Università degli Studi di Verona

Department of Computer Science
Doctoral Program in Computer Science
Cycle XXXIV, 2021

Discovering phase and causal dependencies on manufacturing processes

Author:
Giovanni Menegozzo

Supervisor:
Paolo Fiorini

Co-Supervisor:
Diego Dall'Alba

Submitted in fulfillment of the requirements for the degree of Doctor of Philosophy in Computer Science at University of Verona

S.S.D. INF/01

March 16, 2022

Discovering phase and causal dependencies

on manufacturing processes

by

Giovanni Menegozzo

Submitted to the department of Computer Science at University of Verona
on March 16, 2022, in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy in Computer Science
S.S.D. INF/01
Cycle XXXIV/2018

Abstract

Small and Medium Enterprises (SMEs) represent 90% of businesses and more than 50% of employment worldwide. While large companies lead long-term innovation strategies with dedicated resources for new technologies, SMEs struggle to manage the increased complexity of processes due to funds shortages, low managerial skills, and lack of personnel. To achieve a sustainable fourth industrial revolution, it is crucial to develop systems that consider SMEs' conditions. Monitoring technologies provide numerous benefits for SMEs without burdening the production process in use. Monitoring systems facilitate process management, improve product quality and relief operators. In this thesis, we provide two contributions to process monitoring: the first one is an automatic system for recognizing the process phases, while the second one consists of developing a forecasting procedure for product quality. The automatic recognition system uses a supervised deep learning method to capture long and complex actions. This is made possible by stacking expanded convolutional filters on the input data. The second proposed contribution is a predictive procedure that combines the causal flow of the product with domain expert knowledge to achieve an efficient and flexible forecasting system. We introduce a neural network architecture named Separable Temporal Convolutional Network (S-TCN), which efficiently exploits causal precursors to obtain distant temporal information. The proposed methods are tested in numerical experiments and in a controlled environment that replicates manufacturing tasks. Finally, the predictive procedure has been applied to a medium-sized manufacturing company.

Thesis Advisor:
Paolo Fiorini
Title: Full Professor

Thesis Co-Advisor:
Diego Dall'Alba
Title: Doctor

Segmentazione di fasi del processo e relazioni causali per impianti manifatturieri continui

Di
Giovanni Menegozzo

Sommario

Le piccole e medie imprese (PMI) rappresentano il 90% delle imprese e più del 50% dell'occupazione nel mondo. Mentre le grandi aziende conducono strategie di innovazione a lungo termine con risorse dedicate alle nuove tecnologie, le PMI faticano nel gestire la maggiore complessità dei processi a causa della scarsità di fondi, delle basse competenze manageriali e della penuria di personale. Per raggiungere una quarta rivoluzione industriale sostenibile, è fondamentale sviluppare sistemi che considerino le condizioni delle PMI. Le tecnologie di monitoraggio forniscono numerosi benefici alle PMI senza appesantire il processo produttivo in uso. I sistemi di monitoraggio facilitano la gestione del processo, migliorano la qualità del prodotto e soccorrono gli operatori. In questa tesi, forniamo due contributi al monitoraggio dei processi: il primo consiste in un sistema automatico di riconoscimento delle fasi del processo, mentre il secondo consiste nello sviluppo di una procedura di previsione per una caratteristica qualitativa chiave del prodotto. Il sistema di riconoscimento automatico utilizza un metodo di apprendimento profondo supervisionato per catturare movimenti lunghi e complessi. Ciò è reso possibile dall'utilizzo di filtri convoluzionali sui dati di input. Il secondo contributo proposto consiste in una procedura predittiva che combina il flusso causale del prodotto con la conoscenza degli esperti del dominio per ottenere un sistema di previsione efficiente e flessibile. Introduciamo un'architettura di rete neurale chiamata Separable Temporal Convolutional Network (S-TCN), che sfrutta in modo efficiente i precursori causali per ottenere informazioni temporali distanti. I metodi proposti sono testati in esperimenti numerici e in un ambiente controllato che replica le attività di produzione. Infine, la procedura predittiva è stata applicata a un'azienda manifatturiera di medie dimensioni.

Relatore:
Paolo Fiorini
Titolo: Professore

Co-Relatore:
Diego Dall'Alba
Titolo: Dottore

"*To all those who enjoy the pursuit of causal insight*"

Elements of Causal Inference
Foundations and Learning Algorithms

by Peters, Janzing and Schölkopf

Acknowledgements

First, I would like to thank professor Paolo Fiorini for letting me explore a wide spectrum of research fields pursuing my interests. He guided me to model practical issues into theoretical problems and taught me the importance of formalization. I would like to thank professor Tania Cerquitelli and professor Emanuele Frontoni for the helpful reviews and for improving the readability of the thesis.

A special thanks for all the work done goes to Diego Dall'Alba. I cannot fully express my gratitude for the support he has given to me during the whole academic journey. He taught me how to conduct experiments, to work in a team and he provided me with countless valuable corrections. I thank my colleagues, Stefano, Eraldo and Matteo, from the ultra-processed food company. They allowed me to comprehend the differences between academia and industry.

Finally, I want to thank Tania for all the patience and encouragement she has given me during these years, my family, my colleagues and my friends. Their support and love are the roots of everything that I have achieved and that I am.

Contents

1 Research goals and challenges **23**

1.1 The importance of SMEs for the European manufacturing sector . . . 24

1.2 Risks and challenges for SMEs 26

1.3 Process management: modeling and monitoring with AI 32

1.3.1 Process monitoring in manufacturing processes 35

1.3.2 Challenges in existing solutions 37

1.4 Thesis description and organizations 39

1.4.1 Summary of the thesis 40

1.4.2 Summary of contributions 41

1.5 Conclusions 43

2 State of the art **45**

2.1 Process phase recognition 45

2.1.1 Automatic gesture recognition for Surgical Robotic Systems . 49

2.1.2 Decomposition method for industrial processes 51

2.2 Forecasting models 53

2.2.1 Forecasting with Neural Networks 56

2.2.2 Predictive models in manufacturing 60

2.3 Causal models 60

2.3.1 Structural causal model 62

2.3.2 Causal discovery for time-series 64

2.4 Conclusions 71

3 Phase recognition with Time Delay Neural Network 73

3.1 Automatic process phase classification . . . 74

3.2 Time delay neural network . . . 74

3.3 Experiments . . . 78

3.3.1 Datasets . . . 79

3.3.2 Evaluation . . . 84

3.3.3 Metrics . . . 85

3.3.4 Results . . . 86

3.3.5 Discussion . . . 87

3.4 Conclusions . . . 89

4 Predictive monitoring with causal precursors 91

4.1 Causal modeling of manufacturing plant . . . 92

4.1.1 PCMCI . . . 93

4.1.2 Experiments on Causal Discovery . . . 95

4.2 Predictive procedure . . . 103

4.3 Forecasting with S-TCN . . . 108

4.3.1 S-TCN . . . 108

4.3.2 Experiment on S-TCN . . . 110

4.4 Conclusions . . . 117

5 Case Study: An ultra-processed food SME 119

5.1 Executive summary . . . 119

5.2 Ultra-processed food manufacturer . . . 121

5.3 Experiments on Causal Discovery . . . 123

5.3.1 Dataset description . . . 123

5.3.2 Evaluations . . . 124

5.3.3 Results . . . 124

5.3.4 Discussion on Causal Discovery . . . 126

5.4 Experiments on predictive procedure . . . 128

5.4.1 Dataset description . . . 128

5.4.2 Evaluations 129
5.4.3 Results 129
5.4.4 Discussion on predictive monitoring 130
5.5 Conclusions 132

6 Future works and Conclusions **133**
6.1 Future works 133
6.2 Conclusions 135

A Numerical Experiments **137**

List of Figures

1-1 Percentages of firms that consider financing the biggest problem for the company, adapted from [1]. 26

1-2 Projection on jobs lost due to default by size changes 2021/2019. Effect of Covid adapted from [2]. 27

1-3 The digitalization of Italian companies from CERVED 2020 report [3]. 29

1-4 Average Ratings for Technology Benefits and Challenges on UK SMEs survey from [4]. Monitoring technologies are emphasized. 31

1-5 Examples of continuous and batch processes, from [5]. 35

2-1 Time-series causal graph models. Red arrows represent instantaneous effect, blue arrows entails causation between variables at different time lag. 65

3-1 TDNN network architecture from [6]. 77

3-2 3x3 convolution kernels with different dilation rate as 1, 2, and 3 on two dimensional input [7]. 78

3-3 1) ICRT setup from [6]. ICRT dataset setup. (A) Intel RealSense, (B) Panda robot, (C) Leap Motion, (D) ArUco marker, (E) Drop point for robot, (F) Tool for interacting with ring. 2) JIGSAWS setup from [8]. (A) suturing, (B) knot-tying,(C) needle-passing 79

3-4 1) The hardware training console used by one of the students during data acquisition. 2) Example of the Virtual task considered in the VIT-MR dataset: four colored rings need to be placed in the corresponding peg from [6]. 81

3-5 Normalized confusion matrix, color indicates the probability of classification as represented by the scalar number, index are referred in Table 3.1 (1) ICRT, (2) VIT-MR. 86

4-1 Summary of chapter 4. 92

4-2 Synthetic model with ultra-processed food manufacturing features: rectangles represent phases, each phase is represented by one or more variables. The arrows in blue represent linear causal relationships with coefficient β. In green the non-linear and non-polynomial causal links. In red causal links with increasing non-linearity with α coefficient.The different shade of red are related to the strength of the coefficents (i.e., more red-colored arrow entails a stronger causal relationship). This graph is described by the structural causal equation 4.2. 97

4-3 Results of the PCMCI method with Partial Correlation independence test. 101

4-4 Results of the PCMCI method with CMI independence test. 101

4-5 (1) Graph of the example described in 4.2. (2) Structural Causal Model (SCM) for the example, where $\delta_{1...6}$ are coefficients (for a full description of SCM refer to section 2.3.1). (3) Database table of the example. 104

4-6 In the first step of the procedure, PCMCI is used to build the causal vector. We select the causal precursors of the response variable Y and their temporal activation between lagged values. τ_{max} is the potential maximum time connection of the underlying system that in the manufacturing entails the length of the production process. The colored rectangles represent the different machines of the process while black nodes are causal precursors for Y. Arrows show causal connections between time-lagged variables and Y. As example variable X^4 is causal for variable Y after 3 time lags. It means that if we intervene on variable X^4, the values of Y changes after three temporal instances. Arrows with the same causal activation time have identical colors. The variables that are not causal precursors of Y are set to 0. If PCMCI detect causation at multiple time delay as for variable X^1 in the figure, we select the median time lag (i.e., one time lag). The causal vector report for each variables the corresponding time lag. Experts operator can change the detected causal vector adjusting the time lags and modifying the vector. 107

4-7 The architecture of the Depth-Separable block described by eq. (4.3.1). The dilatation coefficient c given as input to the block controls the size of the receptive field. 109

4-8 The architecture of the S-TCN is created from the structure of the causal vector. Depending on the cardinality of $\mathbf{G}$ which represents the groups of variables with the same effect lag on the response variable (i.e., c_{-}), $|\mathbf{G}|$ depth-separable blocks are generated where the kernel dilation corresponds to the lag c_{-}). In each of these blocks, variables with a specific dilation are processed, producing a features map S of dimension batch size $*$ the maximum time length of the process $(\tau_{max}) * 50$. The output of these blocks will contain information about the signals at a specified time delay. These new feature maps are concatenated into a vector (layer 0 in the Figure) which have batch size$^*\tau_{max}$ $*|\mathbf{G}|$ dimension. This new input is processed by a TCN of 4 layers where the kernel dilation d is set to $[2, 4, 8, 16]$ respectively. The TCN allows increasing the network's ability to detect non-linear relationships on the new input. Finally, a fully connected layer with a single output is used to predict the quality variable. 111

4-9 Box plots of numerical experiments. The colored lines represent the average of the result of the ten different realizations at corresponding maximum time activation. The more the τ_{max} increases, the more likely distant causal connections occurs. The results for each experiment with the correct temporal activation are attached in the appendix. . . 115

5-1 Summary of chapter 4 and 5. 121

5-2 Results on real case scenario of the PCMCI method with Partial Correlation independence test. For more details refer to section 5.3.3. 125

5-3 Results of the PCMCI method with CMI independence test. For more details refer to section 5.3.3. 126

5-4 Proposed causal interaction model. Rectangles represent the five selected phases of the ultra-processed food production process. The red links are correct causal relationships inter-phases that allow the flow recognition of the process. Grey links are false positives. Blue links represent the causal interactions between the parameters in the same phase 127

5-5 12 hours of prediction for the response variable in the real case scenario. Above comparison between methods that do not use causal vectors. Below three different causal vectors on the S-TCN model. . 131

List of Tables

1.1 EU recommendation 2003/361 on enterprises category. 25

1.2 Description of the features of SMEs and MNEs. Refer to [9] for the original source and more details 30

2.1 Summary of causal discovery methods presented in section 2.3.2. Provides indication of assumptions of sufficiency, faithfulness as well as whether the method is suitable for high dimensional data. The output can be Binary (cause-effect), CPDAG (completed partially directed acyclic graph), DAG (directed acyclic graph) and partial ancestral graph (PAG). 70

3.1 Phases and ID for each datasets 82

3.2 Features for each datasets 84

3.3 Cross validation for each dataset 85

3.4 Result for macro and micro accuracy for the datasets. For JIGSAWS refer to [10, 11] 87

4.1 Precision, Recall and False positive rate for the synthetic scenario percentage values 102

4.2 Ground truth causal vector corresponding to a single realization with $\tau_{max} = 30$. Example explaned in section 4.3.2 112

5.1 number of in-line intervention on the set point of the machine's parameters 128

5.2 Proprieties of the real case scenario datasets for the predictive procedure and the causal discovery experiments. 129
5.3 Result on real case study with three different causal vector. In bold the lowest mean squared error obtained is highlighted. 130

List of Acronyms

General

GDP Gross Domestic Product

US United States

USA United States of America

I4.0 Industry 4.0

EU European Union

EU27 European Union after Brexit (2020)

SMEs micro, small, medium enterprises

MNEs multi-national enterprise

SM Smart Manufacturing

SCADA Supervisory Control and Data Acquisition

PLC Programmable Logic Controller

AI Artificial Intelligence

DL Deep Learning

SCM Structural Causal Model

DAG Direct Acyclic Graph

PM Predictive Model

RELU REctified Linear Unit

RMSE Root-Mean Squared Error

CPDAG Completed Partially-Direct Acyclic Graph

MAG Maximum Ancestral Graph

SRS Surgical Robotics System

PCA Principal Component Analysis

HMM Hidden Markov Model

SVM Support Vector Machine

ARIMA Autoregressive Integrated Moving Average

Methods and Algorithms

ANN Artificial Neural Network

CNN Convolutional Neural Network

RNN Recurrent Neural Network

FNN Feedfoward Neural Network

MLP Multi-layer Perceptron Neural Network

LSTM Long-Short Term Network

GRU Gated Recurrent Unit

TDNN Time-Delay Neural Network

TCN Temporal Convolutional Network

S-TCN Separable-Temporal Convolutional Network

PC Peter-Clark

FCI Fast Causal Inference

GES Greedy Equivalent Search

LiNGAM Linear non-Gaussian Acyclic model

TiMINo Time Series Models with Independent Noise

TCDF Temporal Causal Discovery Framework

SLARAC Subsampled Lin-ear Auto-Regression Absolute Coefficient

QRBF Quantiles of Ridgeregressed Bootstrap Sample

CCM onvergent cross-mapping

CD-NoD Causal Discovery from heterogeneous/NOnstationary Data

PCMCI PC-Momentary conditional independence

Chapter 1

Research goals and challenges

The manufacturing sector accounts for roughly 15% of the global Gross Domestic Product (GDP) and it reached $13.9 trillion US dollars in 2018 [12, 13]. These enormous numbers are difficult to conceive, however, it is sufficient to consider all the products that surround us to understand the impact of manufacturing in our life. Mass production has expanded, thus the number of manufactured objects has steadily increased. Nowadays, technological discoveries and globalization push industries to adapt the production process to customers' needs by overcoming constraints. The so-called fourth industrial revolution is taking over, blurring the boundaries between the physical, digital and biological domains [14]. The fourth industrial revolution refers to the technological transformation that society is undergoing in the 21st Century [15]. Indeed, since our lives are strongly related to manufacturing goods, the fourth industrial revolution will not involve only products, but it will affect our whole experience. This enormous transformation cannot be relegated to companies, but it must be sustained by all society players as universities and governments. In 2011, Germany was the first nation to lunch a plan to promote a sustainable fourth industrial revolution. The "Industrie 4.0" plan encourages synergy between research centers and stakeholders to introduce technologies and build the proper entrepreneurial attitude [16]. "Industrie 4.0" promotes a holistic concept of industrial production with shared responsibility between companies and society for developing interconnected factories and cities. It focuses on skills distribution and seeks to spread the fourth

industrial revolution thrust throughout the community. Following, Europe adopted the "Industry 4.0" plan inheriting from Germany the concept of combining industrial with social-economic development. In 2015, China began the "Made in China 2025" plan to promote the fourth industrial revolution. Differently from European Union (EU), they propose a top-down approach using substantial financial investments to restructure the entire industry [17]. The United States of America (USA) presented the "Advanced Manufacturing Partnership" project and favored public services, political, education, and training policies. As a result, many governments have adopted policies to manage the introduction of new technologies into industries. Researchers compared innovative policies approved by different nations for Industry 4.0 (I4.0) [18, 19]; the different choices made by governments have strong effects on the evolution of the manufacturing ecosystem. For example, some investments in training and education return after many years with higher qualified operators, while others have immediate results as financial aid to companies. This situation requires a flexible assessment for I4.0 achievement. Therefore, before proposing innovations, it is necessary to consider the manufacturing context analyzed, its needs and limitations. Without proper consideration of the circumstances, the proposed innovation may be unaffordable for some companies or, even worse, impact negatively on social welfare. For this reason, we start this thesis by describing the characteristics of the European and Italian manufacturing context.

1.1 The importance of SMEs for the European manufacturing sector

The realization of I4.0 is heavily affected by companies' size. The EU has divided the businesses into four categories shown in Table 1.1. The size and turnover of a company suggest diverse roadmaps to Industry 4.0. I4.0 technologies need to reach all companies to enable healthy and sustainable industrial growth [20]. Therefore, it is necessary to consider the different characteristics between micro, small, medium en-

Company category	**Staff headcount**	**Turnover**	or	**Balance sheet total**
Large/Multi-national	> 250	$>$ € 50 m		$>$ € 43 m
Medium	< 250	$\leq$ € 50 m		$\leq$ € 43 m
Small	< 50	$\leq$ € 10 m		$\leq$ € 10 m
Micro	< 10	$\leq$ € 2 m		$\leq$ € 2 m

Table 1.1: EU recommendation 2003/361 on enterprises category.

terprises (SMEs) and multi-national enterprises (MNEs) to develop effective methodologies [9]. SMEs worldwide account for 99% of all undertakings and employ 60% of the workforce in the private sector [21]. The International Energy Agency (IEA) highlights that SMEs create almost 50% of the global gross value added [22]. According to [23] European SMEs employ approximately 90 million people in total, and this Figure increases by 1.1 million every year. Although their smaller dimension and lower turnover, SMEs are indeed the true economic backbone of the EU. Applying the I4.0 concept without considering SMEs' challenges is not acceptable in the EU because if the I4.0 technologies would be suitable only for large manufacturers, the consequences could be harmful. For example, proposing solutions that require numerous specialized employers would benefit large firms, while SMEs would not access these technologies. The gap between SMEs and large companies would increase with a consequent loss of competitiveness for most industries.

The presence of SMEs is particularly significant in Italy. SMEs generate the 63% of the total value-added for enterprises compared to the average of 53% in EU27 (after Brexit) [24]. Micro-enterprises in Italy (95%) are over the EU average (93%). Only 0.09% of Italian companies have more than 250 employees, compared to 0.14% in France, 0.19% in Europe, and even 0.48% in Germany (five times as many) [25]. Accordingly, in Italy, micro-companies account for 45% of the workforce, compared to 30% in France, 19% in Germany and 29.5% in the EU [25]. For large companies, the workforce employed is only 21% in Italy, compared to 33% in France, 37% in Germany, and 33% in the EU [26]. These data present Italy as a reference for the adoption of policies towards SMEs. The Italian situation allows to understand the effectiveness of I4.0 practices for SMEs and justifies focusing on this class of companies.

In the next section, the substantial differences between SMEs and large enterprises are exposed. This thesis aims to promote enabling technologies for I4.0 in the context of SMEs. As described in this section, the research focus is motivated by the high numbers of SMEs in the European and Italian panorama.

1.2 Risks and challenges for SMEs

As shown in Table 1.1, in addition to the number of employees, a characteristic feature for SMEs is a limited turnover. The financial aspect, indeed, is quite restrictive for SMEs and represents a primary obstacle [1]. Although it has gradually improved since the mid-2010s, for SMEs' access to finance remains challenging. The financing situation of euro area firms was particularly severe for SMEs, with some differences across sectors. Figure 1-1 show that the percentage of firms that perceived access to finance as their main problem was consistently higher for SMEs than for large companies [3]. During the 2009-12 period, about 15% of EU SMEs that were looking

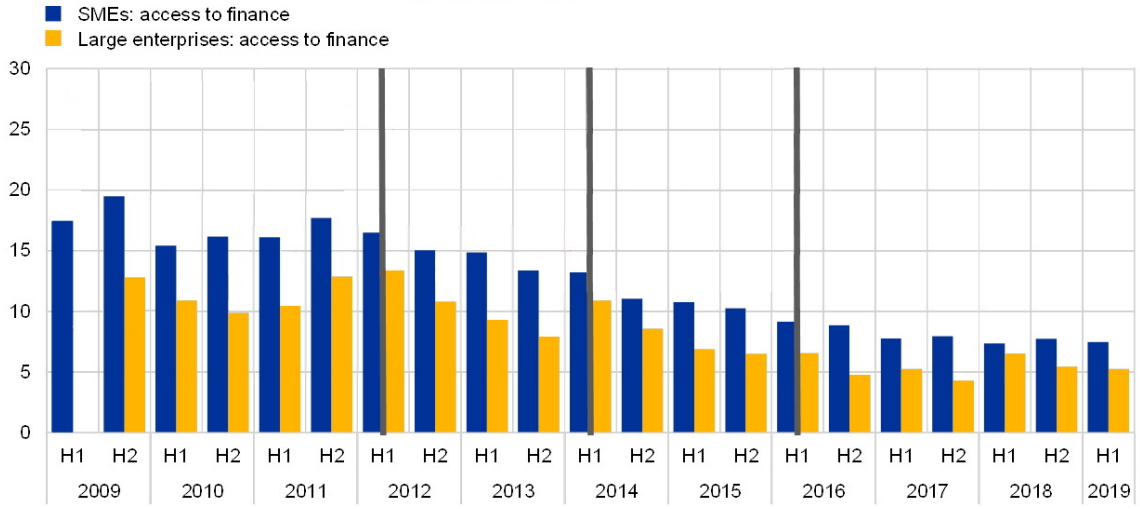


Figure 1-1: Percentages of firms that consider financing the biggest problem for the company, adapted from [1].

for bank loans as a founding source was also constrained in obtaining a bank loan. Today it is currently stabilizing around 8% [3]. The financial uncertainty has been crucial with the arrival of the pandemic, further increasing the gap between SMEs

and large enterprises. If we consider the Italian scenario that predicts the number of jobs lost due to Covid, the consequences are expected to be less critical for large companies than for SMEs (Figure 1-2). On a prospect of 125 thousand lost jobs, large companies expect a loss of 1.1 percentage points lower than SMEs (almost half loss) [27]. The gap between MNEs and SMEs leads to unfair competition between

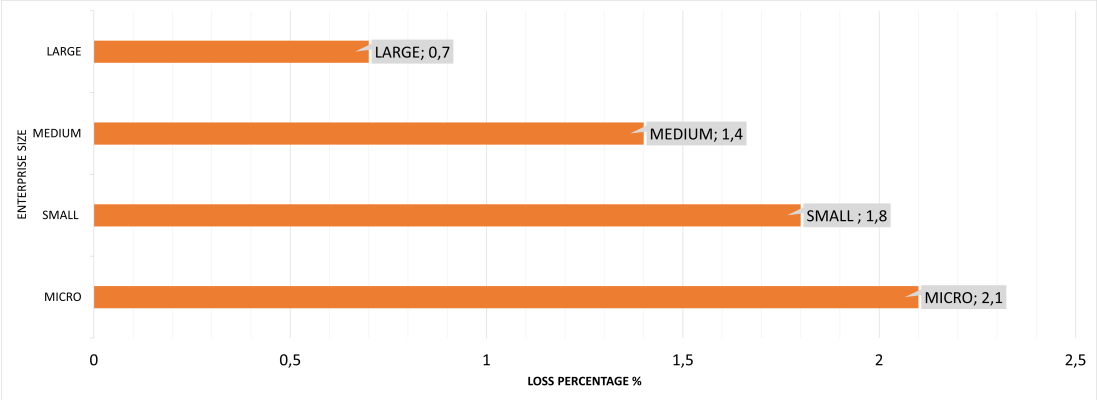


Figure 1-2: Projection on jobs lost due to default by size changes 2021/2019. Effect of Covid adapted from [2].

industries and it favors large companies at the expense of SMEs with a consequent reduction in social welfare distribution. In recent years, to facilitate the achievement of I4.0 in SMEs, strong funding has been issued by governments. As reported at the beginning of this section, the adopted policies (particularly the EU with the NextGeneration EU plan) gradually fund SMEs and consequentially decrease financial pressure. Specifically, in Italy, there will be a unique opportunity as in addition to the new "Transizione 4.0" plan and the NextGeneration EU project, with "Piano Italia" an endowment of around 200 billion will be granted, just under two-thirds in the form of loans and the remainder in grants [28]. These funds allow companies to access expensive technologies through facilitated financial plans. For example, during the "Industria 4.0" project (the first plan adopted by Italy for I4.0), hyper-amortization (150%) has proved to be an extremely effective tool in facilitating the purchase of advanced technologies for SMEs. It emerges that between 2017 and 2018, the value of investments generated by hyper-amortization in Italy exceeds 25 billion euros overall [29]. Therefore, although limited financial capabilities are a feature of

SMEs, many efforts have been made to alleviate the economic pressure and in the coming years the access to finance should further improve.

However, other barriers slow down the adoption of I4.0 in SMEs. In [9], Mittal et al. reviewed Smart Manufacturing (SM) and I4.0 maturity models and they analyzed specific requirements of SMEs. They identify three main research gaps that pertain to SMEs:

- different starting conditions between SMEs and large firms;
- disconnection between maturity models and self-assessment readiness tools;
- differences in preparing the next step after maturity and readiness are assessed.

The above study highlights the importance of cultural maturation in SMEs. Even if industries can obtain funding thanks to governments' support plans, without I4.0 cultural awareness, funding can be lost on inadequate technology. Often in SMEs, enabling technologies are erroneously perceived as tools to improve the current production without exploiting the true potential provided by the interaction of these systems. Without adequate cultural and managerial competence, I4.0 technologies are not recognized as revolutionary and fail to switch the running paradigm in autonomous and re-configurable industries. The result of the survey carried out by [30] highlight this risk while drawing a comparison between different nations: *The majority of German companies does not assume their production processes have achieved a high degree of digitalization, while the Slovenian and Italian companies believe that their production processes have achieved a medium degree of digitalization. So it is not strange if the Italian and Slovenian companies agree that new technologies can be implemented gradually with contained investments, even without radical changes, and that implementation does not require significant investments. In contrast, the Germans think that implementation of Industry 4.0 requires major investments and that these will cost* [30]. This behavior describes an inconsistency, given that digitalization is higher in Germany than in the other two states [31]. Digitization and cultural maturity towards I4.0 are correlated. These discrepancies are also recognizable between

large companies and SMEs. Figure 1-3 shows that Italian manufacturer's digitization is strongly tilted in favor of large firms [32]. Again, the main risk is that the

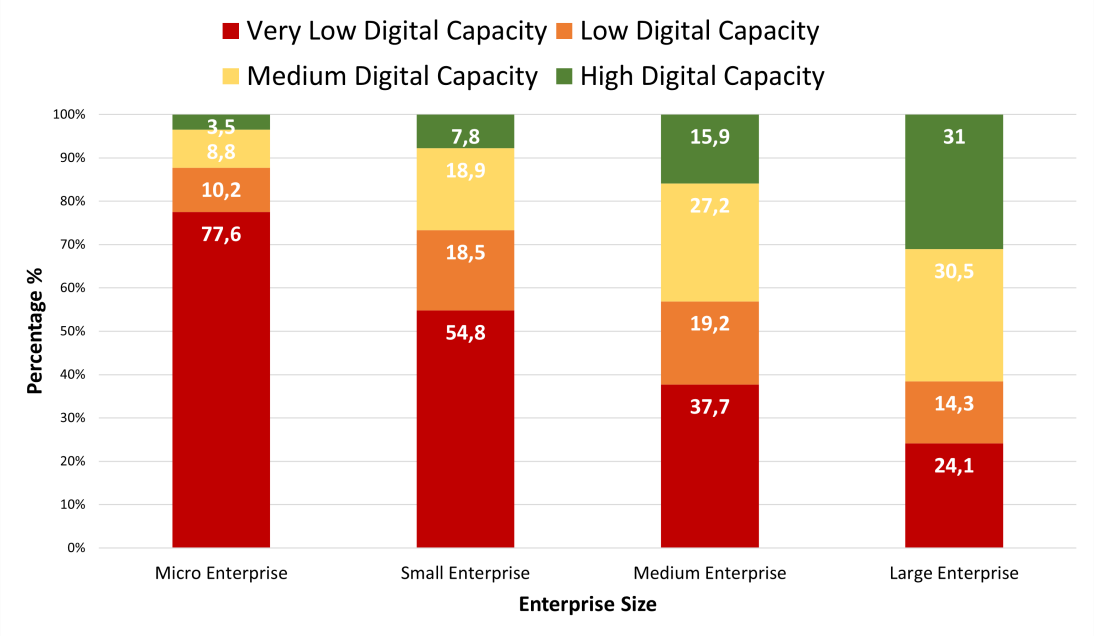


Figure 1-3: The digitalization of Italian companies from CERVED 2020 report [3].

improvement of advanced technologies benefits large firms as they are more prepared for the transition, not succeeding in intercepting the majority of companies. Therefore, it is suggested that although I4.0 technologies have revolutionary potential, their introduction into the companies must be accompanied by a cultural shift.

In addition to the financial and managerial circumstances, another feature of SMEs is related to business strategies. Table 1.2 reports the different characteristics of the SME and Multi-National Enterprises (MNE), adapted from [9]. SMEs must quickly adapt the industrial process to respond to customer needs and produce highly specialized products [33]. A large product portfolio has a negative influence on the operational performance [34]. The request for a flexible production increases the overall complexity of the management. Moreover, in order to promptly customize the production, companies struggle to manage long-term strategies [35]. The need for flexibility and lack of long-term strategy prevents research and development. The absence of standardization and protocols due to large variations in production processes hinders the training time of new operators. Therefore, a small number of expert

N.	Feature	SMEs	MNEs
1	Financial Resources	Low	High
2	Advanced Manufacturing Technology	Low	(Very) High
3	Software Umbrella	Low	High
4	Research and Development	Low	High
5	Nature of Product Specialization	High	Low
6	Standards consideration	Low	High
7	Organization culture and Leadership flexibility	Low	High
8	Company Strategy	Dictated by Instinct Of Leader (Owner)	Market Research and Accurate Analyses
9	Decision Making	Restricted to Leader	Board of Advisors and Consultants
10	Organizational Structure	Less Complex And Informal	Complex And Formal
11	Human Resources Engagement	Multiple Domains	Specialized Domains
12	Exposure to Human Resource Development	High in The Industry Low Outside The Industry	Low Within Industry High Outside the Industry
13	Knowledge and Experience Industry	Focused In A Specific Area	Spread Around Different Areas
14	Alliances with Universities Research Institutions	Low	High
15	Important Activities	Outsourced	Internal to the Organization
16	Dependence on Collaborative Network	High	Low
17	Customer/Supplier Relations	High (Strong)	Low (Not So Strong)

Table 1.2: Description of the features of SMEs and MNEs. Refer to [9] for the original source and more details

operators find themselves managing the increasing complexity of SMEs without the capability to train or hire new practitioners. For example, according to a 2015 survey of the manufacturing sector in Japan, the main obstacles to investing in data-mining (that is an advanced technology for I4.0) are related to lack of human resources and planning [36]. To enable I4.0 in SMEs is necessary to improve the management of processes reducing overall complexity. Reducing complexity allows devoting more resources to knowledge transfer, testing new technologies and improving production operations.

Even if efforts have been made to increase funding and maturity culture, the

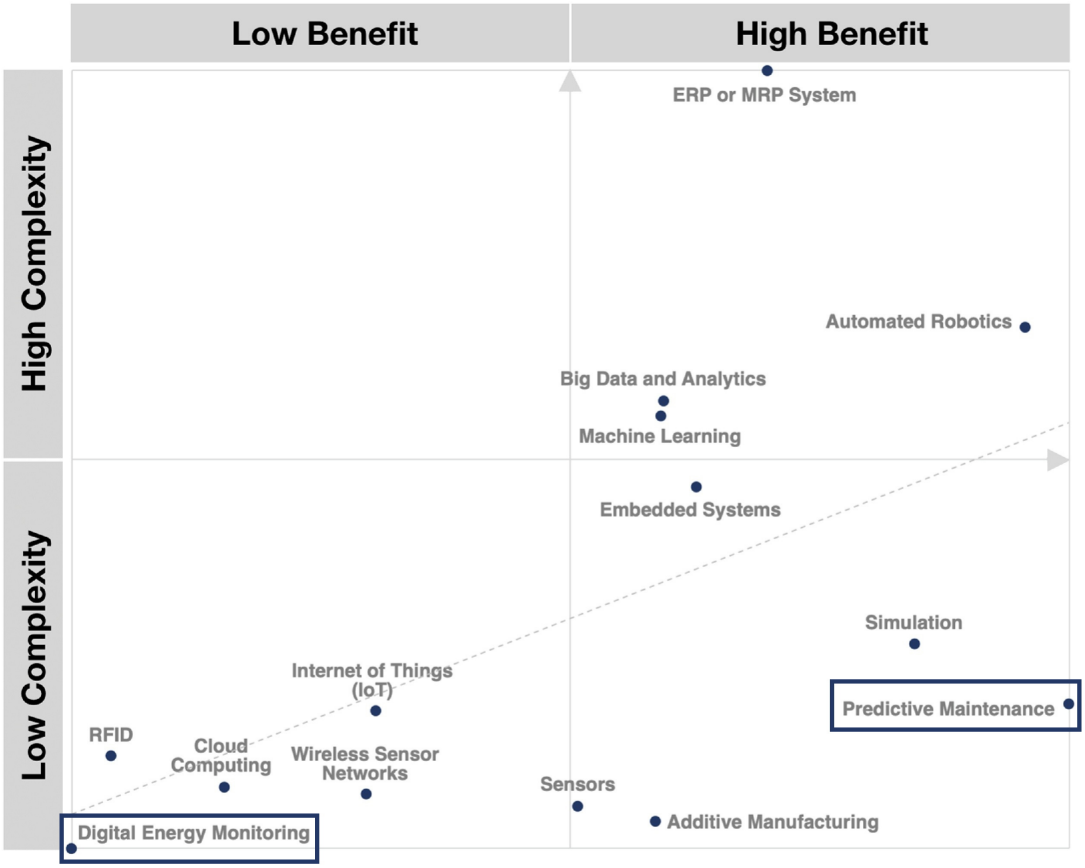


Figure 1-4: Average Ratings for Technology Benefits and Challenges on UK SMEs survey from [4]. Monitoring technologies are emphasized.

demand for product customization and flexibility increases business complexity in SMEs. Considering SMEs' needs is thus important to propose methods that aim to reduce complexity and that can be gradually integrated into companies. In a survey on UK SMEs, the authors compare benefits and complexity for I4.0 technologies [4]. The result shows that the most promising technology in terms of benefit and complexity are related to monitoring (i.e., predictive maintenance and digital energy monitoring). Process monitoring is effective for [37, 38, 39]:

- reducing complexity;
- reduction of waste;
- improve employee awareness;
- increase understanding of processes;

- gains the involvement of people (operators, engineers, managers);
- reduce service delivery time;

In the next section, we introduce the advantages of process monitoring in SMEs. Then, we present the limitations of current monitoring systems and the contribution of this thesis. We aim to improve monitoring technologies and promote a sustainable fourth industrial revolution without burdening the gap between SMEs and MNEs.

1.3 Process management: modeling and monitoring with AI

Process monitoring has been an active area of research starting from the beginning of the last century. In [40], the evolution of monitoring systems is described, dividing it into three categories from past (detection), present (diagnosis), and future (prognosis). Detection belongs to the challenges faced in the past and it can be referenced as monitoring. Monitoring the production system and its environment can be achieved today by using various connected sensors [40]. For example, different elements of a production system can issue warnings if there is an inconsistency in achieved performances. Sensors can monitor the process, detecting faults whenever a fixed threshold is exceeded. The monitoring function can also be implemented as historical analysis [41]. With a memory system, the sensors can recognize past trends, such as an increasing or decreasing behavior, identifying faults. Detection is applied in fault detection, or identification [42]. It includes univariate and multivariate statistical process monitoring methods, which are based on adopting a certain probabilistic model that is suitable for identifying trends in a wide class of industrial processes. These general-purpose methods typically do not require specific information about the process structure, other than the parameter estimates.

Diagnosis is the feature of the process monitoring solution under development. These models are not limited to recognizing a deviation in the process, but they estimate the causes. Such causal structure is absent from detection methods and

may lead to an ambiguous fault recovery. For example, the same erroneous trend for the final product can be generated by faults in multiple process phases. While the detection system recognizes only the abnormal behavior of the outcome, the diagnosis feature allows identifying the source fault. Therefore, it allows the operator to intervene to restore the process. Diagnosis systems can also be referred as process recovery or faults diagnosis. These methods use the information derived from the sequentiality of the process with both: knowledge or data-driven approaches.

In the future, the process prognosis will integrate a predictive dimension. The goal of prognosis systems is to predict potential failures before they occur. Unlike detection, to forecast faults, the underlying causal structure of the system is required and a statistical model of the plant is provided. Prognosis systems recognize patterns in the past features that will cause a non-normal behavior of the outcome. Prognosis is closely related to "Equipment Health," which monitors equipment performance evolution over time based on operational effectiveness and failure rate.

Detection, diagnosis and prognosis describe three features for monitoring with increasing complexity. Monitoring systems are also affected by the type of data analyzed, whether the objective is to detect diagnosis or predict faults. In [40], the authors highlighted different characteristics of monitoring methods that, depending on the underlying structure of the data, must be considered. Each of these adds several levels of complexity:

- *From Univariate to Multivariate, to High-Dimensional ("Mega-Variate")* It suggests an increased complexity due to the simultaneous analysis of one, several and many sensors with a consequent growth in the dimensionality of the model;

- *From Static to Dynamic, to Non-Stationary* Highlights processes that either remain stable over time, change following a recurrent pattern or are irregular;

- *From Monitoring the Mean to Dispersion, to Correlation* Describe the different types of relationships between sensors that can have similar behavior in magnitude, in their evolution over time (i.e., are correlated) or in their distribution. Generally, the more significant is the variability, the greater the challenge in

recognizing patterns in the data;

- *From Unstructured to Structured Process Monitoring* Denotes the increasing complexity given by using structured data that are consistently ordered and related or unstructured data (for example, text or non-formatted values);

- *From Homogeneous Data Tables to Heterogeneous Datasets* Using multiple datasets may increase the complexity of the problem due to missing data o different record information.

Thus, there are many monitoring systems with different features combinations. These characteristics are often interrelated, such as Mega-Variate and Dispersion (the more data, the more the diversity). Therefore specific sub-trends in monitoring systems research began to develop. In particular, we highlit the following research trend topics on monitoring systems:

Health monitoring aim at the detection of faults after the occurrence of certain failures (diagnosis) and predictions of the future working conditions and the remaining useful life (prognosis). To this category belong the works on fault detection and predictive maintenance [43].

Plant-Wide Monitoring, focus on the integration of monitoring systems at a plant and business level. Such systems are usually physically interconnected, very large in scale, geographically dispersed, and have hierarchical structures. The responsibilities of the plant-wide monitoring systems mainly include plant-wide performance evaluation, plant-wide fault diagnosis and prognosis, as well as (in the general concept) plant-wide performance optimization with maintenance [44].

To summarize, we reported that the scope of research for systems monitoring is vast and varies depending on the type of industrial processes. We can state that the detection deals with the monitoring, the diagnosis adds past models with constraints, and the prognosis foresees the future state. In this thesis, we will cover the detection, diagnosis and prognosis features for the industrial process of SMEs on both synthetic and real scenarios.

1.3.1 Process monitoring in manufacturing processes

There are generally two types of manufacturing processes, batch and continuous. In a batch process, the raw material is charged into the system at the beginning, and the product is discharged all at once at the end as two separate operations. In continuous processes, the material and product are continuously charged and discharged from the system throughout the duration of the process [5]. The above definitions can be applied to an individual machine or an entire manufacturing process consisting of a series of machines as shown in Figure 1-5. A process phase may consist of a set of machines or a single machine. In a continuous process, the quality of the finished product is the result of all the previous phases. In the case of batch production, the product can be controlled and inspected separately. The process flow is defined by the

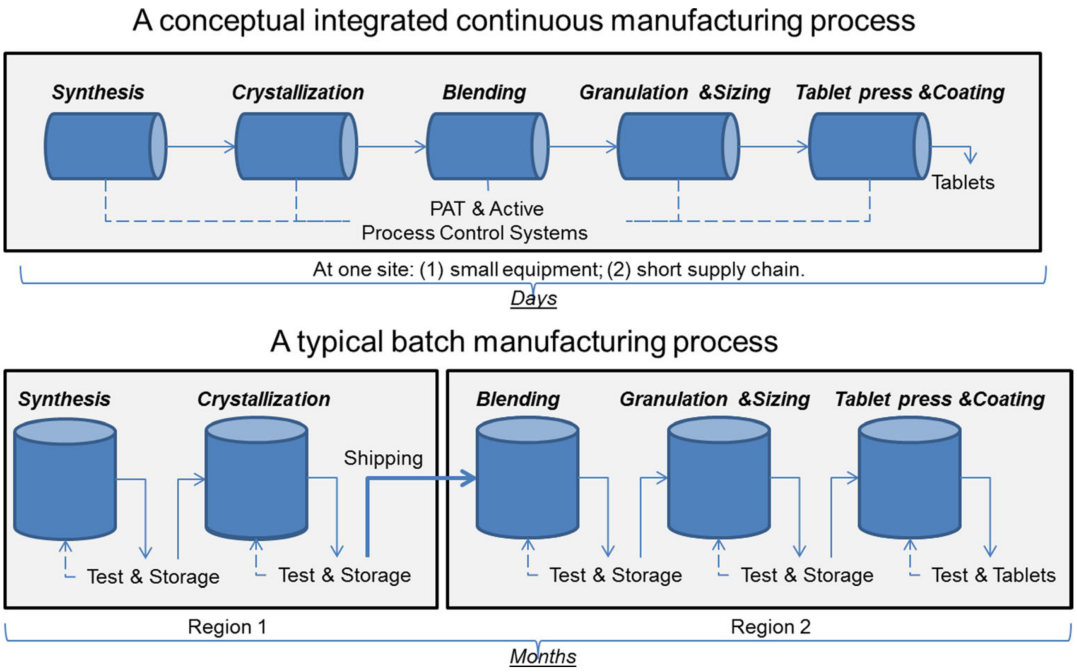


Figure 1-5: Examples of continuous and batch processes, from [5].

order of the phases where the raw material pass through to become a finished product. There are many ways to describe an industrial process as graphical models, petri-nets, ontologies, computer languages and many others [45]. These models generally use states for representing phases and transition functions between them. Therefore they can model the interaction between phases displaying the industrial process flow.

The requirement for these types of methods is the distinction between states as it must be possible to recognize the actual phase of the system. For example, we must distinguish an assembly phase from a packaging phase and describe the production process into separate phases that will represent the states of the model. Although this procedure is trivial in an automated and temporally organized system, being able to distinguish the phases in a process where there is an extensive human intervention, for example, in an automated process based on collaborative robotic, is very complex [46]. In one part of this thesis, we will attempt to distinguish the phases of a process during collaborative procedures.

In a continuous process, however, there is another constraint. If a variation occurs upstream in the production, it will propagate into the subsequent phases affecting the final product. Therefore the outcome is conditioned by the past phases of the process at lagged instants of time. In statistical control, a process is defined when it is possible to predict the future trend of one or more variables based on past observations with a probabilistic confidence. Therefore, given some variations on past observation, we can guess the system's behavior with some probability. Regressive models estimate the relationships between a "response" variable and one or more independent variables. These types of methods, instead of modeling the transition function between states, the system's behavior (also called "response") is modeled given other production process variations. Those variations are disturbances that negatively affect the quality of the final product or interventions made by the operator to stabilize the production. An effective control strategy for continuous process has to reduce the risk of these potential disturbances to process stability and product quality. This definition involves identifying past states that already happened in the past and a future state that can be identified with the vector of response variables $\mathbf{Y}$ [47]. The response variables vector is given by the evolution of the past states within a probabilistic confidence. Following [42] in a manufacturing process, $\mathbf{Y}(t)$ can be a quality measure obtained over time, space or states and $x_1(s), ..., x_n(s)$ is a set of process variables that influences the final quality of the product. Without loss of generality, we use a single output variable $y(t)$ over time. For example, $y(t)$ can

be the temperature of the final product while $[X_1(t-1), \ldots, X_n(t-1), \ldots, X_1(t-j), \ldots, X_n(t-j)]$ can be the energetic consumption of machines that influence the final temperature of the product. We assume that the $y(t)$ can be modeled by a non-linear combination of covariates as follows [42]:

$$y(t) = \sum_{j=1}^{p} g_j(\int x_j(s)\beta_j(s,t)ds) + e(t) \tag{1.1}$$

where $t \in R$ is a functional intercept t (e.g., the signal sampling in temporal instances), g_j is a non-linear function, $x(s)$ with $s \in R$ is j^{th} functional input, $\beta_j(s,t)$ is the jth functional parameter and $e(t)$ is a random noise. The goal is to estimate the functional parameters $\beta(s,t)$, given a set of observations. In a second part of the thesis we will deal with product quality forecasting. We propose a procedure that involve causal discovery, regressive model and domain knowledge to integrate the causal flow typical of continuous production in a prognosis system for a quality variable of the product. To summarize, we present two types of the manufacturing process: the batch process that requires a distinction between phases and the continuous process that requires past states. Following, we highlight the limitation of current monitoring solution manufacturing processes. Finally, at the end of the chapter, we present our contribution to overcoming these barriers, also considering the challenge in SMEs.

1.3.2 Challenges in existing solutions

In the industrial domain, the amount of data and their heterogeneity is enormous. As a result, regression systems on data dimensionality (such as neural networks) have had a major impact on research as described in section 2.2.1. Recently many monitoring systems that apply neural networks to historical data have been proposed [48]. Nevertheless, given the complexity of industrial data, neural networks need many parameters to infer systems behavior, thus, they are generally referred as black-boxes. Black-box models are non-transparent and their predictions are not traceable by humans given the large number of variables used to parametrize the model. This paragraph focuses on three main limits of existing solutions that use a black-box

approach for monitoring. In particular, we highlight the following limitations:

- Black-box approach lacks process understanding.
- Knowledge-based approach is not considered for improving methods flexibility.
- The discretization of the process in phases is assumed.

After an initial period in which black-box systems were strongly used to create models and obtained outstanding results in many research areas, several critical issues towards this type of system are emerging. It is becoming evident that black-box models are limiting the achievement of intelligent data fusion because they do not allow the reconstruction of the decisional path. Therefore, it is useful to set constraints on the models by exploiting domain information and reasoning policies congruent to the context analyzed. The aim is to give more semantic meaning to the results obtained and the pattern recognized in the input features. Semantic knowledge of the process means understanding how and why the models make some decision (explainable artificial intelligence), inferring the causality of the process (causal modeling) and integrating background knowledge. One of the main interests for companies is, indeed, to infer which are the causes of a deviation in product quality. Monitoring system that aims to regress heterogeneous data accepting or denying the natural state of the process without considering the interpretability of the decisional path, is limiting for the management of the process. Using black-box models, it is not possible to infer causation and perform faults' prognosis [49].

Another limitation of the so-called black-box models applied for process monitoring is the lack of flexibility. In the context of SMEs, many production processes are exploited in order to obtain customizable products. Moreover, all the unrecorded values and operators' knowledge cannot be integrated into the system as they are not contained in the data. Therefore, black-box monitoring systems are suitable for companies with lots of data, substantial computational resources and standardized processes. These features generally belong to large companies, thus, further increasing the gap between SMEs and MNEs. For this reason, it is important to develop systems

that integrate the operators' knowledge to reach higher flexibility and adaptability to the different needs of the industry.

Finally, a weakness of current works in literature is the acknowledgment of process phases. In the classical industry, the phases of a process can be easily identified through low-level signals acquired from the plant with Supervisory Control and Data Acquisition (SCADA) or Programmable Logic Controller (PLC) that record changes in the set point. The subdivision of the process in phases can then be made simply by checking setpoint variations. The transition between phases is discrete and easily distinguishable. In an Industry 4.0 perspective where robots and humans collaborate, the distinction in process phases is far from trivial. Recognizing and segmenting a collaborative process into different stages is necessary for state recognition and process description.

1.4 Thesis description and organizations

This thesis aims to contribute to process monitoring methods specifically designed for SMEs to reduce the industrial processes' management complexity. We propose two contributions for recognizing process phases and forecasting a quality variable of the final product. The predictive procedure allows the preemptive supervision of irregular operating conditions without expert operators' intervention, exploiting the process's causal flow. The automatic recognition of phases allows monitoring process's phases, including collaborative tasks between humans and robots. This thesis proposes the following scientific contributions:

1. Uses a causal process model rather than a temporal one. We follow the flow of the product instead of the recording timestamps for machines. This approach leads to the optimization of the data used as input to a predictive procedure that exploits the causal model to represent distant temporal relationships precisely.

2. Extends the recognition of process phases by considering an I4.0 setting. In contrast to a standard automation system where set-point changes define phases,

we recognize the process phases using a supervised machine learning approach. Even in collaborative situations between humans and machines, it can identify the actual running phase.

3. We propose a monitoring procedure based on both data-driven and knowledge-based approaches. Integrating domain experts' knowledge allows reaching adequate flexibility for SMEs even in situations of missing information. The proposed procedure can therefore account for various industrial environments and possible unrecorded causes. On the other hand, the data-driven approach relieves expert operators from continuous active control of the process.

1.4.1 Summary of the thesis

The thesis is organized as following: in chapter 2, we present state-of-the-art in the following directions. Firstly, we report past work on process phase recognition at two levels: robot and human actions recognition and industrial process block decomposition. For gesture recognition, we will also include state-of-the-art in surgical robotics, which turns out to be one of the most advanced in the field of action segmentation. Then, we will describe regressive models that determine the future behavior of the system, given a set of past observations. We will describe the currently used models for forecasting with particular attention to deep learning systems. Finally, we will focus on causality, which is the main innovation of this thesis. We will define the framework used and describe the algorithms to infer cause-effect relationships from the data.

In chapter 3, we present the first contribution on automatic phase recognition. The method, described in section 3.2, deals with the recognition of phases in a collaborative process. It is a supervised deep-learning method that describes movements through a large receptive temporal context. The method is validated using three different datasets, two of which specifically designed to reconstruct a typical I4.0 production process.

The second contribution described in chapter 4 proposes a predictive procedure

for monitoring a process quality variable. This procedure exploits the causal flow of the process inferred from the data in combination with a novel supervised deep learning architecture to achieve a more efficient prediction. We performed specific experiments for both the causal flow discovery and the predictive procedure on the synthetic dataset. In section 4.1 we present the method used for causal discovery and the experiments on synthetic datasets that reproduce the characteristics of a manufacturing process. In section 4.3.1 we present the proposed neural network architecture that is valitated with extensive numerical experiments in section 4.3.2. The procedure for combining the inferred causal flow of the process and experts domain experts knowledge in the predictive model is described in section 4.2.

After presenting the methods and the results obtained on synthetic datasets, in chapter 5, we show the work done with a local SME that produces ultra-processed food. Thanks to this collaboration, it has been possible to test the predictive procedure and the causal discovery method on data obtained from a real continuous process. The collaboration, which lasted for nearly two years, has led to the development of a predictive application for the company and in collecting realistic data for models evaluations. The results obtained with the company are presented and discussed at the end of chapter 5.4.

Finally, in chapter 6, we outline future works, highlighting the potential of causal theory for future monitoring applications and proposing extensions for the presented methods.

1.4.2 Summary of contributions

In this thesis, we gave the following contributions to the problem of process monitoring for SMEs:

- In chapter 3 we propose a method for automatic process phases recognition based on supervised deep learning. We apply an ANN architecture named Time Delay Neural Network (TDNN) to recognize the phases in a collaborative process. We demonstrated that TDNNs are suitable for recognizing particularly

complex movements and interpreting signals from multiple sensors on three different datasets.

- In chapter 4 we propose a predictive procedure that integrates the causal relationships of the industrial plant and the domain's knowledge. We have combined a causal discovery algorithm with a robust false positive control and a specific ANN architecture capable of reaching distant temporal information efficiently. The procedure allows building more accurate predictive monitoring systems while also providing operators control over the monitoring system.

- In chapter 5 we tested the causal discovery alghoritm and the predictive procedure in collaboration with a local SME. The methods proved to be effective in proposing new online interventions protocols and forecasting an important feature for product quality.

Publications: The following research resulted in the publication of two journal papers and 3 conference papers. An additional journal article was submitted on 01/03/2022 and is currently under review. Finally, this research has led to collaboration with an industry company and the release of data available to the research community.

- G. Menegozzo, D. Dall'Alba and P. Fiorini, "CIPCaD-Bench: Continuous Industrial Process datasets for benchmarking Causal Discovery methods" in IEEE Robotics and Automation Letters, Under Review.

- G. Menegozzo, D. Dall'Alba and P. Fiorini, "Industrial Time Series Modeling With Causal Precursors and Separable Temporal Convolutions," in IEEE Robotics and Automation Letters, vol. 6, no. 4, pp. 6939-6946, Oct. 2021, doi:10.1109/LRA.2021.3095907.

- G. Menegozzo, D. Dall'Alba and P. Fiorini, "Causal interaction modeling on ultra-processed food manufacturing," 2020 IEEE 16th International Conference

on Automation Science and Engineering (CASE), 2020, pp. 200-205, doi: 10.1109/CASE48305.2020.9216973.

- G. Menegozzo, D. Dall'Alba, A. Roberti, and P. Fiorini, "Automatic process modeling with time delay neural network based on low-level data.," Procedia Manufacturing, vol. 38, pp. 125–132, 2019, doi: https://doi.org/10.1016/j.promfg.2020.01.017.

- G. Menegozzo, D. Dall'Alba, C. Zandonà and P. Fiorini, "Surgical gesture recognition with time delay neural network based on kinematic data," 2019 International Symposium on Medical Robotics (ISMR), 2019, pp. 1-7, doi: https://doi.org/10.1109/ISMR.2019.8710178.

- M. Bombieri, D. Dall'Alba, S. Ramesh, G. Menegozzo, C. Schneider and P. Fiorini, "Joints-Space Metrics for Automatic Robotic Surgical Gestures Classification," 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2020, pp. 3061-3066, doi: https://doi.org/10.1109/IROS45743.2020.9341094.

1.5 Conclusions

To summarize in chapter 1, we have introduced the importance of SMEs for the EU manufacturing industry, particularly in the Italian context. We have highlighted three main challenges for SMEs: financial access, cultural maturity towards I4.0, and increasing process management complexity. We propose process monitoring as the most suitable technology to facilitate management. We presented the limitations of current monitoring solutions: lack of process understanding, the demand of process's phase recognition methods and integration of knowledge-based strategy for improving flexibility. Finally, we propose incorporating prior knowledge and Artificial Intelligence (AI) methods to achieve monitoring that considers the causal process flow, automatic phases recognition, and operators' knowledge. At the end of the chapter we reported the organizational structure of the thesis.

Chapter 2

State of the art

2.1 Process phase recognition

Thanks to a continuous automation and optimization, production processes have become a concatenation of almost simultaneous tasks. As a result, distinguishing between production phases has become challenging due to faster interactions. Contrary to automated phases, manual operations, in which the operator carries out different functions, do not always have a pre-established order and are hard to accelerate. Moreover, sensitive operations performed by operators are difficult to bound in regular time intervals. Therefore, it is useful to build an automatic system that recognizes the state of the process. Recognizing, automating, and integrating manual and automated operations will be further accentuated in future production that integrates robotic automation components with human-performed actions. In the last years, since the introduction of the I4.0 concept, there has been an increase of industrial applications exploiting Human Robot Collaboration (HRC) [50]. Several works compared collaborative-robotic cells with manual or robotic work-stations. Considering the production process subdivided into different tasks, it is clear that some assignments could be performed more proficiently by humans or robots alone, others collaboratively [51]. While in automated processes, it is straightforward to define the different process phases thanks to the changes of setpoints recorded by the programmable logic controller (PLC), in the manual or collaborative tasks, it is necessary

to recognize the actions of the operators. Action segmentation is crucial for applications ranging from collaborative robotics to analysis of activities of daily living. Machine learning (ML) methods have shown excellent results for pattern recognition from data and are therefore helpful in finding differences between phases. Following notation reported in [52], an ML model can exploit a training dataset S of n observations, where S is defined as

$$S = \{X_i,\ c_i \in R\}_{i=1}^{n} \tag{2.1}$$

with X_i, the i^{th} observation, consisting of a set of p time series defined as $X_i = [x_i^{(1)}(t), \ldots, x_i^{(j)}(t), \ldots, x_i^{(p)}(t)],\ t \in [0, 1]$ and y_i is a class target value $\in \mathbf{C}$. $\mathbf{C}$ is the set of process classes, i.e., the different phases. For example, in our scenario X can be a set of images of an operator assembling an object, or, a series of the kinematics extracted from a cobot on different set of phases contained in $\mathbf{C}$ where $\mathbf{C} =$ [*object-pickup*, *displacement*, *object-storage*]. Generally, machine learning problems are divided into three groups depending on $\mathbf{C}$. If the labels are not available, it is an unsupervised machine learning problem. If the class labels are usable only for some data, it is called a self-supervised method; otherwise, if all data points have the corresponding class labels available in the training set, it is a supervised problem. Following, we will briefly describe these three approaches.

- **Unsupervised** methods do not use labels; therefore, they discriminate between classes using learned features. They generally use the distance metrics on input data or density on distribution [53]. These metrics create clusters of classes and segment the data into groups [54]. The main advantage of unsupervised learning is that it does not require labeling the data (except during model evaluation) and is therefore easily scalable. Unsupervised methods are suitable where the labeling procedure is done manually and requires much effort. The features that represent each class should be homogeneous in the dataset with few outliers. Consequently, these methods generally can not reach the same

precision or accuracy as supervised and semi-supervised learning. According to [55] unsupervised methods are grouped into three categories:

1. *Clustering:* Clustering is the process of splitting a set of entities so that the entities in a group (called cluster [55]) are similar to each other than those in the other groups.
2. *Anomaly detection:* The detection of an anomaly, which can also be referred to as outliers, noise, deviations, novelties, and exceptions [56], is the recognition of items or events which do not comply with a predicted design or other items in a dataset [55].
3. *Artificial Neural Networks (ANN):* These methods exploit NN to find patterns in the data features. Some of these methods are Autoencoders, Deep Belief Network (DBN), Hebbian Learning, Generative Adversarial Network (GAN), self-organizing maps (SOM) [57, 58, 59, 60].

The methods can also be divided into parametric (i.e., they assume a distribution in the data) or nonparametric methods. Among the parametric methods in action recognition, we can find Gaussian Mixture Model, Gaussian Hidden Markov Model, Coresets and Transition State Clustering (TSC) [61]. Nonparametric models require fewer a priori assumptions about the data; for instance, we can find PCA, Histograms, Kernel density estimator and k-nearest neighbor [62, 63, 64].

- **Semi-supervised** classification methods are particularly relevant in scenarios where labeled data is scarce [65]. In those cases, it may be challenging to construct a reliable supervised classifier. This situation occurs in application domains where labeled data is expensive or difficult to obtain, like computer-aided diagnosis, drug discovery and speech recognition [65]. Semi-supervised methods use algorithms based on unsupervised methods to increase the labeled required for supervised training. They are thus halfway between the two approaches. In [65], authors distinguish semi-supervised classification methods, considering

the primary objective of the approach (transductive versus inductive learning) and the way unlabelled data is used (i.e. wrapper methods, unsupervised preprocessing, and intrinsically semi-supervised methods). In the field of action recognition, most of the work has been done on video or images as they represent the most straightforward way to capture movements [66]. A review of semi-supervised methods is reported in [67, 68].

- **Supervised** ML approaches are the most popular and commonly used technique [69]. They are used in many research fields, such as medical imaging, video surveillance, gait recognition, home care center and intelligent vehicle system [70]. Intuitively, supervised methods are the most reliable but they require extensive and onerous data labeling. We can divide them into machine learning models and deep learning models that use deep neural networks. Among the former, we find hidden Markov model (HMM), Ada-Boost classifier, Support vector machine, Bag of words [69]. To the second group belongs Convolutional neural network (CNN), Recurrent neural vector (RNN) and multi-layer perception (MLP).

Neural Network models can also be classified into the following two distinct categories: discriminative and generative. A discriminative model is a bottom-up approach in which data flows from the input layer via the hidden layers to the output layer. They are used in supervised training for problems like classification and regression [71]. Discriminative models learn the boundaries between classes or labels in a dataset. Generative methods on the other hand, can generate new data occurrences. Generative models focus on the distribution of classes in a dataset and fit training data points in distribution that can be used to generate new data.

This thesis focuses on discriminative supervised learning methods as they generally obtain higher accuracy results in classification and regression problems. The first proposed method addresses the challenge of phases recognition in a production process. The labeling procedure, which is the main disadvantage of supervised methods, can occur in two ways: during production through real-time labeling by operators

or offline by historical data analysis. Offline labeling is generally more precise as it allows an evaluation by multiple expert operators reducing possible errors. Online labeling, however, speeds up the process of datasets creation. In our experiments, we use both approaches. The process segmentation into phases should be sufficiently generic to withstand potential factory re-configurations. In this thesis, a comparison between supervised deep learning and machine learning methods is provided, however, a comparison between unsupervised or semi-supervised methods is not addressed. Following, we cover supervised machine learning methods for both: process phase classification in chapter 3 and product quality forecasting in chapter 4.

2.1.1 Automatic gesture recognition for Surgical Robotic Systems

Many research on process segmentation uses video or images as they are the most straightforward way to capture motions. Process segmentation related to action and gesture recognition represents a major challenge in several contexts (i.e., human tracking, video surveillance, self-driving vehicles). One of the most advanced research area for phases recognition is in surgical robotic systems (SRSs). Automatic gesture recognition during surgical procedures is an enabling technology for developing advanced assistance features in SRSs. In SRSs, similarly to I4.0 settings, heterogeneous data sources are used, ranging from robot kinematics to machines status and operator tracking systems. Automatic surgical gesture recognition and classification are of paramount importance to enable the forecast of possible dangerous situations and suggest corrective actions before their critical consequences [72]. Moreover, automatic process recognition allows the evaluation of operators online during the procedure, improving the training process and overall performance thanks to surgeon-specific feedback and prompt detection of excessive fatigue and cognitive overloading [73]. In robotic surgery, as in the industrial process, the subdivision of procedure into sub-tasks (i.e., gestures, maneuvers, action) from heterogeneous data is required. Breaking down the process into tasks allows measuring local metrics facilitating the

management of the process (in the industrial case) or the procedure (in the surgical environment). Below a description of the state-of-the-art methods used in the context of SRSs is drawn.

One of the first works in surgical procedure segmentation dates back to 2013 when Zappella et al. proposed gesture recognition from video data using linear dynamical systems (LDSs) and bags of words (BoW) [73]. Kinematic data acquired from robotic manipulators provide accurate measurements for gesture recognition since these data directly correlate to the surgeon's commands and movements of instruments. Kinematic data have been firstly modeled with Hidden Markov Model (HMM) [74], then other approaches were proposed with Condition Random Field (CRF) and skip-chain CRF (SKCRF) [75, 76]. SKCRF classification results are influenced by selected kinematic values and their representation (i.e., relative distances and velocities between objects and instruments), as demonstrated in [77]. Recently, Deep Neural Networks (DNN) have emerged as powerful feature extractors that can be learned automatically from the data, in contrast with hand-crafted filters based on domain-specific knowledge [78]. In DNNs, layers of computational units are stacked using multiple kernels to extract features patterns at different frequencies/resolutions. In addition, specific DNNs have been developed to capture temporal dependencies. Recurrent Neural Networks (RNN) can obtain long-term non-linear dynamics in surgical kinematic data [79]. In-depth analysis and comparison of different architectures (e.g., simple Bidirectional RNNs, Bidirectional Long Short Term Memory (LSTM), Bidirectional Gate Recurrent Units (GRUs) and Bidirectional Mixed History RNNs) showed that LSTMs and GRUs are less sensitive to hyper-parameter choice and achieve the best recognition performance [80]. Recent results indicate that convolutional architectures can outperform recurrent networks [81]. A systematic evaluation of generic convolutional and recurrent architectures for sequence modeling indicates that a simple convolutional architecture outperforms recurrent canonical networks such as LSTMs across diverse tasks and datasets while demonstrating longer effective memory. One of the most successful convolutional networks for action modeling is Temporal Convolutional Network (TCN) [76]. With TCN, the convolutions in the architecture are

causal, meaning that there is no information "leakage" from future to past [76]. This thesis proposed a modified version of TCNs, which consists of time-delay neural networks (TDNNs). Our proposed method for phase recognition will be presented in Chapter 3.

2.1.2 Decomposition method for industrial processes

From an I4.0 perspective, many works have been presented for phase recognition in the area of collaborative robots safety. Safety is one of the significant advantages of using co-bots instead of classic industrial manipulators because they do not require cages and allow operators to collaborate inside the work cell. In a recent review on Human–Robot Collaboration (HRC), the authors analyze and discuss the HRC trends within the smart manufacturing environment and how phase recognition methods contribute towards enabling safe and efficient HRC [82]. They highlighted the following usage of recognition and classifications techniques:

- *Human detection* Detecting the presence of humans can be useful to adapt the movements of the robot and ensure safety;
- *Training operators for HRC* Recognizing the phases of a process allows to improve operators' skills, highlighting the most difficult tasks and providing data insight;
- *Force detection* Recognizing the force applied by operators allow to avoid a harmful collision;
- *Human physiology recognition* Allows the study of ergonomics and mobility in the workplace;
- *Motion planning* Optimizes the path of movement of robots by increasing safety or reducing their consumption;
- *Gesture recognition* Allows to recognize a movement and highlight the correct execution of the task;

- *Speech recognition* Allows to describe dangerous situations or intervene through voice commands to control the robot from distance;

All of these algorithms aim to recognize phases and situations during an industrial collaborative process. For a more detailed description of HRC refer to [83, 82].

Other works focus on the decomposition of the process at the plant level regardless it contains HRC. These methods aim to infer the propagation and the production flow (e.g., error propagation in a plant). A plant-wide decomposition process does not necessarily use all the available data since some parts do not interact with others. Therefore, a commonly used modeling strategy is firstly to decompose the whole process into different phases [84]. Knowledge-based methods were initially used in the subdivision of the process into phases (i.e., block decomposition). Later, the first data-driven approach for decomposing the process was based on principal component analysis (PCA). Recently, a new multi-block method uses mutual information technique to recognize different blocks in plant-wide processes taking into account both linear and non-linear relations [85]. After the division, the PCA-based fault detection and the root cause diagnosis are performed in each sub-block. In [86] a fault detection method based on distributed canonical correlation analysis is used to infer blocks using correlation information of the neighboring nodes. Yanan et al. [87] propose a community discovery (CD) algorithm with Bayesian inference and principal component analysis (PCA) for a plant-wide process monitoring scheme. For a review on process decomposition methods, refer to [84].

To summarize, we presented the state-of-the-art for recognizing process phases for both: plant-wide and collaborative task scenarios. Then, we highlight the methods used in SRS, one of the main area of research for task recognition and process segmentation. Next, we report the researches done in the industrial contest for HRC and process block decomposition. In the next paragraph, we will present the state-of-the-art for regressive models.

2.2 Forecasting models

One of the most common purpose towards regressive models is the prediction in the output $y(t)$ of the formula 1.3.1 given the past values $x_j(s)$ while estimating the parameters $\beta_j(s,t)$. This process is called forecasting or prediction. Indeed, the forecast capability of a method depends on how well a given model maintains its structure beyond a single collection of data [88]. Thus, monitoring model parameter stability over time is an essential element in evaluating a forecasting model. Stability, which leads to predictability, is related to how well the system preserves its behavior through time [47]. In time series forecasting, data are obtained at regular time intervals and are used to predict future patterns. The simplest form of forecasting is moving average, which uses an average of past patterns to forecast future patterns [88]. To forecast the time series value at time t an average of past values on a sliding window δ is applied. An adequate width for δ has to be selected by the modeler. Small δ implies minor weight from recent past information while a large δ is more suitable for slow changes over time. Exponential smoothing models trade the influence on prediction between recent and past observations via a smoothing constant. The formula for exponential smoothing is shown below, where y_{t+i} is the forecasted value, a is the smoothing constant, and X_t, $X_t - 1$,... X_{t-n} are past observations in a time-series of length n [88]:

$$y_{t+i} = aX_t + a(1-a)X_{t-1} + a(1-a)^2X_{t-2} + a(1-a)^3X_{t-3} \cdots + a(1-a)^nX_{t-n} \quad (2.2)$$

Exponential smoothing is convenient when recent data are more informative than older data for the time series, or the information is not equally distributed in δ.

A typical time series behavior is autocorrelation, i.e., the future value of actual observation is related to the previous value at a specific delay or lag. The autocorre-

lation formula is shown below:

$$Autocorr(m) = \frac{\sum_{i=1}^{N-m}(X_i - X_{mean})(X_{i+m} - X_{mean})}{\sum_{i=1}^{N}(X_i - X_{mean})^2} \tag{2.3}$$

Where $X_1, X_2, ..., X_N$ is a series of observations, X_{mean} is the mean of the observations, and m is the delay as past instances computed. ARIMA (AutoRegressive Integrated Moving Average) models combine autoregression and moving average models. In an ARIMA model, the future value of a variable is assumed to be a linear function of several past observations and random errors. Following notation in [89], the underlying process that generates the time series of length n has the form

$$X_n + a_1 X_{n-1} + \cdots + a_p X_{n-p} = \varepsilon_n + b_1 \varepsilon_{n-1} + \cdots + b_q \varepsilon_{n-q} \tag{2.4}$$

where ε_n is a purely random process and a, b are constant. This $ARIMA(p, q)$ process becomes AutoRegressive for $q = 0$ and MovingAverage for $p = 0$. p and q are integers where $p, q \in \mathbf{N}$ and are often referred to as orders of the model. One of the main challenges for ARIMA is to determine the appropriate values for p and q. ARIMAX models extend ARIMA models through the inclusion of exogenous and multivariate variables. Therefore, ARIMAX can consider a set of variables described in the process or even external to the production to predict the response variables. However, these linear models are only a coarse approximation of real-world complex systems and generally fail to predict the evolution of non-linear and non-stationary processes [90]. Most of the techniques developed for time series analysis and prediction assume stationary data, i.e., mean, variance, and autocorrelation of the time series do not change over time. For example, considering outdoor production factory, the temperature of the final product is affected by external temperatures varying with the seasons, thus, the time series of product temperature will not be stationary during the year. Moreover, ARIMA/ARIMAX models are linear and many real-world relationships are instead non-linear, therefore, other models have been proposed. The

Dickey-Fuller test allows checking if the data are non-stationary [91], however, to control non-stationarity a careful adjustment on the data should be carried out during preprocessing. For example, the data can be standardized separately for each season or the dataset can be split into multiple parts. However, adjusting for non-stationarity is hard and it is highly dependent on the data analyzed. To handle non-linearity, in [92] a Threshold AR model (TAR) is proposed that assumes piece-wise linearity. Non-overlapped partition of the input space can be specified in terms of a threshold variable.

Another approach to modeling non-linearity is to transform non-linear spaces into linear spaces. It project data into another dimension so that the data can be classified. SVM-based forecasting methods use a class of generalized regression models, such as Support Vector Regression (SVR) and Least-Squares Support Vector Machines, that are parameterized using convex quadratic programming methods [93]. An SVM maps the inputs $\mathbf{X}$ into a higher-dimensional feature space $\phi(\mathbf{X})$. If $\phi(\mathbf{X})$ entails specific behaviors from the data to the response variable, then the forecasting model will obtain better accuracy. Ensembles methods, generate multiple mapping in the higher-dimensional space $\phi(\mathbf{X})$, adding also randomicy. Then the new space with the most probable output is used.

An ensembles random forest is a collection of tree predictors $h(\mathbf{X}; \Phi_i), I = 1, ..., i$ where $\mathbf{X}$ represents the observed input vector and Φ_i are independent identically distributed random vectors where for each tree casts a unit vote for the most popular class at input X_i [94]. The random forest prediction is the unweighted average over the collection: $\hat{h}(x) = (1/I) \sum_{k=1}^{I} h(\mathbf{X}; \Phi_k)$. While the ensemble methods as random forest are remarkable for providing latent features spaces that describe patterns in the data, the struggle with the curse of dimensionality due to the prohibitive computational effort, memory requirements, and large data sizes hamper their applicability to many real-world problems, especially for online process monitoring [90].

Forecasting methods that avoid generating random vectors in latent space are Hidden Markov Models (HMM). An HMM where the observed time series $y(t)$ (i.e. the response variable from formula 1.3.1) is treated as a function of the underlying, unob-

served states vector X_t. Following notation in [95] a state vector may be reconstructed from autoregressive terms of $y(t)$. An HMM is composed of two state sets and three probability matrices [95]. Two sets are the hidden-state set, $H = \{h_1, h_2, \ldots, h_n\}$, and the observable state set $O = \{o_1, o_2, \ldots, o_m\}$, where hidden states are related to the observable states, and n and m are the number of hidden states and observable states, respectively. The probability matrices are used to describe the relation between hidden and observable states and normally represented asa parameter set (π, A, B) of the HMM. π, A, and B can be defined mathematically as follows:

$$\pi = [\pi_i], \pi_i = \mathrm{P}((h_i, t)), \quad 1 \leq i \leq n \tag{2.5}$$

$$A = [a_{ij}], a_{ij} = \mathrm{P}((h_j, t)|(h_i, t-1)), \quad 1 \leq i, j \leq n \tag{2.6}$$

$$B = [b_{ij}], b_{ij} = \mathrm{P}((o_j, t)|(h_i, t)), \quad 1 \leq i \leq n, \quad 1 \leq j \leq m. \tag{2.7}$$

π is the probability vector of the initial state, where π_i is the probability of the hidden state h_i occurring at the initial time step $t = 1$. The $n \times n$ state-transition matrix A provides the probability about the relation of two contiguous hidden states from time $t-1$ to t. The $n \times m$ matrix B is the confusion matrix, which characterizes the probability of observing a state o_j given the hidden state h_i at time t. Therefore, a particular HMM can be characterized by $\mathrm{P}((h_i, t)|(h_x, t-1))$ and $\mathrm{P}((o_y, t)|(h_i, t))$.

2.2.1 Forecasting with Neural Networks

In recent years, thanks to strong investment and advancement in computation technology together with the support of open-source frameworks such as Tensorflow and Pytorch [96, 97], models that map inputs in bigger features spaces have been more successful than traditional forecasting methods (e.g., ARIMA or HMM). The development of these models culminated with the extensive use of artificial neural networks (ANN). ANNs can be understood as a non-linear mapping method, which is inspired by the human brain [98]. The common supervised learning algorithm used for ANN is based on the following steps: present training data to the neural network and de-

termine the error between its output and the expected output on the selected input. Next, the parameters of the network are tuned to minimize the error. The most common algorithm for training is the Back-Propagation (B-Prop) algorithm [99]. The B-Prop algorithm is a gradient-based optimization algorithm that minimizes the output error for a learning data set [100]. Iterating error minimization with parameter optimization using the back-propagation algorithm defines the training phase of the network. The interaction ends when a threshold is reached, which can be placed in the number of interactions or the error's magnitude. The model obtained is then validated on new data that have been used in the training phase.

In this paragraph, an overview of neural network architecture for forecasting is drawn. We review the most relevant types of supervised deep learning (DL) networks, dividing them into feed-foward fully connected neural networks, recurrent neural networks, and convolutional networks.

- **Feed-forward Neural Network models (FNNs)** FNNs parameterized with a back-propagation algorithm have been employed for non-linear time series forecasting. Multi-Layer Perceptron (MLP) is the most basic type of feed-forward ANN. FNN architecture comprises a three-block structure: an input layer, hidden layers, and an output layer. Hidden layers are composed of nodes. MLP networks only have forward connections between neurons at different layers. A deeper neural network is generally able to model more sophisticated patterns at the expense of interpretability [101]. Features are created from a linear combination of the input, while the activation function that models the transitions between layers adds non-linearity to the model. The most common activation functions are: sigmoid, tanh and rectified linear unit (ReLU) [102]. An MLP network with a non-linear activation function can represent any non-linear model with only one hidden layer [103]. However, the success of neural networks is related to the combination of several layers and activation functions in specific architectures that addresses pattern in the data [103]. Similarly to SVM, each hidden layer represents latent variables in an augmented space.

- **Recurrent neural networks (RNNs)** In these ANNs, the temporal problem is transformed into a spatial architecture able to encode the time dimension, thus effectively capturing the underlying dynamical patterns of time series. RNNs were introduced as a variant of ANN for time-dependent data. The network recognizes one observation at a time and can learn information about the previous observations and how relevant the actual observation is to forecasting. Therefore, the network not only parametrizes relations between input and output but also between output and a sequence of past observations thanks to the parametrization of the internal memory. This characteristic makes RNNs one of the most common ANN used for time-series data. Following, we will present three types of RNN: Long Short Term Memory (LSTM), Gated Recurrent Unit (GRU), and Bi-directional GRU/LSTM.

  - LSTM: LSTMs can model temporal dependencies on larger horizons without forgetting, thus, mitigating the vanishing gradient problem. A multiplicative input gate unit is introduced to protect the memory contents (containing past observation) from perturbation by irrelevant inputs. Likewise, a multiplicative output gate unit is introduced to protect other units from perturbation by currently irrelevant memory contents [104]. A typical LSTM cell is configured mainly by four gates trained as weights: input gate, input modulation gate, forget gate and output gate. Input gate takes a new input from outside, i.e., processes newly coming data. The memory cell input gate takes input from the output of the LSTM cell in the last iteration. Forget gate decides when to forget the output results and thus selects the optimal time lag for the input sequence. Output gate takes all results calculated and generates output for the LSTM cell. A linear regression layer is applied to the output layer of the LSTM cell. For a full description of LSTM cell refer to [104]

  - GRU: A typical GRU cell is composed of two gates: reset gate r and update gate z. Similar to LSTM cell, hidden state output at time t is computed

using the hidden state of time $t-1$ and the input time series value at time t. In [105] a comparison between GRU and LSTM is drawn. GRU's has fewer tensor operations; therefore, they are a little faster to train than LSTM's.

- Bidirectional LSTM/GRU: Neither LSTM nor GRU can encode information from back to front. This feature allows the model to learn relationships in a-causal settings, supposing that the output is known for training the network. Bi-directional long short-term memory (Bi-LSTM) and Bi-directional gated recurrent unit(Bi-GRU) solve this problem. They are formed by stacking forward and backward LSTM or GRU to better capture bidirectional semantic dependence. Bi-LSTM or Bi-GRU is usually better than LSTM or GRU, but the training will be more time-consuming.

- **Convolutional Neural Network**

  - Convolutional Neural Networks (CNNs) are a family of ANNs considered as the state-of-the-art for many problems in classifications and pattern recognition. CNN learns to extract meaningful features from the data using the convolutional operation, maintaining features invariance propriety [106]. Unlike standard MLP networks, each node is connected only to a range of the input, which is known as the receptive field. The receptive field is regulated by the kernel dimension. These networks are based on three principles: local connectivity, shared weights, and translation equivariance [107]. A more detailed description is provided in section 3.2. These special properties allow CNNs to have a much smaller number of trainable parameters compared to a RNN [81].

  - Temporal convolutional network architecture is inspired by the Wavenet autoregressive model, originally designed for audio generation problems [108]. The convolutions are causal to prevent leakage from future features, and the architecture can process a sequence of any length and map it to an output of the same length. The TCN architecture is based on dilated causal

convolutions to enable the network to learn the long-term dependencies present in time series. TCNs are explained in more detail in section 4.3.1.

2.2.2 Predictive models in manufacturing

Many forecasting models focus on manufacturing data. In [109] a "smart process analytics" framework for predictive models is presented, which empowers the users to focus on goals rather than on methods and automatically transforms manufacturing data into intelligence. The authors suggest using different forecasting methods depending on the type of historical data instead of model proprieties. In particular, they divide the characteristics of the data according to non-linearity, collinearity and dynamics. They suggest different models for each group, demonstrating their effectiveness with case studies for experimental datasets from a variety of process systems.

Regarding the use of neural networks several architectures have been proposed in order to obtain a more advanced feature extraction. In [110], authors propose a TCN integrating regularly updated multi-region operations based on principal component analysis and hierarchical clustering for an industrial methanol production process. The hierarchical clustering method extracts hidden temporal multi-region features to enhance a TCN. A combination of CNN and GRU is applied to extract spatio-temporal features from the supervisory control and data acquisition (SCADA) of wind turbines in [111]. In [112], the authors present a Multivariate-TCN that constructs a sub-model for each feature in the input data and the overall prediction is accomplished by a combination of all sub-models.

2.3 Causal models

Differently from the previously presented topics (i.e., process monitoring, phase recognition and temporal modeling), causality has origin from the philosophy with authors such as Aristotle, Ovid, Hume and many others [49]. Causality is a topic addressed in medicine, economics and social sciences and it is a broad concept investigated with

multiple approaches in different fields. For example, in physics, causality is related to the speed of light and direction of time [113], while in medicine, it is related to the effectiveness of drugs or harms induced by interventions [114]. Indeed, the following state-of-the-art focuses on methods for inferring causality from data using the computer science approach. There are currently two main causal frameworks that have been shown equivalent in computer science [115]: the potential outcome and the structural causal model framework respectively developed by Rubin and Pearl [2, 116]. The potential outcome is used by practitioners to learn causal relationships observing treatment and its effect on the outcome. Nevertheless, Gylmour in [115] states that the potential outcomes framework is essentially a special case of a structural causal model (SCM). The SCM is a framework that allows modeling three levels of hierarchical abstraction also called the ladder of causality [117]:

1. **Association:** Association is the ability to extract statistical relationships from data. This type of correlation is estimated with probabilistic models, usually inferred with conditional probability. Theoretically, with infinite data, associations can be inferred optimally;

2. **Intervention:** With intervention, we mean the ability to intervene on the system and observe the results. Conditioning on a variable is different from intervening. Conditioning belongs to the probabilistic framework, while intervention belongs to causal theory. Conditioning is related to the system's structure, while when an intervention is performed, the underlying causal structure is changed. The intervention breaks the control on the target variable of the intervention and sets a constraint from that point on. For example, during the code execution, an intervention manually sets a variable to a specific value; therefore, the value for that variable will not be affected by the system. This type of abstraction is performed with do calculus [116]. It requires the possibility to intervene in the system and observe the outcome.

3. **Counterfactual:** Counterfactuals hypothesize an intervention and evaluate the result even if it is not happening. Therefore, the intervention is carried out in

a hypothetical environment rather than on the actual system. In other terms, counterfactual answer to the question "what would have happened if"? A type of counterfactual question could be, for example: Given that the pandemic has occurred and overseas vacations have been canceled, if the pandemic had not arisen, where would you be on vacation? This type of abstraction is obtained with the functional relations or structural [118].

Regarding process monitoring with causality, in the introduction, we describe the concept of prognosis. We state that prognosis is the feature of process monitoring that seeks to describe the process also considering the connection between different phases. A causal model is able to describe the causal relations between variables and infer a flow that follows the product rather than the time. Thus, the concept of causality is much broader than correlation and describes a higher lever of abstraction. It uses association between the quality features of the product as well as interventions made by the operators or hypothetical outcomes.

2.3.1 Structural causal model

Structural causal model is a mathematical framework for modeling causality with multiple variable under certain assumptions. Following notation in [119] a SCM ϱ with graph $\textcircled{C} \rightarrow \textcircled{E}$

$$C := N_c, \tag{2.8}$$

$$E := f_E(C, N_E), \tag{2.9}$$

where $N_E \perp\!\!\!\perp N_C$, that is N_E is independent of N_C. This model show the random variable C as cause and the E as effect. C is a direct cause of E and the causal graph $\mathbf{G}$ is represented with $\textcircled{C} \rightarrow \textcircled{E}$. The use of the assignment operator := makes explicit the asymmetric nature of these equations. In other words, they are not to be rearranged to solve for their inputs [120]. One of the advantages of SCM is that they can be represented by graphical models. This representation is done through direct acyclic graph (DAG). Pearl proposed a graphical model called Bayesian network in

which each node represents a variable, and the arrows represent dependencies. Thus, the graph entails a joint probability distribution across the variables. Causal graphical models entails also intervention denoted with the *do* notation. Following notation in [120] we give the definition of causal graphical models:

Given $\mathbf{X} = (X_1, ..., X_d)$ random variables a graph $\mathbf{G}$ and a collection of function $f_j(x_j, x_{\mathbf{PA}_j^G})$ that integrates to 1, these function induce a distribution P_X over $\mathbf{X}$ via

$$p(x_1, ..., x_d) = \prod_{j=1}^{d} f_j(x_j, x_{\mathbf{PA}_j^G}) \tag{2.10}$$

Where p have strictly positive, continuous density, $\mathbf{PA}$ are the parents of the node and P_X over $\mathbf{X}$ is Markovian. The previous equation allows us to define the independencies between the graph nodes and associate them with conditional distributions. This assumption allows using the D-separation rule to define independencies and causal relationships between variables stating for independence. For Reichenbach's principle, indeed, if two random variables X and Y are not independent then there exists a third variable Z that causally influences both. (As a special case, Z may coincide with either X or Y.) Furthermore, this variable Z screens X and Y from each other in the sense that given Z, they become independent, $X \perp\!\!\!\perp Y|Z$.

To deduce causation with graphical models the assumptions on faithfulness and causal minimality must be satisfied. The two proprieties of the system are describe as follow:

Consider a distribution P_X and a DAG $\mathbf{G}$.

1. P_X is faithful to the DAG $\mathbf{G}$ if

$$A \perp\!\!\!\perp B|C \rightarrow A \perp\!\!\!\perp_G B|C \tag{2.11}$$

   for all disjoint vertex sets A, B, C.

2. A distribution satisfies causal minimality with respect to $\mathbf{G}$ if it is Markovian with respect to $\mathbf{G}$, but not to any proper subgraph of $\mathbf{G}$.

Causal faithfulness relates statements about conditional independence that hold in the graph structure. It states that a set of independencies that hold in the distribution, can be extracted from the structure of the graph. This assumption can be violated if there is a causal chain with opposite coefficients that cancel each other out.

Causal Sufficiency states that "for every pair of variables which have their observed values in a given data set, all their common causes also have observations in the data set". This condition defines that a pair is causally sufficient if all the common causes of a pair of variables are measured, meaning that there are no hidden causes. If the data is obtained in a closed system, we can justify that there are no latent variables [121].

Extending SCM to time-series is fairly straightforward, supposing that the temporal process is stationary and that the above assumptions are met. Indeed, the causal structure is consistent with the temporal order. Special attention should be given to time series with instantaneous effects as they entail instantaneous cause-effect relationships and the assumption on causal Markov condition may not hold. However, with a careful choice of sampling rate on the data, we can discretize each time series with instantaneous effects into a time series without instantaneous relationships by oversampling. The key concept behind time series SCM is that the causal relationship is repeated over time. Figure 2-1 shows a causal graph representation in time series with instantaneous effect (red arrows). A formal extension of SCM to time series is done in [120].

2.3.2 Causal discovery for time-series

The field of causal discovery or structure learning aims to find causes in joint distributions and to infer a causal structure from the data (causal learning). There are multiple approaches to causal discovery. Arguably one of the best methods is to use randomized controlled experiments (RCE) that isolate the desired behaviors in controlled environments. Others methods involve performing an intervention on the system (treatment) and checking for different outcomes (effect). These methods, however, assume having control of the system, which is not always feasible. For ex-

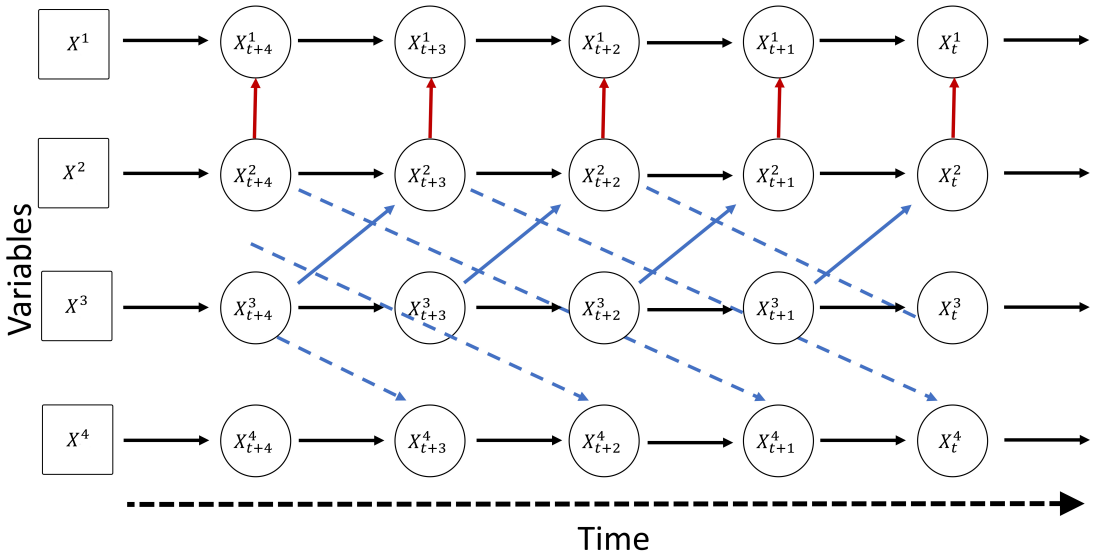


Figure 2-1: Time-series causal graph models. Red arrows represent instantaneous effect, blue arrows entails causation between variables at different time lag.

ample, while working with a company's historical data, it is not possible to make new interventions on data since the process has already occurred. Moreover, even if it is possible to perform the intervention and record the outcomes while the production is running, it may be too expensive for the company as it requires wasting products and resources.

This led to the development of methods based on causal discovery from observational data. In the literature we can find numerous reviews for causal inference from data [122, 123]. Most of these divide causal discovery methods into four types: methods that use prediction improvements, those exploiting structural asymmetries, score-based, and constraint-based.

1. **Prediction improvement methods:** These types of models use outcomes' prediction to estimate the influence of the cause variables under the aforementioned assumption. To this class of methods belongs Granger causality. The idea of Granger causality is that X has a causal influence on Y if the prediction of Y from its own past is improved by additionally accounting for X [119]. That is:

$$X\ Granger - Cause\ Y : \Longleftrightarrow\ Y_t \not\perp\!\!\!\perp X_{past(t)} | Y_{past(t)} \tag{2.12}$$

which means that X has a causal influence on Y if the prediction of Y from its own past is improved by additionally accounting for X. Even if Granger causality is based on prediction, Eichler in [124] gives a theoretical justification by relating the concept to other theoretical causality measures. To measure the quantity of information added by the time series X on Y transfer entropy can be used [125]. Moreover, in literature, it has been proposed extension to non-linear and multivariate models [126]. This method is the most widely used in time series casual discovery and among the oldest, but it has limitations. It is necessary to find the correct delay between variables to obtain a meaningful causal prediction. Moreover, since external variables may influence multiple observed variables, to affirm causality, it is necessary to condition all potentially influential variables. Convergent cross-mapping (CCM), instead, assume that interactions occur in an underlying dynamical system and attempt to uncover causal relationships based on Takens' theorem and non-linear state-space reconstruction. A causal relationship between two dynamical variables X and Y can be established if they belong to a common dynamical system, which can be reconstructed from time-delay embedding of each of the observed time series. More specifically, if variable X can be predicted using the reconstructed system based on the time-delay embedding of variable Y, then we know that X had a causal effect on Y [127]. Therefore, while CCM is preferred for non-linear system with few variable, Granger causality is suitable for heterogeneous time series with linear relationships. In 2019, a challenge on causal structure learning from time-series data (Causality 4 Climate) was presented at the Conference on Neural Information Processing Systems 2019 (NeurIPS) [128]. Subsampled Linear Auto-Regression Absolute Coefficients (SLARAC) and Quantiles of Ridge regressed Bootstrap Samples (QRBS) are the two algorithms that won the competition and they both use prediction improvements. These methods regress present on past values and inspect the regression coefficients to decide whether one variable is a Granger-cause of another. SLARAC fits a VAR model on bootstrap samples of the data each time choosing a random number of lags

to include; QRBS considers bootstrap samples of the data and Ridge-regresses time-deltas $X(t)-X(t-1)$ on the preceding values $X(t-1)$; Even these methods are linear since they use OLS they obtain the best performance on non-linear datasets [129]. Causal Discovery Framework (TCDF) uses attention-based convolutional neural networks combined with a causal validation step. TCDF can also discover the time delay between a cause and its effect by looking at the attention weight of the convolutional networks. It can include confounders and instantaneous effects. Moreover, the attention weight estimate the strength of the causal link [130]. The advantages of TCDF relay on the relaxed assumption on the data and the uses of deep neural network to exploit dependence.

2. **Asymmetry methods:** Asymmetry methods test which nodes are more likely to be a cause or effect using asymmetry in the distributions. These asymmetries can be exploited in multiple ways. If some dependant noise in added to one variable, if this variable is a cause it will propagate to its effect variable. While if the variable is an effect, the system is not going to report any changes to the cause variable. However, these types of methods are generally difficult to scale to many variables. Linear non-Gaussian Acyclic models (LINGAM) use the independence between the noise and cause, which holds for only one direction implying asymmetry between cause and effect. This method is based on functional causal models. However, when both noise and variable are gaussian, the model is not identifiable as no asymmetry arises. This method moreover works with linear relationships on a small number of variables. Time Series Models with Independent Noise (TIMINO), similarly to LINGAM, require independent residual time series and use the additive model with structural equation models to find non-linear dependencies. It handles hidden confounders by staying undecided instead of inferring any (possibly incorrect) causal relationship, but it cannot scale to large numbers of variables [131].

3. **Score methods:** Score-based methods search over the space of possible graphs trying to maximize a score function that reflects the most suitable graph to fit

the data. This score is typically related to the likelihood of the graph given the data. However, the number of possible graphs is super-exponential to the number of variables. In [122] the objective function is descrived as:

$$\tilde{G} = argmax_{G\,OVER\,X} S(D, G) \tag{2.13}$$

where D represents the empirical data for variables X, G the graph and S is the scoring function. Greedy equivalent search (GES) is one of the most used score-based algorithms []; instead of exploring the optimal DAG, it chooses a node and analyses possible neighbors. Then, it keeps adding dependencies between nodes until it reaches a maximum for each node. In the second step, it removes dependencies and stops at an equivalence DAG. Bayesian information criterion (BIC), Z-score or statistical hypothesis give the score for selecting neighbors. Therefore it starts from an empty set of edges and increases the Markov equivalence class iteratively. As for FCI, many variations of GES can be found in literature [132]

4. **Constraint methods:** Constraint based method uses independence and dependence constraints obtained from statistical tests to narrow down the candidate graphs that may have produced the data. They use independence tests to remove possible false causal relationships and orientation rules to create a subset of potential causal graph structures. These constraint-based causal discovery methods can allow for the presence of latent confounders, feedback cycles and the utilization of several (partially overlapping) observational or experimental data sets [123]. Constraint-based methods can infer some causal orientations on the basis of v-structures (unshielded colliders). The analyzed series can then be represented as a DAG. However, not all DAGs have causal value. We refer to [122] for the differences between DAGs and causal DAGs. One of the first methods proposed for constraint based causal discovery is the Peter Clark (PC) algorithm [133]. PC provides a search architecture that uses statistical procedures. The PC algorithm is guaranteed to converge to the valid Markov

equivalence class (i.e., a set of possible causal graphs) assuming, in addition to the properties described in the paragraph 2.3.1, the absence of unmeasured confounders. The return of the PC algorithm is a completed partially direct acyclic graph (CPDAG). It is composed of two steps. In the first step, it learns from data a skeleton graph, which contains only undirected edges. In the second step, it orients the undirected edges to form an equivalence class of DAGs. In section 4.1.1, a more in-depth explanation of the algorithm will be provided. Following the Fast Causal Inference (FCI) algorithm have been proposed [134]. FCI evolves from PC but allows the presence of unobserved variables and thus does not assume causal sufficiency. It returns a maximal ancestral graph (MAG) instead of a CPDAG that uses bidirectional edges. Different versions of FCI have been proposed to optimize the algorithm computations [134]. FCI compared to PC thus optimizes speed and assumption recruitment. Other works propose algorithms to relax the assumption on the data. In [135], authors propose a Constraint-based causal Discovery from heterogeneous/NOnstationary Data (CD-NOD) algorithm. CD-NOD is a nonparametric causal discovery method with no hard restrictions on data distributions and causal mechanisms and that does not rely on window segmentation. CD-NOD introduces a surrogate variable into the causal system to characterize hidden quantities that lead to the changes across domains or over time. Including a variable in the causal system provides a convenient way to unpack distribution shifts to causal representations and, therefore, to adjust for non-stationarity or multiple domains. Finally PCMCI is a causal discovery constraint based method for large-scale time series datasets. PCMCI evolve from PC alghoritm as FCI but it allows a strong control on false positive thanks to a validation step with Momentary conditional independence between lagged causal parents. PCMCI is explaned in section 4.1.1.

In Table 2.1 we report the above methods with the corresponding characteristics. Research in causal discovery is very active and numerous algorithms are proposed every year with different goals as relaxed assumptions, scalability or accuracy. We

refer to the following literature reviews for a more extensive description of causal discovery methods [122, 121, 125, 136].

Method	Type	Sufficency	Faithfulness	Large multivariate time series	Multi-Domain	Output
PC	Const.	Yes	Yes	Yes	No	CPDAG
FCI	Const.	No	Yes	Yes	No	PAG
GES	Score.	Yes	Yes	Yes	No	CPDAG
LiNGAM	Asym.	Yes	No	No	No	DAG
TiMINo	Asym.	Yes	No	No	No	DAG
TCDF	Pred.	Yes	Yes	Yes	No	Binary
SLARAC	Pred.	Yes	Yes	Yes	No	Binary
QRBF	Pred.	Yes	Yes	Yes	No	Binary
CCM	Pred.	Yes	Yes	No	No	Binary
CD-NOD	Const.	No	Yes	Yes	Yes	CPDAG
PCMCI	Const.	Yes	Yes	Yes	No	DAG
Granger	Pred.	Yes	Yes	Yes	No	Binary

Table 2.1: Summary of causal discovery methods presented in section 2.3.2. Provides indication of assumptions of sufficiency, faithfulness as well as whether the method is suitable for high dimensional data. The output can be Binary (cause-effect), CPDAG (completed partially directed acyclic graph), DAG (directed acyclic graph) and partial ancestral graph (PAG).

In this thesis, we used PCMCI causal discovery method. We choose a constraint-based method as it is most suitable for our case study. Constraint methods apply structural causal models that rely on independence instead of the Granger causal approach based on prediction. Since our goal is to use the causal model as a precursor for a monitoring system, using a prediction-based approach could have reduced the advantages of testing independence in the causal graph and dependence with the regressive model. We discarded score methods based on minimizing a function that could lead to local models. A local model could have introduced a suboptimal equivalent class. The methods based on asymmetries between cause and effect were unable to satisfy the scalability of many variables, which is an important feature in an I4.0 manufacturing plant perspective.

Compared to other constraint-based methods, PCMCI allows an additional control on false positive. This means that opposed to PC, CD-NOD and FCI, it allows recreating the process flow without erroneous phase ordering. As explained in section 1.3.1, from the perspective of developing a monitoring system, reducing erroneous

causation is more critical than detecting all possible relationships. Furthermore, multi-domain flexibility is not a feature of interest for the predictive model. If the domain is changed (e.g., through factory reconfiguration), the predictive model will need to be adapted to the new domain making multi-domain adaptation unnecessary. On the other hand, the causal sufficiency assumption remains a limitation for PCMCI as it means that, for monitoring the product, all variables affecting the process must be considered. Especially in SMEs, it is possible that some variables affecting product characteristics (e.g., external temperature) are not monitored, and it may lead to incorrect identification of cause and effect variables. Nevertheless, as reported in section 4.1.1, fo our case scenario PCMCI is the most suitable algorithm for causal discovery. We apply causal discovery to reconstruct the flow of a process. The rigorousness of the constraint-based methods and the additional control on the causal parents allows reaching high levels of accuracy as demonstrated in the experiments in section 4.1.2. In addition, for PCMCI, a unique framework for time series has been provided, which efficiently manages the computation of causal relationships at different delays.

2.4 Conclusions

This chapter initially presents the state-of-the-art methods adopted in process phases recognition. The problem is represented as a classification task that can employ supervised, semi-supervised or unsupervised machine learning. We subsequently focused on supervised methods used in Surgical Robotic Systems as they represent an area of excellence for recognizing the process phases. Then, we focus on state-of-the-art methods applied specifically in the I4.0 context. Therefore, we highlighted the recognition of several operations during a human robotic procedure and the decomposition of plant-wide process phases in the I4.0 domain.

In the second section, we presented state-of-the-art predictive models describing the different challenges addressed. We focused on the use of deep learning, which is widely used to solve forecasting problems given the capability of modeling com-

plex non-linear relationships. We presented various architectures for neural networks such as convolutional networks, feedforward networks and recurrent networks. Subsequently, we presented some research on forecasting applied in the manufacturing sector.

Finally, in the last section, we introduced the concept of causality by defining structural causal models. We have highlighted the assumptions that must be considered to infer causal discovery from the data. We focused on causal discovery for time series by displaying state-of-the-art algorithms. In section 2.3.2 we highlight the advantages and limitation between the the various methods for causal discovery on .

Chapter 3

Phase recognition with Time Delay Neural Network

In this chapter, we introduce a novel method for industrial process phase recognition. We design the recognition of phases as a classification task from a heterogeneous set of signals. The contribution given in this chapter is to apply a supervised deep learning method that uses a convolutional architecture to process time series with distant temporal recurrences. We validated the method on two datasets specifically created to replicate industrial processes adopting HRC and a third dataset used as a benchmark in the context of phase recognition. As a result, we achieve higher accuracy by recognizing characteristic movements and actions performed in each phase.

The chapter is organized as follows: in section 3.1 we describe the recognition of phases as a supervised classification problem. In section 3.2, we present the time delay neural network (TDNN) model; we display the TDNN architecture and the differences with a standard convolutional models. The experiments performed are outlined in section 3.3. In the same section, we present the datasets, the evaluation methodology and the metric used. Following, we display the results obtained, discussing them in detail. The conclusions of the chapter are made in section 3.4.

3.1 Automatic process phase classification

In continuous manufacturing processes, the raw materials advance through several phases to create the final product as reported in section 1.3.1. Therefore, distinguishing between phases becomes necessary to follow the product's progress. The advantages of dividing the process into phases are multiple. For example, it allows to describe the process, compute explicit metrics (lead times, product's variation, failure rate), activate specific safety protection during dangerous situations, or optimize the process in general. As mentioned in the section 1.3.1, generally, the transition between phases takes place within temporal synchronization, through rotors or conveyor belts that determine the speed of product progress. In these cases, it is possible to distinguish phases analyzing the PLCs that regulate the movements in the plant. However, when the industrial process has collaborative phases or non-deterministic steps, recognizing different process phases is complicated. It is impracticable to synchronize events using temporal constraints in a similar scenario. Indeed, the duration and sequence of the phases could vary depending on how the user performs these gestures. For example, in the case of the collection of an object by an operator, we need to discretize the phases of collection, movement and deposit of the object using action recognition. The operator may perform the gestures differently, affecting the duration of the movements or skip some phases (e.g, with a defective product). Since it is not possible to use time, it is necessary to recognize production phases from the actions and gestures performed by robots or operators as well as from sensors of the surrounding environment. The proposed method is a discriminative supervised algorithm which interprets the situation and automatically classifies the phases for heterogeneous data sources.

3.2 Time delay neural network

Assuming that the number of phases is known, we can represent an automatic phase recognition system as a classification problem. Using Equation 2.1 and following

notation proposed in [52], we can define a classification dataset as:

$$S = \{X_i,\ c_i\}_{i=1}^{n} \tag{3.1}$$

with $X_i = [x_i^{(1)}(t), \ldots, x_i^{(j)}(t), \ldots, x_i^{(p)}(t)]$, where p is the number of sensor used as input, i represent the i^{th} istance in the dataset composed of n elements, $c \in \mathbf{C}$ with $\mathbf{C}$ being the known set of phases of the process also called classes. The dataset S is divided into two parts such that:

$$S = S^{train} \cup S^{test} \quad \text{with} \quad S^{train} \cap S^{test} = \emptyset \tag{3.2}$$

Supervised machine learning models denoted as f are built minimizing the loss function during the training phase using S_{train} data. The loss function can be defined as:

$$Loss = \sum_{i=1}^{n^{train}} d\left[f\left(X_i^{train}\right), c_i^{train}\right] \tag{3.3}$$

Where d is a defined distance metric. The model is validated in the testing phase using the data from S_{test}. As already mentioned in section 2.2.1, a deep neural network is a composition of L parametric functions referred to as layers where each layer is considered a representation of the input domain [137]. One layer l_k, such as $k \in 1 \ldots L$, contains neurons, which are small units that compute one element of the layer's output [138]. The layer l_k takes as input the output of its previous layer l_{k-1} and applies non-linearity to compute its own output. The behavior of these non-linear transformations is controlled by a set of parameters W_k for each layer. In the context of DNNs, these parameters are called weights which link the input of the previous layer to the output of the current layer. Hence, given an input X_i, a neural network performs the following computations to predict the class:

$$f_L(W_L, x) = g_{L-1}(W_{L-1}, g_{L-2}(W_{L-2}, \ldots, g_1(W_1, x))) \tag{3.4}$$

were g represent non linear function [138]. Convolutional neural networks are based

on a convolution operator that produces an output feature map S by sliding a kernel (w) over the input X. The kernel's dimension is fixed for each layer L; its value is given by the weights W learned with training. Each element in the output feature map is obtained with the element-wise multiplication between the layer's input and the kernel. The number of kernels (filters) M used in a convolutional layer determines the depth of the output volume (i.e., the number of output feature maps). Considering a 1D time-series X_p and a one dimensional kernel $K(W_1)$, the i^{th} element of the convolution between X_p and w is:

$$s(i) = (X_p * K(w))(i) = \sum_{j=0}^{|k|-1} x(i-j)w(j) \tag{3.5}$$

As reported in [107] the main three advantages of the convolutional neural network are the following:

- Local connectivity: each hidden neuron is connected to a subset of input neurons that are close to each other
- Parameter sharing: the weights used to compute the output neurons in a feature map are the same, so that the same kernel is used for each location.
- Translation equivariance: the network is robust to an eventual shifting of its input

When applied to time series, convolutional operation occurs between an instant of time X_t of the input signal or eventually of the feature map and the instant of time X_{t-1}. As the application of convolutional filters in an image produces a distortion on the output for specific features, in the case of signals, the convolution allows elaborating the input signal at different frequencies resulting in enhanced temporal modeling. Time delay neural network (TDNN) uses temporal convolution to model features from different sensors at varying frequencies. TDNN has a pyramidal structure due to a wider temporal context [6]; the initial transforms are learnt on narrow contexts and the deeper layers process the latent variables from using kernel's dilatation, as shown in Figure 3-1.

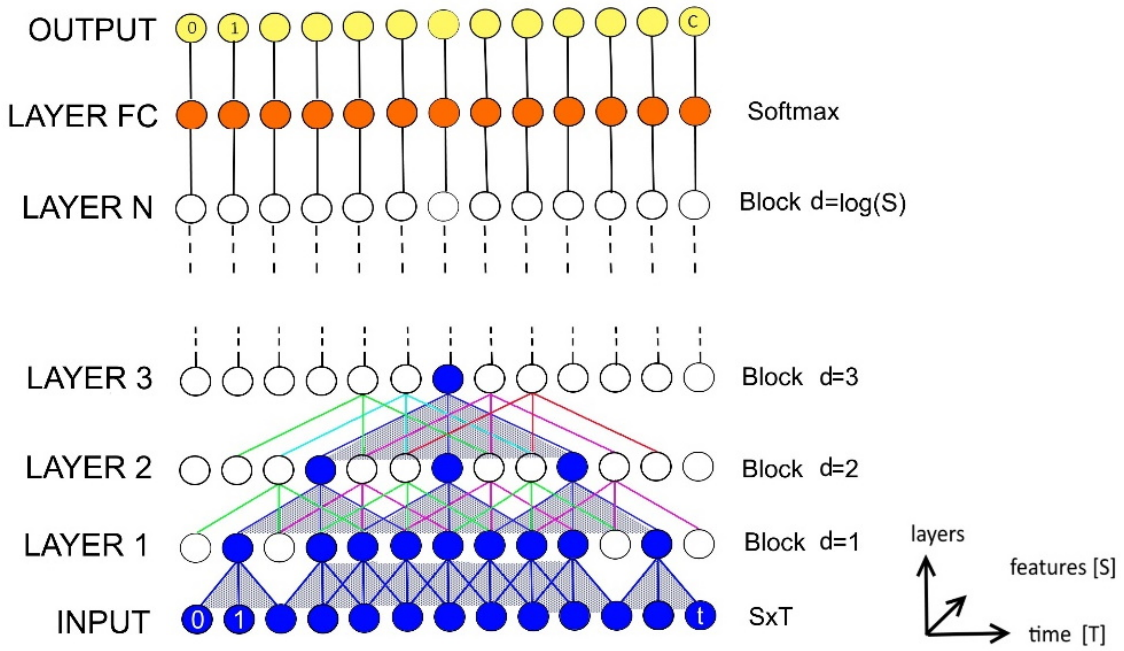


Figure 3-1: TDNN network architecture from [6].

They were firstly proposed for speech recognition in 1989 by Waibel [139]. The deeper layers can learn wider temporal relationships, thus providing a higher feature abstraction. This enables the recognition of longer time features thanks to wider receptive field, thus modelling gestures and movements considering different temporal context. During back-propagation, due to the pyramidal structure, the network is updated by a gradient accumulated over different temporal resolution of the input. Thus, the lower layers of the network are forced to learn translation invariant feature transformation [6]. TDNN uses dilated convolutions that employ a padded kernel to learn more distant features. Given a dilation factor d the convolution on a single time series with a one dimensional kernel is described as:

$$s(i) = (X_p *_d K(w))(i) = \sum_{j=0}^{|k|-1} x(i - dj)w(j) \tag{3.6}$$

A padded kernel on a two dimensional feature is represented in Figure 3-2. This is a major advantage of stacked dilated kernels is that while for standard convolutions the receptive field r grows linearly with the depth of the network $r = k(L - 1)$ with

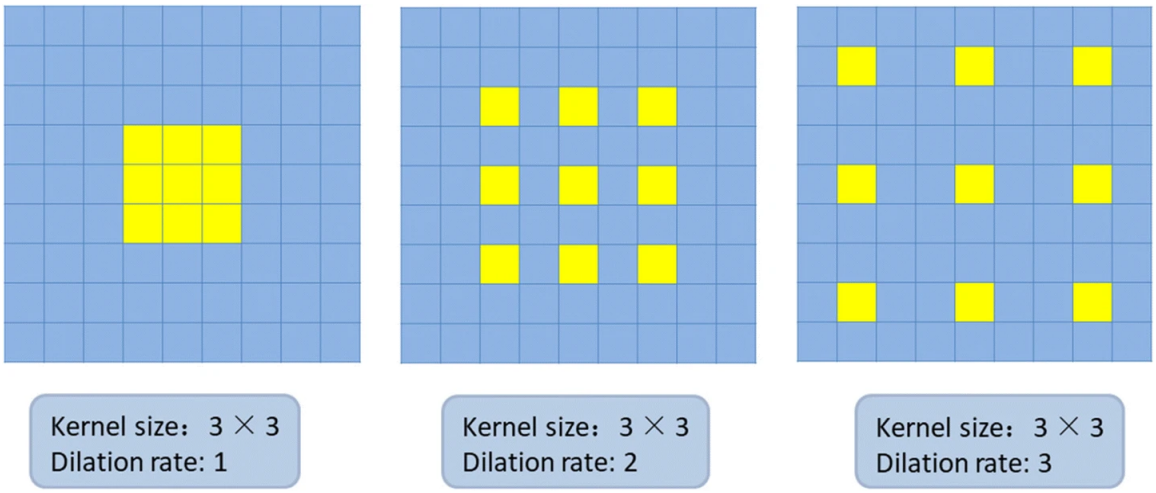


Figure 3-2: 3x3 convolution kernels with different dilation rate as 1, 2, and 3 on two dimensional input [7].

in the dilated convolutions the dependence is exponential $r = 2^{L-1}k$ with increasing dilation factor d. This ensure that a much larger history size is used by the network. We define a block B as a tuple $< s(i), Re >$ were $s(i)$ is dilated convolution and Re is a Rectified Linear Unit activation function. The block are stacked with an increasing dilatation rate. The number of blocks N used in our network depends on the number of features S contained in the input vector. A fully-connected layer with softmax activation function (i.e., a generalizes logistic function) is applied to obtain classification results, i.e. Y vector.

3.3 Experiments

We apply TDNN on three different datasets and evaluate the proposed method ability for phase recognition. Two of them were created specifically for simulating a collaborative industrial process, while the other is extensively used as a benchmark dataset to recognize actions in SRSs. In particular the first presented dataset aim to describe an I4.0 industrial process with numerous sensors, the second describe precision movements perform by an expert operator and the third compare TDNN performance with state-of-the-art models. In this section we present the datasets, the metrics and the cross validation used.

3.3.1 Datasets

- **ICRT (Industrial Collaborative Robot-human Task)** The ICRT dataset employ four devices for the interaction between a human operator and a collaborative robot. The sensors used are a Leap Motion device [140], an ArUco marker [141], an Intel RealSense [142] camera and a Panda robot [143]. A total of 144 features are extracted from these devices simulating an I4.0 environment with heterogeneous sensors. Thus, we used data extracted directly from machines (robot kinematics) and data obtained from the supervising sensors such as Leap Motion and ArUco marker. The product is represented with a ring to avoid complications from manipulating the object, such as incorrect grasping. The product is grasped using a tool that extends the experiment to scenarios where the object is unsuitable for human contact. The setup is shown in Figure

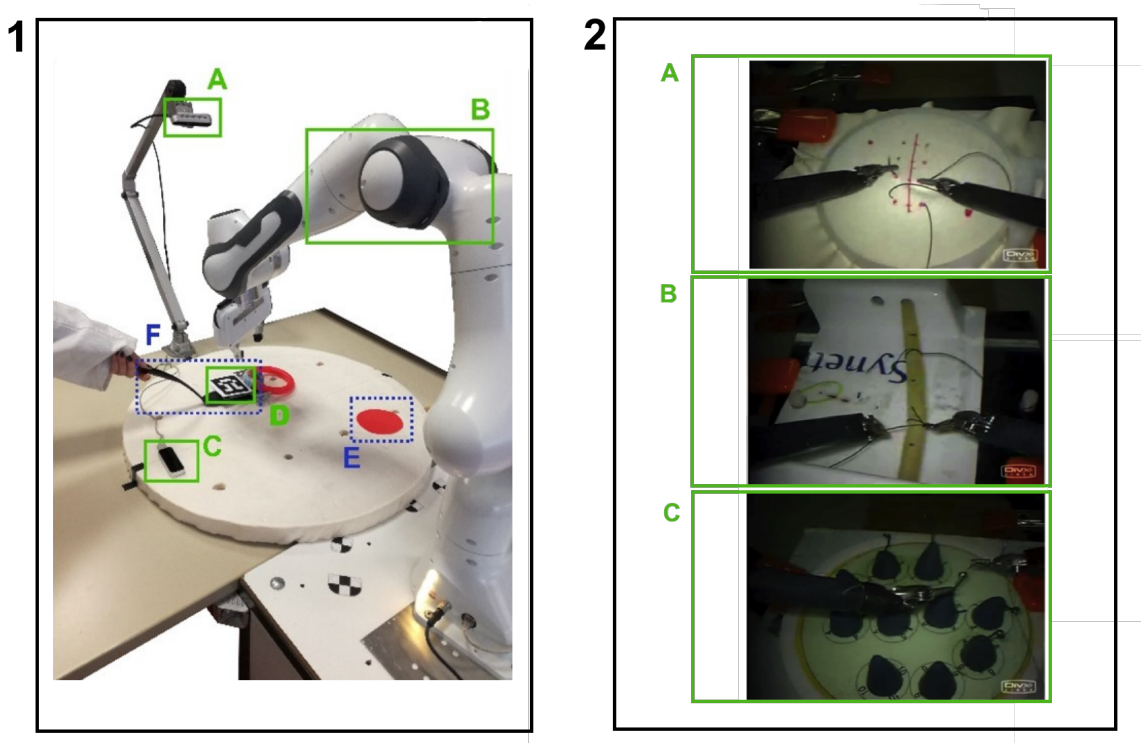


Figure 3-3: 1) ICRT setup from [6]. ICRT dataset setup. (A) Intel RealSense, (B) Panda robot, (C) Leap Motion, (D) ArUco marker, (E) Drop point for robot, (F) Tool for interacting with ring. 2) JIGSAWS setup from [8]. (A) suturing, (B) knot-tying,(C) needle-passing

3-3, highlighting the sensors used and the salient points of the task. The ex-

periment is described as follows: in the first phase, a human operator picks the ring through a tool to avoid direct contact with the hand. Then the user moves it to an arbitrary position and releases the ring. The second phase involves the scene segmentation and ring recognition by the RealSense camera sensor, and afterward, the robot picks up the ring and then places it to the drop point, then it will release the ring. When the robot ends the task, it returns to a ready position waiting for a new task to be executed. The entire task will be described by:

  - The Leap Motion data to capture hand movements
  - The robot joints position and the end-effector pose to capture the motion of the robot
  - The ArUco to track the movements of the ring.

  These values are labelled and detailed in Table 3.2. This task has been repeated 40 times by a single human operator. The dataset and related detailed documentation is available at `gitlab.com/altairLab/ICRT.git`

- **VIT-MR (Virtual Industrial Task Master-slave Robot)** The VIT-MR dataset was created to replicate all the steps of the robotic-assisted teleoperated manipulation process practiced in high precision small-scale manufacturing. The user controls remote slave manipulators from Leo master console [144] visible in Figure 3-4, a compact hardware device integrating two masters manipulators, high-definition stereo viewer and foot-pedals tray. The console guarantees an immersive user experience, which enables ergonomic slave manipulators controls and enhanced magnified vision. In addition, simulated slave manipulators provide high-dexterity and movement scaling to ensure precise and stable components manipulation during the assembly process. We have implemented a simulated environment using a research version of Xron (BBZ srl, Verona, Italy), a realistic virtual simulator suitable for high fidelity applications, such as medical training. We have used this experimental setup to reproduce kine-

Figure 3-4: 1) The hardware training console used by one of the students during data acquisition. 2) Example of the Virtual task considered in the VIT-MR dataset: four colored rings need to be placed in the corresponding peg from [6].

matic variables similar to a real master-slave robotic setting in the industrial field. The manipulation task involves positioning a set of colored rings in their correct position on a pegboard, as shown in Figure 3-4. The exercise consists in lifting a ring with one of the robotic tools, passing the ring on the other robotic arm and positioning it in the corresponding pole. A group of 17 users without significant experience in robotic assisted manipulation has been enrolled in this study. All the users have no more than one hour of experience in using similar robotic systems. Each subject had a time slot of one hour, the first half is dedicated to practicing with the specific platform and the second half is dedicated to the recorded trials. Each subject performed from a minimum of ten to a maximum of twenty trials, resulting in 256 sequences. Multiple users with different levels of experience allow the model to be robust to specific bias in the movements execution and enhance phases recognition. The dataset and related detailed documentation is available at `gitlab.com/altairLab/VIT-MR.git`.

- **JIGSAWS (JHU-ISI Gesture and Skill Assessment Working Set)** JIGSAWS is a dataset widely used in SRS action recognition, being the reference benchmark dataset. We decided to include this dataset because of the complexity of the gestures performed by surgeons and the large number of recent methods to compare with. Three robotic-assisted surgery simulation exercises

ID	JIGSAWS	ICRT	VIT-MR
0	Reaching for needle with right hand	Hand pick the tool with the ring	Colleting the ring
1	Positioning needle	Move the ring with the tool at an arbitrary point	Passing the ring from the right arm to the left arm
2	Pushing needle through tissue	Release the tool	Posing the ring in the correct pole
3	Transferring needle from left to right	Robot identify and pick the ring	Failing grabbing the ring
4	Moving to center with needle in grip	Let the robot move the ring at drop point	Failing passing the ring from right arm to left arm
5	Pulling suture with left hand	Robot release the ring	Failing posing the ring in the correct pole
6	Pulling suture with right hand		
7	Orienting needle		
8	Using right hand to help tighten suture		
9	Loosening more suture		
10	Dropping suture at end and moving to end points		
11	Reaching for needle with left hand		
12	Making C loop around right hand		
13	Reaching for suture with right hand		
14	Pulling suture with both hands		

Table 3.1: Phases and ID for each datasets

(suturing, knot-tying and needle passing) were performed on a da Vinci surgical system at Johns Hopkins University [8]. Motion data collected from the da Vinci API were collected and made available online. The JIGSAWS consists of 39 trials of Suturing task, 36 trials of Knot tying, and 28 trials of Needle Passing. Each task was performed by eight users. The previous experience with the da Vinci surgical system vary significantly between users. Some users had fewer than 10 hours as past experience with SRS while others have more than 100 hours. Experienced operators generally perform smoother movements and therefore are easier to recognize. In [8] the experience levels for each user are reported. The three tasks include:

- Suturing (SU): the subject picks up needle, proceeds to the incision (designated as a vertical line on the bench-top model), and passes the needle through the tissue, entering at the dot marked on one side of the incision and exiting at the corresponding dot marked on the other side of the incision. After the first needle pass, the subject extracts the needle out of the tissue, passes it to the right hand and repeats the needle pass three more times;
- Knot-Tying (KT): the subject picks up one end of a suture tied to a flexible tube attached at its ends to the surface of the bench-top model, and ties a single loop knot;
- Needle-Passing (NP): the subject picks up the needle (in some cases not captured in the video) and passes it through four small metal hoops from right to left. The hoops are attached at a small height above the surface of the bench-top model.

The motion is described by a local frame attached at the distal end using 19 kinematic variables; therefore, there are 76-dimensional data considering the four manipulators involved: left and right for master and slave side. The 19 kinematic variables for each manipulator include Cartesian position, rotation matrix, linear velocities, angular velocities, and instrument gripper angle. The

JIGSAWS	
Column indices	Description of variables
1-3	Left MTM tool tip position
4-12	Left MTM tool tip rotation matrix
13-15	Left MTM tool tip linear velocity
16-18	Left MTM tool tip rotational velocity
19	Left MTM gripper angle velocity
20-38	Right MTM kinematics
39-41	PSM1 tool tip position
42-50	PSM1 tool tip rotation matrix
51-53	PSM1 tool tip linear velocity
54-56	PSM1 tool tip rotational velocity
57	PSM1 gripper angle velocity
58-76	PSM2 kinematics
VIT-MR	
Column indices	Description of variables
0	Process phase
1-15	Rotation and translation Right
16	Gripping Angle Right
17-28	Rotation and translation Left
29-31	Cartesian position Left
32	Gripping angle Left
ICRT	
Column indices	Description of variables
0	Gesture
1-7	ArUco Marker Cartesian position and orientation in quaternions.
8-16	Robot joint angles
17	Timestamp
18-25	End-effector cartesian and orientation
26-114	Leap motion hand feature (87)

Table 3.2: Features for each datasets

classification classes are fifteen and are defined in Table 3.1. The dataset and related detailed documentation is available at `https://cirl.lcsr.jhu.edu/research/hmm/datasets/jigsaws_release/`.

3.3.2 Evaluation

To evaluate our model, we use Leave One User Out (LOUO) evaluation. LOUO cross-validation is performed by splitting the dataset in the training and testing set.

During the training, we exclude a single user that is later used for testing. This procedure is repeated for all users. The reported result will be the average between the result of each user. For the ICRT dataset, instead of LOUO, leave-5-out was used since a single user is available. Consequently, five repetitions of the task excluded as testing and the remaining for training, thus obtaining eight different splits. To facilitate understanding, we report in Table 3.3 the information on the evaluation methodology applied.

Dataset	**Cross-validation**	**repetition**	**N.task in total**
ICRT	Leave-5-Out	8	40
VIT-MR	Leave-One-User-Out	17	256
Suturing	Leave-One-User-Out	8	39
Needle-Passing	Leave-One-User-Out	8	28
Knot-tying	Leave-One-User-Out	8	36

Table 3.3: Cross validation for each dataset

3.3.3 Metrics

As metrics, we used the macro and micro accuracy with standard deviation and training time for each repeated task. Micro-average is preferable for classification on unbalanced classes since it aggregates the contributions of all classes to compute the average metric, while macro-average computes the metric independently for each class and then takes the average. For ICRT and VIT-MR we also report the confusion matrix in the discussion. Following notation in [11], a confusion matrix $C_f(f = \{1, 2, \ldots, F\})$ of size $[n, n]$ is computed as

$$C_f[i, j] = \text{number of } class\ i \text{ samples predicted as } class\ j. \tag{3.7}$$

The complete confusion matrix C is the sum of all of the confusion matrices.

$$C = C_1 + C_2 + \ldots + \ldots C_F. \tag{3.8}$$

Given the complete confusion matrix, the Micro average is computed as the average of total correct predictions across all classes

$$\text{Micro} = \frac{\sum_{i=1}^{n} C[i,i]}{\sum_{i,j=1}^{n} C[i,j]} \tag{3.9}$$

the Macro average is performed as:

$$\text{Macro} = \frac{1}{n} \sum_{i=1}^{n} \frac{C[i,i]}{\sum_{j=1}^{n} C[i,j]} \tag{3.10}$$

3.3.4 Results

The results are reported in Table 3.4 and Figure 3-5. For the JIGSAWS dataset we present the results only for the TDNN as many other models can be referenced in literature for comparison. In [10, 11] a review of the performances obtained by each method on JIGSAWS Suturing is outlined.

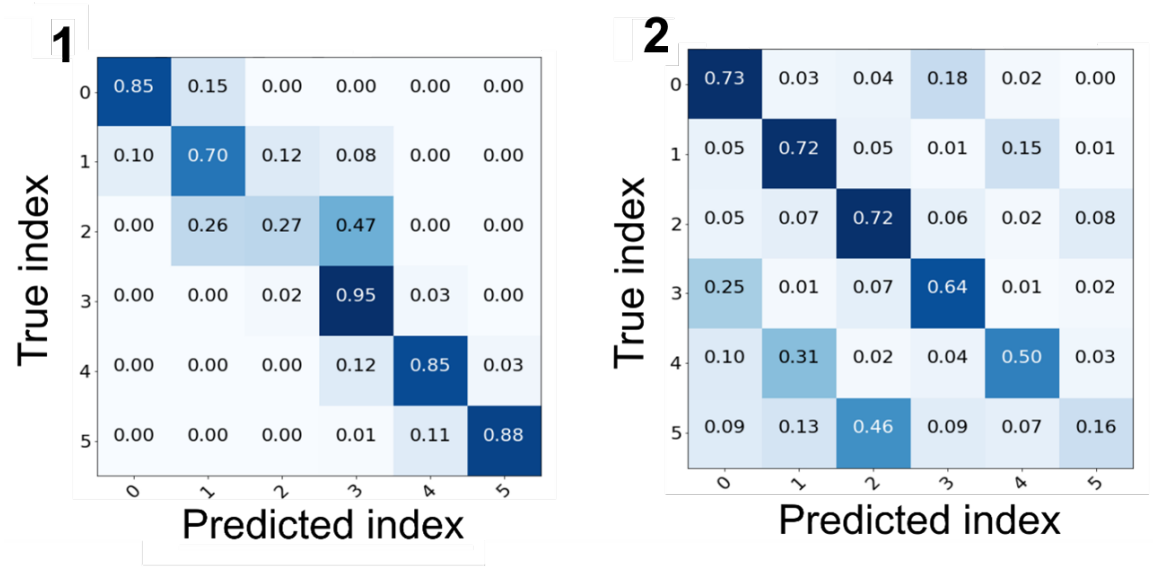


Figure 3-5: Normalized confusion matrix, color indicates the probability of classification as represented by the scalar number, index are referred in Table 3.1 (1) ICRT, (2) VIT-MR.

Method	Dataset	
	VIT-MR	
	Micro-Accuracy	Macro Accuracy
Time Delay Neural Network	69.76 ± 6.08	53.85 ± 3.508
Long-Short Term Memory	60.86 ± 8.60	42.35 ± 3.50
Random Forest	62.16 ± 8.914	42.7 ± 3.792
Support Vector Machine	54.55 ± 10.42	32.11 ± 5.254
	ICRT	
	Micro-Accuracy	Macro Accuracy
Time Delay Neural Network	86.95 ± 3.693	79.04 ± 4.902
Long-Short Term Memory	65.83 ± 13.18	51.98 ± 15.07
Random Forest	86.92 ± 3.558	80.32 ± 5.126
Support Vector Machine	41.19 ± 1.72	6.865 ± 0.28
	JIGSAWS Sututing	
	Micro-Accuracy	Macro Accuracy
Time Delay Neural Network	74.4 ± 7.41	53.89 ± 8.39
GMM-HMM [11]	80.83	65.03± 33.07
LDS [11]	73.64	51.75±32.91
BiLSTM [10]	84.7	-
MS-RNN [10]	90.2	-
	JIGSAWS Knot-Tying	
	Micro-Accuracy	Macro Accuracy
Time Delay Neural Network	73.44 ± 9.58	70.27 ± 12.17
GMM-HMM [11]	78.44	72.68±21.31
LDS [11]	71.42	63.99±24.51
	JIGSAWS Needle-passing	
	Micro-Accuracy	Macro Accuracy
Time Delay Neural Network	64.36 ± 11.65	48.26 ± 7.83
GMM-HMM [11]	66.22	62.70±16.38
LDS [11]	47.96	32.59±29.74

Table 3.4: Result for macro and micro accuracy for the datasets. For JIGSAWS refer to [10, 11]

3.3.5 Discussion

This experiment aims to validate the TDNN for industrial process phases recognition.

In particular, we focus on addressing two main challenges:

- Infer phases of the process from heterogeneous sets of sensors.

- Recognize complex actions composed of multiple movements in a collaborative task.

TDNN presents the best results on the VIT-MR dataset, obtaining the highest accuracy on both: micro and macro accuracy. The proposed method can interpret the kinematic sequence of data by defining the temporal correlation as a fundamental characteristic for recognizing gestures. In the VIT-MR dataset, the data are associated with the movements performed by the users on the Leo master console (i.e., master manipulator kinematics). Since the data doesn't contain information from the simulated environment, the phases are described exclusively by spatial and temporal features on the movements. Figure 3-5, confirms that the majority of errors are made between a phase and its corresponding failed action (e.g., pick-up/fail pick-up); therefore, the network infers specific characteristics of the movements. In VIT-MR, it is difficult to detect errors from kinematics since the ring's actions are not recorded. For example, recognizing a recovery action is challenging if the ring is stuck or placed in the wrong pole. Moreover, fewer instances of error classes are available in the dataset. The lower accuracy in the error classes justifies the difference in magnitude between micro and macro accuracy. In general, in addition to a suitable recognition method is essential to keep track of the surrounding environment using different sensors for correct recognition of process phases.

On the ICRT dataset, TDNN performs similarly to random forest while the accuracy is sensibly higher than LSTM and SVM. ICRT is a smaller dataset compared to VIT-MR and presents less variation as represented by the standard deviation of random forest and TDNN. It is reasonable that the small dimension of the dataset penalizes the TDNN, since ANN generally performs better with lots of data [103]. Nevertheless, TDNN shows a remarkable ability to extract features from multiple sensors. The wide diversity of signals analyzed in the ICRT datasets prove that TDNN is suitable for various industrial scenarios. Devices as infrared cameras, Aruco markers or robot kinematics can provide distinct information on a wide variety of HRC pro-

cesses. Moreover, the performance increases significantly when the task is performed only by an automated system. The confusion matrix in Figure 3-5 shows that the phases of the process performed by the operator in the ICRT task (i.e. 0,1,2) are more difficult to recognize than those performed independently by the robot (i.e. 3,4,5). The automated phases have an accuracy of around 90%. These results demonstrate that the use of systems for the recognition of process phases in an industrial context can be easily extended to robotic procedures without overloading the operators in integrating these systems with the PLC.

The movements performed by surgeons in the JIGSAWS dataset, being from a surgical context, are particularly complex. JIGSAWS, contains detailed actions performed by surgeons (called surgemes) that are more intricate than regular motion movements. The results show that the TDNN reaches state-of-the-art accuracy and recognizes the correct movement in more than 70% of cases for micro accuracy. However, since it is an extensively used dataset for action recognition for SRS, some state-of-the-art methods outperform TDNN. Different factors have to be considered in the comparison. First, in [10], the results focus exclusively on the suturing exercise. The goal of our experiment is to apply TDNN over a wide range of movements rather than to optimize the result for a single exercise, therefore, it is reasonable that TDNNs do not reach the highest level of accuracy available in the literature. In [145] TDNN trained exclusively on suturing reached, indeed, a micro accuracy of 80%. Bidirectional LSTM and MS-RNN achieves higher accuracy on kinematics at the expense of model complexity and higher parameter tuning time. The results also highlight that the standard deviations for JIGSAWS datasets are higher than other datasets. This is mainly due to the substantial variance between operators' experience with SRS.

3.4 Conclusions

In chapter 3, we applied TDNN in the context of industrial phases classification. The pyramidal structure allows features to be analyzed at different scales, enhancing pattern recognition from data. This architecture has proven to be efficient in interpreting

the actions of both operators and robots. Its capability was demonstrated on three datasets. The experiments suggest that TDNN can describe the phases of an I4.0 process from the data by interpreting movements and action, even in cases of very complex and articulated movements.

Publications: Most of the results presented in the paragraph have been already published in the following papers:

- G. Menegozzo, D. Dall'Alba, A. Roberti, and P. Fiorini, "Automatic process modeling with time delay neural network based on low-level data.," Procedia Manufacturing, vol. 38, pp. 125–132, 2019, doi: `https://doi.org/10.1016/j.promfg.2020.01.017`.
- G. Menegozzo, D. Dall'Alba, C. Zandonà and P. Fiorini, "Surgical gesture recognition with time delay neural network based on kinematic data," 2019 International Symposium on Medical Robotics (ISMR), 2019, pp. 1-7, doi: `https://doi.org/10.1109/ISMR.2019.8710178`.

Chapter 4

Predictive monitoring with causal precursors

In this chapter we introduces a procedure for monitoring a key quality variable in manufacturing SMEs. The main contributions in the presented procedure are the following

- Inferring the flow of the production plant by reconstructing the causal relationships between machine's parameters.
- Develop a predictive procedure based on supervised deep learning that integrate causal precursors and domain's experts knowledge.

In section 4.1 we infer the flow of the production using causal discovery. We present the PCMCI algorithm and prove its effectiveness in a synthetic scenario that replicates the characteristics of an industrial plant. PCMCI allows to infer cause-effect relationships between phases and reconstruct the flow of the process. In section 4.2 we present the predictive procedure that exploits the inferred causal model to build a specific ANN architecture for forecasting a quality feature of the final product. The deep learning architecture called S-TCN is explained in section 4.3. Similarly, we present and discuss the experiment done on the S-TCN architecture. In Figure 4-1 we present a summary of the fourth chapter highlighting the predictive procedure which combines causal discovery and ANN.

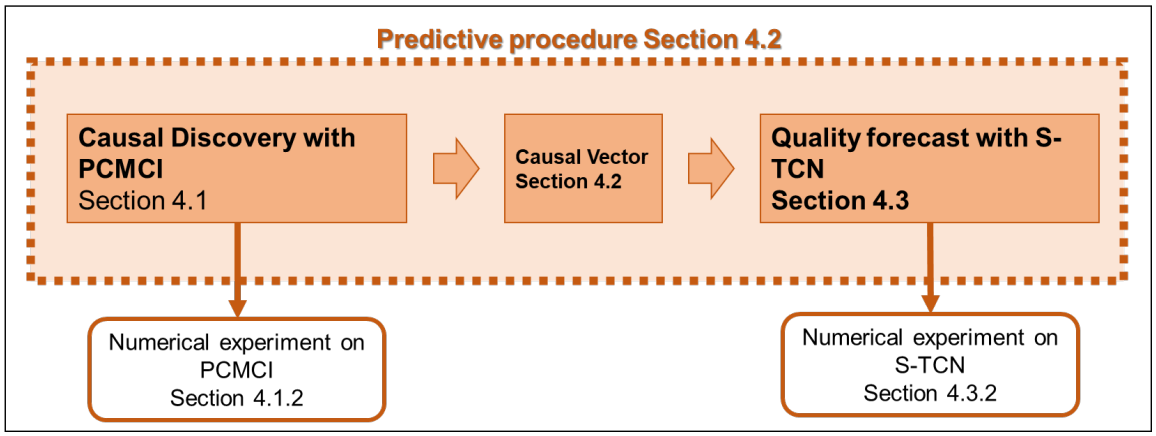


Figure 4-1: Summary of chapter 4.

4.1 Causal modeling of manufacturing plant

In chapter 3 we presented an automatic system for recognizing different process phases. In continuous processes, besides partitioning the process into phases, it is essential to establish an organized execution sequence that defines the flow of the process. Since the product evolves over time, starting from raw material and ending with the finished product, to describe the process, it is crucial recognize also the relationships between phases. Indeed, each phase affects specific characteristics of the product, thus, modifications made on the product are propagated to subsequent phases, refining the quality in the final product. Causal discovery allows to reconstruct the cause-effect relationships within a plant. In this section, we propose to use a causal discovery method to reconstruct the process flow of a plant from signals extracted from machines during the production. The sequentiality of continuous manufacturing processes assures that the causal relationships between machines are consistent with the process flow. We exploit the idea that a variation that occurs in a specific phase will propagate in the following, so it has a causal effect on the remaining production; vice versa, if the variation is not propagated to other machines, the phase belongs to the final stage of the production.

Reconstructing the causal model of the plant has multiple advantages for the company. In this thesis, we demonstrate the use of causal relationships to improve

the monitoring system for a characteristic quality variable of the final product. The causal model of the plan can also be used to suggest additional protocols for in-line intervention performed by operators.

4.1.1 PCMCI

Causal discovery in multivariate time series aims to identify the relevant cause-effect relationships and find their effective activation between time lagged values [146]. With time lagged values we define the number of temporal instances that occur between intervening on a cause variable and the manifestation of the change in the quality of the final product. PCMCI is a constraint-based method that uses independence tests to exploit causal effects between time series. PCMCI is based on a condition-selection step, followed by a momentary conditional independence (MCI) test [147, 148]. The first step is based on Peter-Clark (PC) algorithm [134], while the MCI step has been introduced in [148] to account for autocorrelation, leading to improved control of false-positive rates. The PC algorithm learns from data a (partially) directed graph that represents the Markov equivalence class of the underlying distribution [134]. It starts with a fully connected undirected graph thus assuming as a starting condition that all the variables are dependent on each other. Then it iteratively tests pairs of variables for conditional independence given conditioning sets of increasing size. PC defines that X is a direct cause of Y with respect to a set of variables $\mathbf{Z}$ if changing the value of X results in changes in the probability distribution of Y, assuming that all the values of variables in $\mathbf{Z}$ are held constant [149]. We use upper-case letters to denote random variables and bold-face to denote variable sets. During each iteration, the PC algorithm checks the independence of two variables and then adds a variable in $\mathbf{Z}$. If conditional independence is maintained with all variables in $\mathbf{Z}$, PC removes the arc between the two nodes as they are considered independent. The result will be an undirected graph [148]. The significance level of these tests is given by α parameter. Then, a set of four orientation rules is applied recursively to orient the undirected edges and obtain an equivalence class of partially directed graph. The final output of PC algorithm is a partially directed graph. In the second phase, the momentary

conditional independence (MCI) is used to test whether $x^i_{(t-\tau)} \rightarrow x^j_t$ with

$$MCI : x^i_{(t-\tau)} \not\perp\!\!\!\perp x^j_t | \mathcal{P}(x^j_t) \setminus x^i_{(t-\tau)} , \mathcal{P}(x^i_{(t-\tau)}) \tag{4.1}$$

Where $\mathcal{P}$ stands for parents, t runs through the samples up to the time series length, τ depends on the application and can be chosen according to the maximum causal time lag expected in the complex system, $\not\perp\!\!\!\perp$ indicates dependence and $\rightarrow$ causation between two variable $x^i, x^j \in \mathbf{X}$ where $i \neq j$ and $\mathbf{X}$ is the set of all variables. The additional condition on the lagged parents $\mathcal{P}(x^i_t)$, $\mathcal{P}(x^i_{(t-\tau)})$ accounts for autocorrelation, leading to correctly control false-positive rates [148]. We decide to apply this method because it is superior to PC or to full conditional independence testing (FullCI) as demonstrated in [148]. A diminished number of false positives leads to a reduction in the amount of detected cause-effect relationships. This aspect is interesting for monitoring systems that can integrate the reduced outcome of the causal discovery algorithm for improved efficiency. We apply two different types of independence tests:

- *Partial correlation:* Partial correlation of $X \perp\!\!\!\perp Y | \mathbf{Z}$ is estimated in two-stage procedure: a multivariate regression of X and Y on $\mathbf{Z}$, followed by a correlation test on the residuals based on T-student test. It is therefore applicable only to the multivariate Gaussian case which can only capture linear dependencies.

- *Conditional mutual information:* Non-linear dependencies are evaluated with a non-parametric test for continuous data based on conditional mutual information combined with a local permutation scheme. The conditional mutual information (CMI) is zero if and only if $X \perp\!\!\!\perp Y | \mathbf{Z}$. An estimator for mutual information was developed by Kraskov from the nearest-neighbour entropy estimator [150].

The PCMCI output is a $|\mathbf{X}| * |\mathbf{X}| * \tau_{max}$ tridimensional matrix that returns the causal precursors, for each variable $x^- \in \mathbf{X}$ and for each corresponding time lagged delay τ. In practice, if the industrial plant setting remains unchanged, a cause variable should manifest its effect always at the same time delay. The experiments done on

causal discovery with PCMCI are described in section 4.1.2.

To summarize, PCMCI is a constraint-based method with strong control on false positive that uses independence test to exploit causal effect between time series as explained in section 2.3.2. The analysis framework used in this work was developed by Runge [147]. The method has three free parameters:

1. α*:* The α value that is the significance level of the PC algorithm. The higher the value of α, the higher the number of edges which will be kept in the model.

2. τ*:* The τ value that indicates the maximum time lag for MCI test. This is given by the maximum causal time lag expected in the complex system.

3. *Independence test:* The independence test that are described above: partial correlation and conditional mutual information.

In this thesis, as explained in section 2.3.2, we use PCMCI to perform causal discovery and reconstruct the flow of a production plant from data. We chose to use PCMCI for the following reasons:

- PCMCI is suitable for high-dimensional data. In particular, in combination with the partial correlation independence tests it is scalable on time series with large cause-effect delays.

- Compared to score-based methods it does not use a minimization function which presents local minima.

- Compared to methods that exploit asymmetries, it is scalable on many variables and cofounders differently from Lingam, Timino and CCM methods.

- Compared to predictive methods that apply granger causality it allows a strong controls on false positive.

4.1.2 Experiments on Causal Discovery

In this experiment we built a synthetic model of a continuous manufacturing process to validate the use of PCMCI in an industrial context. The goal is to reconstruct

cause-effect relationships between variables from process data. Typical manufacturing lines consists of multiple phases where different operations take place. Each phase is composed of one or more machines that contribute to the advancement of the product. Each phase brings an effect on the subsequent phases, thus the production line is governed by the cause-effect principle.

Dataset

We create the synthetic model represented in Figure 4-2 based on the following criteria:

- The cause-effect relationships follow the production process flow therefore a downstream phase in the process cannot be the cause of a previously occurred phase.

- The cause-effect relationships are stronger if two phases follow each other in the production process flow. Increasing the distance between the two phases will decrease the causal effect present. Moreover, we expect an increased non-linearity between distant phases.

- Each phase is composed of one or multiple interacting variables. They represent the values of a single machine or several machines belonging to the same process phase. These interactions are described by cause and effect relationships, however, cycles are not allowed.

We added a source node that only affected the first three variables to test different types of non-linear dependencies. The proposed synthetic model is described by the structural causal model 4.2.

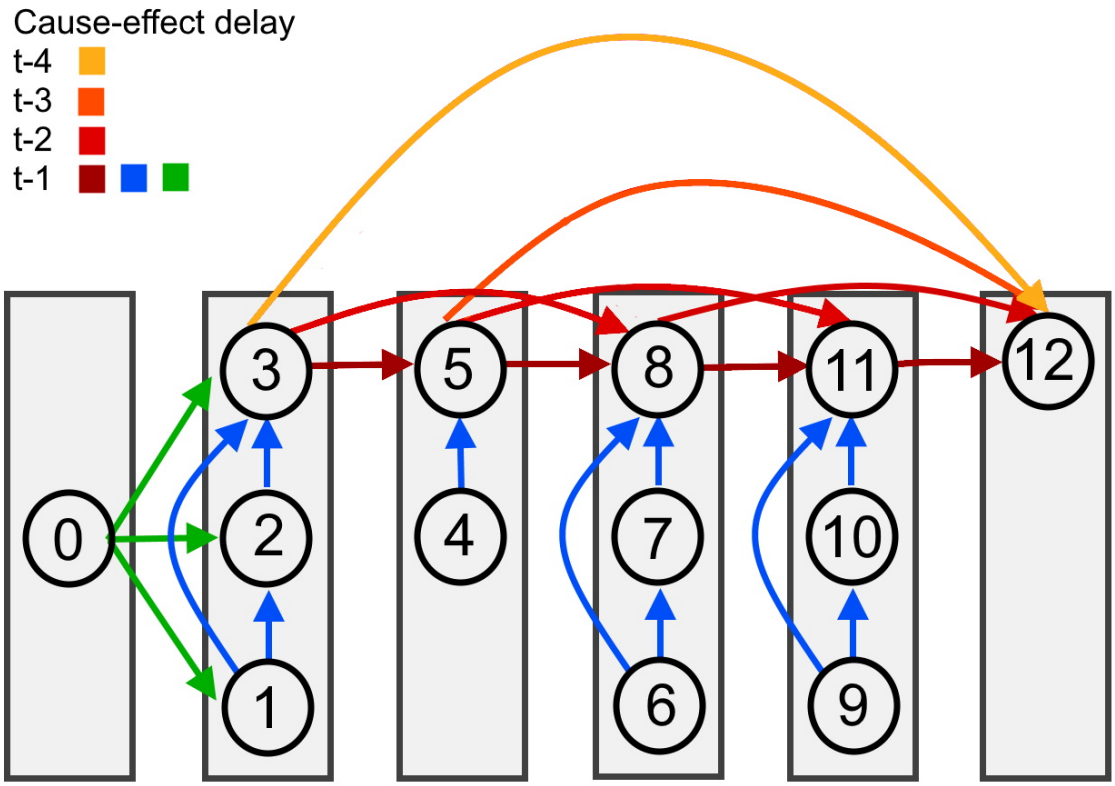


Figure 4-2: Synthetic model with ultra-processed food manufacturing features: rectangles represent phases, each phase is represented by one or more variables. The arrows in blue represent linear causal relationships with coefficient β. In green the non-linear and non-polynomial causal links. In red causal links with increasing non-linearity with α coefficient.The different shade of red are related to the strength of the coefficents (i.e., more red-colored arrow entails a stronger causal relationship). This graph is described by the structural causal equation 4.2.

$$
\begin{aligned}
X_t^0 &= \mathcal{N}^0 \\
X_t^1 &= 1 - 4e^{-(X_{t-1}^0)^2/2} + \mathcal{N}^1 \\
X_t^2 &= 1 - 4(X_{t-1}^0)^3 e^{-(X_{t-1}^0)^2/2} + \beta_2^1 X_{t-1}^1 + \mathcal{N}^2 \\
X_t^3 &= e^{-(X_{t-1}^0)^3/3} + \beta_3^1 X_{t-1}^1 + \beta_3^2 X_{t-1}^2 + \mathcal{N}^3 \\
X_t^4 &= \mathcal{N}^4 \\
X_t^5 &= \sigma_5^3 X_{t-1}^3 + \beta_5^4 X_{t-1}^4 + \mathcal{N}^5 \\
X_t^6 &= \mathcal{N}^6 \\
X_t^7 &= \beta_7^6 X_{t-1}^6 + \mathcal{N}^7 \\
X_t^8 &= \sigma_8^5 X_{t-1}^5 + \sigma_8^3 (X_{t-2}^3)^2 + \beta_8^7 X_{t-1}^7 + \beta_8^6 X_{t-1}^6 + \mathcal{N}^8 \\
X_t^9 &= \mathcal{N}^9 \\
X_t^{10} &= \beta_{10}^9 X_{t-1}^9 + \mathcal{N}^{10} \\
X_t^{11} &= \sigma_{11}^8 X_{t-1}^8 + \sigma_{11}^5 (X_{t-2}^5)^2 + \sigma_{11}^3 (X_{t-3}^3)^3 + \\
&\quad + \beta_{11}^{10} X_{t-1}^{10} + \beta_{11}^9 X_{t-1}^9 + \mathcal{N}^{11} \\
X_t^{12} &= \sigma_{12}^{11} X_{t-1}^{11} + \sigma_{12}^8 (X_{t-2}^8)^2 + \sigma_{12}^5 (X_{t-3}^5)^3 + \\
&\quad + \sigma_{12}^3 (X_{t-4}^3)^4 + \mathcal{N}^{12}
\end{aligned}
\tag{4.2}
$$

$\mathcal{N}^i$ is an independent and identically distributed Gaussian noise with zero mean and unit variance that is used to simulate i process variable. Linear relationships between intra-phase variables are modeled using β_i^v coefficients, where i indicates the variable that has a linear link with v. β_i^v are randomly chosen between $[-1; 1]$. The maximum time lag τ for these links is fixed to 1 because variables in the same phase have faster causal connections. We modeled causal links between variables in different process phases introducing increasing non-linear relationships. The σ_i^v represents the relationship between variable i and v, and its value is chosen from the set $[0.8, 0.6, 0.4, 0.2]$. In the proposed model, non-linearity is added by decreasing the

value of σ while simultaneously increasing the degree of the function and the delay k within 0 and τ. According to (2), this means that the causal link between parameter 3 and 5 has a coefficient equal to 0.8, a first degree function and a $k = 1$ while the causal link between 3 and 12 have a coefficient equal to 0.2, a fourth degree function and a $k = 4$. All the generated time series have a total length $t_{max} = 500$, unit variance and zero mean. They are assumed to be stationary and the maximum time lag τ is set to 4. To summarize, there are three type of causal links:

1. *Link from variable 0:* They are used to test different types of non-linearities with non-polynomial functions ranging from the source node to the first three nodes. In Figure 4-2 they are represented in green.

2. *Link intra-phases:* They are in the form $\beta_i^v X_{t-1}^v$ where β_i^v is a random coefficient chosen between [-1,1] linking variables i and v belonging to the same phase. In Figure 4-2 they are represented in blue.

3. *Link inter-phases:* They are in the form $\sigma_i^v (X_{t-k}^v)^k$ where the coefficient σ_i^v is chosen from [0.8,0.6,0.4,0.2] linking variables i and v belonging to different phases. σ_i^v decreases with the increasing distance between the belonging phases of the two nodes. k increases for each phase skipped by the causal link up to a maximum value given by τ. In Figure 4-2 they are represented at different shades of red.

Metrics

An ideal causal model allows us to recognize the highest number of true causal links keeping a small amount of false positives. Thanks to the realizations generated with the synthetic model we can evaluate the proposed method in a manufacturing synthetic environment. We used the following evaluation metrics:

$$Precision = \frac{\#True\,learned\,edges}{\#All\,learned\,edges} \tag{4.3}$$

$$Recall = \frac{\#True\,learned\,edges}{\#All\,true\,edges} \tag{4.4}$$

The precision indicates the percentage of correct links on all links found and therefore a high precision means that causal links found are reliable. The recall instead allows us to understand how many existing links have not been found. A high recall means that the method is not missing a significant number of connection. We also defined the false positive rate as:

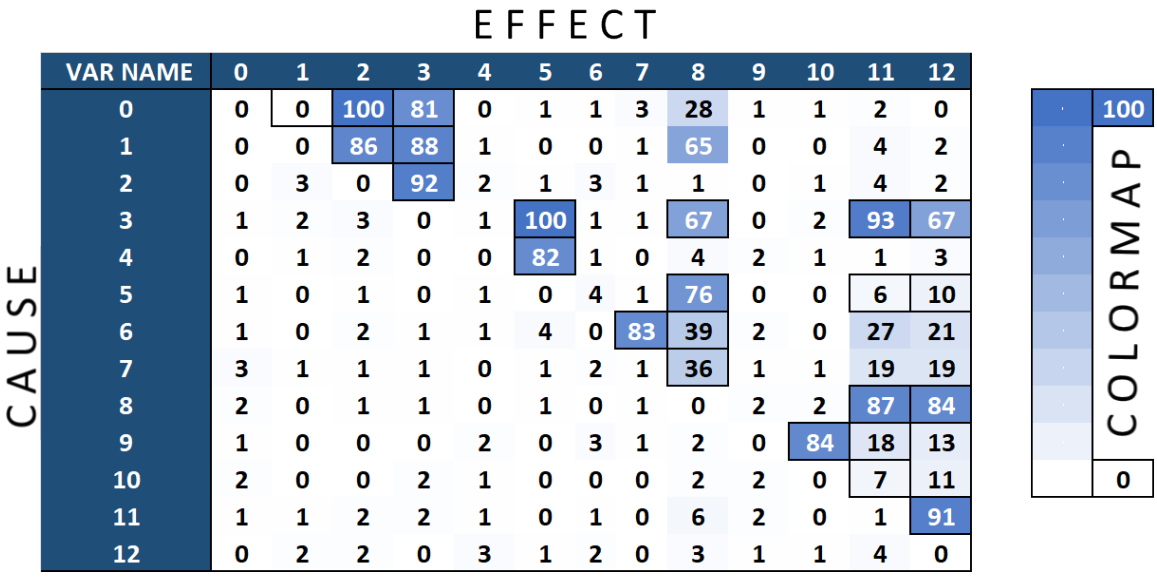
$$FalsePositiveRate = \frac{\#False\,learned\,edges}{\#All\,false\,edges\,possible} \tag{4.5}$$

This metric allows to verify false causal relationships (i.e. false positives) that lead to incorrect interpretations of the model. It is therefore essential to keep this metric close to 0.

In this experiment, unlike for the predictive procedure, we did not measure the temporal lag between cause-effect relationships. Therefore, in assessing whether a variable X is causal for the variable Y (i.e., $(\mathrm{X}) \rightarrow (\mathrm{Y})$), we investigate if at least one cause-effect relationship exists independently from the specific temporal delay between 1 and 4. The objective of this experiment is to reconstruct the flow of a plant from the data and not to recognize the temporal aspect.

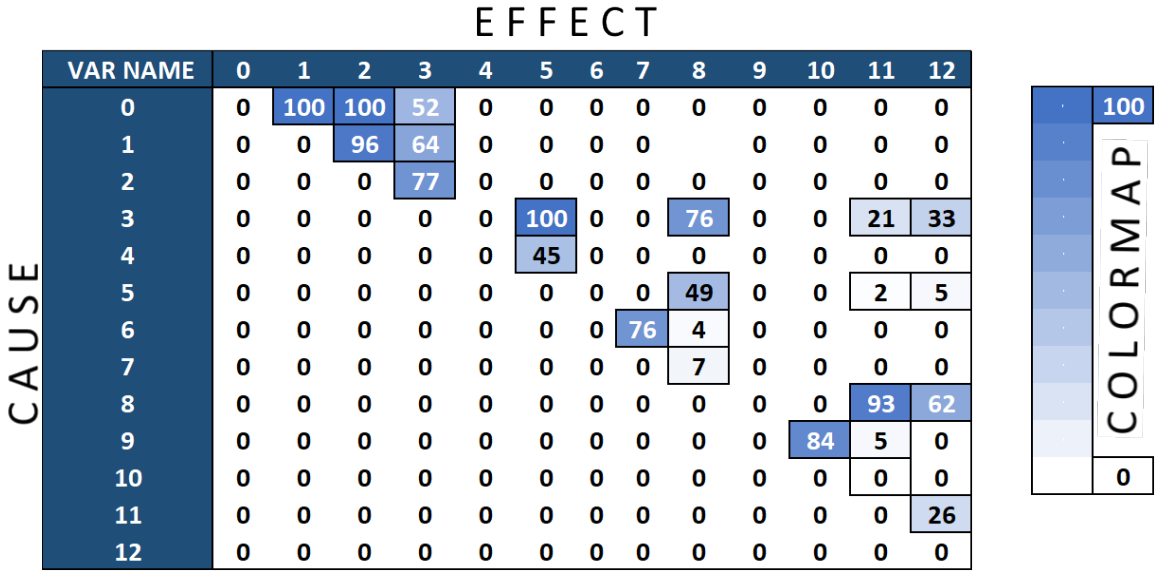
Result on Causal Discovery

To obtain a more reliable result, we created a synthetic dataset composed of 100 realizations using the parameters described by 4.2. For the PCMCI we used $\alpha = 0.05$ to obtain a lower number of false positives and therefore a sparser model. We selected $\tau = 4$ as explained in section 4.1.2. We tested both the partial correlation and CMI independence test on the considered synthetic dataset. Figures 4-3 and 4-4 sums the result for 100 realization. Bold cells are true connection on the realizations. Cell $[lines, col]$ indicates that the variable *lines* has (1) or hasn't (0) cause significance for variable *col*. A causal link is considered if PCMCI detects as significant at least one causal relationship between the two variables at any delay τ. The cells with the black border represent the causal connections which have been described in 4.2 as ground truth. Table 4.1 shows the average result for each independence test using



VAR NAME	0	1	2	3	4	5	6	7	8	9	10	11	12
0	0	0	100	81	0	1	1	3	28	1	1	2	0
1	0	0	86	88	1	0	0	1	65	0	0	4	2
2	0	3	0	92	2	1	3	1	1	0	1	4	2
3	1	2	3	0	1	100	1	1	67	0	2	93	67
4	0	1	2	0	0	82	1	0	4	2	1	1	3
5	1	0	1	0	1	0	4	1	76	0	0	6	10
6	1	0	2	1	1	4	0	83	39	2	0	27	21
7	3	1	1	1	0	1	2	1	36	1	1	19	19
8	2	0	1	1	0	1	0	1	0	2	2	87	84
9	1	0	0	0	2	0	3	1	2	0	84	18	13
10	2	0	0	2	1	0	0	0	2	2	0	7	11
11	1	1	2	2	1	0	1	0	6	2	0	1	91
12	0	2	2	0	3	1	2	0	3	1	1	4	0

Figure 4-3: Results of the PCMCI method with Partial Correlation independence test.



VAR NAME	0	1	2	3	4	5	6	7	8	9	10	11	12
0	0	100	100	52	0	0	0	0	0	0	0	0	0
1	0	0	96	64	0	0	0	0		0	0	0	0
2	0	0	0	77	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	100	0	0	76	0	0	21	33
4	0	0	0	0	0	45	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	49	0	0	2	5
6	0	0	0	0	0	0	0	76	4	0	0	0	0
7	0	0	0	0	0	0	0	0	7	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	93	62
9	0	0	0	0	0	0	0	0	0	0	84	5	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0	26
12	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 4-4: Results of the PCMCI method with CMI independence test.

the precision, recall and false positive rate metrics previously described.

Table 4.1: Precision, Recall and False positive rate for the synthetic scenario percentage values

Independence Test	Precision	Recall	FPR
Partial Correlation	0.79	0.60	0.024
CMI	1	0.51	0

Discussion on Causal Discovery

This experiment aims to utilize PCMCI to reconstruct the industrial process flow. In order to recognize a causal flow, two types of mistakes can be committed: failing to detect a causal relationship between two variables (false negative, type I error) or inferring an incorrect causal relationship (false positive error, type II). Depending on the kind of application, one mistake can have more dangerous consequences than the other. If, for example, we use PCMCI to create a decision support system for interventions, a type II error could have consequences for other phases of the process as we are intervening on a wrong variable. Indeed, in addition to failing to adjust the desired characteristic of the final product, it can affect other variables that were correctly set with supplementary adverse effects. In contrast, a type I error would only fail the adjustment of the final product quality (i.e. ineffective intervention) without interfering with other phases. In addition other decision path could be proposed to overcome suggesting alternative intervention. Therefore the consequences for false negative are more severe than false positive in a decision support system. If the causal discovery algorithm is used as features selection or preprocessing as in this thesis, failing to select a variable with a causal meaning (type I error) would have substantial consequences because it would eliminate a relevant part of the input for the response variable. On the contrary, if we insert a sensor that is not significant for the process, it would increase the dimensionality of the input without other consequences for the system. The additional dimensionality is therefore handled by the regression model.

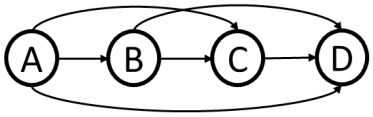
The results obtained, presented in Table 4.1, prove the capability of PCMCI to control false positives (type II). For causal relationship detection, PCMCI with partial correlation test seems to perform better than PCMCI with CMI as it shows an higher recall. On the other hand, the CMI have outstanding result on false positive. PCMCI, thanks to the additional condition on the parents performed in the second step of the algorithm, manages to ensure the absence of false positive relationships or, in the case of partial correlation, a deficient number. Figure 4-3 shows that a partial correlation test can also detect non-linear relationships from variable 0. Moreover, it can detect more relationships between intra-phases links (variables $1, 6, 7$). Recall shows 40% missing arc for a single realization, which can lead to ignoring the correct ordering for some parameters. Depending on the application, then it is possible to choose the most suitable independence test. The results show that through PCMCI, it is possible to construct more than 50% of the results without errors. These results are very encouraging for the use of PCMCI for applications such as decision support systems or feature selection for monitoring. However, in these experiments, we test the ability of PCMCI to infer the causal relationships between variables without considering the propagation delay (the time lag between cause and effect). In section 4.3 we use PCMCI to estimate the lag times between cause and effect.

4.2 Predictive procedure

When applied to industrial processes, monitoring system easily fail to depict the overall complexity of a production plant. Multiple factors can interfere during the production process and the requirements in accuracy, safety and efficiency are very high. To fully achieve the potential of monitoring, it is useful to integrate typical characteristics of the manufacturing process and build procedures that account for the domain peculiarities. Causal discovery allows to consider variables that causally influence an important quality feature of the product exploiting causation instead of correlation. In section 4.1, we use causal discovery to infer the production flow from data. In this section we present a predictive procedure which uses the causal model

discovered with PCMCI to optimize a predictive model (PM). Instead of standard diagnosis features for process monitoring, which focus on discovering root causes after a non-normal behavior occurred, we use a specific deep learning architecture that exploit causal precursors to forecast a key quality feature of final product. Therefore we optimize the monitoring system considering only the relevant variables at the correct time lag. This constraint can be applied in continuous processes where the quality of the finished product results from all the previous phases. To give an example let's consider the following four variables A, B, C, D such that $(A) \rightarrow (B) \rightarrow (C) \rightarrow (D)$, $(A) \rightarrow (D)$, $(B) \rightarrow (D)$, $(A) \rightarrow (B)$ and A, B, C are variables of machines belonging to different phases of an industrial process which all influence the quality feature D of the final product. The product transits through a phase at each time instant and thus takes four instants from phase A to phase D. The time series for the variables are defined by $[t_1, t_2, t_3, t_4]$ such that the product starts in the phase described by variable A at time t_4 and end in D at time t_1. A representation of the graph, the SCM and the database for the example is given in Figure 4-5.

1



2

$$A_t = \eta^A$$
$$B_t = \delta_1 A_{t-1} + \eta^B$$
$$C_t = \delta_2 A_{t-2} + \delta_3 B_{t-1} + \eta^C$$
$$D_t = \delta_4 A_{t-3} + \delta_5 B_{t-2} + \delta_6 C_{t-1} + \eta^D$$

3

Database	A	B	C	D
t_4	A_{t_4}	B_{t_4}	C_{t_4}	D_{t_4}
t_3	A_{t_3}	B_{t_3}	C_{t_3}	D_{t_3}
t_2	A_{t_2}	B_{t_2}	C_{t_2}	D_{t_2}
t_1	A_{t_1}	B_{t_1}	C_{t_1}	D_{t_1}

Figure 4-5: (1) Graph of the example described in 4.2. (2) Structural Causal Model (SCM) for the example, where $\delta_{1...6}$ are coefficients (for a full description of SCM refer to section 2.3.1). (3) Database table of the example.

Following formula 1.3.1 standard ANN model to predict the values of variable D at time t_1 consider the past value from t_4 to t_1 for each variable such that

$$D(t_1) = \sum_{j=2}^{4} g_j(\int A, B, C, D\beta_j(t_j)dt) + e(t) \tag{4.6}$$

However the constrain given by the production flow allows to state that the product at time t_1 is not affected by variable A at time $\{t_3, t_2, t_1\}$, B at time $\{t_4, t_2, t_1\}$ and C at time $\{t_4, t_3, t_1\}$. This suggest that we can use causal information to optimize the predictive model. Indeed, the final product is causally affected from the other machines only at specific time lag. Thanks to the causal model inferred with PCMCI it is possible to estimate which variables influence the final product and their specific lag delay between cause and effect.

This thesis introduces a two-step forecasting procedure that uses the causal relationships typical of a manufacturing process to improve predictive models (PM).

1. In the first step, we identify causal variables and their actual temporal activation through PCMCI, optionally, including domain expert knowledge. This means that we infer from data which are the variables that can potentially change the outcome of the final product as they are cause for the key quality feature in the final product.

2. In the second step, we integrate causal precursors detected with a causal discovery algorithm (i.e., variables that generate an effect with a specific time delay) in the Separable Temporal Convolutional Network (S-TCN) to forecast the desired response variable. The architecture of the S-TCN is adapted to the causal model and and, as demonstrated in section 4.3.2, allows to reach temporal recurrences more efficiently than standard TCNs that explore the entire temporal space of the variables.

Causal vector

The transition between the two steps is managed through the causal vector that allows to bind the output of PCMCI with the architecture of the S-TCN and integrate expert operators' knowledge. From the output of PCMCI we build a vector of size $|\mathbf{X}|$ denoted as "causal vector" for a specific variable that represent the response variable in the predictive models (i.e. the desired quality variable to be monitored). In the causal vector, each input feature $x^- \in \mathbf{X}$ is associated with the corresponding

causal activation time on the response variable. If the input time series is not a causal precursor for the response variable, the time lag is set to 0. Note that in a manufacturing setting, it is likely that many variables could be associated with the same cause-effect temporal activation, for instance, when they belong to the same phase of the process, they generate their effect at the same time lag on the response variable. In Figure 4-6 we represent an example of the first step of the procedure. It is possible that PCMCI detect causation at multiple time delay for the same variable. We select the median time lag (i.e., one time lag). Alternatively, the causal link with the highest statistical confidence value could be chosen. We gave more importance to the proximity of multiple time gaps as it allows a more robust selection than the confidence value for real data. It is straightforward for a domain expert to interpret the proposed causal vector and to check and eventually correct the identified causal relationship. The advantage of causal vector is therefore twofold: it checks that the causal relationships found by PCMCI are correct, eliminating any false positives with domain knowledge, and at the same time it proposes a set of alternative for intervening on the key quality feature on the product. The causal system can debunk some assumptions made by operators using the correlation between machine parameters. In the next section we explain how the S-TCN use the causal vector to obtain more distant temporal modeling.

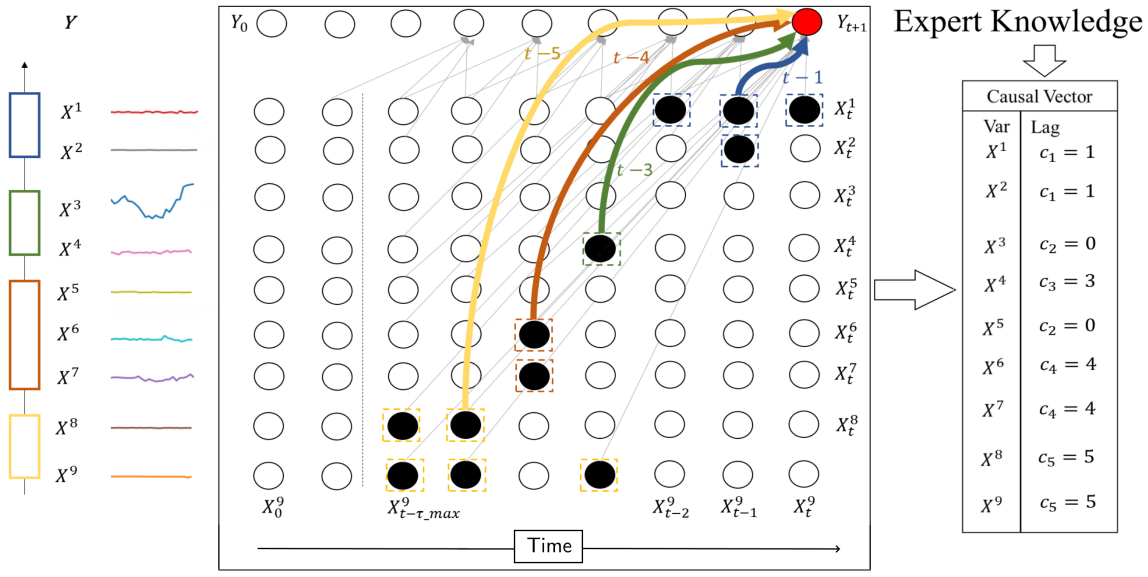


Figure 4-6: In the first step of the procedure, PCMCI is used to build the causal vector. We select the causal precursors of the response variable Y and their temporal activation between lagged values. τ_{max} is the potential maximum time connection of the underlying system that in the manufacturing entails the length of the production process. The colored rectangles represent the different machines of the process while black nodes are causal precursors for Y. Arrows show causal connections between time-lagged variables and Y. As example variable X^4 is causal for variable Y after 3 time lags. It means that if we intervene on variable X^4, the values of Y changes after three temporal instances. Arrows with the same causal activation time have identical colors. The variables that are not causal precursors of Y are set to 0. If PCMCI detect causation at multiple time delay as for variable X^1 in the figure, we select the median time lag (i.e., one time lag). The causal vector report for each variables the corresponding time lag. Experts operator can change the detected causal vector adjusting the time lags and modifying the vector.

In the next section we present the S-TCN architecture. In section 4.3.2 we present a large number of numerical experiments done to validate the proposed architecture and compare the results with state-of-the-art methods.

4.3 Forecasting with S-TCN

In the second step of the procedure, we use a temporal neural network named S-TCN to build a predictive model (PM) for response variable Y that exploits the inferred causal relationships and effectively learns temporal distant activation. Enhanced temporal modeling (i.e. wider receptive field) is obtained in standard TCN by adding layers, resulting in an increased complexity and reducing the interpretability of the PM. Instead, the proposed S-TCN architecture uses depth-separable blocks that directly convolves distant temporal activations optimizing the receptive field for specific features rather than exploring the entire temporal space. This is made possible by the causal vector calculated in the previous step, which contains causal relationships including the distance of the temporal connections. The proposed S-TCN architecture is represented in Fig. 4-8.

4.3.1 S-TCN

All TCN models are based on dilated causal convolution, which applies a dilated kernel to gain a wider receptive field and prevent leakage from future input. In a standard TCN, the dilation factor d increases exponentially to the layer's number depending on the chosen dilation coefficient c, such that $d = c^l$ for layer l. With an exponentially increasing dilation factor d, a network with stacked dilated convolutions can operate on a coarser scale without loss of resolution or coverage [151, 22]. The receptive field of a standard 1D TCN with a kernel of dimension K, L layers and dilation coefficient c is given by:

$$ReceptiveField = 1 + \sum_{l=0}^{L} (K-1) \cdot c^l. \tag{4.7}$$

If we set the kernel dimension $K = 2$ and the numbers of layers to $L = 1$ we can control the receptive field of the layer through the dilation coefficient c. With this setting, the TCN network is simply a single 1D causal convolution on the temporal

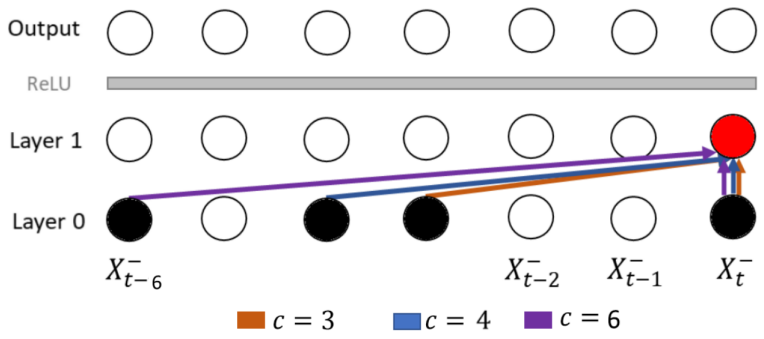


Figure 4-7: The architecture of the Depth-Separable block described by eq. (4.3.1). The dilatation coefficient c given as input to the block controls the size of the receptive field.

axes where the kernel convolves value t and $t-c-1$ and the receptive field is of length $c+2$. 1D causal convolution applies a convolution operation on the input signal where the kernel is padded to prevent leakage from the future. This convolution is applied on all the channels of the inputs mapping signals into features.

The depth-separable block is composed by a similar convolution and a ReLU activation function. Given a multi-dimensional input sequence $\mathbf{X}$, the output features s at location t, with dilation coefficient c and kernel k we describe the depth-separable block with the following formula:

$$s(t,c) = ReLU(X *_c K)(t) = ReLU(\sum_{w=0}^{w=1} k(w)X(t-(c)\cdot k)) \tag{4.8}$$

where ReLU indicates the rectified non-linear unit function and the convolution is assumed to be causal. The peculiarity of the depth-separable block relies on setting the dilatation coefficent c according to the precise estimated temporal delay.

Given the causal vector, we group each input variable that has the same causal activation time on the response variable. For each identified group $g_k \in \mathbf{G}$, we create a unique depth separable block where the dilation coefficient c_k corresponds to the causal activation time contained in the causal vector. Each group will be separately convolved at specific dilation given by c_k. For instance, as represented in Fig. 4-6, the two variables x^6, x^7 have the same effect on the response variable at time t_{-4}, they will

be processed separately from the other variables by a specific separable block with $c = 4$. This operation takes place also for non causal variables that are convolved with a standard kernel and can be identified in the causal vector by their causal activation time set to 0. The outputs of depth-separable blocks are concatenated on the feature's axes to create a single feature map. The size of this new input is given by the output of a single depth-separable block (that in our setting is 50 features) multiplied by the number of identified groups. This new input contains the aligned temporal information thanks to the different dilatation factor c_k applied by the previous separable blocks layer.

We apply a TCN on this new input to model non-linear relationships between different groups. We have limited the TCN depth to four layers, since using the new input instead of the original features allows reaching the furthest connections without increasing the number of layers. The goal of the TCN is to model the temporal relationship among features that belong to different machines and add non-linearity through the activation functions.

4.3.2 Experiment on S-TCN

We validated the S-TCN architecture with an extensive numerical experiment. The main purpose of the numerical experiment is to validate the proposed architecture and measure its limits on modeling temporal dynamics. We compared the proposed approach with different state of the art methods. Given the difficult interpretability of TCN and the significant resources required for training neural network models, rarely extensive numerical tests are carried out on synthetic models to analyze the actual predictive potential given by temporal features. We show that the S-TCN architecture learns connections over a wider temporal frame compared to different machine learning algorithms on a large class of synergetic non-linear discrete-time stochastic processes.

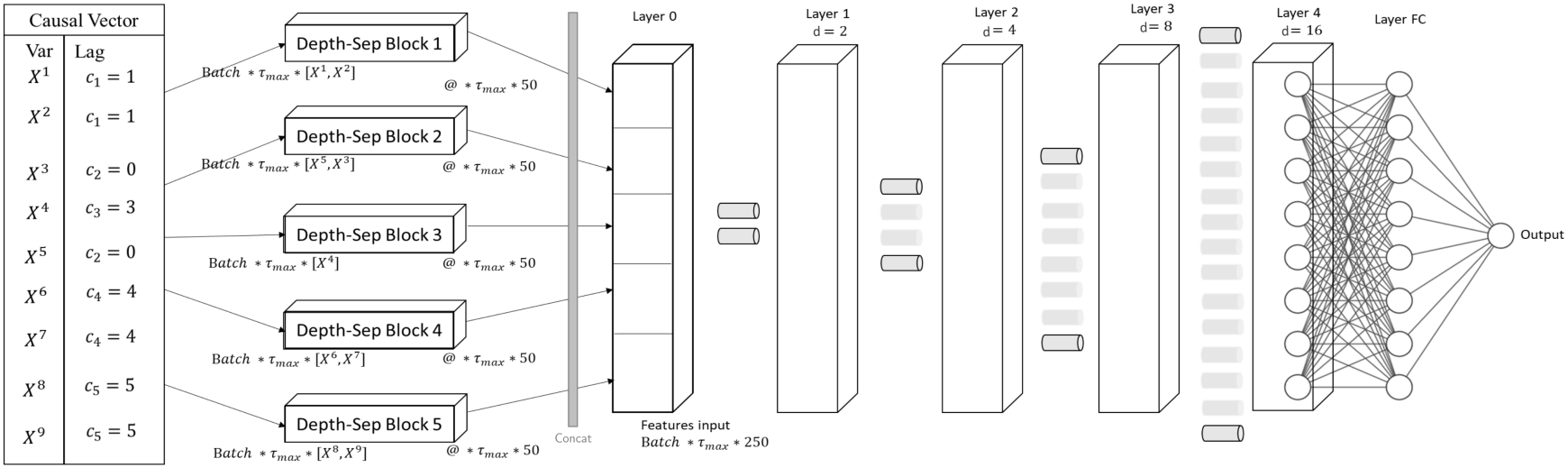


Figure 4-8: The architecture of the S-TCN is created from the structure of the causal vector. Depending on the cardinality of $\mathbf{G}$ which represents the groups of variables with the same effect lag on the response variable (i.e., $c_{\_}$), $|\mathbf{G}|$ depth-separable blocks are generated where the kernel dilation corresponds to the lag $c_{\_}$). In each of these blocks, variables with a specific dilation are processed, producing a features map S of dimension batch size $*$ the maximum time length of the process $(\tau_{max}) *$ 50. The output of these blocks will contain information about the signals at a specified time delay. These new feature maps are concatenated into a vector (layer 0 in the Figure) which have batch size$^*\tau_{max} *|\mathbf{G}|$ dimension. This new input is processed by a TCN of 4 layers where the kernel dilation d is set to $[2, 4, 8, 16]$ respectively. The TCN allows increasing the network's ability to detect non-linear relationships on the new input. Finally, a fully connected layer with a single output is used to predict the quality variable.

Dataset

Similarly to [47], we define:

$$Y_t = \mathrm{a} \cdot \mathrm{g}_{\mathrm{lin}}(\mathbf{X}_\tau^-) + \mathrm{b} \cdot \mathrm{g}_{\mathrm{syn}}(\mathbf{X}_\tau^-) + \eta_t \tag{4.9}$$

The linear function $\mathrm{g}_{\mathrm{lin}}$ is the sum of 10 randomly chosen subprocesses $\mathbf{X}_{t-\tau}^{(\cdot)}$ where $1 \leq \tau \leq \tau_{max}$ and τ belongs to a subset $\mathbf{T}_{\tau_1-\tau_5}$ of 5 random lags. On the other hand, the non-linear function $\mathrm{g}_{\mathrm{syn}}$ is the product of 5 randomly chosen subprocesses (excluding process Y and the ones already included in the linear term). The coefficients are fixed to $a = 0.2$, $b = 2$. The stochastic noise driving response variable Y is represented by η. We report in Table 4.2 the ground truth causal vector corresponding to an example realization of the model with $\tau_{max} = 30$. We randomly choose 5 values for $\mathbf{T} = \{\tau_1 = 1, \tau_2 = 9, \tau_3 = 15, \tau_4 = 20, \tau_5 = 22\}$ with $1 \leq \tau \leq 30$. The causal vector is generated assigning to each variable $\in \mathbf{X}^-$ a corresponding time lag $\in \mathbf{T}$. The response signal Y is generated following the formula (4.3.2) so that each signal causes the effect on the response variable with a specified delay always inferior to τ_{max}.

Var	**Time-Lag**	**Var**	**Time-Lag**	**Var**	**Time-Lag**
X_1	20	X_2	9	X_3	15
X_4	1	X_5	22	X_6	15
X_7	22	X_8	22	X_9	15
X_{10}	9	X_{11}	20	X_{12}	1
X_{13}	9	X_{14}	20	X_{15}	1

Table 4.2: Ground truth causal vector corresponding to a single realization with $\tau_{max} = 30$. Example explaned in section 4.3.2

Metrics

For the validation of our models, we use mean squared error (MSE) and consider a training composed of 40000 samples while the test set has 10000 samples. Since the

response variable is dependent only by the described equation (4.3.2) the prediction in the accuracy is given exclusively by the efficiency of the algorithm to model temporal dependency and by the correct identification of the causal precursors. This allows us to effectively evaluate the temporal modeling capabilities of the proposed S-TCN architecture focusing only on temporal features and providing a demonstration of its potential over a wide time interval.

Comparison

Following, we present the machine learning methods we used as benchmark:

- **LSTM**: Long Short-term Memory (LSTM) is a common recurrent neural network that demonstrated strong ability on modeling temporal information. Recurrent neural networks are more suitable for time series modeling than convolutional networks since they use memory gates. However, their complexity, requires more computation time, the parameters optimization is challenging and the temporal context interpretation is not intuitive.

- **Short-TCN** A TCN consists of a CNN architecture with a 1D kernel in which each layer widens the temporal window analyzed. In this experiment with "Short-Tcn" we designate a TCN with only four layers identical to the one used in the final part of our S-TCN architecture presented in Fig. 4-8. This TCN is able to model time series up to a maximum τ of 32 that is its maximum receptive field. Therefore if the temporal activations are more distant than the maximum receptive field the performance will be the same as the trivial case (see below).

- **Long-TCN** We indicate the TCN with the maximum receptive field. In our experiment, this neural network has a depth of 9 levels, increasing the receptive field exponentially up to $\tau_{max} = 256$. This TCN has the standard setup used in predictive model [22].

- **Trivial** The trivial model always outputs the mean of the response variable.

The trivial case is used as a reference for the other models to test whenever they are learning from the temporal context.

- **RF** In an ideal setting with the causal vector we could skip the temporal convolution and align the features in input to apply a non-linear regression without considering the temporal axes. We tested random forests even if they are not suited for temporal modeling. The input to this network is indeed shifted directly to the precise delay of the signal thus eliminating the temporal context and reducing the stochastic process to an instantaneous non linear configuration. Random forests in this setting allow a reference for results obtained by an optimal temporal prediction.

Given the huge number of models and realizations to compare, it was not possible to optimize each neural network individually. In the numerical experiment validation, the parameters of the neural networks are standardized to obtain comparable results rather than optimized as absolute values. Therefore we have kept the same number of nodes per layer within the networks, the same number of epochs, optimizer and learning rate. We used 50 nodes for each layer, a learning rate of 0.01 with ADAM optimizer [42]. We train each network for 20 epochs decreasing the learning rate after 15 epochs with a batch size of 32. The development framework used is Pytorch with an Nvidia 1060 GPU.

Result on S-TCN

To demonstrate the ability of the S-TCN architecture to model more distant cause-effect relationships, following the formula (4.3.2), we generated 70 realizations reported in the appendix. We choose 7 increasing τ_{max} from 30 to 210, with steps of 30. For each maximum time lag we created 10 realizations for a total of 70 realizations. In Fig. 4-9 we show the boxplot of the results for each maximum time lag τ_{max}.

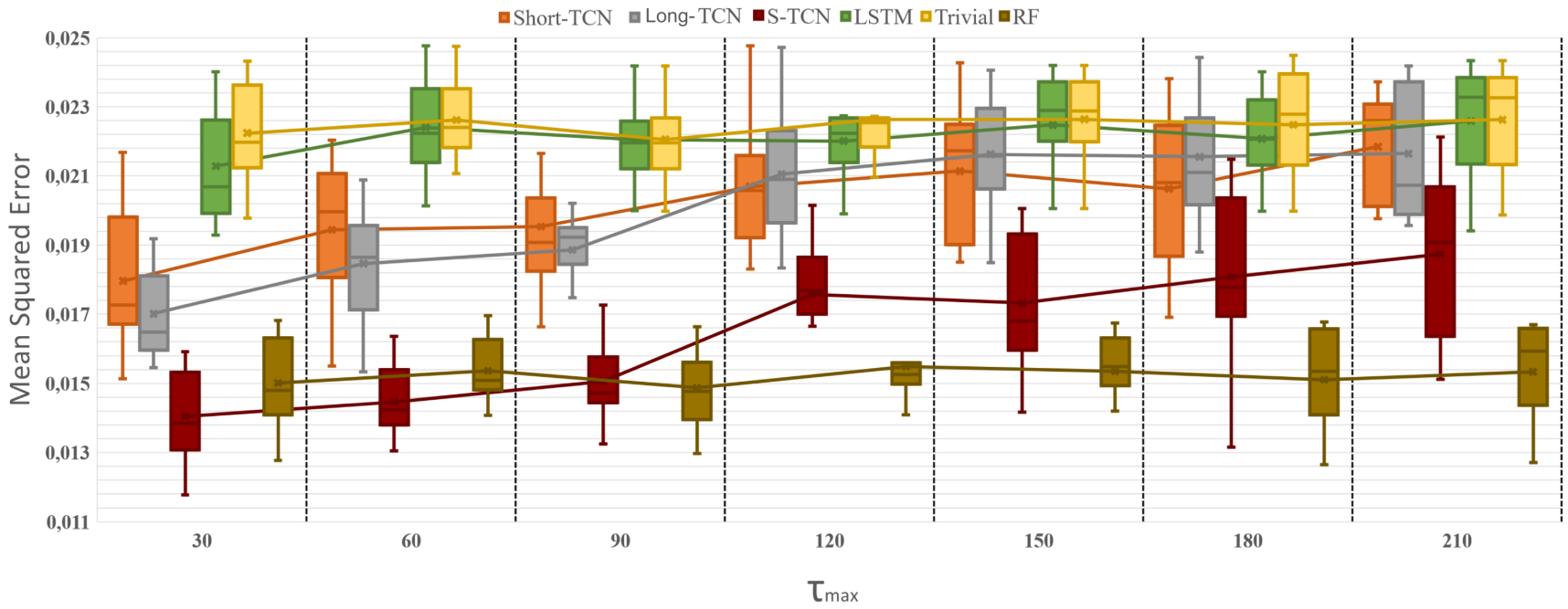


Figure 4-9: Box plots of numerical experiments. The colored lines represent the average of the result of the ten different realizations at corresponding maximum time activation. The more the τ_{max} increases, the more likely distant causal connections occurs. The results for each experiment with the correct temporal activation are attached in the appendix.

Discussion on S-TCN

In this experiment, we focus on testing if the introduction of causal precursors allows achieving the same accuracy as a deeper network that analyzes the signals in its entire temporal context. It means demonstrating that the information contained in the input can be modeled more efficiently thanks to the knowledge of the underlying causal structure. Therefore, the experiment must demonstrate that the combination of causal discovery and S-TCN architecture is more convenient than a standard multi-level TCN. The results presented in the appendix show that more efficient modeling of the dimensionality of the input variables results in better model accuracy and greater efficiency. Accuracy decreases as the maximum distance of τ increases for all the algorithms, and S-TCNs can learn distant temporal connections efficiently. We expected the S-TCN network to perform better than the Short-TCN as the latter is limited to forecast time gaps closer than $\tau = 32$. On the other hand, the Long-TCN network, thanks to its high depth of 9 levels and $\tau_{max} = 256$, should be able to model all the considered realizations. However, the numerical experiments show that S-TCN can manage larger temporal connections more efficiently and it is less affected than Long-TCN by the increase in complexity given by a larger receptive field. Numerical experiments show that the random forest algorithm achieves the best overall accuracy. This is not surprising as it runs in a different configuration where the input features are aligned during pre-processing, as explained above. Since it is not modeling the temporal axes, it fits a reduced dimensionality and obtains better predictions in the synthetic setting regardless of temporal distances. A reader might observe that it is still convenient to perform pre-processing to remove the time dimension as done for random forest, instead of training a specific temporal model. However, performing such a complicated pre-processing in an industrial scenario, is very challenging as the productions could have clogging or stops that should be properly addressed.

4.4 Conclusions

In this chapter, we presented a two-step procedure that uses a causal discovery method, called PCMCI, and a deep learning network, called S-TCN, to build a prognosis system for a key quality feature of the final product in a continuous manufacturing process. The advantages of the presented procedure are the following:

- Integrating the product flow in the monitoring system
- Integrating expert domain knowledge
- A novel neural network architecture

These features allow the prognosis of the process even in SMEs, reducing the complexity of the management.

In section 4.1 we use causal discovery to reconstruct the causal relationships in a plant. We first present PCMCI, a causal discovery method with a strong control on false positives. We created a synthetic model of a manufacturing process and proved the ability of PCMCI to discover both: linear and non-linear causal relationships. The results showed that it is possible to reconstruct 50% of the causal relationships without generating false positives. Causal relationships make it possible to establish an order between the different phases and reconstruct the production flow. In section 4.2 we present the predictive procedure that combines the output of PCMCI with a predictive model through the use of a causal vector constructed on the inferred causal model. In section 4.3 we present the predictive model that evolves from TCN and is called S-TCN. The S-TCN can use the causal precursors given by the causal vector to reach distant temporal information more efficiently. We build an extensive numerical experiment reported in section 4.3.2 ad in the appendix. S-TCN outperformed state-of-the-art predictive models, especially in the case of distant temporal recurrence.

Publications: Most of the results presented in the paragraph have been already published in the following papers:

- G. Menegozzo, D. Dall'Alba and P. Fiorini, "Industrial Time Series Modeling With Causal Precursors and Separable Temporal Convolutions," in IEEE Robotics and Automation Letters, vol. 6, no. 4, pp. 6939-6946, Oct. 2021, doi:10.1109/LRA.2021.3095907.

- G. Menegozzo, D. Dall'Alba and P. Fiorini, "Causal interaction modeling on ultra-processed food manufacturing," 2020 IEEE 16th International Conference on Automation Science and Engineering (CASE), 2020, pp. 200-205, doi: 10.1109/CASE48305.2020.9216973.

Chapter 5

Case Study: An ultra-processed food SME

5.1 Executive summary

This thesis underlines the importance of achieving a fourth industrial revolution which accounts for SMEs as they represent most companies in the European and Italian context. In chapter 1 is highlighted that SMEs struggle to handle the increasing complexity of the process compared to large enterprises. Therefore monitoring technologies are proposed as support for process management.

In a first contribution presented in chapter 3, an automatic recognition method for industrial process' phases is presented. This system is based on TDNN that learn specific features to classify phases, facilitating process analysis. TDNN use convolutional operation on the temporal axis to recognize complex movements and action from an heterogeneous set of sensors. The management of the process is therefore improved by the automatic system as it allows to subdivide the production in multiple distinct phases from data. This feature is useful in collaborative process between human and robots or in production were the phases cannot be synchronized using fixed temporal constraints. TDNN are validated on three dataset, two of which simulate an industrial scenario while the other is used as benchmark.

In Chapter 4, firstly a reconstruction of the product flow using causality is pro-

posed. Modeling causal relationships within a plant allows describing how changes are propagated throw the production. Therefore, given the inferred causal model of the plant, a predictive procedure that monitors the product's quality is suggested. The predictive procedure allows anticipating deviations during production. The advantage for companies are twofold:

- New protocols are proposed, suggesting how to intervene to improve product qualities through causal relationships;
- Product quality is monitored following process flow and integrating operator experience;

The validity of the contributions proposed in chapter 4 has been proved with numerical experiments on synthetic models that reproduce the characteristics of an industrial plant. These experiments have demonstrated the effectiveness of the predictive model (S-TCN) in processing information distant in time and of the causal discovery method (PCMCI) in recognizing the flow of an industrial process from the data.

Synthetic models, however, fail to simulate behaviors due to human interventions. Indeed, while in a synthetic experiment, the underlying causal model describes the causal relationships, in SMEs, the operators' protocols for interventions can vary according to multiple factors (correlation between parameters, accessibility to the machine, cost of the intervention, and so on). Therefore, it is necessary to compare the causal model deduced through PCMCI in a real case scenario to evaluate its effectiveness in proposing unexplored causal structures.

The predictive procedure must also be assessed in a real case scenario. In particular, applying the procedure to a real case allows comparing the improvement on the monitoring system given by data-driven causal discovery method rather than domain's experts.

In this chapter we apply the contribution presented in chapter 4 in a medium enterprise which produce ultra-processed food. The manufacturer considered does not present collaborative processes, therefore, the automatic system for the recognition

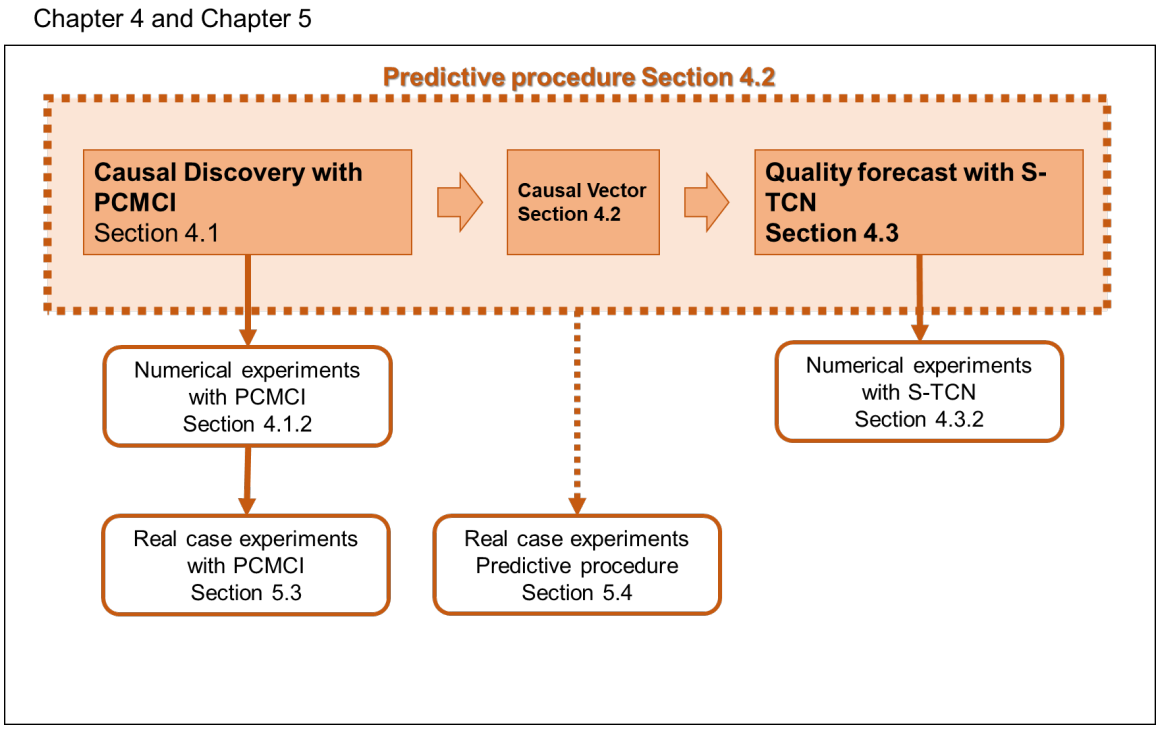


Figure 5-1: Summary of chapter 4 and 5.

of phases presented in chapter 3 was not implemented. In section 5.2 we introduce the context of ultra processed food manufacturing. In section 5.3 we present the result of applying causal discovery on the process proposing a new set of intervention for the company. In section 5.4 we use the predictive procedure for a key quality variable of the final product. The conclusion are outlined at the end of the chapter. In Figure 5-1 we report a summary of chapter 4 and 5 to facilitate understanding of the experiments performed.

5.2 Ultra-processed food manufacturer

The food and drink industry is the largest manufacturing sector in the EU economy employing directly 4.25 million workers [152]. It processes 70% of EU agriculture, besides being the largest global exporter of food and drink products [153]. The food and drink industry is the EU's most significant manufacturing sector in terms of jobs and value-added [153]. In Italy, the agro-food industry is the second largest national

manufacturing sector. With 132 billion euros of turnover, it covers 13% of national industrial production and the value of exports amounts to 29 billion [154]. The European Prospective Investigation Into Cancer (EPIC) showed that in the Nordic and central European regions, highly processed foods are the dominant source of nutrients, accounting for between 50% and 90% of nutrient intakes, with the exceptions of just 2 nutrients, vitamin C and beta-carotene [155]. Ultra-processed food are industrial formulations with 5 or more (usually many) ingredients[156]. Ultra-processed food companies are specialized in reducing natural raw material variations and provide customers with products with more homogeneous characteristics cite124. Due to the high variability in natural ingredients, it is complex to develop production protocols to obtain uniform outcomes. Thus, all along the production chain is frequently necessary that an expert operator modifies the machine's parameters in upstream stations (cause) to obtain better downstream products properties (effect). In ultra-processed food manufacturers, indeed, the production process is generally continuous, and the final product is strongly affected by the industrial process [157]. The frequent interventions by operators, the variability of raw materials and the final product's homogeneity requirement represent important characteristics for the development of monitoring systems. Thus, monitoring systems allow identifying and predicting possible variations and maintaining a stable quality for final products.

In this case study, we have considered an Italian SME that produces ultra-processed food. This medium-sized company represents an excellent reference for the use of causal discovery and predictive monitoring procedure. The automatic recognition of phases however was not considered for the real case scenario. The company does not present collaborative processes between man and machine and the process phases are pre-defined. Below we will present the experiments carried out in the company, the data used and the results obtained.

5.3 Experiments on Causal Discovery

In this experiment we used PCMCI described in chapter 4 to reconstruct cause-effect relationships within the plant using a data-driven approach. We test the ability of PCMCI to describe the product flow through the different production phases. In the next section we present the dataset used and the pre-processing operations carried out, following, we will describe the validation methodology and the results obtained.

5.3.1 Dataset description

The raw data are obtained from Programmable Logic Controller (PLC) and Supervisory Control And Data Acquisition (SCADA) of an industrial plant for ultra-processed food production and contain both set points and parameters. The sampling was carried out every 5 minutes while the total production cycle takes approximately 3 hours, from raw ingredients to final semi-finished products. Data has been acquired for three consecutive weeks, for an overall 363 hours of production. The production process consists of ten sequential phases. We apply a pre-processing guided by both distribution of data and expert's domain knowledge. From the ten phases, we have removed those that do not affect the characteristics of the final product. Therefore, we consider five relevant sequential phases with a total of 65 parameters. We further reduce the parameters (from 65) to 35 by ignoring feature that are unique keys, features with zero variance and categorical features. Moreover, we remove binary parameters and integrate domain experts' knowledge to select variables that don't contain useful information. These are recognizable from a data-driven perspective since they are kept constant for at least 60 hours. Furthermore, considering the restricted sampling period, we eliminated non-stationary variables. In three weeks, there are no product features variations and temporal deviation is caused by external noise. To test the stationarity of the remaining features, we applied the Augmented Dickey-Fuller unit root test with a p value of 10% [158]. We select a large p value as data are noisy and present unlikely values. The automatic identification of outliers is not possible because we do not have the specific confidence intervals for each parameter, therefore we

applied a moving window averaging with 30 minutes support to reduce the influence of improbable values. It is possible to describe the complete production of a specific semi-finished product with 6 temporal instances. The resulting dataset is composed of 5 phases represented by 14 process variables normalized with unit variance. The total number of time samples after pre-processing is 924.

5.3.2 Evaluations

To compute the metrics used in the synthetic realization the true causal model of the plant is required. However it is not possible to reconstruct the set of equations of the industrial plant in the real scenario. Based on expert's domain knowledge, we reconstruct the belonging of parameters to different process phases and we sort them according to the sequential order given by the process flow. We use PCMCI with CMI and partial correlation independence test as explained in section 4.1.1. In this experiment, we focused on identifying a possible cause-effect between two variables without considering the exact time lag between cause and effect. We identified a cause-effect relationship between two variables if there are at least two different causal links for each pair of variables. This means that the variable X has a valid causal effect on the variable Y if a causal link is found at $\tau = a$ and $\tau = b$ with a, b between 1 and 6. This is equivalent to considering only the causal effect that has at least two time gap distances for causal effect propagation. The output of PCMCI will be a bi-dimensional matrix is a $|\mathbf{X}| * |\mathbf{X}|$ without the τ_{max} dimension. In other words, we do not try to estimate the time lag of the causal effect between two variables but we infer the existence and consistency of the causal effect between two variables at different time gaps.

5.3.3 Results

We, therefore, show the tables of causal links found highlighting which phases they belong to. The parameters chosen for PCMCI are $\alpha = 0.05$ and $\tau = 6$. The choice of τ is explained in the dataset description and it entails the lenght of the product. We

choose a small α to obtain more reliable results as explained in section 4.1.1. Figures 5-3 and 5-2 show results obtained, with the inferred causal links. Cell $[lines, col]$ indicates that the variable *lines* has (green) or hasn't (white) cause significance for variable *col*. Different colors used on diagonal shows belonging phase of variables. Phases are ordered according to expert's knowledge to ease the interpretation of causal link found. For example the green cell $[1, 7]$ indicates that the variable 1 which belong to the first phase of the process have a causal effect on the variable 7 which belong to the second phase of the second phase of the process. Figures 5-3 and 5-2 recall Figures of section 4.1.2, however instead of the bold cells representing the ground truth cause-effect relationships, the belonging phases assumed from expert knowledge for each parameter is highlighted.

Figure 5-4, show the proposed causal interaction model for in line-intervention. The model is build on the result of PCMCI with CMI independence test. We use expert knowledge to choose the direction of the arrows in ambiguous cases where the PCMCI algorithm cannot distinguish between cause and effect (i.e., (A) $\rightarrow$ (B) and simultaneously (B) $\rightarrow$ (A)).

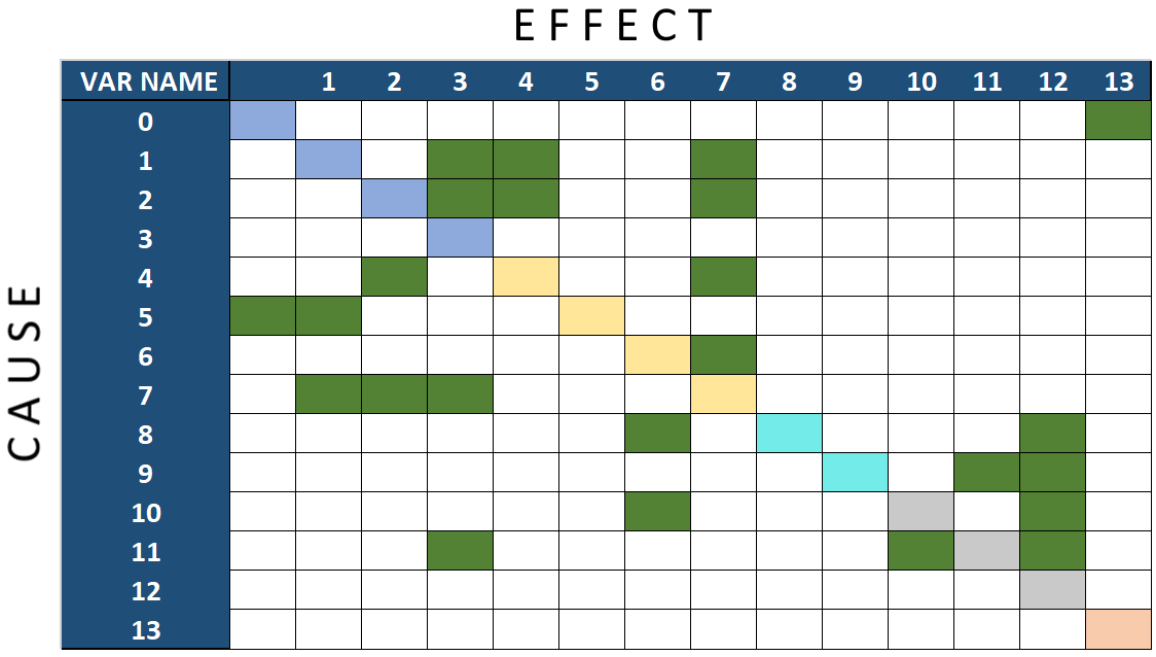


Figure 5-2: Results on real case scenario of the PCMCI method with Partial Correlation independence test. For more details refer to section 5.3.3.

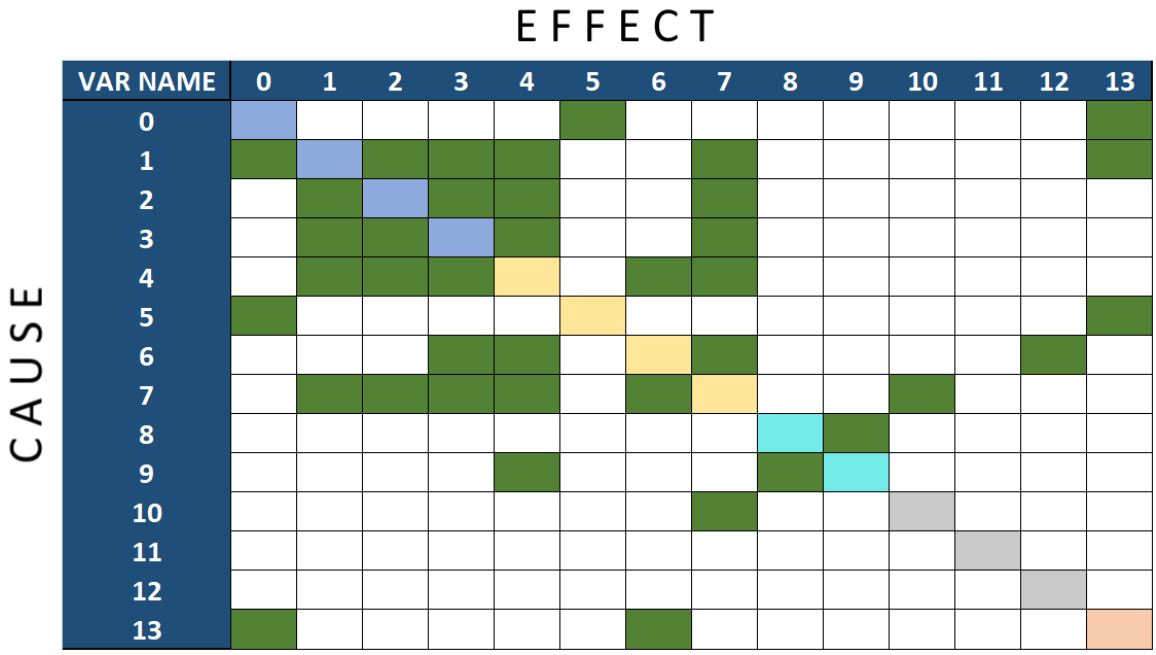


Figure 5-3: Results of the PCMCI method with CMI independence test. For more details refer to section 5.3.3.

5.3.4 Discussion on Causal Discovery

In this experiment, we used PCMCI to infer the flow of a product throw the industrial plant of an SME from the data. Since, as explained in section 5.3.2, it is not possible to precisely establish all causal relationships within the plant, we cannot provide metrics for the evaluations. However, given the known machines' orders in the production process, we can provide the violations of the inter-phases link for the real case scenario. Looking at Figures 5-3 and 5-2 it is evident that CMI independence test shows a higher detection rate than the Partial correlation. In addition, the CMI is more symmetrical, highlighting that the method recognizes relationships between variables but struggles to distinguish the cause and effect. To facilitate the interpretation of the results, in Figure 5-4, we reconstruction the industrial plant obtained with PCMCI and CMI. We used expert knowledge to group the parameters belonging to the same phases. Furthermore, if a bidirectional link is found, we have chosen the arc that followed the direction of the process provided by the experts. Many causal links have been found in the first two phases considered. The link between variables 9 and 4 can lead to the

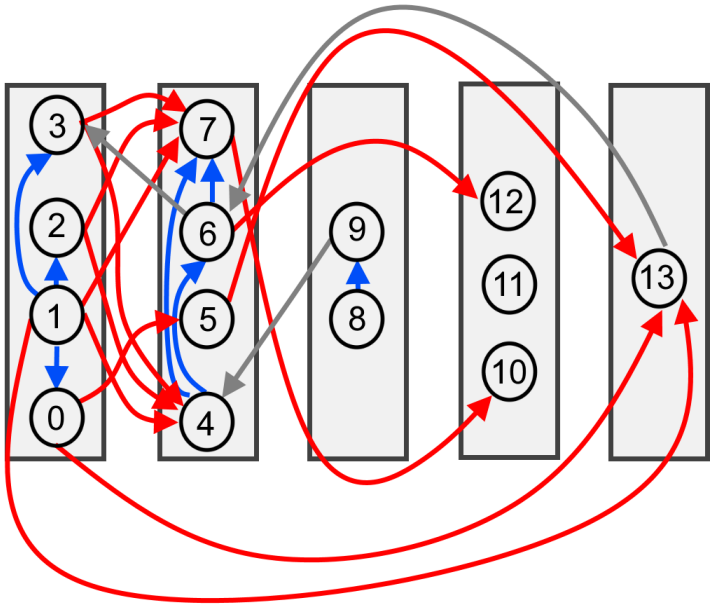


Figure 5-4: Proposed causal interaction model. Rectangles represent the five selected phases of the ultra-processed food production process. The red links are correct causal relationships inter-phases that allow the flow recognition of the process. Grey links are false positives. Blue links represent the causal interactions between the parameters in the same phase

incorrect exchange of phases 3 and 2 as no other link has been found. On the other hand, the product flow for phases $1, 2, 4, 5$ is correctly detected. It is more complex to evaluate the correctness of the proposed intra-phase links. However, the absence of cycles allows us to set an order of intervention even among internal parameters of the same phase. We establish a set of interventions to modify the product's final variable and compare them with those used in the company. In Table 5.1 we report the number of interventions applied on the line during data collection. PCMCI has found fewer causal links where the operators intervene most frequently for changing set points. The operators intervene in the last phases of the process to have more direct consequences on the final product features. Modification of the set-points seems to negatively affect data-driven causal discovery due to the manual update of the parameters setting. However, the analysis of this aspect is out of the scope of this work and it will be the research goal of future works. The proposed causal model can provide an alternative intervention protocol for changing the production

Var Id	N. Interventions	Var Id	N. Interventions
0	4	7	4
1	5	8	16
2	5	9	15
3	4	10	32
4	4	11	32
5	4	12	37
6	5	13	6

Table 5.1: number of in-line intervention on the set point of the machine's parameters

set-points. Thanks to the inferred causal interaction model, it would be possible to carry out interventions on upstream processing stations of the process line and therefore intervene with greater timing and precision on the final product properties.

5.4 Experiments on predictive procedure

In the real case study, the predictive accuracy suggests the ability of the procedure to model a manufacturing process. The validation on a real case demonstrates the effectiveness of the two-step procedure by considering both the contribution given in feature selection with causal precursors and the efficiency in the temporal modeling of the S-TCN.

5.4.1 Dataset description

The dataset used to validate the predictive procedure differs from that used for causal discovery. In fact, for the use of neural networks, a larger number of data is required. The final dataset contains a total of 11 features with 26000 samples that entail more than 100 days of continuous production. The data are standardized using a robust scaler that removes the median and scales the data according to the quantile range. This scaling algorithm is required to reduce the effect of outliers. In agreement with experienced operators we have established a τ_{max} of 45 to capture the total production cycle. The main difference between the dataset presented in section 5.3.1 and the

dataset used for prediction are summarized in Table 5.2.

Experimental Dataset	**Causal discovery**	**Predictive procedure**
N.samples	924	26000
Days of production	19	100
N.Variables	14	11
Standardization	Min-Max	Robust
Frequency	30 min	5 min

Table 5.2: Proprieties of the real case scenario datasets for the predictive procedure and the causal discovery experiments.

5.4.2 Evaluations

We use K-fold validation with $K = 3$. We choose a small K values to include a huge portion of the data in each train/test split and mitigates the effect of non-stationarity due to unrecorded external causes. The presented results are the means and standard deviations of the K validations. The metric used is root mean squared error (RMSE). We compare the presented procedure with the comparison methods in section 5.3.2. For S-TCN and Random Forest models we compare also the accuracy with three different causal vectors:

- a PCMCI causal vector that uses the output of PCMCI. We used partial correlation independence test as CMI is not scalable on long timeseries.
- an expert-based causal vector that is obtained directly by expert operators using a knowledge based causal map
- an empty set causal vector that assumes all temporal activations to 0. The machine learning algorithms tested in this setting are not affected by the causal vector removing the first step of the procedure.

5.4.3 Results

Table 5.3 presents the RSME obtained with standard deviation for each run.

Causal Vector	Empty Set	PCMCI	Expert-Based
LSTM	0,508 ± 0,04		
Short-TCN	0,493 ± 0,05		
Long-TCN	0,535 ± 0,03		
S-TCN	0,549 ± 0,03	**0,490 ± 0,02**	0,504 ± 0,03
Trivial	0,879 ± 0,03		
RF	0,753 ± 0,02	0,765 ± 0,01	0,762 ± 0,03

Table 5.3: Result on real case study with three different causal vector. In bold the lowest mean squared error obtained is highlighted.

5.4.4 Discussion on predictive monitoring

Unlike in the synthetic scenario where the causal model is well-defined, for the predictive procedure, the outcomes are affected by multiple factors. First, the results demonstrate how, in practice, temporal models (i.e., LSTM, Short-TCN, Long-TCN and S-TCN) perform better due to feature analysis over time. This behavior validates the use of S-TCN over RF with preprocessing on input presented for the synthetic dataset. In addition, as reported in Table II, the effect of different causal vectors on the random forest is reduced if compared to the S-TCN. This result suggests that S-TCN gives more importance to the causal relations of the industrial plant. Another observation is the strong autocorrelation in the real data. The results obtained from Short-TCN and Long-TCN denote that most of the information about the prediction is contained near the forecast horizon. In addition, PCMCI on the S-TCN shows better performance than the causal vector recommended by experts. PCMCI, indeed, is particularly suited to handle parameter autocorrelation.

In Fig. 5-5 we show an example of prediction on the real case scenario. In the upper part of the Figure, all methods with the empty-set causal vector (i.e. not using causal precursors) are compared. Even though temporal information allows a strong improvement of predictions with respect to the empty-set random forest or the trival model. However, the trends of the predictions for the different autoregressive temporal methods are similar. S-TCN has a predictive behavior similar to TCN while LSTM is showing a different prediction trend that could be related to a diverse internal features modeling. In the bottom part of the Figure, we show the comparison

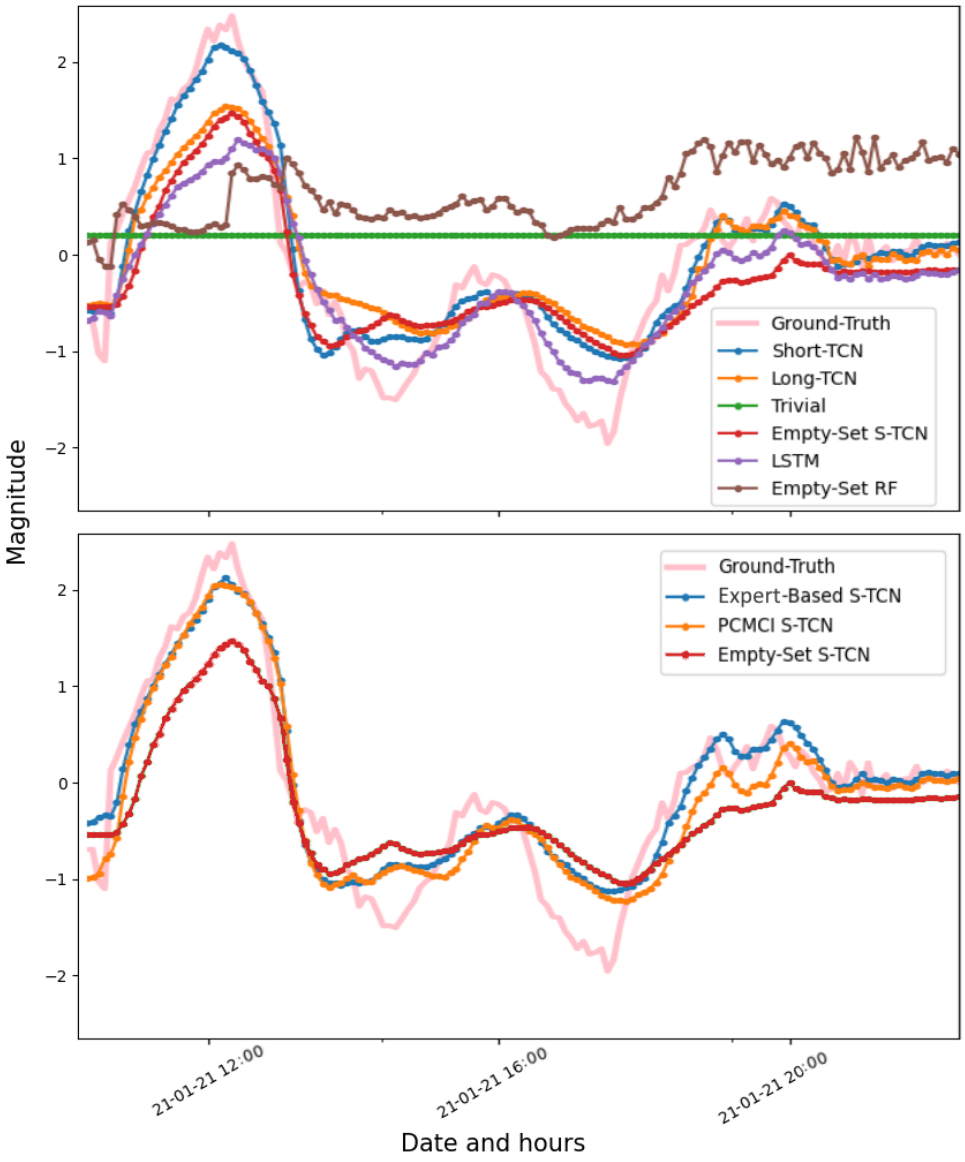


Figure 5-5: 12 hours of prediction for the response variable in the real case scenario. Above comparison between methods that do not use causal vectors. Below three different causal vectors on the S-TCN model.

between S-TCN models with different causal vectors to investigate their contribution to predictions. We have excluded from the Figure the comparison between different causal vectors on random forest to allow a better reading of the graph. Both the Figure and the results in Table 5.3 confirm that the causal vector strongly affects predictions and forces the S-TCN to model features diversely. PCMCI features selection fully exploits the potential of the S-TCN structure by obtaining the best predictions. When

working with data extracted from an industrial plant, forecasting product parameters is extremely challenging due to unregistered confounders that can affect the predicted variable and its causal precursors.

5.5 Conclusions

In this chapter, we applied the contribution presented in chapter 4 in the case of a medium-sized company producing ultra-processed food. In section 5.1, an executive summary outline the theoretical contributions proposed in the previous chapters and motivate the validation of the models through a real case scenario. In section 5.3 we used PCMCI to reconstruct the production flow of the company, and we created a causal model to intervene on the machines' parameters. We compared the inferred intervention protocols with the common intervention performed in the production. In section 5.4 we applied the whole predictive procedure for the monitoring of a product quality variable. We have shown that using a causal discovery method based on a data-driven approach allows the most suitable selection of causal precursors. Finally, we compared the predictive monitoring procedure proposed in this thesis with state-of-the-art methods.

Chapter 6

Future works and Conclusions

6.1 Future works

In this thesis, for the real case scenario, in the predictive procedure we apply PCMCI with partial correlations independence test that considers only linear relationships. This is due to the onerous computational requirements for CMI independence test, which is not scalable to long time series. Constraint-based causal discovery algorithms heavily depend on the independence test applied. Therefore, as future developments, it is important to develop new non-linear independence tests for high dimensional time series. In addition, in a real case scenario, industrial time-series data are generally non-stationarity due to unrecorded factors. Recently, causal discovery algorithms have been proposed to handle non-stationarity. In [135], authors suggest that non-stationarity and multiple domains can help in identifying causal structures benefiting from information carried by changing distributions. Using these methods could facilitate data collection because, besides including non-stationary time series, data from multiple products can be used. For SMEs it is essential to develop causal maps that are product-independent. The outcome of multi-domain causal discovery algorithms can help to create long-term protocols and speed up training for new operators.

While improving causal discovery algorithms is essential for recognizing cause effect relationships, the introduction of causal theory brings other benefits than causal discovery. For example, the notation of structural causal models allows formalizing

the interventions made during the production. Thanks to SCADA systems, setpoint changes are recorded and stored along with the production. It is, therefore, possible to formalize intervention with the do-notation developed by Pearl [49] to optimize online intervention.

Regarding the S-TCN architecture it is important to improve accuracy and interpretability. For example, the addition of attention gates before concatenation can estimate the strength of the causal link between two variables, increasing the model's interpretability. Nauta did a similar use of attention gates in [130]. Furthermore, in the final part of the S-TCN, which is composed of a four-layer TCN as explained in section 4.3.1, we did not implement any residual blocks. Residual blocks could improve the predictive capability of the architecture at the expense of increased computational cost.

In order to improve the phase recognition, as already mentioned in the discussions, it is first necessary to increase the number of sensors to recognize all the variables affecting the scene. If this is not possible, it is required to integrate the domain's knowledge. An interesting technique used in [159] for SRS has been to define surgical procedures at multiple levels of granularity using a hierarchical approach. Similarly, micro-phases and macro-phases can be defined for the industrial process. For example, we can define process phases that describe the production flow throw machines and tasks that describe the actions performed during an HRC. The action performed during HRC task (micro phases) is a subset of a single-phase used to describe the process (macro phases). Micro and macro phases can then be controlled with a hierarchical finite state machine built using domain knowledge. The number of possible classes for micro phases is reduced thanks to the constraint given by the macro phases. Using the recognition of phases at different levels of granularity would allow reducing classification errors.

To summarize we propose the following improvements as future works:

- Improving causal discovery algorithms by considering industrial process characteristics such as non-stationarity, non-linearity, and scalability;

- Using Pearlanian $do - notation$ to validates operators' in-line interventions;
- Enhancing S-TCN architecture using attention gates to improve interpretability and residual block to improve accuracy;
- Building an hierarchical finite state machine to improve phase recognition at different level of granularity;

6.2 Conclusions

SMEs are of vital importance for the global and European industrial context. The fourth industrial revolution needs to reach SMEs through the introduction of sustainable technologies. Therefore, it is necessary to develop technologies capable of intercepting the challenges addressed by these types of companies. Market flexibility, employers shortage and limited economic resources are increasing the management complexity of SMEs. Developing monitoring systems is helpful to reduce the complexity of the company, improving product quality and process management. However, current monitoring technologies are limited to detecting process behaviours without considering characteristic features of the manufacturing process. These features are the process subdivision into phases, the sequential flow of the product throw machines ruled by cause effect relationships, and the integration of domain knowledge. Considering these aspects allows building more accurate and specific monitoring systems for the industrial manufacturing context. In this thesis we have proposed the following contributions in this direction:

- We propose a method for automatic process phases recognition based on ANN architecture named Time Delay Neural Network (TDNN) to recognize the phases in a collaborative process.
- We propose a predictive procedure that integrates the causal relationships of the industrial plant and the domain's knowledge. The procedure allows building more accurate predictive monitoring systems while also providing operators control over the monitoring system.

Appendix A

Numerical Experiments

In these tables we presents the results of the numerical experiment described in 4.3.2. We set 7 increasing maximum time lag τ_{max} from 30 to 210. For each maximum time lag we create 10 realization for a total of 70 realizations. For each machine learning model we show the obtained mean squared error and we underline the best model performance for each realization. The result are summarized in Figure 4-9. In the tables the causal vectors generated following equation 4.3.2 are reported. The configuration of the S-TCN varies depending on the causal vector as explained in section 4.3.1. The configuration of the other models is kept constant on all experiments with the exception of the Random Forest (RF) which represents the optimal reference for temporal modeling as explained in 4.3.2.

τ_max = 30		
Causal Vector	14 17 17 8 25 25 8 25 14 14 13 17 13 13 8	9 18 9 4 12 12 18 26 26 18 26 12 4 9 4
Id	1	2
Short-TCN	0,016740207	0,018297825
Long-TCN	0,016310273	0,019189185
S-TCN	0,011772394	0,015631685
LSTM	0,019351326	0,023893999
Trivial	0,019777063	0,024229262
RF	0,012779907	0,016591849
Causal Vector	9 13 27 2 2 9 2 3 13 9 13 27 27 3 3	8 8 4 26 27 8 1 27 1 26 4 27 4 26 1
Id	3	4
Short-TCN	0,016739113	0,015132897
Long-TCN	0,015454015	0,015579456
S-TCN	0,014484654	0,013048103
LSTM	0,022330094	0,019978594
Trivial	0,02243745	0,021528358
RF	0,015315971	0,014077481
Causal Vector	8 8 17 18 8 12 11 18 12 17 17 18 12 11 11	28 17 3 1 12 28 12 28 3 12 1 17 17 1 3
Id	5	6
Short-TCN	0,020829942	0,02030777
Long-TCN	0,018219931	0,018793626
S-TCN	0,015505982	0,015918253
LSTM	0,024009345	0,022703271
Trivial	0,024015192	0,024319418
RF	0,01664039	0,016825952
Causal Vector	8 12 12 8 20 27 8 2 12 20 2 27 27 2 20	13 22 6 6 29 13 22 18 29 29 6 22 18 13 18
Id	7	8
Short-TCN	0,015409178	0,01777734
Long-TCN	0,016274134	0,017804403
S-TCN	0,013032189	0,014729392
LSTM	0,019902825	0,020568991
Trivial	0,020976804	0,022272693
RF	0,013946713	0,015504802
Causal Vector	13 23 17 13 5 23 17 13 23 28 28 28 5 5 17	1 28 1 28 4 28 4 4 17 8 8 8 17 1 17
Id	9	10
Short-TCN	0,021686755	0,01670453
Long-TCN	0,015850754	0,016649049
S-TCN	0,013212477	0,01314592
LSTM	0,02078958	0,019287439
Trivial	0,021686295	0,021135854
RF	0,014265771	0,01413759

τ_max = 60		
Causal Vector	55 55 18 7 59 36 55 59 36 59 7 18 36 18 7	8 2 31 2 8 31 50 8 50 24 50 24 2 31 24
Id	11	12
Short-TCN	0,016984187	0,01883273
Long-TCN	0,018564392	0,018995948
S-TCN	0,013974804	0,013785251
LSTM	0,02012887	0,021946011
Trivial	0,02229798	0,0219907
RF	0,014882783	0,014795026
Causal Vector	46 30 10 38 30 43 43 10 46 46 38 10 38 43 30	25 34 25 45 47 2 2 2 47 25 47 45 34 34 45
Id	13	14
Short-TCN	0,021238344	0,022034572
Long-TCN	0,019754376	0,02088992
S-TCN	0,014597238	0,01574547
LSTM	0,022515824	0,024011448
Trivial	0,022516899	0,02400657
RF	0,015280453	0,016589761
Causal Vector	12 4 4 19 12 2 9 12 2 9 19 4 2 9 19	51 48 4 35 4 48 51 9 35 9 9 48 51 35 4
Id	15	16
Short-TCN	0,01974971	0,015508059
Long-TCN	0,016745938	0,01666244
S-TCN	0,014484967	0,013137369
LSTM	0,022794247	0,021267926
Trivial	0,022792324	0,021237483
RF	0,015525156	0,014081634
Causal Vector	3 6 8 8 22 47 6 22 6 3 47 8 47 3 22	28 11 27 27 2 28 2 11 2 28 39 39 11 27 39
Id	17	18
Short-TCN	0,02018658	0,021061832
Long-TCN	0,018247465	0,02076376
S-TCN	0,015662977	0,016359325
LSTM	0,0237677	0,024765365
Trivial	0,023756888	0,024753543
RF	0,016512189	0,016966113
Causal Vector	51 29 45 29 33 51 35 51 33 35 29 35 33 45 45	12 53 15 7 53 23 23 12 23 7 15 53 7 12 15
Id	19	20
Short-TCN	0,021061914	0,017799586
Long-TCN	0,015340879	0,018712943
S-TCN	0,013049193	0,0138759
LSTM	0,021076065	0,021779453
Trivial	0,021062052	0,021771554
RF	0,014126994	0,014893095

τ_max = 90		
Causal Vector	57 57 46 43 46 18 18 75 18 57 75 75 43 46 43	11 14 70 70 70 47 76 11 76 14 11 76 47 47 14
Id	21	22
Short-TCN	0,020370388	0,021647852
Long-TCN	0,018423153	0,020208167
S-TCN	0,014721273	0,014565024
LSTM	0,022265472	0,021658437
Trivial	0,022271411	0,021648329
RF	0,015670677	0,014085071
Causal Vector	67 85 69 53 69 61 67 53 53 85 61 69 67 61 85	25 40 25 64 64 82 77 82 25 77 64 82 77 40 40
Id	23	24
Short-TCN	0,016636433	0,018885974
Long-TCN	0,016661337	0,019402692
S-TCN	0,013246855	0,013989549
LSTM	0,020875622	0,02267359
Trivial	0,020900728	0,022784468
RF	0,013862827	0,014847517
Causal Vector	62 62 79 6 17 62 34 34 6 79 6 34 17 79 17	44 23 44 75 80 23 44 75 75 9 23 80 9 9 80
Id	25	26
Short-TCN	0,020314947	0,01925837
Long-TCN	0,019954257	0,019209519
S-TCN	0,015796637	0,015648866
LSTM	0,021578126	0,022346035
Trivial	0,021577144	0,022369998
RF	0,014680827	0,01544072
Causal Vector	5 44 44 10 10 5 73 21 44 73 5 73 10 21 21	18 17 15 17 42 6 17 15 6 42 6 18 42 15 18
Id	27	28
Short-TCN	0,01797656	0,023594523
Long-TCN	0,01953374	0,019259837
S-TCN	0,014727272	0,017271787
LSTM	0,01999837	0,024180971
Trivial	0,01997688	0,024173455
RF	0,012970506	0,016641243
Causal Vector	20 71 1 71 38 66 38 20 1 20 38 1 66 66 71	28 29 68 24 28 29 29 28 82 82 82 68 68 24 24
Id	29	30
Short-TCN	0,018151978	0,018508747
Long-TCN	0,01747282	0,018504748
S-TCN	0,01441409	0,01625048
LSTM	0,021084577	0,023777848
Trivial	0,02107365	0,023782019
RF	0,013911661	0,016568556

τ_max = 120		
Causal Vector	14 58 14 14 5 58 70 70 99 5 99 58 99 70 5	2 44 96 96 72 44 2 72 18 18 96 2 44 72 18
Id	31	32
Short-TCN	0,020995356	0,019584771
Long-TCN	0,020970361	0,02169991
S-TCN	0,019302908	0,018083291
LSTM	0,02473773	0,021843154
Trivial	0,024718795	0,021818586
RF	0,016648108	0,014993365
Causal Vector	100 100 114 80 23 114 23 57 114 57 80 23 80 57 100	100 100 114 80 23 114 23 57 114 57 80 23 80 57 100
Id	33	34
Short-TCN	0,021594338	0,021594338
Long-TCN	0,020814992	0,020814992
S-TCN	0,017412543	0,017412543
LSTM	0,022531286	0,022531286
Trivial	0,022538796	0,022538796
RF	0,015587775	0,015587775
Causal Vector	53 118 38 53 78 78 118 78 118 79 79 38 53 79 38	13 72 51 13 72 72 13 51 103 103 36 36 103 36 51
Id	35	36
Short-TCN	0,018305285	0,024768222
Long-TCN	0,01833673	0,024714246
S-TCN	0,01314004	0,020142075
LSTM	0,017716924	0,024920251
Trivial	0,02096757	0,024905205
RF	0,014087309	0,017913357
Causal Vector	4 61 4 4 109 61 119 61 109 3 3 109 3 119 119	11 23 7 117 7 66 23 11 117 66 23 66 117 7 11
Id	37	38
Short-TCN	0,019042935	0,020171847
Long-TCN	0,022516506	0,018883884
S-TCN	0,01686375	0,01882537
LSTM	0,019900294	0,021230765
Trivial	0,022562461	0,02158884
RF	0,015077455	0,014539229
Causal Vector	11 101 18 33 33 101 101 11 75 18 18 33 11 75 75	54 54 88 39 88 55 15 88 55 39 15 54 55 39 15
Id	39	40
Short-TCN	0,019097002	0,021754805
Long-TCN	0,022533666	0,019249093
S-TCN	0,017958133	0,0166445
LSTM	0,022735758	0,021923486
Trivial	0,022722421	0,021907287
RF	0,015446352	0,014985356

τ_max = 150		
Causal Vector	143 143 75 59 93 83 83 143 93 59 75 59 93 83 75	48 149 104 48 139 80 139 104 149 80 80 48 104 149 139
Id	41	42
Short-TCN	0,021696717	0,01853536
Long-TCN	0,022165481	0,020897955
S-TCN	0,017794639	0,015637815
LSTM	0,022173658	0,019256754
Trivial	0,02217452	0,020961024
RF	0,014996222	0,014200199
Causal Vector	19 79 14 14 111 19 140 111 14 140 111 79 19 79 140	75 115 115 124 115 139 89 75 139 75 139 89 124 124 89
Id	43	44
Short-TCN	0,023178976	0,022641057
Long-TCN	0,021215793	0,023209505
S-TCN	0,019842483	0,016507361
LSTM	0,024146896	0,023276206
Trivial	0,024153316	0,02328366
RF	0,016737271	0,015547902
Causal Vector	59 54 47 59 54 59 20 20 18 18 54 47 20 47 18	57 127 71 127 59 59 59 57 40 40 71 71 127 40 57
Id	45	46
Short-TCN	0,02427683	0,021760667
Long-TCN	0,024054738	0,018485673
S-TCN	0,020062506	0,016612325
LSTM	0,024192687	0,023205115
Trivial	0,024193818	0,023190992
RF	0,016671922	0,015439026
Causal Vector	78 27 78 3 3 45 87 45 45 27 3 87 27 78 87	103 112 103 103 112 11 131 131 112 32 32 11 11 131 32
Id	47	48
Short-TCN	0,022025973	0,018649489
Long-TCN	0,023779951	0,021922538
S-TCN	0,01981837	0,015783427
LSTM	0,023872074	0,021948176
Trivial	0,023859711	0,021930732
RF	0,016536946	0,014920787
Causal Vector	104 104 8 114 135 135 114 80 104 80 114 80 8 8 135	70 15 43 15 15 74 43 74 70 70 47 43 47 47 74
Id	49	50
Short-TCN	0,018502682	0,020126397
Long-TCN	0,020534981	0,019982494
S-TCN	0,016992453	0,014161357
LSTM	0,022583151	0,02005193
Trivial	0,02257701	0,020050805
RF	0,015678111	0,012722083

τ_max = 180		
Causal Vector	148 124 136 6 148 136 63 136 6 124 63 6 148 63 124	21 87 3 23 3 23 3 21 88 88 88 21 23 87 87
Id	51	52
Short-TCN	0,022473253	0,0229197
Long-TCN	0,022577839	0,02089803
S-TCN	0,018520901	0,020484708
LSTM	0,024009237	0,022861097
Trivial	0,024002826	0,022860913
RF	0,016408997	0,015344414
Causal Vector	179 179 6 109 177 69 109 109 6 179 177 69 177 6 69	155 81 162 81 155 48 155 58 81 58 162 48 162 48 58
Id	53	54
Short-TCN	0,02145457	0,016911324
Long-TCN	0,024419177	0,019874394
S-TCN	0,021482069	0,013159671
LSTM	0,021406224	0,019983398
Trivial	0,024481151	0,019975403
RF	0,016774357	0,012835714
Causal Vector	129 97 129 141 29 141 29 41 141 41 129 97 29 41 97	59 176 121 94 16 59 94 176 176 16 16 94 121 121 59
Id	55	56
Short-TCN	0,018281473	0,022450175
Long-TCN	0,019934375	0,02417477
S-TCN	0,01702593	0,019980386
LSTM	0,020016605	0,023319928
Trivial	0,019997634	0,024277784
RF	0,012649562	0,016734571
Causal Vector	112 163 24 112 159 147 159 147 112 147 163 163 24 24 159	84 131 175 90 90 175 131 84 94 131 94 94 84 175 90
Id	57	58
Short-TCN	0,019865906	0,017954566
Long-TCN	0,018794555	0,021283638
S-TCN	0,017001899	0,015324134
LSTM	0,022711758	0,021305965
Trivial	0,022717766	0,021347826
RF	0,015364515	0,014248996
Causal Vector	113 32 2 32 113 89 89 130 2 113 130 130 32 89 2	61 109 129 144 61 87 109 61 129 144 129 87 144 109
Id	59	60
Short-TCN	0,023806479	0,020168843
Long-TCN	0,022713251	0,02086677
S-TCN	0,02086508	0,016921224
LSTM	0,023806378	0,021313487
Trivial	0,02380339	0,0212918
RF	0,016621432	0,014040287

τ_max = 210		
Causal Vector	207 40 81 81 66 7 40 40 207 7 66 81 207 7 66	187 117 195 57 187 117 112 117 195 195 112 187 57 112 57
Id	61	62
Short-TCN	0,019763105	0,023120092
Long-TCN	0,019855049	0,024182387
S-TCN	0,017874595	0,021227805
LSTM	0,019417044	0,024266595
Trivial	0,019877741	0,024249744
RF	0,012715453	0,016647882
Causal Vector	192 104 6 192 84 6 152 84 192 152 104 152 6 104 84	99 80 117 117 205 80 99 205 142 142 80 205 117 99 142
Id	63	64
Short-TCN	0,02236835	0,023420338
Long-TCN	0,01982952	0,02373343
S-TCN	0,01585782	0,018596115
LSTM	0,022801492	0,02380668
Trivial	0,022785898	0,023844948
RF	0,015335462	0,016576801
Causal Vector	15 74 87 87 94 87 74 74 15 94 94 127 15 127 127	175 133 175 64 88 152 133 88 64 152 88 175 64 133 152
Id	65	66
Short-TCN	0,020028451	0,02371418
Long-TCN	0,019557856	0,023680035
S-TCN	0,015116327	0,020364363
LSTM	0,019966966	0,023740342
Trivial	0,019971944	0,023729429
RF	0,012845031	0,016589726
Causal Vector	59 87 87 50 50 59 11 87 50 59 11 184 184 11 184	56 56 56 173 81 101 81 49 101 101 173 81 173 49 49
Id	67	68
Short-TCN	0,022839874	0,019827599
Long-TCN	0,020003274	0,020784685
S-TCN	0,020782473	0,015859522
LSTM	0,02280682	0,020866089
Trivial	0,02279929	0,020842442
RF	0,015337157	0,014039627
Causal Vector	146 84 9 15 9 146 146 9 42 42 15 15 84 84 42	82 180 180 64 133 133 67 67 64 180 64 133 82 67 82
Id	69	70
Short-TCN	0,02038135	0,022974687
Long-TCN	0,020672534	0,024160428
S-TCN	0,019540688	0,022129824
LSTM	0,023855513	0,024332525
Trivial	0,0238488	0,024325933
RF	0,016519708	0,016697284

Bibliography

[1] J. A. G. Katarzyna Bańkowska, Annalisa Ferrando, "Access to finance for small and medium-sized enterprises since the financial crisis: evidence from survey data," *ECB Economic Bulletin, Issue 4/2020*, vol. 4, 2020.

[2] D. B. Rubin, "Causal inference using potential outcomes," *Journal of the American Statistical Association*, vol. 100, no. 469, pp. 322–331, 2005.

[3] cerved, "Cerved pmi 2020 report," 2020. `https://know.cerved.com/wp-content/uploads/2020/11/RAPPORTO-CERVED-PMI-2020-2.pdf`.

[4] T. Masood and P. Sonntag, "Industry 4.0: Adoption challenges and benefits for smes," *Computers in Industry*, vol. 121, p. 103261, 2020.

[5] S. L. Lee, T. F. O. Connor, X. Yang, C. N. Cruz, L. X. Yu, and J. Woodcock, "Modernizing Pharmaceutical Manufacturing : from Batch to Continuous Production," pp. 191–199, 2015.

[6] G. Menegozzo, D. Dall'Alba, A. Roberti, and P. Fiorini, "Automatic process modeling with time delay neural network based on low-level data.," *Procedia Manufacturing*, vol. 38, pp. 125–132, 2019. 29th International Conference on Flexible Automation and Intelligent Manufacturing (FAIM 2019), June 24-28, 2019, Limerick, Ireland, Beyond Industry 4.0: Industrial Advances, Engineering Education and Intelligent Manufacturing.

[7] Y. Wang, S. Hu, G. Wang, C. Chen, and Z. Pan, "Multi-scale dilated convolution of convolutional neural network for crowd counting," *Multimedia Tools and Applications*, vol. 79, pp. 1057–1073, Jan 2020.

[8] Y. Gao, S. Swaroop Vedula, C. E. Reiley, N. Ahmidi, B. Varadarajan, H. C. Lin, L. Tao, L. Zappella, B. Béjar, D. D. Yuh, C. Chiung, G. Chen, R. Vidal, S. Khudanpur, and G. D. Hager, "JHU-ISI Gesture and Skill Assessment Working Set (JIGSAWS): A Surgical Activity Dataset for Human Motion Modeling,"

[9] S. Mittal, M. A. Khan, D. Romero, and T. Wuest, "A critical review of smart manufacturing & industry 4.0 maturity models: Implications for small and medium-sized enterprises (smes)," *Journal of Manufacturing Systems*, vol. 49, pp. 194–214, oct 2018.

[10] B. van Amsterdam, M. J. Clarkson, and D. Stoyanov, "Gesture recognition in robotic surgery: A review," *IEEE Transactions on Biomedical Engineering*, vol. 68, p. 2021–2035, Jun 2021.

[11] N. Ahmidi, L. Tao, S. Sefati, Y. Gao, C. Lea, B. B. Haro, L. Zappella, S. Khudanpur, R. Vidal, and G. D. Hager, "A dataset and benchmarks for segmentation and recognition of gestures in robotic surgery," *IEEE Transactions on Biomedical Engineering*, vol. 64, no. 9, pp. 2025–2041, 2017.

[12] W. B. national accounts data and OECD, "Manufacturing, value added (% of gdp)." `https://data.worldbank.org/indicator/NV.IND.MANF.ZS?most_recent_value_desc=true`.

[13] W. B. national accounts data and OECD, "Manufacturing, value added (current us)." `https://data.worldbank.org/indicator/NV.IND.MANF.CD?most_recent_value_desc=true`.

[14] S. K, ed., *The Fourth Industrial Revolution: What It Means, How to Respond.* World Economic Forum, 2016. `https://www.weforum.org/agenda/2016/01/the-fourth-industrial-revolution-what-it-means-and-how-to-respond/`.

[15] P. Ross and K. Maynard, "Towards a 4th industrial revolution," *Intelligent Buildings International*, vol. 13, no. 3, pp. 159–161, 2021.

[16] E. commission, "Digital transformation monitor germany: Industrie 4.0," 2017. `https://ati.ec.europa.eu/sites/default/files/2020-06/DTM_Industrie\%204.0_DE.pdf`.

[17] E. chamber, "China manufacturing 2025,putting industrial policy ahead of market forces," 2017. `http://www.cscc.it/upload/doc/china_manufacturing_2025_putting_industrial_policy_ahead_of_market_force\%5Benglish-version\%5D.pdf`.

[18] L. Li, "China's manufacturing locus in 2025: With a comparison of "made-in-china 2025" and "industry 4.0"," *Technological Forecasting and Social Change*, vol. 135, pp. 66–74, 2018.

[19] T. O. Isabel Castelo-Branco, Frederico Cruz-Jesus, "Assessing industry 4.0 readiness in manufacturing: Evidence for the european union," *Computers in Industry*, vol. 107, pp. 22–32, 2019.

[20] M. D.T. and R. E., "Sme 4.0: The role of small- and medium-sized enterprises in the digital transformation.," *Journal of Manufacturing Systems*, 2020.

[21] OECD, *OECD SME and Entrepreneurship Outlook 2019*. 2019.

[22] I. (2017), "Policy pathways brief accelerating energy efficiency in small and medium-sized enterprises 2017 iea paris," 2017. `https://www.iea.org/reports/policy-pathways-brief-accelerating-energy-efficiency-in-small-and-medium-sized-enterprises-2017`.

[23] E. B. . P. Research, "Do smes create more and better jobs?," 2017. https://ec.europa.eu/growth/sites/default/files/docs/body/do-smes-create-more-and-better-jobs_en.pdf.

[24] E. U. commission, "2021 sme country fact sheet european union," 2021. https://ec.europa.eu/docsroom/documents/46060.

[25] OECD, "Enhancing the contributions of smes in a global and digitalised economy," 2017. https://www.oecd.org/industry/C-MIN-2017-8-EN.pdf.

[26] F. O. Giorgia Sali, "Le pmi nell'ecosistema imprenditoriale italiano: un confronto con l'ue, osservatorio innovazione digitale nelle pmi," 2020. https://blog.osservatori.net/it_it/pmi-ecosistema-imprenditoriale-italiano-confronto-ue.

[27] S. Mittal, M. A. Khan, D. Romero, and T. Wuest, "A critical review of smart manufacturing & Industry 4.0 maturity models: Implications for small and medium-sized enterprises (SMEs)," *Journal of Manufacturing Systems*, vol. 49, pp. 194–214, oct 2018.

[28] G. I. M. dello sviluppo economico, "Transizione 4.0," 2019. https://www.mise.gov.it/index.php/it/incentivi/impresa/transizione-4-0/traniszione-4-0-2019-2020.

[29] L. R. C. S. Confindustria, "Crescono gli occupati grazie agli investimenti agevolati in tecnologie 4.0," 2020. https://www.confindustria.it/home/centro-studi/temi-di-ricerca/valutazione-delle-politiche-pubbliche/tutti/dettaglio/crescono-gli-occupati-grazie-agli-\\investimenti-agevolati-in-tecnologie-4.0.

[30] J. Johnson, "Eu: digitalization level 2020, by country," 2020. https://www.statista.com/statistics/1245595/eu-digitalization-level/.

[31] daCompletare, "dacompletare," 2017. daCompletare.

[32] J. M. Müller, O. Buliga, and K.-I. Voigt, "Fortune favors the prepared: How smes approach business model innovations in industry 4.0," *Technological Forecasting and Social Change*, vol. 132, pp. 2–17, 2018.

[33] L. Raymond, "Operations management and advanced manufacturing technologies in SMEs," *Journal of Manufacturing Technology Management*, vol. 16, pp. 936–955, dec 2005.

[34] T. D. Brunoe and K. Nielsen, "Complexity management in mass customization smes," *Procedia CIRP*, vol. 51, pp. 38–43, 2016. 3rd ICRM 2016 International Conference on Ramp-Up Management.

[35] N. O'Regan and A. Ghobadian, "Short- and long-term performance in manufacturing smes," *International Journal of Productivity and Performance Management*, vol. 53, pp. 405–424, Jan 2004.

[36] M. Prause, "Challenges of industry 4.0 technology adoption for smes: The case of japan," *Sustainability*, vol. 11, no. 20, 2019.

[37] N. Grigg, "Statistical process control in uk food production: an overview," *International Journal of Quality & Reliability Management*, vol. 15, no. 2, 1998.

[38] G. Cartwright and B. Hogg, "Measuring processes for profit," *TQM Magazine*, vol. 8, no. 1, 1996.

[39] A. J. Rungasamy S. and G. S., "Critical success factors for spc implementation in uk small and medium enterprises: some key findings from a survey," *TQM Magazine*, vol. 14, no. 4, 2002.

[40] M. S. Reis and G. Gins, "Industrial process monitoring in the big data/industry 4.0 era: from detection, to diagnosis, to prognosis," *Processes*, vol. 5, no. 3, 2017.

[41] A. Moeuf, R. Pellerin, S. Lamouri, S. Tamayo-Giraldo, and R. Barbaray, "The industrial management of smes in the era of industry 4.0," *International Journal of Production Research*, vol. 56, no. 3, pp. 1118–1136, 2018.

[42] M. Reisi Gahrooei, K. Paynabar, M. Pacella, and J. Shi, "Process modeling and prediction with large number of high-dimensional variables using functional regression," *IEEE Transactions on Automation Science and Engineering*, vol. 17, no. 2, pp. 684–696, 2020.

[43] R. Kothamasu, S. H. Huang, and W. H. VerDuin, *System Health Monitoring and Prognostics – A Review of Current Paradigms and Practices*, pp. 337–362. London: Springer London, 2009.

[44] J. Downs and E. Vogel, "A plant-wide industrial process control problem," *Computers & Chemical Engineering*, vol. 17, no. 3, pp. 245–255, 1993. Industrial challenge problems in process control.

[45] B. Curtis, M. I. Kellner, and J. Over, "Process modeling," *Commun. ACM*, vol. 35, p. 75–90, sep 1992.

[46] H. Liu, T. Fang, T. Zhou, and L. Wang, "Towards robust human-robot collaborative manufacturing: Multimodal fusion," *IEEE Access*, vol. 6, pp. 74762–74771, 2018.

[47] M. Pacella, *Il controllo di qualità per processi manifatturieri tramite l'impiego di un algoritmo neurale basato sulla teoria della risonanza adattativa*. PhD thesis, Polytechnic University of Milan, 2003.

[48] Z. Ge, Z. Song, and F. Gao, "Review of recent research on data-based process monitoring," *Industrial & Engineering Chemistry Research*, vol. 52, no. 10, pp. 3543–3562, 2013.

[49] J. Pearl, *Causality, Models Reasoning and Inference*. Cambridge Press, 2009.

[50] A.-L. F, "The industry 4.0 revolution and the future of manufacturing execution systems (mes)," *J Innov Manag*, vol. 3, no. 4, pp. 16–21, 2015.

[51] A. D. Bruno G., "Dynamic task classification and assignment for the management of human-robot collaborative teams in workcells," *Int J Adv Manuf Technol*, vol. 98, p. 2415–2427, 2018.

[52] G. A. Susto, A. Schirru, S. Pampuri, and S. McLoone, "Supervised aggregative feature extraction for big data time series regression," *IEEE Transactions on Industrial Informatics*, vol. 12, no. 3, pp. 1243–1252, 2016.

[53] T. Hastie, R. Tibshirani, and J. Friedman, *Unsupervised Learning*, pp. 485–585. New York, NY: Springer New York, 2009.

[54] B. Rao, "Machine Learning Algorithms: A Review," *International Journal of Computer Science and Information Technologies*, vol. 7, no. 3, pp. 1174–1179, 2016.

[55] H. U. Dike, Y. Zhou, K. K. Deveerasetty, and Q. Wu, "Unsupervised learning based on artificial neural network: A review," in *2018 IEEE International Conference on Cyborg and Bionic Systems (CBS)*, pp. 322–327, 2018.

[56] Z. Ma, J. Tavares, R. Jorge, and T. Mascarenhas, "A review of algorithms for medical image segmentation and their applications to the female pelvic cavity," *Comput Methods Biomech Biomed Engin.*, vol. 13, no. 2, pp. 235–46, 2010.

[57] M. A. Salama, A. E. Hassanien, and A. A. Fahmy, "Deep belief network for clustering and classification of a continuous data," in *The 10th IEEE International Symposium on Signal Processing and Information Technology*, pp. 473–477, 2010.

[58] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[59] F. Q. Lauzon, "An introduction to deep learning," in *2012 11th International Conference on Information Science, Signal Processing and their Applications (ISSPA)*, pp. 1438–1439, 2012.

[60] T. Kohonen, "The self-organizing map," *Proceedings of the IEEE*, vol. 78, no. 9, pp. 1464–1480, 1990.

[61] S. Krishnan, A. Garg, S. Patil, C. Lea, G. Hager, P. Abbeel, and K. Goldberg, "Transition state clustering: Unsupervised surgical trajectory segmentation for robot learning," *The International Journal of Robotics Research*, vol. 36, no. 13-14, pp. 1595–1618, 2017.

[62] M. Ringnér, "What is principal component analysis?," *Nature Biotechnology*, vol. 26, pp. 303–304, Mar 2008.

[63] Z. I. Botev, J. F. Grotowski, and D. P. Kroese, "Kernel density estimation via diffusion," *The Annals of Statistics*, vol. 38, no. 5, pp. 2916 – 2957, 2010.

[64] L. E. Peterson, "K-nearest neighbor," *Scholarpedia*, vol. 4, no. 2, p. 1883, 2009. revision #137311.

[65] van Engelen, J. E., Hoos, and H. H., "A survey on semi-supervised learning," *Machine Learning*, vol. 109, no. 2, p. 373–440, 2020.

[66] M. Guillaumin, J. Verbeek, and C. Schmid, "Multimodal semi-supervised learning for image classification," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 902–909, 2010.

[67] O. Chapelle, B. Scholkopf, and A. Zien, Eds., "<emphasis emphasistype="bold">semi-supervised learning</emphasis> (chapelle, o. et al., eds.; 2006) [book reviews]," *IEEE Transactions on Neural Networks*, vol. 20, no. 3, pp. 542–542, 2009.

[68] J. C. Ang, A. Mirzal, H. Haron, and H. N. A. Hamed, "Supervised, unsupervised, and semi-supervised feature selection: A review on gene selection," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 13, no. 5, pp. 971–989, 2016.

[69] R. Saravanan and P. Sujatha, "A state of art techniques on machine learning algorithms: A perspective of supervised learning approaches in data classification," in *2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS)*, pp. 945–949, 2018.

[70] K. M.A., M. M., Goyal, and L.M., "A deep survey on supervised learning based human detection and activity classification methods," *Multimedia Tools Applications*, vol. 80, no. 18, p. 27867–27923, 2021.

[71] A. Shrestha and A. Mahmood, "Review of deep learning algorithms and architectures," *IEEE Access*, vol. 7, pp. 53040–53065, 2019.

[72] L. HC, S. I, Y. D, and H. GD, "Towards automatic skill evaluation: detection and segmentation of robot-assisted surgical motions," *Comput Aided Surg*, vol. 11, no. 5, pp. 220–30, 2006.

[73] G. I. Lee, M. R. Lee, I. Green, M. Allaf, and M. R. Marohn, "Surgeons' physical discomfort and symptoms during robotic surgery: a comprehensive ergonomic survey study," *Surgical Endoscopy*, vol. 31, no. 4, pp. 1697–1706, 2017.

[74] J. Rosen, B. Hannaford, C. Richards, and M. Sinanan, "Markov modeling of minimally invasive surgery based on tool-tissue interaction and force-torque signatures for evaluating surgical skills," *IEEE Transactions on Biomedical Engineering*, vol. 48, pp. 579–591, may 2001.

[75] L. Tao, L. Zappella, G. D. Hager, and R. Vidal, "Surgical gesture segmentation and recognition," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 8151 LNCS, pp. 339–346, Springer, Berlin, Heidelberg, 2013.

[76] C. Lea, G. D. Hager, and R. Vidal, "An Improved Model for Segmentation and Recognition of Fine-Grained Activities with Application to Surgical Training Tasks," in *2015 IEEE Winter Conference on Applications of Computer Vision*, pp. 1123–1129, IEEE, jan 2015.

[77] N. Ahmidi, L. Tao, S. Sefati, Y. Gao, C. Lea, B. B. Haro, L. Zappella, S. Khudanpur, R. Vidal, and G. D. Hager, "A dataset and benchmarks for segmentation and recognition of gestures in robotic surgery," *IEEE Transactions on Biomedical Engineering*, vol. 64, pp. 2025–2041, sep 2017.

[78] B. van Amsterdam, M. J. Clarkson, and D. Stoyanov, "Gesture recognition in robotic surgery: A review," *IEEE Transactions on Biomedical Engineering*, vol. 68, no. 6, pp. 2021–2035, 2021.

[79] R. DiPietro, C. Lea, A. Malpani, N. Ahmidi, S. S. Vedula, G. I. Lee, M. R. Lee, and G. D. Hager, "Recognizing surgical activities with recurrent neural networks," 2016.

[80] R. DiPietro and G. D. Hager, "Chapter 21 - deep learning: Rnns and lstm," in *Handbook of Medical Image Computing and Computer Assisted Intervention* (S. K. Zhou, D. Rueckert, and G. Fichtinger, eds.), The Elsevier and MICCAI Society Book Series, pp. 503–519, Academic Press, 2020.

[81] S. Bai, J. Z. Kolter, and V. Koltun, "An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling," 2018.

[82] J. Arents, V. Abolins, J. Judvaitis, O. Vismanis, A. Oraby, and K. Ozols, "Human–robot collaboration trends and safety aspects: A systematic review," *Journal of Sensor and Actuator Networks*, vol. 10, no. 3, 2021.

[83] E. Matheson, R. Minto, E. G. G. Zampieri, M. Faccio, and G. Rosati, "Human–robot collaboration in manufacturing applications: A review," *Robotics*, vol. 8, no. 4, 2019.

[84] Q. Jiang, X. Yan, and B. Huang, "Review and perspectives of data-driven distributed monitoring for industrial plant-wide processes," *Industrial & Engineering Chemistry Research*, vol. 58, no. 29, pp. 12899–12912, 2019.

[85] "Plant-wide process monitoring based on mutual information–multiblock principal component analysis," *ISA Transactions*, vol. 53, no. 5, pp. 1516–1527, 2014. ICCA 2013.

[86] Q. Jiang and X. Yan, "Monitoring multi-mode plant-wide processes by using mutual information-based multi-block pca, joint probability, and bayesian inference," *Chemometrics and Intelligent Laboratory Systems*, vol. 136, pp. 121–137, 2014.

[87] Y. Li, X. Peng, and Y. Tian, "Plant-wide process monitoring strategy based on complex network and Bayesian inference-based multi-block principal component analysis," *IEEE Access*, vol. 8, pp. 199213–199226, 2020.

[88] T. Mitsa, *Temporal Data Mining*. Chapman & Hall/CRC, 1st ed., 2010.

[89] S. Mehrmolaei and M. R. Keyvanpour, "Time series forecasting using improved arima," in *2016 Artificial Intelligence and Robotics (IRANOPEN)*, pp. 92–97, 2016.

[90] C. Cheng, A. Sa-Ngasoongsong, O. Beyca, T. Le, H. Yang, Z. J. Kong, and S. T. Bukkapatnam, "Time series forecasting for nonlinear and non-stationary processes: a review and comparative study," *IIE Transactions*, vol. 47, no. 10, pp. 1053–1071, 2015.

[91] D. A. Dickey and W. A. Fuller, "Distribution of the estimators for autoregressive time series with a unit root," *Journal of the American Statistical Association*, vol. 74, no. 366a, pp. 427–431, 1979.

[92] H. Tong, *Non-linear time series. A dynamical system approach*. 1990.

[93] B. Schlkopf, A. J. Smola, and F. Bach, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. The MIT Press, 2018.

[94] L. Breiman, "Random forests," *Machine Learning*, vol. 45, pp. 5–32, Oct 2001.

[95] R. Liu and W. Guo, "Hmm-based state prediction for internet hot topic," in *2011 IEEE International Conference on Computer Science and Automation Engineering*, vol. 1, pp. 157–161, 2011.

[96] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas,

O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015. Software available from tensorflow.org.

[97] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32* (H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, eds.), pp. 8024–8035, Curran Associates, Inc., 2019.

[98] P. Lara-Benítez, M. Carranza-García, and J. C. Riquelme, "An experimental review on deep learning architectures for time series forecasting," *CoRR*, vol. abs/2103.12057, 2021.

[99] H. Leung and S. Haykin, "The complex backpropagation algorithm," *IEEE Transactions on Signal Processing*, vol. 39, no. 9, pp. 2101–2104, 1991.

[100] R. HECHT-NIELSEN, "Theory of the backpropagation neural network**based on "nonindent"," in *Neural Networks for Perception* (H. Wechsler, ed.), pp. 65–93, Academic Press, 1992.

[101] B. Chakraborty, B. Shaw, J. Aich, U. Bhattacharya, and S. K. Parui, "Does deeper network lead to better accuracy: A case study on handwritten devanagari characters," in *2018 13th IAPR International Workshop on Document Analysis Systems (DAS)*, pp. 411–416, 2018.

[102] X. Hu, P. Niu, J. Wang, and X. Zhang, "A dynamic rectified linear activation units," *IEEE Access*, vol. 7, pp. 180409–180416, 2019.

[103] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[104] S. Hochreiter and J. Urgen Schmidhuber, "LONG SHORT-TERM MEMORY," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[105] P. T. Yamak, L. Yujian, and P. K. Gadosey, "A comparison between arima, lstm, and gru for time series forecasting," in *Proceedings of the 2019 2nd International Conference on Algorithms, Computing and Artificial Intelligence*, ACAI 2019, (New York, NY, USA), p. 49–55, Association for Computing Machinery, 2019.

[106] P. Lara-Benítez, M. Carranza-García, and J. C. Riquelme, "An experimental review on deep learning architectures for time series forecasting," *International Journal of Neural Systems*, vol. 31, no. 03, p. 2130001, 2021. PMID: 33588711.

[107] A. Gasparin, S. Lukovic, and C. Alippi, "Deep learning for time series forecasting: The electric load case," 2019.

[108] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. W. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," *CoRR*, vol. abs/1609.03499, 2016.

[109] W. Sun and R. D. Braatz, "Smart process analytics for predictive modeling," *Computers & Chemical Engineering*, vol. 144, p. 107134, 2021.

[110] Y. Wang, K. Yang, and H. Li, "Industrial time-series modeling via adapted receptive field temporal convolution networks integrating regularly updated multi-region operations based on pca," *Chemical Engineering Science*, vol. 228, p. 115956, 2020.

[111] Z. Kong, B. Tang, L. Deng, W. Liu, and Y. Han, "Condition monitoring of wind turbines based on spatio-temporal fusion of scada data by convolutional neural networks and gated recurrent units," *Renewable Energy*, vol. 146, pp. 760–768, 2020.

[112] R. Wan, S. Mei, J. Wang, M. Liu, and F. Yang, "Multivariate temporal convolutional network: A deep neural networks approach for multivariate time series forecasting," *Electronics*, vol. 8, no. 8, 2019.

[113] D. H. Mellor, "Causation and the direction of time," *Erkenntnis (1975-)*, vol. 35, no. 1/3, pp. 191–203, 1991.

[114] J. Williamson, "Establishing causal claims in medicine," *International Studies in the Philosophy of Science*, vol. 32, no. 1, pp. 33–61, 2019.

[115] C. Glymour, "Counterfactuals, graphical causal models and potential outcomes: Response to lindquist and sobel," *NeuroImage*, vol. 76, pp. 450–451, 2013.

[116] J. Pearl, "Causal inference," in *Proceedings of Workshop on Causality: Objectives and Assessment at NIPS 2008* (I. Guyon, D. Janzing, and B. Schölkopf, eds.), vol. 6 of *Proceedings of Machine Learning Research*, (Whistler, Canada), pp. 39–58, PMLR, 12 Dec 2010.

[117] J. Pearl, "The seven tools of causal inference, with reflections on machine learning," *Commun. ACM*, vol. 62, p. 54–60, Feb. 2019.

[118] J. Pearl, "Structural counterfactuals: a brief introduction.," *Cognitive science*, vol. 37, no. 7, p. 977–985, 2013.

[119] J. Peters, D. Janzing, and B. Schölkopf, *Elements of Causal Inference: Foundations and Learning Algorithms*. Cambridge, MA, USA: MIT Press, 2017.

[120] C. A. F. Ana Rita Nogueira, João Gama, "Causal discovery in machine learning: Theories and applications," *Journal of Dynamics & Games*, vol. 8, no. 3, pp. 203–231, 2021.

[121] H. Wang and D.-Y. Yeung, "A survey on bayesian deep learning," *ACM Comput. Surv.*, vol. 53, Sept. 2020.

[122] M. J. Vowels, N. C. Camgoz, and R. Bowden, "D'ya like dags? a survey on structure learning and causal discovery," 2021.

[123] K. Kuang, L. Li, Z. Geng, L. Xu, K. Zhang, B. Liao, H. Huang, P. Ding, W. Miao, and Z. Jiang, "Causal Inference," *Engineering*, vol. 6, no. 3, pp. 253–263, 2020.

[124] M. Eichler, "Causal inference with multiple time series: Principles and problems," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 371, no. 1997, 2013.

[125] R. Moraffah, P. Sheth, M. Karami, A. Bhattacharya, Q. Wang, A. Tahir, A. Raglin, and H. Liu, "Causal Inference for Time series Analysis: Problems, Methods and Evaluation," 2021.

[126] Y. Sun, J. Li, J. Liu, C. Chow, B. Sun, and R. Wang, "Using causal discovery for feature selection in multivariate numerical time series," *Machine Learning*, vol. 101, no. 1-3, pp. 377–395, 2015.

[127] G. Feng, K. Yu, Y. Wang, Y. Yuan, and P. M. Djurić, "Improving convergent cross mapping for causal discovery with gaussian processes," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3692–3696, 2020.

[128] J. R. J. Muñoz-Marí, G. Mateo and G. Camps-Valls, "Causeme: An online system for benchmarking causal discovery methods.," 2019. `https://causeme.uv.es/`.

[129] S. Weichwald, M. E. Jakobsen, P. B. Mogensen, L. Petersen, N. Thams, and G. Varando, "Causal structure learning from time series: Large regression coefficients may predict causal links better in practice than small p-values," vol. 123 of *Proceedings of the NeurIPS 2019 Competition and Demonstration Track, Proceedings of Machine Learning Research*, pp. 27–36, PMLR, 2020.

[130] M. Nauta, D. Bucur, and C. Seifert, "Causal discovery with attention-based convolutional neural networks," *Machine Learning and Knowledge Extraction*, vol. 1, no. 1, pp. 312–340, 2019.

[131] J. Peters, D. Janzing, and B. Schölkopf, "Causal inference on time series using restricted structural equation models," in *Advances in Neural Information Processing Systems* (C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, eds.), vol. 26, Curran Associates, Inc., 2013.

[132] P. Spirtes, "An anytime algorithm for causal inference," *Proceedings of AISTATS*, pp. 213–231, 2001.

[133] P. Spirtes, C. Glymour, and R. Scheines, *Causation, Prediction, and Search, 2nd Edition*, vol. 1 of *MIT Press Books*. The MIT Press, 2001.

[134] C. Glymour and P. Spirtes, "An Algorithm for Fast Recovery of Sparse Causal Graphs," *Social Science Computer Review*, vol. 9, no. 1, pp. 62–72, 1991.

[135] B. Huang, K. Zhang, J. Zhang, J. Ramsey, R. Sanchez-Romero, C. Glymour, and B. Schölkopf, "Causal discovery from heterogeneous/nonstationary data," *Journal of Machine Learning Research*, vol. 21, no. 89, pp. 1–53, 2020.

[136] C. Glymour, K. Zhang, and P. Spirtes, "Review of causal discovery methods based on graphical models," *Frontiers in Genetics*, vol. 10, p. 524, 2019.

[137] N. Papernot and P. McDaniel, "Deep k-nearest neighbors: Towards confident, interpretable and robust deep learning," 2018.

[138] H. Ismail Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Muller, "Deep learning for time series classification: a review," *Data Mining and Knowledge Discovery*, vol. 33, pp. 917–963, Jul 2019.

[139] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K. Lang, "Phoneme recognition using time-delay neural networks," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, pp. 328–339, mar 1989.

[140] "GitHub - ros-drivers/leap_motion: Leap Motion ROS integration."

[141] "GitHub -pal-robotics-aruco-ros: Software package and ROS wrappers of the Aruco Augmented Reality marker detector library."

[142] L. Keselman, J. I. Woodfill, A. Grunnet-Jepsen, and A. Bhowmik, "Intel realsense stereoscopic depth cameras," *CoRR*, vol. abs/1705.05548, 2017.

[143] S. Sidhik, "justagist/franka_ros_interface: Controller and Interface API for Franka Emika Panda Robot Manipulator," Nov. 2020.

[144] D. Zerbato and D. Dall'Alba, "Role of virtual simulation in surgical training," *Journal of Visualized Surgery*, vol. 3, no. 3, 2017.

[145] G. Menegozzo, D. Dall'Alba, C. Zandonà, and P. Fiorini, "Surgical gesture recognition with time delay neural network based on kinematic data," in *2019 International Symposium on Medical Robotics (ISMR)*, pp. 1–7, 2019.

[146] Y. Sun, J. Li, J. Liu, C. Chow, B. Sun, and R. Wang, "Using causal discovery for feature selection in multivariate numerical time series," *Machine Learning*, vol. 101, no. 1-3, pp. 377–395, 2015.

[147] J.Runge, *tigramite framework*.

[148] J. Runge, P. Nowack, M. Kretschmer, S. Flaxman, and D. Sejdinovic, "Detecting and quantifying causal associations in large nonlinear time series datasets," *Science Advances*, vol. 5, no. 11, 2019.

[149] P. Spirtes, "Introduction to causal inference," *Journal of Machine Learning Research*, vol. 11, no. 54, pp. 1643–1662, 2010.

[150] A. Kraskov, H. Stögbauer, and P. Grassberger, "Estimating mutual information," *Physical Review E - Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics*, vol. 69, no. 6, p. 16, 2004.

[151] J. SHI, "Quality science for smart manufacturing in the era of data-driven automation,case 2021 keynote speaker," 2021. https://case2021.sciencesconf.org/resource/page/id/19.

[152] J. Ludvik, "Diversity europe - gr iii - czech republic, reference: Ccmi/129-eesc-0000." `https://www.eesc.europa.eu/en/our-work/opinions-information-reports/opinions/food-and-drinks-sector`.

[153] E. Commission, "Food and drink industry." `https://ec.europa.eu/growth/sectors/food-and-drink-industry_it`.

[154] M. ministero sviluppo economico, "industria-alimentare." `https://www.mise.gov.it/index.php/it/impresa/competitivita-e-nuove-imprese/industria-alimentare`.

[155] N. Slimani, G. Deharveng, D. A. T. Southgate, C. Biessy, V. Chajès, M. M. E. van Bakel, M. C. Boutron-Ruault, A. McTaggart, S. Grioni, J. Verkaik-Kloosterman, I. Huybrechts, P. Amiano, M. Jenab, J. Vignat, K. Bouckaert, C. Casagrande, P. Ferrari, P. Zourna, A. Trichopoulou, E. Wirfält, G. Johansson, S. Rohrmann, A.-K. Illner, A. Barricarte, L. Rodríguez, M. Touvier, M. Niravong, A. Mulligan, F. Crowe, M. C. Ocké, Y. T. van der Schouw, B. Bendinelli, C. Lauria, M. Brustad, A. Hjartåker, A. Tjønneland, A. M. Jensen, E. Riboli, and S. Bingham, "Contribution of highly industrially processed foods to the nutrient intakes and patterns of middle-aged populations in the european prospective investigation into cancer and nutrition study," *European Journal of Clinical Nutrition*, vol. 63, pp. S206–S225, Nov 2009.

[156] E. Martínez Steele, L. G. Baraldi, M. L. d. C. Louzada, J.-C. Moubarac, D. Mozaffarian, and C. A. Monteiro, "Ultra-processed foods and added sugars in the us diet: evidence from a nationally representative cross-sectional study," *BMJ open*, vol. 6, pp. e009892–e009892, Mar 2016. 26962035[pmid].

[157] M. J. Gibney, "Ultra-processed foods: Definitions and policy issues," *Current developments in nutrition*, vol. 3, pp. nzy077–nzy077, Sep 2018. 30820487[pmid].

[158] D. A. Dickey and W. A. Fuller, "Distribution of the estimators for autoregressive time series with a unit root," *Journal of the American Statistical Association*, vol. 74, no. 366a, pp. 427–431, 1979.

[159] Y. Qin, M. Allan, J. W. Burdick, and M. Azizian, "Autonomous hierarchical surgical state estimation during robot-assisted surgery through deep neural networks," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 6220–6227, 2021.