement with its embedded support for reliability mechanisms such as "at least once"and "exactly once"modalities for ensuring message delivery. Naturally, this comes at the expense of the additional complexity brought by the need to deploy and manage Zookeeper and Kafka clusters, thus a trade-off point should be decided upon based on the requirements of the use case at hand.

Chapter 7

Discussion and Conclusions

After this journey through the design, implementation and demonstration of the IDARTS framework for the development of PMS, the last chapter of this document aims to provide a venue for the discussion of first and foremost the contributions achieved throughout this process, then of the validation of each of the hypotheses put forward in Chapter 2 to answer the research questions, and finally of existing limitations, opportunities and the future outlook regarding this research effort.

Ultimately, this dissertation aims to support the development and promote the adoption of PMS through the design of a generic reference framework to serve as a guideline for future implementation efforts in the field. Such systems can assist manufacturers in making the most out of the increasingly large volumes of data being generated every day in modern manufacturing environments.

To this end, an in-depth literature review conducted through a combination of machine learning techniques and empirical knowledge was presented, providing the means to unearth current emergent trends regarding the topic of predictive manufacturing in the context of smart factory environments, as well as current gaps and opportunities for further research (**Contribution 1**).

Afterwards, following a consensus-based design science research methodology, sets of goals, as well as functional and non-functional requirements were derived in order to guide the design of PMS, providing the main orientation, system constraints and the functionalities required to see them through (**Contribution 2**). The specification of this requirements followed a set of good practices to ensure that quality characteristics of the requirements were taken into account, particularly regarding their traceability concerning the aforementioned system goals.

With the requirement specification concluded, the IDARTS framework was designed and specified in Chapter 4, proposing what is, to the extent of the author's knowledge based on the literature review, a previously non-existent common guideline for the development of PMS (**Contribution 3**). Furthermore, to assist in the comprehension and realization of the framework, a pilot implementation is also described for each of the framework's components (**Contribution 4**), which was then mapped to the previously established requirements.

Lastly, three different scenarios are presented for the demonstration of the framework at work, comprising a total of five test cases (**Contribution 5**), one serving as a proof of concept at a laboratory level regarding the implementation presented in Chapter 5, with the remaining four pertaining to real-world applications of varied implementations of the IDARTS framework.

7.1 Confirmation of the Hypotheses

The first research question formulated at the beginning of this research process was formalized as follows:

RQ1: *In which way can the core components and principles of a Predictive Manufacturing System be identified, in the context of enabling it to provide a business advantage to manufacturers, while coping with the current market requirements of flexibility and agility?*

Consequently, in an effort to provide verifiable potential answer to this research challenge the following hypothesis was formulated:

H1: *If a baseline is established based on the current PMS literature, followed by a refined survey of current applications of PMS in the context of smart factory environments, sufficient information will be acquired to thoroughly identify common and critical requirements and components that should be encompassed in modern PMS.*

Concerning this first hypothesis, it can be concluded from the results of Section 4.1.2 that through the thorough literature review and the identification of relevant requirement sources for the artefact-based elicitation, it was possible to identify the key requirements concerning the constraints to be imposed on a PMS (set of NFRs) and the functional components necessary to achieve the predictive goals of the system (set of FRs).

From this first point the second research question and respective hypothesis were formulated, focusing on the formalization and realization of the framework:

RQ2: *How can we define a generic framework to guide the full realization of an intelligent, proactive and connected PMS solution for smart factory environments?*

H2: *If the common ground between existing narrow approaches is studied, it will be possible* *to formalize a generic PMS framework based on the combination of recent advancements in regards to Cyber-Physical Production Systems, data analytics and data management. From this framework implementations of PMS can be generated to be employed in varied application fields.*

In relation to this second hypothesis, from the design process described in Chapter 4 it can be verified that it was possible to produce an innovative artefact in the form of the full specification of the generic IDARTS framework, encompassing as hypothesised the combination of the elements of a CPPS, data analytics and knowledge management. Moreover, in Chapter 6 the application of different IDARTS implementations were demonstrated in five heterogeneous use cases, spanning a varied array of environments and showcasing the generic nature of the framework's design.

7.2 Peer-Reviewed Scientific Contributions for Knowledge Transfer

Several scientific publications in international conferences and journals were generated from the work comprised in this thesis and its related research activities. A summary of these results can be seen in Table 7.1.

Table 7.1: Scientific Publications

Conference Papers	Journal Articles	Book Chapters
6	3	2

List of Conference Publications:

All the conference publications listed below are from international, IEEE sponsored peer-review conferences indexed mostly to the Web of Science or Scopus.

- Rocha, A. D., Peres, R. S., Flores, L., & Barata, J. (2015, December). A multiagent based knowledge extraction framework to support plug and produce capabilities in manufacturing monitoring systems. In 2015 10th International Symposium on Mechatronics and its Applications (ISMA) (pp. 1-5). IEEE. (A. D. Rocha et al., 2016)

- Peres, R. S., Parreira-Rocha, M., Rocha, A. D., Barbosa, J., Leitão, P., & Barata, J. (2016, October). Selection of a data exchange format for industry 4.0 manufacturing systems. In Iecon 2016-42nd annual conference of the ieee industrial electronics society (pp. 5723-5728). IEEE. (R. S. Peres et al., 2016)

- Peres, R. S., Rocha, A. D., Coelho, A., & Oliveira, J. B. (2016, October). A highly flexible, distributed data analysis framework for industry 4.0 manufacturing systems. In International Workshop on Service Orientation in Holonic and Multi-Agent Manufacturing (pp. 373-381). Springer, Cham. (R. S. Peres, Rocha, et al., 2017)

- Peres, R. S., Rocha, A. D., & Barata, J. (2017, May). Dynamic Simulation for MAS-based Data Acquisition and Pre-processing in Manufacturing using V-REP. In Doctoral Conference on Computing, Electrical and Industrial Systems (pp. 125-134). Springer, Cham. (R. S. Peres, Rocha, & Barata, 2017)
- Peres, R., Rocha, A. D., Matos, J. P., & Barata, J. (2018, July). GO0DMAN Data Model-Interoperability in Multistage Zero Defect Manufacturing. In 2018 IEEE 16th International Conference on Industrial Informatics (INDIN) (pp. 815-821). IEEE. (R. Peres et al., 2018)
- Rocha, A. D., Peres, R. S., Barata, J., Barbosa, J., & Leitão, P. (2018, September). Improvement of Multistage Quality Control through the Integration of Decision Modeling and Cyber-Physical Production Systems. In 2018 International Conference on Intelligent Systems (IS) (pp. 479-484). IEEE. (A. D. Rocha, Peres, Barata, Barbosa, & Leitão, 2018)

List of Journal Publications:

The articles listed below were published in Q1 journals according to the Scimago Journal Ranking, with the exception of the second item in the list which was published on *Procedia Manufacturing* (Q2).

- Peres, R. S., Rocha, A., Leitão, P., Barata, J. (2018). IDARTS: Towards intelligent data analysis and real-time supervision for industry 4.0. *Computers in Industry*, vol. 101 pp. 138-146 (R. S. Peres, Rocha, Leitao, & Barata, 2018);
- Angione, G., Barbosa, J., Gosewehr, F., Leitão, P., Massa, D, Matos, J., Peres, R. S., Wermann, J. (2017). Integration and Deployment of a Distributed and Pluggable Industrial Architecture for the PERFoRM project. *Procedia Manufacturing*, (June), 896-904. (Angione et al., 2017).
- Peres, R. S., Barata, J., Leitao, P., & Garcia, G. (2019). Multistage Quality Control Using Machine Learning in the Automotive Industry. *IEEE Access*, vol. 7 (R. S. Peres, Barata, et al., 2019).

List of Book Chapter Publications:

- Architectural Elements: PERFoRM Data Model, Chapter 4 in *Digitalized and Harmonized Industrial Production Systems: The PERFoRM Approach* (R. S. Peres, Rocha, Barata, & Colombo, 2019);

- Use Case: Electric Vehicles, Chapter 12 in *Digitalized and Harmonized Industrial Production Systems: The PERFoRM Approach* (Introzzi et al., 2019).

7.3 Limitations, Opportunities and Future Outlook

One possible barrier to the success of such a predictive solution in the long term is the possibility of drastic changes in the underlying distributions of the data being fed to the predictive models. An example of this could be regarding the automotive use case in Section 6.3 referring to the dimensional characteristics of the cars. This can happen for instance due to a change in the materials' suppliers or the replacement of parts in the stations before the first stage considered in this study. This is typically known as *Concept Drift*, referring to the change in relationships between the input and output data of the underlying problem over time (Žliobaitė, Pechenizkiy, & Gama, 2016). This phenomenon can invalidate the deployed models and cause them to perform poorly, having been the centre of considerable research efforts over the last few years.

A possible solution in the occurrence of this case during production would be through online monitoring and/or training of the models using for instance an architecture similar to the one showcased in (R. S. Peres et al., 2018) leveraging the core functionalities of the IDARTS framework, as illustrated in Figure 6.12. However, it can represent a pitfall for implementations which do not consider this before-hand, thus also making it a very important point for more intuitive improvement in future research.

On a different note, an important aspect that is outside the scope of this thesis is that of security. In the current information age cyber-security is taking an increasingly larger role in modern Industry 4.0 systems, with considerable effort being put into the research and development of security mechanisms to ensure the privacy and integrity of manufacturing data. While this aspect is partially embedded in some of the technologies used for the implementation proposed in Chapter 5, with the advent and increasing adoption of technologies such as blockchain further research should be conducted in order to ensure this crucial aspect within the context of PMS.

Regarding the inference mechanisms, while for the pilot implementation the simple inference regarding states and timespans serves the purpose of providing a guideline for future implementations, implementing it in Java from scratch makes it fairly difficult to maintain and update in runtime as the system becomes more complex. As such, exploring possibilities with embedded rule engines such as CLIPS, Drools and Jess can be an interesting effort to improve this part of the PMS.

As a closing note for future research building on the results from this dissertation, particularly those with a bigger emphasis on applied research, it will be extremely valuable to revisit the RE process described in Chapter 4 and go over new iterations of the proposed methodology involving industrial stakeholders in order to further refine and improve the requirements for the design and implementation of successful PMS.

Bibliography

Angione, G., Barbosa, J., Gosewehr, F., Leitão, P., Massa, D., Matos, J., ... Wermann, J. (2017). Integration and Deployment of a Distributed and Pluggable Industrial Architecture for the PERFoRM Project. *Procedia Manufacturing*, *11*(June), 896–904. doi:10.1016/j.promfg.2017.07.193

Babiceanu, R. F., & Seker, R. (2016). Big Data and virtualization for manufacturing cyber-physical systems: A survey of the current status and future outlook. *Computers in Industry*, *81*, 128–137. doi:10.1016/j.compind.2016.02.004

Bastian, M., Heymann, S., & Jacomy, M. (2009). Gephi: An Open Source Software for Exploring and Manipulating Networks. *Third International AAAI Conference on Weblogs and Social Media*, 361–362. doi:10.1136/qshc.2004.010033

Bellifemine, F., Caire, G., & Greenwood, D. (2007). *Developing Multi-Agent Systems with JADE*. John Wiley & Sons, Ltd.

Bellifemine, F., Poggi, A., & Rimassa, G. (1999). JADE–A FIPA-compliant agent framework. *Proceedings of PAAM*, 97–108. doi:10.1145/375735.376120

Blondel, V. D., Guillaume, J. L., Lambiotte, R., & Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, *2008*(10), 1–12. doi:10.1088/1742-5468/2008/10/P10008. arXiv: 0803.0476

Bouma, G. (2009). Normalized (Pointwise) Mutual Information in Collocation Extraction. *Proceedings of German Society for Computational Linguistics (GSCL 2009)*, 31–40.

Breiman, L. (2001). Random forests. *Machine learning*, *45*(1), 5–32.

Brettel, M., Friederichsen, N., Keller, M., & Rosenberg, M. (2014). How Virtualization, Decentralization and Network Building Change the Manufacturing Landscape: An Industry 4.0 Perspective. *International Journal of Information and Communication Engineering*, *8*(1), 37–44. doi:10.1016/j.procir.2015.02.213. arXiv: arXiv:1011.1669v3

Bumblauskas, D., Gemmill, D., Igou, A., & Anzengruber, J. (2017). Smart Maintenance Decision Support Systems (SMDSS) based on corporate big data analytics. *Expert Systems with Applications*, *90*, 303–317. doi:10.1016/j.eswa.2017.08.025

Canito, A., Fernandes, M., & Pra, I. (2017). An Architecture for Proactive Maintenance in the Machinery Industry. In *8th international symposium on ambient intelligence (isaml 2017)* (Vol. 615, *ISAmI*). doi:10.1007/978-3-319-61118-1

Chan, A., & McNaught, K. R. (2008). Using Bayesian networks to improve fault diagnosis during manufacturing tests of mobile telephone infrastructure. *Journal of the Operational Research Society*, *59*(4), 423–430. doi:10.1057/palgrave.jors.2602388

Chang, C.-C., & Lin, C.-J. (2011). Libsvm: A library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, *2*(3), 27.

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). Smote: Synthetic minority over-sampling technique. *Journal of artificial intelligence research*, *16*, 321–357.

Chen, T., & Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (pp. 785–794). ACM.

Consortium, I. I. (2015). *Industrial internet reference architecture.*

Cupek, R., Ziebinski, A., Zonenberg, D., & Drewniak, M. (2018). Determination of the machine energy consumption profiles in the mass-customised manufacturing. *International Journal of Computer Integrated Manufacturing*, *31*(6), 537–561. doi:10.1080/0951192X.2017.1339914

Deekshatulu, B., Chandra, P. et al. (2013). Classification of heart disease using k-nearest neighbor and genetic algorithm. *Procedia Technology*, *10*, 85–94.

Deloitte. (2014). *Industry 4.0–challenges and solutions for the digital transformation and use of exponential technologies.*

Deng, C., Guo, R., Liu, C., Zhong, R. Y., & Xu, X. (2018). Data cleansing for energy-saving: a case of Cyber-Physical Machine Tools health monitoring system. *International Journal of Production Research*, *56*(1-2), 1000–1015. doi:10.1080/00207543.2017.1394596

FIPA. (2000). FIPA Contract Net Interaction Protocol Specification. Retrieved from http://www.fipa.org/specs/fipa00029/

FIPA. (2002). FIPA Request Interaction Protocol Specification. Retrieved from http://www.fipa.org/specs/fipa00026/SC00026H.pdf

Flath, C. M., & Stein, N. (2018). Towards a data science toolbox for industrial analytics applications. *Computers in Industry*, *94*, 16–25. doi:10.1016/j.compind.2017.09.003

Frei, R., Barata, J., & Onori, M. (2007). Evolvable Production Systems Context and Implications. In *Ieee international symposium on industrial electronics, 2007. isie 2007* (pp. 3233–3238). doi:10.1109/ISIE.2007.4375132

Gandomi, A., & Haider, M. (2015). Beyond the hype: Big data concepts, methods, and analytics. *International Journal of Information Management*, *35*(2), 137–144. doi:10.1016/j.ijinfomgt.2014.10.007

Gao, R., Wang, L., Teti, R., Dornfeld, D., Kumara, S., Mori, M., & Helu, M. (2015). Cloud-enabled prognosis for manufacturing. *CIRP Annals - Manufacturing Technology*, *64*(2), 749–772. doi:10.1016/j.cirp.2015.05.011. arXiv: 0208024 [gr-qc]

Ge, Z. (2018). Distributed predictive modeling framework for prediction and diagnosis of key performance index in plant-wide processes. *Journal of Process Control*, *65*, 107–117. doi:10.1016/j.jprocont.2017.08.010

Gilchrist, A. (2016). Introducing Industry 4.0. In *Industry 4.0* (pp. 195–215). Apress.

GO0D MAN Project (ID:723764). (2017). *Deliverable 2.1 - Multi-Agent Architecture Specification*. Retrieved from http://go0dman-project.eu/wp-content/uploads/2016/10/GO0D-MAN-Deliverable-2.1.pdf

GO0D MAN Project (ID:723764). (2018). *Deliverable 2.3 - Multi-Agent Infrastructure*. Retrieved from http://go0dman-project.eu/wp-content/uploads/2018/05/GO0D-MAN-Deliverable-2.3.pdf

GO0D MAN Project (ID:723764). (2019a). *Deliverable 6.1 - VWAE Use Case Prototype*.

GO0D MAN Project (ID:723764). (2019b). *Deliverable 7.1 - Zannini Use Case Prototype*.

GO0D MAN Project (ID:723764). (2019c). *Deliverable 8.1 - Electrolux Use Case Prototype*.

Golkarnarenji, G., Naebe, M., Badii, K., Milani, A. S., Jazar, R. N., & Khayyam, H. (2018). Support vector regression modelling and optimization of energy consumption in carbon fiber production line. *Computers and Chemical Engineering*, *109*, 276–288. doi:10.1016/j.compchemeng.2017.11.020

Guazzelli, A., Zeller, M., Lin, W., & Williams, G. (2009). PMML: An open standard for sharing models. *The R Journal*, *1*(1), 60–65. doi:doi=10.1.1.538.9778. arXiv: 1011.1669

Guo, W., & Banerjee, A. G. (2017). Identification of key features using topological data analysis for accurate prediction of manufacturing system outputs. *Journal of Manufacturing Systems*, *43*, 225–234. doi:10.1016/j.jmsy.2017.02.015

H.Gomaa, W., & A. Fahmy, A. (2013). A Survey of Text Similarity Approaches. *International Journal of Computer Applications*, *68*(13), 13–18. doi:10.5120/11638-7118. arXiv: arXiv:1011.1669v3

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). The weka data mining software: An update. *ACM SIGKDD explorations newsletter*, *11*(1), 10–18.

Hazen, B. T., Boone, C. A., Ezell, J. D., & Jones-Farmer, L. A. (2014). Data quality for data science, predictive analytics, and big data in supply chain management: An introduction to the problem and suggestions for research and applications. *International Journal of Production Economics*, *154*, 72–80.

He, Y., Gu, C., Chen, Z., & Han, X. (2017). Integrated predictive maintenance strategy for manufacturing systems by combining quality control and mission reliability analysis. *International Journal of Production Research*, *55*(19), 5841–5862. doi:10.1080/00207543.2017.1346843

Hebert, J. (2016). Predicting rare failure events using classification trees on large scale manufacturing data with complex interactions. In *2016 ieee international conference on big data (big data)* (pp. 2024–2028). IEEE.

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. *MIS quarterly*, *28*(1), 75–105.

Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *science*, *313*(5786), 504–507.

Industrial Internet Consortium. (2019). *The Industrial Internet of Things Volume G1: Reference Architecture* (tech. rep. No. June). Retrieved from https://www.iiconsortium.org/pdf/IIRA-v1.9.pdf

International Electrotechnical Commission. (n.d.-a). IEC 61512 Batch Control Parts 1-4.

International Electrotechnical Commission. (n.d.-b). IEC 62264 Enterprise-control system integration Parts 1-5.

Introzzi, R., Perlo, P., Grosso, M., Pozzato, S., Biasiotto, M., Penserini, D., ... Gepp, M. (2019). Use case: Electric vehicles. In A. W. Colombo, M. Gepp, J. Barata, P. Leitao, J. Barbosa, & J. Wermann (Eds.), *Digitalized and harmonized industrial production systems: The perform approach* (Chap. 12). CRC Press.

ISO/IEC/IEEE. (2017). Iso/iec/ieee international standard - systems and software engineering–vocabulary. *ISO/IEC/IEEE 24765:2017(E)*, 1–541. doi:10.1109/IEEESTD.2017.8016712

Jardine, A. K., Lin, D., & Banjevic, D. (2006). A review on machinery diagnostics and prognostics implementing condition-based maintenance. *Mechanical systems and signal processing*, *20*(7), 1483–1510.

Jia, F., Lei, Y., Lin, J., Zhou, X., & Lu, N. (2016). Deep neural networks: A promising tool for fault characteristic mining and intelligent diagnosis of rotating machinery with massive data. *Mechanical Systems and Signal Processing*, *72*, 303–315.

Jimenez, J. F., Bekrar, A., Trentesaux, D., & Leitão, P. (2016). A switching mechanism framework for optimal coupling of predictive scheduling and reactive control in manufacturing hybrid control architectures. *International Journal of Production Research*, *54*(23), 7027–7042. doi:10.1080/00207543.2016.1177237

Kagermann, H., Helbig, J., Hellinger, A., & Wahlster, W. (2013). *Recommendations for Implementing the strategic initiative INDUSTRIE 4.0: securing the future of German manufacturing industry; final report of the Industrie 4.0 working group.*

Kang, P., Lee, H.-j., Cho, S., Kim, D., Park, J., Park, C.-K., & Doh, S. (2009). A virtual metrology system for semiconductor manufacturing. *Expert Systems with Applications*, *36*(10), 12554–12561.

Kanjanatarakul, O., Denœux, T., & Sriboonchitta, S. (2016). Prediction of future observations using belief functions: A likelihood-based approach. *International Journal of Approximate Reasoning*, *72*, 71–94. doi:10.1016/j.ijar.2015.12.004

Kant, G., & Sangwan, K. S. (2014). Prediction and optimization of machining parameters for minimizing power consumption and surface roughness in machining. *Journal of Cleaner Production*, *83*, 151–164. doi:10.1016/j.jclepro.2014.07.073

Khaleghi, B., Khamis, A., Karray, F. O., & Razavi, S. N. (2013). Multisensor data fusion: A review of the state-of-the-art. *Information fusion*, *14*(1), 28–44.

Khan, A. A., Moyne, J. R., & Tilbury, D. M. (2007). An approach for factory-wide control utilizing virtual metrology. *IEEE Transactions on semiconductor Manufacturing*, *20*(4), 364–375.

Kolar, M., Vyroubal, J., & Smolik, J. (2016). Analytical approach to establishment of predictive models of power consumption of machine tools' auxiliary units. *Journal of Cleaner Production*, *137*, 361–369. doi:10.1016/j.jclepro.2016.07.092

Lao, L., Ellis, M., Durand, H., & Christofides, P. D. (2015). Real-Time Preventive Sensor Maintenance Using Robust Moving Horizon Estimation and Economic Model Predictive Control. *AIChE Journal*, *61*(504). doi:10.1002/aic. arXiv: 0201037v1 [arXiv:physics]

Lechevalier, D., Narayanan, A., & Rachuri, S. (2015). Towards a domain-specific framework for predictive analytics in manufacturing. *Proceedings - 2014 IEEE International Conference on Big Data, IEEE Big Data 2014*, (February 2015), 987–995. doi:10.1109/BigData.2014.7004332

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *nature*, *521*(7553), 436.

Lee, J., Bagheri, B., & Kao, H. A. (2015). A Cyber-Physical Systems architecture for Industry 4.0-based manufacturing systems. *Manufacturing Letters*, *3*, 18–23. doi:10.1016/j.mfglet.2014.12.001. arXiv: 1503.07717

Lee, J., Bagheri, B., & Kao, H.-A. (2014). Recent Advances and Trends of Cyber-Physical Systems and Big Data Analytics in Industrial Informatics. *Int. Conference on Industrial Informatics (INDIN) 2014*, (November 2015). doi:10.13140/2.1.1464.1920

Lee, J., Davari, H., Singh, J., & Pandhare, V. (2018). Industrial artificial intelligence for industry 4.0-based manufacturing systems. *Manufacturing letters*, *18*, 20–23.

Lee, J., Kao, H. A., & Yang, S. (2014). Service innovation and smart analytics for Industry 4.0 and big data environment. *Procedia CIRP*, *16*, 3–8. doi:10.1016/j.procir.2014.02.001

Lee, J., Lapira, E., Bagheri, B., & Kao, H.-a. (2013). Recent advances and trends in predictive manufacturing systems in big data environment. *Manufacturing Letters*, *1*(1), 38–41. doi:10.1016/j.mfglet.2013.09.005

Lee, J., Wu, F., Zhao, W., Ghaffari, M., Liao, L., & Siegel, D. (2014). Prognostics and health management design for rotary machinery systems - Reviews, methodology and applications. *Mechanical Systems and Signal Processing*, *42*(1-2), 314–334. doi:10.1016/j.ymssp.2013.06.004

Leitão, P., Colombo, A. W., & Karnouskos, S. (2016). Industrial automation based on cyber-physical systems technologies: Prototype implementations and challenges. *Computers in Industry*, *81*, 11–25.

Leitao, P., Karnouskos, S., Ribeiro, L., Lee, J., Strasser, T., & Colombo, A. W. (2016). Smart agents in industrial cyber–physical systems. *Proceedings of the IEEE*, *104*(5), 1086–1101. Retrieved from http://ieeexplore.ieee.org/abstract/document/7437398/

Leitão, P., & Restivo, F. (2006). ADACOR: A holonic architecture for agile and adaptive manufacturing control. *Computers in Industry*, *57*, 121–130. doi:10.1016/j.compind.2005.05.005

Li, J., & Shi, J. (2007). Knowledge discovery from observational data for process control using causal Bayesian networks. *IIE Transactions*, *39*(6), 681–690. doi:10.1080/07408170600899532

Li, P., Jia, X., Feng, J., Davari, H., Qiao, G., Hwang, Y., & Lee, J. (2018). Prognosability study of ball screw degradation using systematic methodology. *Mechanical Systems and Signal Processing*, *109*, 45–57. doi:10.1016/j.ymssp.2018.02.046

Liu, Q., Dong, M., & Chen, F. F. (2018). Single-machine-based joint optimization of predictive maintenance planning and production scheduling. *Robotics and Computer-Integrated Manufacturing*, *51*(November 2017), 238–247. doi:10.1016/j.rcim.2018.01.002

Loyer, J. L., Henriques, E., Fontul, M., & Wiseall, S. (2016). Comparison of Machine Learning methods applied to the estimation of manufacturing cost of jet engine components. *International Journal of Production Economics*, *178*, 109–119. doi:10.1016/j.ijpe.2016.05.006

Lu, B., & Zhou, X. (2017). Opportunistic preventive maintenance scheduling for serial-parallel multistage manufacturing systems with multiple streams of deterioration. *Reliability Engineering and System Safety*, *168*(May), 116–127. doi:10.1016/j.ress.2017.05.017

Ma, J., Kwak, M., & Kim, H. M. (2014). Demand trend mining for predictive life cycle design. *Journal of Cleaner Production*, *68*, 189–199. doi:10.1016/j.jclepro.2014.01.026

Mobley, R. K. (2002). *An Introduction to Predictive Maintenance* (2nd Editio). Elsevier.

Mullarkey, M. T., Hevner, A. R., Gill, T. G., & Dutta, K. (2019). Citizen data scientist: A design science research method for the conduct of data science projects. In *International conference on design science research in information systems and technology* (pp. 191–205). Springer.

Nagorny, K., Colombo, A. W., & Schmidtmann, U. (2012). A service- and multi-agent-oriented manufacturing automation architecture: An IEC 62264 level 2 compliant implementation. *Computers in Industry*, *63*(OCTOBER), 813–823. doi:10.1016/j.compind.2012.08.003

Onori, M., Lohse, N., Barata, J., & Hanisch, C. (2013). The IDEAS Project: Plug & Produce at Shop-Floor Level. *Assembly Automation*, *32*(2), 124–134.

Pandiyan, V., Caesarendra, W., Tjahjowidodo, T., & Tan, H. H. (2018). In-process tool condition monitoring in compliant abrasive belt grinding process using support vector machine and genetic algorithm. *Journal of Manufacturing Processes*, *31*, 199–213. doi:10.1016/j.jmapro.2017.11.014

Patcha, A., & Park, J. M. (2007). An overview of anomaly detection techniques: Existing solutions and latest technological trends. *Computer Networks*, *51*(12), 3448–3470. doi:10.1016/j.comnet.2007.02.001

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... Duchesnay, É. (2012). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830. doi:10.1007/s13398-014-0173-7.2. arXiv: 1201.0490

Peffers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A design science research methodology for information systems research. *Journal of management information systems*, *24*(3), 45–77.

Peng, H., & Van Houtum, G. J. (2016). Joint optimization of condition-based maintenance and production lot-sizing. *European Journal of Operational Research*, *253*(1), 94–107. doi:10.1016/j.ejor.2016.02.027

Peng, Y., Dong, M., & Zuo, M. J. (2010). Current status of machine prognostics in condition-based maintenance: A review. *The International Journal of Advanced Manufacturing Technology*, *50*(1-4), 297–313.

Penya, Y. K., Bringas, P. G., & Zabala, A. (2008). Advanced fault prediction in high-precision foundry production. *IEEE International Conference on Industrial Informatics (INDIN)*, 1672–1677. doi:10.1109/INDIN.2008.4618372

Peres, R. S., Barata, J., Leitao, P., & Garcia, G. (2019). Multistage quality control using machine learning in the automotive industry. *IEEE Access*, *7*. doi:10.1109/ACCESS.2019.2923405

Peres, R. S., Parreira-Rocha, M., Rocha, A. D., Barbosa, J., Leitao, P., & Barata, J. (2016). Selection of a data exchange format for industry 4.0 manufacturing systems. *IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society*, 5723–5728. doi:10.1109/IECON.2016.7793750

Peres, R. S., Rocha, A. D., & Barata, J. (2017). Dynamic Simulation for MAS-Based Data Acquisition and Pre-processing in Manufacturing Using V-REP. In *Technological innovation for smart systems. doceis 2017* (March, pp. 125–134). doi:10.1007/978-3-319-56077-9

Peres, R. S., Rocha, A. D., Barata, J., & Colombo, A. W. (2019). Architectural elements: Perform data model. In A. W. Colombo, M. Gepp, J. Barata, P. Leitao, J. Barbosa, & J. Wermann (Eds.), *Digitalized and harmonized industrial production systems: The perform approach* (Chap. 4). CRC Press.

Peres, R. S., Rocha, A. D., Leitao, P., & Barata, J. (2018). IDARTS–Towards intelligent data analysis and real-time supervision for industry 4.0. *Computers in Industry*, *101*, 138–146.

Peres, R. S., Rocha, A., Coelho, A., & Barata, J. (2017). A Highly Flexible , Distributed Data Analysis Framework for Industry 4 . 0 Manufacturing Systems. In T. Borangiu, D. Trentesaux, A. Thomas, P. Leitão, & J. Barata (Eds.), *Service orientation in* *holonic and multi-agent manufacturing. sohoma 2017* (March, pp. 373–381). doi:10.1007/978-3-319-51100-9

Peres, R., Rocha, A. D., Matos, J. P., & Barata, J. (2018). Go0dman data model-interoperability in multistage zero defect manufacturing. In *2018 ieee 16th international conference on industrial informatics (indin)* (pp. 815–821). IEEE.

Plattform Industrie 4.0. (2015). *Status Report: Reference Architecture Model Industrie 4.0 (RAMI4.0)* (tech. rep. No. July). VDI/VDE. Retrieved from https://www.zvei.org/fileadmin/user_upload/Presse_und_Medien/Publikationen/2016/januar/GMA_Status_Report___Reference_Archtitecture_Model_Industrie_4.0___RAMI_4.0_/GMA-Status-Report-RAMI-40-July-2015.pdf

Qi, Q., & Tao, F. (2018). Digital Twin and Big Data Towards Smart Manufacturing and Industry 4.0: 360 Degree Comparison. *IEEE Access*, *6*, 3585–3593. doi:10.1109/ACCESS.2018.2793265

Rennie, J. D., Shih, L., Teevan, J., & Karger, D. R. (2003). Tackling the poor assumptions of naive bayes text classifiers. In *Proceedings of the 20th international conference on machine learning (icml-03)* (pp. 616–623).

Ribeiro, L. (2017). Cyber-physical production systems' design challenges. In *2017 ieee 26th international symposium on industrial electronics (isie)* (pp. 1189–1194). doi:10.1109/ISIE.2017.8001414

Rocha, A. (2018). Increase the adoption of agent-based cyber-physical production systems through the design of minimally invasive solutions.

Rocha, A. D., Barata, D., & Orio, G. D. (2015). Technological Innovation for Cloud-Based Engineering Systems. IFIP Advances in Information and Communication Technology, *450*, 101–110. doi:10.1007/978-3-319-16766-4

Rocha, A. D., Barata, D., Orio, G. D., Santos, T., & Barata, J. (2015). PRIME as a Generic Agent Based Framework to Support Pluggability and Reconfigurability Using Different Technologies. In *Technological innovation for cloud-based engineering systems* (Chap. Technologi, pp. 101–110).

Rocha, A. D., Peres, R. S., Barata, J., Barbosa, J., & Leitão, P. (2018). Improvement of multistage quality control through the integration of decision modeling and cyber-physical production systems. In *2018 international conference on intelligent systems (is)* (pp. 479–484). IEEE.

Rocha, A. D., Peres, R. S., Flores, L., & Barata, J. (2016). A multiagent based knowledge extraction framework to support plug and produce capabilities in manufacturing monitoring systems. *ISMA 2015 - 10th International Symposium on Mechatronics and its Applications*. doi:10.1109/ISMA.2015.7373494

Rupp, C., & die SOPHISTen. (2014). *Requirements-engineering und-management: Aus der praxis von klassisch bis agil*. doi:10.3139/9783446443136

Saha, B., Goebel, K., & Christophersen, J. (2009). Comparison of prognostic algorithms for estimating remaining useful life of batteries. *Transactions of the Institute of Measurement and Control*, *31*(3-4), 293–308.

Sahebjamnia, N., Tavakkoli-Moghaddam, R., & Ghorbani, N. (2016). Designing a fuzzy Q-learning multi-agent quality control system for a continuous chemical production line - A case study. *Computers and Industrial Engineering*, *93*, 215–226. doi:10.1016/j.cie.2016.01.004

Santos, M. Y., Oliveira e Sá, J., Andrade, C., Vale Lima, F., Costa, E., Costa, C., ... Galvão, J. (2017). A Big Data system supporting Bosch Braga Industry 4.0 strategy. *International Journal of Information Management*, *37*(6), 750–760. doi:10.1016/j.ijinfomgt.2017.07.012

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural networks*, *61*, 85–117.

SCImago, G. (2007). SJR-SCImago journal & country rank. *Consejo Superior de Investigaciones Cientificas (CSIC), University of Granada, Extremadura, Carlos III (Madrid) and Alcalá de Henares*.

Shin, S.-J., Woo, J., & Rachuri, S. (2014). Predictive Analytics Model for Power Consumption in Manufacturing. *Procedia CIRP*, *15*, 153–158. doi:10.1016/j.procir.2014.06.036

Shrouf, F., Ordieres, J., & Miragliotta, G. (2014). Smart factories in Industry 4.0: A review of the concept and of energy management approached in production based on the Internet of Things paradigm. In *2014 ieee international conference on industrial engineering and engineering management* (pp. 697–701). doi:10.1109/IEEM.2014.7058728

Si, X.-S., Wang, W., Hu, C.-H., Chen, M.-Y., & Zhou, D.-H. (2013). A wiener-process-based degradation model with a recursive filter algorithm for remaining useful life estimation. *Mechanical Systems and Signal Processing*, *35*(1-2), 219–237.

Siafara, L. C., Kholerdi, H. A., Bratukhin, A., Taherinejad, N., & Wendt, A. (2017). SAMBA : A Self-Aware Health Monitoring Architecture for Distributed Industrial Systems. In *Industrial electronics society, iecon 2017-43rd annual conference of the ieee* (pp. 3512–3517). IEEE.

Sick, B. (2002). On-line and indirect tool wear monitoring in turning with artificial neural networks: A review of more than a decade of research. *Mechanical systems and signal processing*, *16*(4), 487–546.

Sommerville, I. (2011). Software engineering 9th edition. *ISBN-10*, *137035152*.

Stock, T., & Seliger, G. (2016). Opportunities of Sustainable Manufacturing in Industry 4.0. *Procedia CIRP*. 13th Global Conference on Sustainable Manufacturing – Decoupling Growth from Resource Use, *40*, 536–541. doi:10.1016/j.procir.2016.01.129

Stonebraker, M., Çetintemel, U., & Zdonik, S. (2005). The 8 requirements of real-time stream processing. *ACM SIGMOD Record*, *34*(4), 42–47. doi:10.1145/1107499.1107504

Susto, G. A., Pampuri, S., Schirru, A., Beghi, A., & De Nicolao, G. (2015). Multi-step virtual metrology for semiconductor manufacturing: A multilevel and regularization methods-based approach. *Computers & Operations Research*, *53*, 328–337.

Susto, G. A., Schirru, A., Pampuri, S., McLoone, S., & Beghi, A. (2015). Machine learning for predictive maintenance: A multiple classifier approach. *IEEE Transactions on Industrial Informatics*, *11*(3), 812–820. doi:10.1109/TII.2014.2349359

Tao, F., & Zhang, M. (2017). Digital Twin Shop-Floor: A New Shop-Floor Paradigm Towards Smart Manufacturing. *IEEE Access*, *5*, 20418–20427. doi:10.1109/ACCESS.2017.2756069

Terrissa, L. S., Meraghni, S., Bouzidi, Z., & Zerhouni, N. (2016). A New Approach of PHM as a Service in Cloud Computing. *2016 4th IEEE International Colloquium on Information Science and Technology (CiSt). Proceedings*, 610–14. doi:10.1109/CIST.2016.7804958

Tuptuk, N., & Hailes, S. (2018). Security of smart manufacturing systems. *Journal of Manufacturing Systems*, *47*, 93–106. doi:https://doi.org/10.1016/j.jmsy.2018.04.007

Upasani, K., Bakshi, M., Pandhare, V., & Lad, B. K. (2017). Distributed maintenance planning in manufacturing industries. *Computers and Industrial Engineering*, *108*, 1–14. doi:10.1016/j.cie.2017.03.027

Waller, M. A., & Fawcett, S. E. (2013). Data science, predictive analytics, and big data: A revolution that will transform supply chain design and management. *Journal of Business Logistics*, *34*(2), 77–84.

Wan, J., Tang, S., Li, D., Wang, S., Liu, C., Abbas, H., & Vasilakos, A. V. (2017). A Manufacturing Big Data Solution for Active Preventive Maintenance. *IEEE Transactions on Industrial Informatics*, *13*(4), 251–295. doi:10.1201/b15906-13

Wang, G., Gunasekaran, A., Ngai, E. W., & Papadopoulos, T. (2016). Big data analytics in logistics and supply chain management: Certain investigations for research and applications. *International Journal of Production Economics*, *176*, 98–110. doi:10.1016/j.ijpe.2016.03.014

Wang, J., Sun, Y., Zhang, W., Thomas, I., Duan, S., & Shi, Y. (2016). Large Scale Online Multi-Task Learning and Decision-Making for Flexible Manufacturing. *IEEE Transactions on Industrial Informatics*, *PP*(99), 1. doi:10.1109/TII.2016.2549919

Wang, J., Zhang, L., Duan, L., & Gao, R. X. (2017). A new paradigm of cloud-based predictive maintenance for intelligent manufacturing. *Journal of Intelligent Manufacturing*, *28*(5), 1125–1137. doi:10.1007/s10845-015-1066-0

Wang, K. (2016). Intelligent Predictive Maintenance (IPdM) system – Industry 4.0 scenario. *WIT Transactions on Engineering Sciences*, *113*, 259–268. doi:10.2495/IWAMA150301

Wang, L., Törngren, M., & Onori, M. (2015). Current status and advancement of cyber-physical systems in manufacturing. *Journal of Manufacturing Systems*, *37*, 517–527.

Wang, P., & Gao, R. X. (2015). Adaptive resampling-based particle filtering for tool life prediction. *Journal of Manufacturing Systems*, *37*, 528–534.

Wang, S., Wan, J., Zhang, D., Li, D., & Zhang, C. (2016). Towards smart factory for industry 4.0: A self-organized multi-agent system with big data based feedback and coordination. *Computer Networks*, *101*, 158–168. doi:10.1016/j.comnet.2015.12.017

Wang, X., Wang, H., & Qi, C. (2016). Multi-agent reinforcement learning based maintenance policy for a resource constrained flow line system. *Journal of Intelligent Manufacturing*, *27*(2), 325–333. doi:10.1007/s10845-013-0864-5

Willett, P. (2006). The Porter stemming algorithm: then and now. *Program*, *40*(3), 219–223.

Wu, D., Jennings, C., Terpenny, J., Gao, R. X., & Kumara, S. (2017). A Comparative Study on Machine Learning Algorithms for Smart Manufacturing: Tool Wear Prediction Using Random Forests. *Journal of Manufacturing Science and Engineering*, *139*(7), 071018. doi:10.1115/1.4036350

Wu, D., Jennings, C., Terpenny, J., Kumara, S., & Gao, R. (2017). Cloud-Based Parallel Machine Learning for Prognostics and Health Management: A Tool Wear Prediction Case Study. *Journal of Manufacturing Science and Engineering*, (March). doi:10.1115/1.4038002

Wu, D., Liu, S., Zhang, L., Terpenny, J., Gao, R. X., Kurfess, T., & Guzzo, J. A. (2017). A fog computing-based framework for process monitoring and prognosis in cyber-manufacturing. *Journal of Manufacturing Systems*, *43*, 25–34. doi:10.1016/j.jmsy.2017.02.011

Wu, D., Rosen, D. W., Wang, L., & Schaefer, D. (2015). Cloud-based design and manufacturing: A new paradigm in digital manufacturing and design innovation. *Computer-Aided Design*, *59*, 1–14.

Xia, T., Jin, X., Xi, L., Zhang, Y., & Ni, J. (2015). Operating load based real-time rolling grey forecasting for machine health prognosis in dynamic maintenance schedule. *Journal of Intelligent Manufacturing*, *26*(2), 269–280. doi:10.1007/s10845-013-0780-8

Xu, X. (2012). From cloud computing to cloud manufacturing. *Robotics and Computer-Integrated Manufacturing*, *28*(1), 75–86. doi:10.1016/j.rcim.2011.07.002

Yan, H., Wan, J., Zhang, C., Tang, S., Hua, Q., & Wang, Z. (2018). Industrial Big Data Analytics for Prediction of Remaining Useful Life Based on Deep Learning. *IEEE Access*, *PP*(99), 1. doi:10.1109ACCESS.2018.2809681

Yasin, H., Mohammad Yasin, M., & Mohammad Yasin, F. (2011). Automated Multiple Related Documents Summarization via Jaccards Coefficient. *International Journal of Computer Applications*, *12*(3), 12–15. doi:10.5120/1762-2415

Zhang, S., Dubay, R., & Charest, M. (2015). A principal component analysis model-based predictive controller for controlling part warpage in plastic injection molding. *Expert Systems with Applications*, *42*(6), 2919–2927. doi:10.1016/j.eswa.2014.11.030

Zhong, R. Y., Xu, C., Chen, C., & Huang, G. Q. (2017). Big Data Analytics for Physical Internet-based intelligent manufacturing shop floors. *International Journal of Production Research*, *55*(9), 2610–2621. doi:10.1080/00207543.2015.1086037

Žliobaitė, I., Pechenizkiy, M., & Gama, J. (2016). An overview of concept drift applications. In *Big data analysis: New algorithms for a new society* (pp. 91–114). Springer.

Zuehlke, D. (2010). SmartFactory—Towards a factory-of-things. *Annual Reviews in Control*, *34*(1), 129–138. doi:10.1016/j.arcontrol.2010.02.008

Zurita, D., Delgado, M., Carino, J. A., Ortega, J. A., & Clerc, G. (2016). Industrial Time Series Modelling by Means of the Neo-Fuzzy Neuron. *IEEE Access*, *4*, 6151–6160. doi:10.1109/ACCESS.2016.2611649