Machine learning for the improvement of a blow-moulding process

Filippo CARA

Director: Nassim BOUDAOUD
Director: Yves GRANDVALET
Advisor: Amélie PONCHET-DURUPT
Advisor: Stéphane GALLIOT

Thesis submitted in partial fulfilment of the requirements
for the degree of
Doctor of Philosophy

Université de Technologie de Compiègne
France
November 2021

Abstract

In the manufacturing industry, especially in the automotive sector, product quality is a major indicator for evaluating the production capacity of a company. Customers are increasingly demanding in term of product quality and providing the customer with a product that comply with the specification is essential in a market that is becoming more and more competitive. In this research work, we propose to make use of machine learning algorithms to improve the overall quality of the manufactured products in an automotive industry context. Machine learning may be used to improve the process control by better understanding how manufacturing process parameters, or features, affect the quality of the finished part. In the same way, we claim that training a learning model able to infer in real-time the quality of a part, without any direct part measurement, could benefits to the overall quality control chain, by ensuring a fast reaction to quality deviation. Both approaches have been tested in a real manufacturing environment, the extrusion blow-moulding for fuel tank production. The experimental evaluation mainly showed two results. The first outcome has highlighted the complexity of applying data-driven methods in an industrial context where it is not possible to take into account all sources of product quality variability. Secondly, we have shown that the introduction of new sensors, such as a thermal camera at the end of the production process, made it is possible to infer in real-time some dimensional characteristics of the finished product that allows for a 100% quality control of the produced parts.

Contents

List of Figures

List of Tables

Introduction

The development of new technologies such as machine learning (and deep learning), IoT and cloud computing are opening up new research perspectives in the manufacturing industry. Industry 4.0 holds the promise of increased flexibility in manufacturing, along with mass customisation, better quality, and improved productivity. In this context, Plastic Omnium Clean Energy System aims to leverage these new technologies in order to keep its leadership in the manufacturing industry of fuel tanks. For Plastic Omnium Clean Energy System, Industry 4.0 is a new way of looking at performance, with a more precise and immediate vision (based on real-time indicators) of the entire production chain, but also the optimisation of production through the use of data-driven methods. In the context of an interconnected plant, the large amount of data collected from different sources —production equipment and systems as well as enterprise— can be helpful in taking decisions and contributes to a continuous improvement process. In particular, Plastic Omnium Clean Energy System thinks that the integration of machine learning models inside a complex industrial process can reduce the non-quality costs with the increase of the Overall Equipment Effectiveness (OEE). This research work focuses on the quality improvement of fuel tanks produced through the extrusion blow-moulding manufacturing process. Extrusion blow-moulding takes a thin-walled tube called a *parison* that has been formed by extrusion, entraps it between two halves of a larger diameter mould, and then expands it by blowing air into the tube, forcing the parison out against the mould. The fuel tank produced through this manufacturing process must respect some dimensional and geometrical constraints to comply with customer specifications. The thickness of the tank over the whole surface must be sufficient to ensure the robustness of the part and therefore its safety, while avoiding an excessive and unnecessary weight of the finished product. Unfortunately, measuring the thickness of a hollow part is a time-consuming operation that requires several minutes of work and that cannot be done online for each part. As a consequence,

only a subset of the produced parts can be measured. One set of statistical tools for applying such a screening is acceptance sampling. Using such tools enables decision makers to determine what action to take on a batch of products. Decisions based on frequency testing, rather than on 100% inspection, are more expedient and cost effective but it cannot guarantee the conformity of all parts of the population from which the sample was drawn. In order to reinforce the control of parts, the tank weight is measured for 100% of the manufactured parts. The weight is an indicator of how much material is composing the part and allows for overall control of part quality. Unlike thickness, which has to be measured in several areas of the tank and cannot be carried out online for all parts, weight can be easily measured for all parts and can provide an overall information about the amount of material composing the fuel tank. This thesis explores how data-driven methods, and in particular machine learning and deep learning, can be applied in the industrial context of the extrusion blow-moulding in order to improve the quality of the fuel tanks produced. Supervised machine learning is used as a tool to discover hidden patterns between process parameters of the machine and quality of the parts that have been manufactured.

In our opinion, the overall quality improvement of the manufactured parts can be achieved in two ways:

- through the manufacturing process optimisation;
- by improving quality control through inspection of all parts, it is possible to react faster to quality non-conformities and avoid sending customers parts that do not comply to the specifications, which may cause a quality recall.

We claim that machine learning, and more generally data-driven methods, can be either used to optimise the process and the quality control. By modelling the relationship between process and quality data, using a data-driven method, it is possible to infer the quality of a part given a new set of input data. Moreover, by leveraging interpretable machine learning algorithms it is possible to identify the parameters that most affect the quality of the final part.

The experimental part is predominant in this research work. Firstly, we rely on experimentation and measurement to get all the data needed to fit the statistical models. The machine has been equipped with new sensors, such as *RGB* or thermal cameras, in order to collect a new data. These new sources of data, combined with process data already available within the company, will constitute the entry point for training our machine learning models. In addition, this

research work has enabled us to develop some industrial software applications that add value to the overall extrusion blow-moulding manufacturing process. The development of these applications is an outcome of our data analysis and is one of the contributions of this research work.

Thesis structure

This PhD thesis is structured as follow: Chapter 1 focuses on detailing the industrial context in which this research work takes place. Extrusion blow-moulding, as well as the key quality characteristics of a fuel tank are described. Then, a literature review of quality control for extrusion blow-moulding process is presented. This allows us to position of our work and to subsequently define the objectives of the project. Chapter 2 describes a general method for quality improvement using a supervised machine learning approach. The second chapter also pays special attention to defining the core concepts and approaches used in machine learning, thus serving as an introduction to the machine learning algorithms and techniques extensively used in the following chapters. Chapter 3 presents a first experimental application of the method described in Chapter 2. Supervised machine learning is used to infer the weight of fuel tanks from the process data measured on the machine. This chapter also presents two software applications developed during the course of the research work presented. A first application uses an RGB camera to measure the length of the parison in real time. The second application allows for the optimisation of some critical phases of the machine such as start-ups and purge cycles. In Chapter 4 we show how thermal imaging, more precisely the measurement of surface-decay temperatures, can be used to infer the thickness of fuel tanks using a learning algorithm. Three data-driven pipelines are proposed to leverage machine learning and deep learning to infer the thickness of some critical areas. Finally, in the general conclusion we summarise our contributions and present some research perspectives that can be addressed in the future to push forward the research on this topic.

Chapter 1

Industrial context and research framework

Contents

In this first chapter, we present the industrial context of this research project. Primarily, we will review the Industry 4.0 paradigm as well as the benefits it can bring to the manufacturing industry through the application of Machine Learning

methods. Subsequently, we will describe our extrusion blow-moulding industrial research framework, and we will formalise and position our research work. Finally, the goal of this research work is presented: the quality improvement of fuel tank production using extrusion blow-moulding process.

1.1 Industry 4.0: a promise for improved manufacturing

Automotive industry is nowadays driven by global competition and the need for fast adaptation of production to the ever-changing market requests. The fourth revolution in industry, Industry 4.0, holds the promise of increased flexibility in manufacturing, along with mass customisation, better quality, and improved productivity (Zhong et al., 2017). As already occurred for the past three revolutions, technical innovations and a new way of perceiving the world, are radically changing the industry. The first industrial revolution at the end of the 18th century introduced steam-powered machines. The second one used electricity to improve productivity and to create mass production. Electronics and information technology, with the introduction of Programmable Logic Controllers (PLC) began the industrial automation and the third industrial revolution. The context of billions of people connected by mobile devices, with unprecedented processing power, storage capacity, and access to knowledge is promoting the emergence of new technologies. Artificial intelligence, robotics, autonomous vehicles, 3-D printing, nanotechnology, biotechnology, materials science, energy storage, and quantum computing are changing the world and today we are on the cusp of the fourth industrial revolution (Schwab, 2016).

The term Industry 4.0 refers to the connection among production departments, tools, machines, "individual things" in general made possible by internet and cyber physical systems (Schläpfer et al., 2015). With the digital revolution, the boundaries between physical and digital worlds are disappearing to create an interconnected factory with strong interactions between employees, machines and products. These connected entities can interact with one another using standard internet-based protocols and analyse data to predict failure, configure themselves, and adapt themselves to changes. According to the estimations made by BCG for German companies, Industry 4.0 will have a positive impact on companies with productivity and revenue growth but also on economy with more investments and with an overall six percent increase in employment during the next ten years. Productivity improvements on conversion costs, which

exclude the cost of materials, will range from 10 to 20% in automotive industry, while productivity gains of 5 to 8% will be achieved if the materials costs are factored in (Lorenz et al., 2016). The revenue growth, as a direct consequence of manufacturers' demand for enhanced equipment and new data applications, as well as consumer demand for a wider variety of increasingly customised products, is estimated at 30 billion euros a year, which is approximately one percent of the German GDP (Rüßmann et al., 2015).

Industry 4.0 is also a new way of looking at performance, with a more precise and immediate vision (based on real-time indicators) of the entire production chain, but also the optimisation of production through the use of artificial intelligence. In interconnected plants, the large amount of data collected from different sources –production equipment and systems as well as enterprise– can be helpful in taking decision and contributes to a continuous improvement process. In particular, we think that the integration of machine learning models inside complex industrial processes can reduce the non-quality costs with the increase of the Overall Equipment Effectiveness (OEE).

In the following, we will present what we consider to be the two key elements that have been contributing most to the fourth industrial revolution: data and machine learning.

1.1.1 Data

For a long time, information was documented on paper while manufacturing was realised by handicraft, therefore, the integration between information technology and manufacturing technology was neither beneficial nor feasible. Since the advent of the first electronic computer in 1940s, the rapid development of information technology (IT) has been driving manufacturing toward informatisation. Since the 1960s, the development of integrated circuits has paved the way for the advancement of computer hardware and software. Since the 1980s, TCP/IP, local area networks (LAN), the World Wide Web (WWW), and search engines emerged one after another to meet the increasing needs for data storage, indexing, processing and exchange. All these information technologies were widely applied in manufacturing. As a result, many advanced manufacturing technologies were put forward, such as computer integrated manufacturing (CIM), computer aided design (CAD), manufacturing execution system (MES), computer aided manufacturing (CAM), enterprise resource planning (ERP), and networked manufacturing (NM), etc. Recently, the rise of New IT technologies such us IoT

(Internet of Things) and cloud solutions provides new sources of data. Due to the deep fusion between IT and manufacturing on going, the degree of manufacturing smartness is progressively elevating. As a result, manufacturing data also becomes increasingly richer (Tao et al., 2018).

As a consequence of the multitude of manufacturing technologies, industrial data comes in very different forms. This implies a lot of heterogeneity in the data which tends to make it harder any data usage or comparison. Moreover, most of data available in manufacturing industry is *unstructured*. In this thesis we consider as *structured* any kind of data that can be stored in form of rows and columns in systems like databases or Excel spreadsheet. Any data that can be stored by respecting this convention, without loosing any information, will be qualified as a structured data. On the other hand, we qualify as *unstructured* any set of data that cannot be stored in a set of rows and columns without loosing information. Some types of data may be difficult to definitely classify into one or the other category. It can actually depend on the use-case and the data processing objective. For example, an image can be represented as a 2D matrix (for black and white images) or as 3D matrix (for colour images). This representation is perfectly structured. Therefore an image, or a video, can be considered as a structured data format for someone willing to conduct a spectral analysis, only interested in the pixel values and positions. Nonetheless, the same image can also be considered unstructured if we focus our interest on the content of the image. Indeed, pixel values can not be easily translated into a structured representation of the actual content.

Unfortunately, dealing with unstructured data is much more challenging in a data science perspective (Blumberg and Atre, 2003; Sagiroglu and Sinanc, 2013; Buneman et al., 1997). It requires highly complex, expensive and time-consuming feature extraction processes and operations (i.e. a feature represents a descriptor (e.g. colour of a car) in a data science context). It is estimated that the average *Information Systems* (IS) roughly contain around 15% of structured data and 85% of unstructured data. Such an assumption seems consistent with the actual status of the manufacturing industry. Furthermore, even if a more optimistic situation is considered, with a balanced rate of 50% structured and 50% unstructured data, it still appears critical to be able to mine, explore, exploit, and search this data. Consequently, we could argue that a big data context is inherently linked to unstructured data. Dealing with manufacturing data implies to use and manage large amounts of human-made data. This data comes with inherent and recurring issues which highly limit its usability without an extensive

pre-processing.

1.1.2 Machine learning and deep learning

As highlighted in the previous section, the amount of available data is exponentially increasing and it can be reasonably considered that humans will not be able, in a near future, to process, by hand, these massive amounts of data any more and perform heavy computations in a parallel manner. Nonetheless, machine learning and deep-learning (see Section 2.4) take advantage of parallel computation capabilities and large data quantities to approach human behaviours and understanding in complex situations. Machine learning, and more generally data-driven methods, provide a new way to deal with manufacturing problems.

In the last decade, the hottest machine learning sub-field, deep learning, has gained a lot of popularity due to its ability to provide state-of-the-art results in multiple domains. Deep learning is not a new idea, most of the recent proposed deep learning architectures rely on advances from the last decades of the 20th century. Deep learning regained attention when Krizhevsky et al. (2012) outperformed by a large margin all the other teams on the ImageNet (Deng et al., 2009) image classification task using a deep convolutional neural network. The reborn popularity of these computational methods can be attributed to the following reasons:

- *Increasing computer power*: GPU (Graphical Processing Unit) computing enabled, in the early 2010s, improved calculation performance in the field of machine learning. Powerful, fast and cheap GPU-devices greatly helped researchers to reach performances never achieved before. Deep learning involves huge amount of matrix multiplications and other operations which can be massively parallelised and thus sped up on GPUs.

- *Larger labeled datasets*: the advent of big data has considerably increased the size of the datasets available in manufacturing companies. The availability of an important amount of data is indispensable for the successful application of deep learning methods that require, in average, more data compared to conventional machine learning approaches. For instance, the ImageNet dataset (Deng et al., 2009) released in 2009, contains more than 14 millions annotated images that can be used for training image classifiers.

- *Advances in deep learning research*: deep learning is one of the most popu-

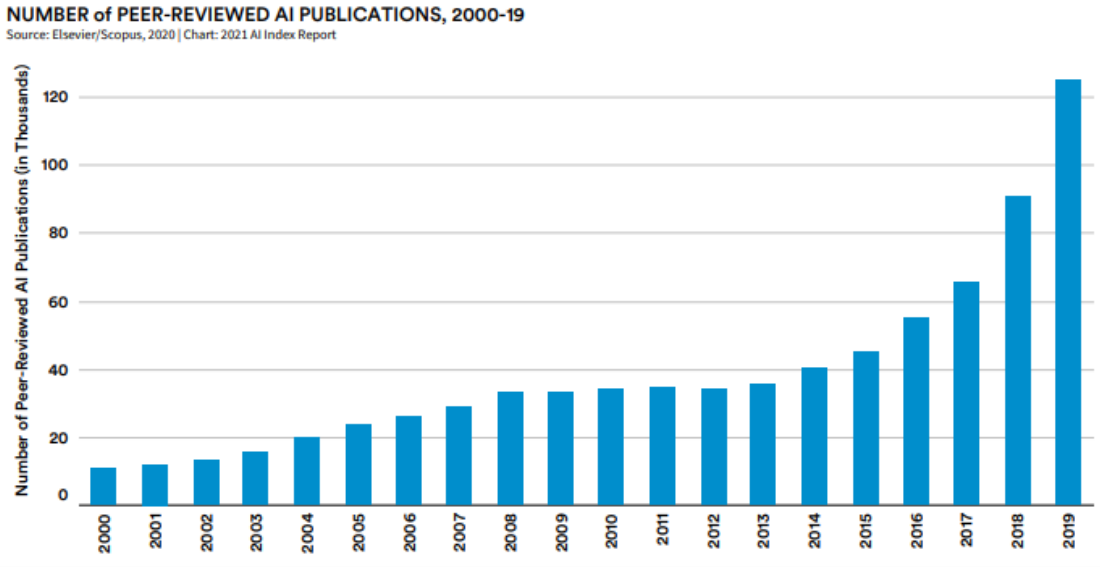

Figure 1.1: Number of peer-reviewed AI publications, 2000-2019 (Zhang et al., 2021)

lar research topics of the moment and interest in this area is growing every year. As pointed by the "AI index 2019 report" (Zhang et al., 2021), between 1998 and 2018, the volume of peer-reviewed AI papers has grown by more than 300%, accounting for 3% of peer-reviewed journal (Figure 1.1) publications and 9% of published conference papers. This increase in the number of researchers leads to faster research progression.

- *Open source tools and models*: the democratisation of open sources frameworks such as *PyTorch* (Paszke et al., 2019), *Tensorflow* (Abadi et al., 2015), *Keras* (Chollet et al., 2015) and *Scikit-Learn* (Pedregosa et al., 2011) allow to apply machine learning and deep learning methods with a few line of code. Moreover, the machine learning community is rather open in sharing results. There are many pre-trained models available online, ready to be used as a starting point for *transfer learning* (2.4.3.3).

1.1.3 Review of opportunities for the manufacturing industry

In this section, we present the result of a literature review we conducted to highlight some of the possible applications where machine learning could be applied in order to create value for manufacturing companies. This study should

Table 1.1: ML opportunities in manufacturing

Domain	Benefits	Bibliography
Quality	Decrease the product failure rate at the end of the production line. Optimise key performance index of the final product to meet customer needs.	(Lieber et al., 2013; Li et al., 2018a; Chen et al., 2008; Nagorny et al., 2017; Haeussler and Wortberg, 1996)
Maintenance	Increase the availability of the production line by preventing the breakdown of equipment in advance. Predict the risk of malfunction of the production line and arrange proactive maintenance.	(Nguyen and Medjaher, 2019; Lee et al., 2017; Einabadi et al., 2019; Li et al., 2017; Liu and Zio, 2016)
Fault Diagnosis	Prognostic diagnose of production line failure event. Identify the malfunction device of the production line. Predict the abnormal behaviours of machines and equipments.	(Toma et al., 2020; Wong et al., 2006; Chen et al., 2014; Malik and Mishra, 2017; Arabacı and Bilgin, 2010)
Scheduling	Logistic management of the production line, which can maximise the throughput of the production line. Buffer control and product routing management.	(Morariu et al., 2020; Woschank et al., 2020; Lolli et al., 2019; Zhang et al., 2019; Gomes et al., 2016)

not, in any case, be considered as an exhaustive list of possibilities. Results are summarised in Table 1.1.3.

1.2 Industrial domain: extrusion blow-moulding

Our industrial process, extrusion blow-moulding, is used to form hollow thermoplastic objects (especially bottles and containers). The process takes a thin-walled tube called a *parison* that has been formed by extrusion, entraps it between two halves of a larger diameter mould, and then expands it by blowing air into the tube, forcing the parison out against the mould. The outside of the thin-

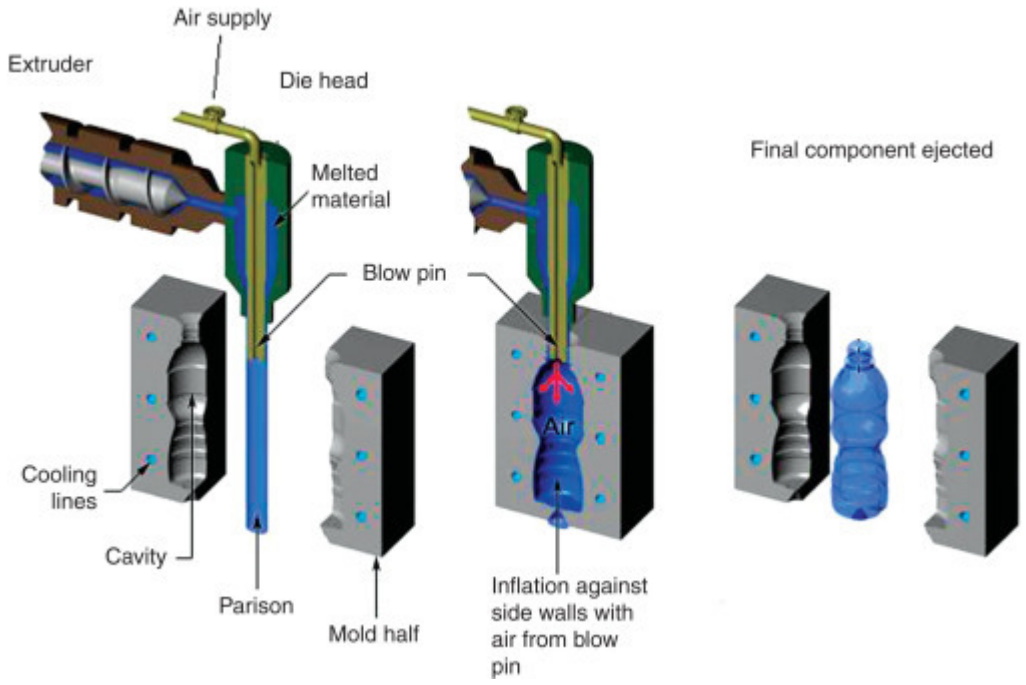

Figure 1.2: Extrusion blow-moulding (Goodship et al., 2015)

walled part takes then the shape of the inside of the mould (Poli, 2001).

As the name suggests, extrusion blow-moulding is composed of two sub-processes: extrusion and blow-moulding.

- *Extrusion:* extrusion is a continuous-flow process where a plastic material feedstock is fed through a hopper onto a feeding transfer screw. The thermal energy provided by the heating clamps as well as the mechanical energy provided by the screw rotation allows the melting of the plastic material.The melted material is then pushed into an extrusion head and the die gap will typically create a tubular extruded cross-section, round or oval depending upon the final shape of the finished blow moulded part. The die gap is the distance between the inner mandrel and the outer bushing. The gap can be varied during the extrusion, due to the tapered nature of the die, by moving either the mandrel or the bushing in a vertical direction. The process of variable die gap extrusion is referred to as parison programming and is utilised to manipulate the thickness distribution in the final part (Diraddo and García-Rejón, 1993).

- *Blow-moulding:* Unlike extrusion, blow-moulding is a discontinuous-flow process. In extrusion blow moulding, the parison is vertically suspended in

the air during which time two mould halves enclose it by the action of a pneumatic or hydraulic mechanism. Internally applied air pressure causes the parison to inflate and take the shape of the inner parts of the moulds. Once the blow operation is completed and the part has been cooled down suitably for ejection, the mould opens and the part is ejected, allowing the parison to be extruded through the mould for the next cycle.

Each phase has parameters that influence the subsequent phases and, ultimately, the characteristics of the finished product. The multiple phases mean that the number of parameters that can be adjusted on the process is fairly high. It is possible to fine-tune temperatures, screw speeds and throughputs for the extrusion, as well as the pressure curves, moulds opening and closing times for the blow-moulding phase. In addition, the extrusion blow-moulding process has a certain dynamic: it takes a certain amount of time for the adjustment of one of the process parameters to have an effect on the products. This dynamic is mainly due to the thermal inertia of the solid tooling. One of the most critical part of the process is the parison formation. In fact, the dimensions of the blow moulded article are directly related to the dimension and thickness of the parison. Furthermore, the thermo-mechanical history of the material during the parison formation stage and the resulting weight and diameter distribution of the parison have a great influence on the characteristics of the subsequent inflation and cooling stages. The shape and the dimensions of the parison are the result of complex interactions between the molten polymer and the thermo-mechanical conditions that influence the melt after it leaves the extruder die. Parison formation is affected by two phenomena knows as *swell* and *sag*. Parison swell, occurring both in diameter and thickness, is due to the nonlinear viscoelastic deformation of the polymer melt in the extrusion die. Sag is caused by gravitational forces that act on the suspended parison (Huang and Liao, 2002).

There are many variations of this process including equipment made to extrude multiple parisons simultaneously, equipment that can extrude multiple layers within the same parison, equipment with rotary capability that will hold several moulds and can provide a continuous nonstop process. Moreover, the extrusion blow-moulding process can be split into two subcategories: intermittent extrusion blow moulding and continuous extrusion blow moulding. With intermittent extrusion blow moulding, the extruder fills a reservoir with plastic. Once the reservoir has been filled, a plunger is activated and pushes the material from the reservoir through the extrusion head. On the other hand, in continuous blow-moulding, the plastic is extruded permanently in a continuous manner

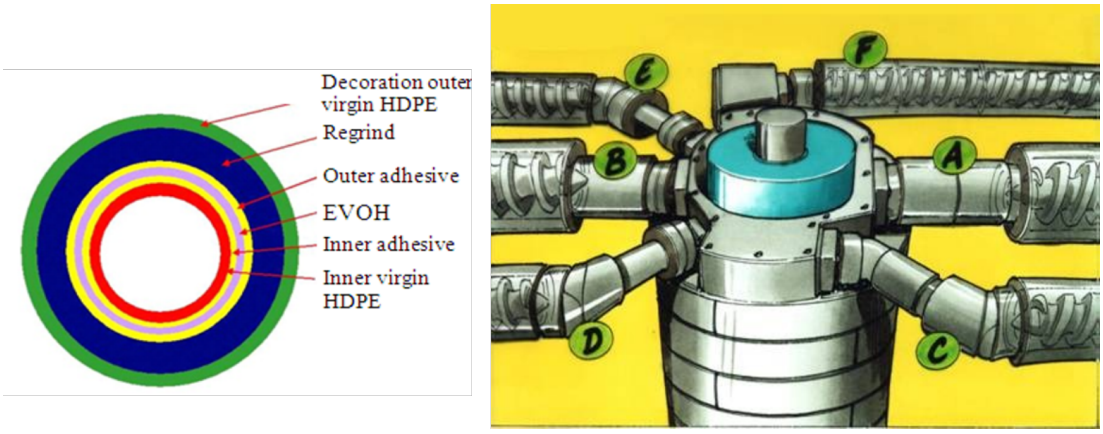

Figure 1.3: Co-extrusion process

while the machine runs.

In this thesis project we have been studying a continuous extrusion blow-moulding process, whose finished products are obtained from the overlay of multiple plastic layers. This particular type of blow-moulding process is known as *co-extrusion*. Co-extrusion was born out of the need of reduce the permeability of fuel tanks. The steps for producing a multi-layer plastic product are the same as those used by the traditional single layer process, except for the number of extruders involved in the manufacturing process. Up to six extruders can be used simultaneously to melt different plastic materials. The goal is to create a multi-layer tank with the use of different materials, as shown in Figure 1.3.

The extrusion blow-molding process taken into account all along this research study is used to produce different tank models. In fact, by simply changing the moulds, it is possible to manufacture tanks with various shapes and sizes. The ratio between the separate plastic layers, as well as the material throughput, need to be adjusted accordingly to take into account the different size or shape of each tank model. This makes the extrusion blow-molding a flexible process which allows for rapid responses to costumers requests. The fuel tanks are usually manufactured in production campaigns of a few days, also known as batches.

The extrusion blow-moulding constitutes one of the various stages necessary to produce finished part. The stages needed to manufacture a finished part may vary depending on the type of product. For instance, a fuel tank container, a plastic bottle or a plastic bumper may require different post-production processes. Further information on how a fuel tank is produced are available in Appendix

4.6.2. As far as our thesis work is concerned, we will only focus on the extrusion blow-moulding stage.

1.2.1 The key parameters of extrusion blow-moulding

Extrusion blow-moulding requires the fine-tuning of multiples adjustable parameters to reach the conditions for producing the final part. One of the most critical process parameter involving the extrusion sub-process is the material throughput. Controlling the amount of material passing through each extruder is crucial for the following reasons:

- It affects the cycle time. The throughput affects the ejection rate of the parison and therefore the cycle time. A cycle time reduction results in an higher manufacturing rate.
- The throughput ratios among the different 6 extruders should be controlled to ensure the correct amount of material on each of the 6 layers composing the tank thickness.

The throughput depends mainly on the rotation speed of the extruder and slightly on the extruder temperature. The temperature of the extruder affects the melting of the plastic material and the material viscosity may change. Depending on the material viscosity the throughput may change given the same extruder speed. For what has been said so far, controlling the temperature of the extruder along the screw length is mandatory to ensure constant and repeatable melting condition and therefore ensure a constant throughput. Another key parameter is the opening of the die gap of the *head* of the machine. By controlling the opening of the gap during the cycle time we are able to distribute more or less material along the parison length. This operation is important to ensure to put enough material in parison zones that are most stretched during the blowing phase. Of course, the parison length plays a key role in ensuring that the material distribution is well positioned relatively to the mould position. Unfortunately, in the production process in question, there is no control system for measuring the parison length. In continuous extrusion blow-moulding, the cycle time triggers the blowing cycle.

Table 1.2: Blow-moulding key parameters

Extrusion			
Parameter	**Range**	**Description**	**Dependencies**
Speed* (in RPM)	[0, 90]	Rotation speed of the screw	
Throughput* (in Kg/h)	[0, 400]	Material throughput in screw	Speed, temperatures
Melt pressure* (in % of motor nominal torque)	[0, 100]	Pressure at the end of the screw	Speed
Feeding temperature* (in °C)	[0, 100]	Temperature at the entrance	
Melt temperature* (in °C)	[0, 250]	Temperature at the end	Feed. temperature, pressure
Cycle time (in s)	[60, 120]	Tank production cycle time	
Parison profile (in %)	[0, 100]	Die gap opening	Cycle time
Parison length (in mm)	[0, 3000]	Length during extrusion	Parison profile, cycle time
Blow-moulding			
Parameter	**Range**	**Description**	**Dependencies**
Blowing pressure** (in bar)	[0, 12]	Blowing pressure inside moulds	
Cooling water temperature (in °C)	[5, 30]	Temperature of cooling water (moulds)	

* For each extruder

** There exist 4 different blowing circuits

With regard to the blow-moulding sub-process, the most critical parameters are the blowing pressure pressures (there exist 4 different air blowing circuits), the cooling water temperature and its throughput. The blowing pressure ensures the parison to inflate and take the shape of the inner parts of the moulds. Without enough air pressure the plastic material does not adhere to the mould surfaces, preventing a correct material cooling. In the same way, the amount of water passing through the cooling circuit, as well as its temperature, affect the cooling capacities of the moulds. Table 1.2 summarises the key parameters of an extrusion blow-moulding process that were identified together with process experts.

1.2.2 The key quality characteristics of a blow-moulded fuel tank

Quality control is generally expressed as the verification of the conformity of the process and the product/service to the requirements of its quality standard. The ISO 9000 standard defines quality control as "a part of quality management focused on fulfilling quality requirements" (ISO, 1992). In our context, the requirements are defined by the customers, and measurements on the product and compared with the customer requirements, if the measurements meet the customer specifications, the part can be sent to the customer, otherwise the part must be rejected. In blow-moulded parts, , the distribution of the material over the entire surface of the part surface plays a key role in ensuring that the finished product meets customer specifications. In the extrusion blow-moulding process, we are mostly interested in the dimensional/geometric characteristics. In fact, the main purpose of the quality control of the blow-moulded part is to asses the integrity of the plastic shell. The thickness of the tank over the whole surface must be sufficient to ensure the robustness of the part and therefore its safety, while avoiding unnecessary excess weight on the finished product. Measuring the thickness of a hollow blow-moulded part is difficult as there is no access to the inner surface. Moreover, the thickness is relatively small, with value ranging from 3 to 8 millimetres. This requires an indirect measurement with an accuracy of at least 0.1 millimetres.

Background in thickness measurement Traditional methods to measure the thickness of hollow parts rely on ultrasonic instruments that provide satisfactory results while avoiding the destruction of parts. The principle of *Ultrasonic Thickness Measurement* (UTM) is to measure the time needed for the

ultrasonic wave to traverse the material. Some of the advantages of UTM over other nondestructive methods are:

- the possibility to measure parts with just one accessible surface,
- the suitability to industrial conditions,
- its sensitivity and accuracy.

These methods are extremely accurate for sample quality control, but they present a major drawback: they cannot be used for online measurement. The measurement of a large number of points, which is necessary to estimate the distribution of the material over the entire surface, is time-consuming and cannot be done online in production. Hence, the quality control of wall thickness is only carried using a sampling approach. Recently, new technologies involving the use of terahertz waves have been developed to accurately measure the thickness of materials without any contact with the material itself. These methods have proven to be extremely powerful for measuring the thickness of automobile paint (Su et al., 2014; Krimi et al., 2016), or pharmaceutical tablets (May et al., 2011). Of all the methods found in the literature, terahertz-based systems seem to be the only ones that can be used to perform real-time thickness measurement, but in order to use them in real-time, the measurement sensor must be installed on a robot, or collaborative robot, which can significantly affect the price of the complete measurement solution.

Another well-known technique for measuring thickness of parts is *Computed Tomography* (CT). Typical areas of use for CT in industry are in the detection of flaws such as voids and cracks, and particle analysis in materials. In metrology, CT allows measurements of the external as well as the internal geometry of complex parts. As stated by De Chiffre et al. (2014), CT is particularly suitable to investigate moulded polymer parts, thanks to the good penetrability of X-rays in this material. Even if CT-based techniques are extremely powerful, they require laboratory conditions and do not lend themselves well to real-time thickness control. Moreover, this equipment may be very expensive, questioning its profitability. Eddy current testing, widely applied for the non-destructive thickness measurement of metallic parts (Cheng, 2017; Mao and Lei, 2016; Wang et al., 2015; Yin and Peyton, 2007) is also non-destructive, but cannot be used as polymers composing the blow-moulded part are not conductive.

In the last decades,*thermal imaging*, a non-contact technology capable of measuring large surfaces in a single shot, has been studied as a possible method to

infer the thickness of a solid element (Sun, 2003, 2006; Choi et al., 2008; Benítez et al., 2008; Zeng et al., 2012; Li et al., 2018b; He et al., 2013). Among the most widely used thermal imaging approaches we can mention: *pulsed thermography* in which a brief controlled thermal stimulation pulse is applied on the tested piece, *step heating thermography*, in which a continuous, uniform heat flow is applied for a long period, and *lockin thermography*, in which a periodic heat input is used. The main idea behind these approaches is to transfer energy to the test piece and to monitor its surface temperature evolution over time. In flash thermal imaging, for example, some flash lamps provide the thermal impulse, and the infrared camera monitors the surface-temperature decay on the heated surface. On the other hand, step-heating thermal imaging is using a long pulse of low intensity heat stimulation. Unlike pulsed thermal imaging, step-heating technology monitors the temperature raise over time while the heat energy is transferred to the test piece. The approach of monitoring the surface-temperature may also be applied without actively providing heat to the test part, especially for parts that are hot after the manufacturing process. The proposition of thermal imaging to measure the thickness of the tank is one of the major contributions of this research work and the proposed approach will be presented in detail in Chapter 4.

Due to the impossibility of measuring the thickness of all parts produced, in practice, weight is often measured as an alternative. The weight is an indicator of how much material is composing the part and allows for an overall control of the quality of the part. The weight has a lower boundary to ensure that sufficient material is composing the tank and an upper boundary to avoid unnecessary weight of the finished product. Unlike the thickness, which has to be measured in several areas of the tank and cannot be carried out online for all the parts, the weight requires a simple weight scale, placed in the area where the blow-moulded part is discharged.

Another quality issue that may occur is the fuel tank contamination. If the extrusion blow-moulding machine screws are not properly emptied following the so called purge procedures, or cycles, some material can remain attached to the screws and can solidify. This may cause the presence of unwanted burned material in the manufactured fuel tank, which generates a visual non-conformity of the part, and which can lead to tank permeability problems. The contamination problem identification completely relies on human visual inspection.

In order to produce a complete fuel system (see Appendix 4.6.2), other manufacturing process are required after the extrusion blow-moulding (see Appendix 4.6.2). Operations such as components welding or the assembly of pieces re-

Table 1.3: Quality indicators of a blow-moulded fuel tank

Indicator	**Unit**	**Value range**	**Description**
Global Thickness	mm	[3, 8]	Thickness of the part measured at several critical points
Weight	Kg	[6, 14]	Total weight

quire further quality checks. The deepening of any quality control that does not directly concern the extrusion blow-moulding is out of the scope of this research work. Table 1.3 summarises the two quality characteristics of a blow-moulded part we are interested in, and which must be monitored to ensure that the parts produced comply with customer requirements. However, in Chapter 3, we will present an initiative which aims to partially reduce the contamination problems.

1.3 Quality control and process monitoring in the extrusion blow-moulding process: a state-of-the-art

A literature review was carried out to identify previous work aimed at improving the overall quality of parts produced by a blow-moulding process. The literature review was carried out using three different databases: *Scopus*, *Google Scholar* and *Crossref*. Subsequently, a screening exercise was carried out to select only the most interesting articles relevant to our scientific problem. Only about ten articles were identified as potentially interesting and strictly related to our research work. Figure 1.4 highlights the recurring words in the title and abstract of the retained articles.

Two main strategies have been developed to improve the quality of the blow-moulded parts: *expertise-based* and the *data-driven* approaches. In the remaining part of the current section, we will investigate these two approaches proposed in the literature and we will discuss the possibility of using these methods in our industrial context.

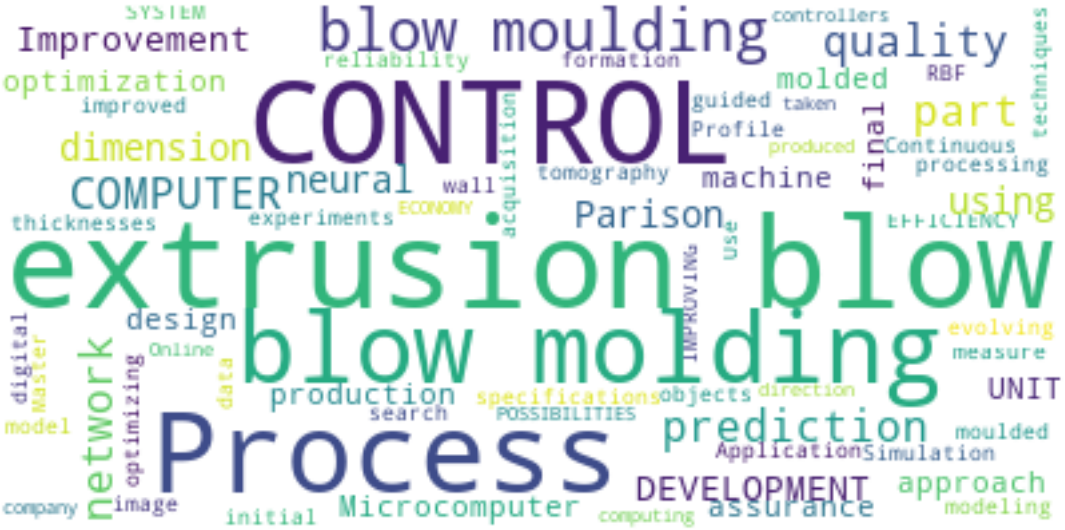

Figure 1.4: Most recurrent words in article titles

1.3.1 Expertise-based approaches

Expertise-based approaches make use of physics and simulation to model the manufacturing process and to fine tune process parameters given the simulated final part characteristics. Different strategies have been used to model the whole process that is the parison extrusion, clamping, inflation and cooling. Lee and Soh (1996) used a finite element model of thin film to simulate blow moulding processes, and applied the feasible direction method to minimise the parison volume at the constraints of part thickness. The proposed parison design simulation is composed of the following stages. The finite element model predicts the thickness of the blow-moulded part from a given parison profile or preform. The resulting thickness distribution of the part is submitted to the optimisation model to generate a new parison profile. The new preform design is compared with the old one. If there is any improvement, the new preform design is again passed to the finite element model, and the loop is repeated until no further design improvement can be achieved. Author showed that the presented approach makes the optimisation algorithm more efficient and reduces the computational requirement drastically.

Other expertise-based methods rely on iterative fine-tuning loop involving the prediction of the final part characteristics, such as the weight or the thickness. Two approaches, with regard to material behaviour during inflation, have been applied for the prediction. The first method assumes that the polymer melt behaves as a viscous or a viscoelastic fluid, whereas the second method assumes

that the melt behaves as an elastic solid. The assumption that the parison behaves as a viscous or a viscoelastic fluid results in a very complex computational formulation. Poslinski and Tsamopoulos (1990) treats the parison as a Newtonian fluid subject to a non-isothermal inflation. Parison position and cooling during the inflation are predicted as a function of time. Some experimental final part thickness distributions are obtained and compared to simulation results for a simple mould geometry and a constant initial thickness preform. Also, the inherent elastic nature of the polymer melt is not considered, since the formulation assumes a Newtonian fluid. Ryan and Dutta (1982) and Khayat et al. (1992) assume a viseoelastic behaviour of the polymer melt. The inflation is modelled as a dynamic process, predicting the parison inflation as a function of time. A free inflation was considered; attempts with confined inflation, that is, employing a mould geometry, have not been handled to date. The approaches discussed by Poslinski and Tsamopoulos (1990); Ryan and Dutta (1982); Khayat et al. (1992) are interesting but make very strong assumptions or deal with very particular cases.

Attar et al. (2008) proposed an approach to assist the development phase of a new product and to optimise the weight of the part and its thickness distribution. Firstly, simulation of the extrusion blow moulding process and preliminary experimental trials were performed concurrently to assist in the development of the part. Once the numerical modelling of the part is done, improvement of the production process is performed based upon the desired objective function, i.e., a uniform part thickness distribution and/or minimal part weight. The optimisation is performed in two sequential steps, weight optimisation then thickness optimisation, by the systematic manipulation of the operating conditions, such as the parison dimensions. A process modelling methodology was employed to demonstrate the reduction in the part development time using the new model-based approach (Figure 1.5). It is a trial and error process, which is time-consuming and produces a lot of material scrap. On the other hand, the concurrent process optimises parameters virtually, and therefore, eliminates scrap, machine downtime and the need for experimental optimisation. The results demonstrate that there is a significant reduction in span time and in effort, since much of the delay and rework is eliminated using the simulation-based development process.

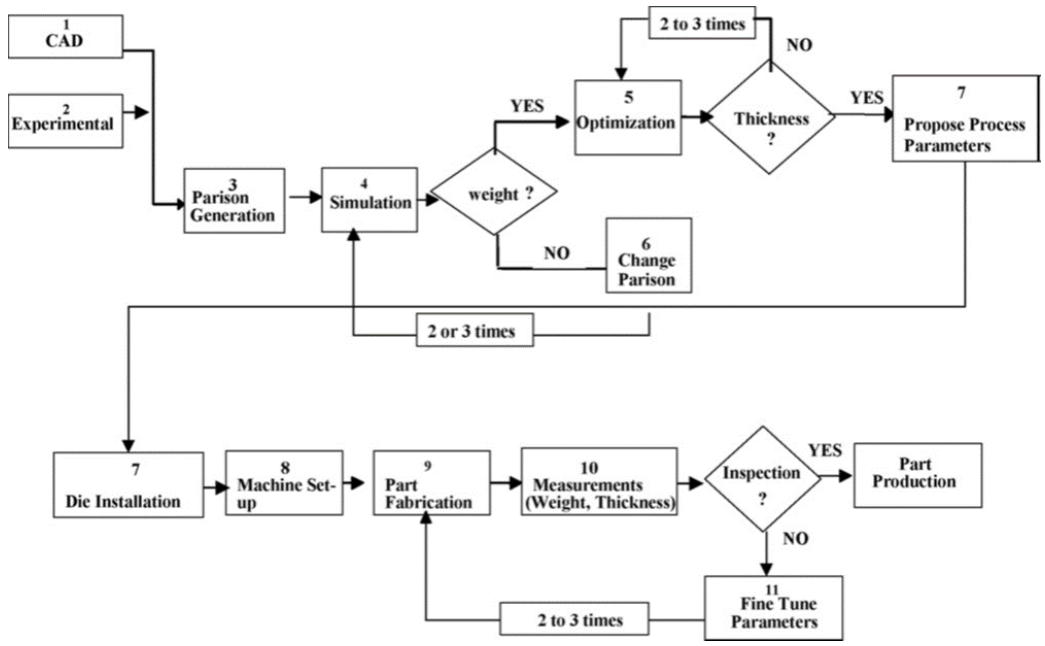

Figure 1.5: Workflow for development process optimisation (Attar et al., 2008)

1.3.2 Data-driven approaches

As the production process is complicated to be modelled physically, most of the work carried out previously makes use of data-driven methods to understand what process parameters affect the most the quality of the blow-moulded parts. The patterns discovered within data allow a subsequent optimisation of the process parameters.

Diraddo and Garcia-Rejon (1993b) have employed an entirely different methodology as a forward predictor of the inflation process. Neural networks are used for the on-line prediction of the final part thickness distribution from the initial process conditions. The fully connected neural network inputs include the initial parison thickness and temperature profiles, the bottle mould geometry and a rheological parameter representative of the raw material. The output data corresponds to 71-dimensional vector representing the thickness of 71 sampling points over the bottle length. The bottle thickness profile was measured by cutting the bottle into segments and measuring each with a hand micrometer. The neural network (see Section 2.4.1.4) is trained by employing a gradient descent optimisation regression approach and mapping a broad range of output and input data. Once trained, the neural network predicts outputs based on new inputs. Authors claimed that the proposed data-driven method has several advantages

over simulation-based methods. On first principles include faster response and the network's ability to update a model to account for process shifts. Neural networks do not allow for an understanding of process fundamentals, they require a great deal of experimental data for the training procedure and problems can arise with extrapolation beyond the range defined by the experimental data. Therefore, the methodology is better suited for on-line applications, where fast response and following of process shifts is crucial. The same authors have employed neural networks for the modelling of the process with the inverse formulation (Diraddo and Garcia-Rejon, 1993a). Compared to the previous approach, they tried to predict the initial parison thickness given the thickness of the final part. It would be valuable to determine the process conditions given the specified final part thickness distribution. The proposed approach was, in most cases, able to predict the constant thickness parison profile required for the specified part thickness distribution.

Ramana and Reddy (2013) propose another use of data mining techniques to identify the factors that significantly affect quality, modelling relationships between input attributes and target attribute (yield, quality, performance index, etc) and predicting quality levels of given input attributes. A clustering analysis is first applied on process data to partition the population in different groups. By comparing these groups with data labels, they observe that rejected parts are classified in different groups than parts that meet the customer's specifications. Naive Bayes and decision tree are then applied with the main purpose of classifying the quality of the parts given the input parameters. The process parameters used as input data are: the process cycle time, the extruder temperatures in different zones, the extrusion die temperature, the expulsion time, the parison length, the parison shape, the blowing pressures and the inflation time. Naive Bayes and clustering models were found to have better accuracy compared to decision trees. The Authors claim that the model deployment has led to a general improvement in the quality of the parts. Unfortunately, the scientific paper does not provide any additional information on how the model was deployed.

1.3.3 Discussion

Two different approaches have been proposed in the scientific literature regarding quality optimisation in extrusion blow-moulding processes: expertise-based and data-driven approaches. Both methods try to predict the quality of the final part given the process conditions with the main purpose of fine-tuning the manufacturing process; they have relative advantages and disadvantages. Expertise-based

approaches need strong assumptions or simplification to account for the overall process complexity. Data-Driven methods are faster and they can be used online. On the other hand, they require a great deal of experimental data for fitting their models and problems can arise with extrapolation beyond the range explored by the experimental data.

The literature also highlight other fundamental aspects:

- Due to the complexity of the studied production process, most recent approaches to improving process control or the quality of manufactured parts use data-driven methods (Diraddo and Garcia-Rejon, 1993b,a; Ramana and Reddy, 2013).
- No articles where found on the process of multi-layers extrusion blow-moulding (Co-extrusion). The literature presented mainly focuses on plastic bottle or simple plastic containers which are commonly produced through mono-layer extrusion blow-moulding. For this reason, some of the proposed methods are not applicable to our production process. For instance, the approach of Diraddo and Garcia-Rejon (1993b) for evaluating the geometrical dimensions of a plastic bottle requires the measurements of the material throughput exiting the extrusion head as well as the rheological parameter representative of the raw material. For technical and economical reasons, this information is impossible to collect in our production process.
- Most of the presented research works have focused on the product development phases, with little attention paid to controlling the quality of the part in production. It would be interesting to identify in real-time those factors that can lead to a degradation of the product quality. This would allow a faster process adjustment and even greater reduction of the scrap rate.

Supported by the scientific literature, and taking into account technological advances in the domain of the data acquisition in the manufacturing environment, we claim that data-driven methods are the right tools to investigate the interactions between extrusion blow-moulding process data and the corresponding product quality data. We claim that this is particularly true in a co-extrusion production process, where there are six extrusion screws, extruding 6 different types of polymers with different physical-chemical properties. In our opinion, physically modelling the co-extrusion production process would be quite complicated as too many assumptions would have to be taken into account. The

interest of using data-driven methods is also confirmed by the scientific literature of the last years involving quality improvement. Data-Driven methods for product quality control have been successfully applied in multiple industrial domains, from steel industry (Lieber et al., 2013; Li et al., 2018a) to plastic industry (Chen et al., 2008; Nagorny et al., 2017, 2018; Haeussler and Wortberg, 1996; Tellaeche and Arana, 2013; Sharma et al., 2017) and to the semiconductor manufacturing processes (Melhem et al., 2016; Lenz and Barak, 2013; Jiang et al., 2020).

1.4 Research objectives and methodology

Because manufacturing processes are becoming more and more complex, and the high level of requirement in the automotive industry regarding safety and environmental impacts, Plastic Omnium is continuously seeking for innovation throughout its different projects that allows the company to remain leader in its field. For Plastic Omnium, the Industry 4.0 paradigm can provide a new way of looking at performance, with a more precise and immediate vision (based on real-time indicators) of the entire production chain, but also the optimisation of production through the use of data-driven methods. An in-depth presentation of the activities of Plastic Omnium is available in Appendix 4.6.2. We have identified four main pillars driving the Industry 4.0 revolution:

- *Smart factory* refers to the set of initiatives that enable for a real-time traceability of what is happening inside production plants. With a real-time system, raw materials, work in progress and finished products are bar coded and tracked throughout the manufacturing process. The digitisation of the data allows also for a better monitoring of the production performance.

- *Digital industrialisation* refers to the set of initiatives that make use of Virtual Reality (VR) and digital twin models to reduce the deployments costs of new machine and to optimise the plant machine layout. Other initiatives aim to optimise the ergonomics.

- *Predictive quality* refers to the use of data-driven methods to reduce the rate of product rejection at the end of the production line. This topic also covers initiatives to improve quality control and reduce destructive testing for quality control purposes.

- *Predictive maintenance* refers to the use of data-driven methods that are meant to analyse equipment status and forecast when maintenance should

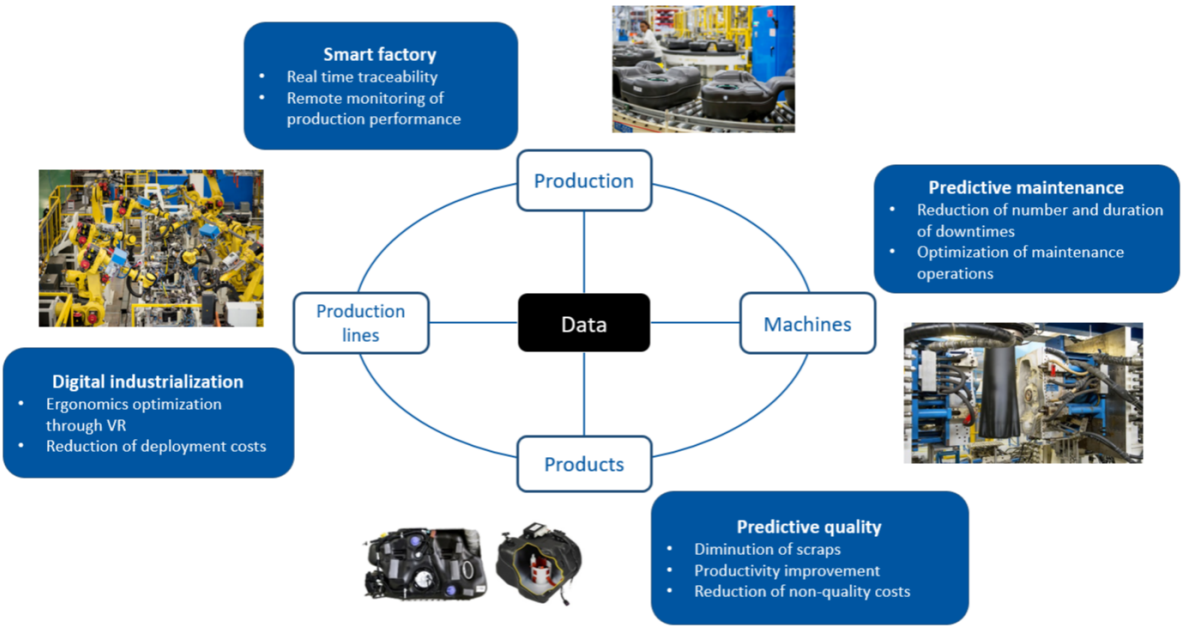

Figure 1.6: Industry 4.0 pillars for Plastic Omnium

be performed. Predictive maintenance aims to reduce the number and the duration of the unplanned down-times and to optimise the maintenance operations.

Figure 1.6 shows how these topics integrate within the research environment of Plastic Omnium. Among the different topics, this research work will focus on "Predictive Quality" topic. For an equipment manufacturer like Plastic Omnium Clean Energy Systems (CES), the "Cost of Non-Quality" (CNQ) is one of the key indicators most used to evaluate the production capacity and bad or "scrap" parts are not acceptable. Therefore, this research work aims at looking for the best way to use the data collected on the machines and the corresponding product in order to propose a data driven approach inferring the quality of a product. The choice to use a data-driven approach is motivated by an ever-increasing availability of data within the manufacturing plants as well as from what has been said in the previous section (see Section 1.3). Using the data that is already available within the company will be an important part of the global study, because it will be the first input going into the developed monitoring system. In the case of Plastic Omnium CES, the data coming from the systems *PES* (Production Execution System) and *DASIP* (Data Acquisition for the Supervision of Industrial Process) will be important. The two systems allow for the traceability of the produced parts as well as a monitoring of the different events happening on the plant's

machines (for PES), and DASIP is monitoring key parameters of production processes. Other data sources will be investigated during this work.

This system will help detecting product non-conformities to plan the corrective actions accordingly. The expected benefits are:

- the reduction of product non-conformities,
- the overall quality control improvement,
- process improvement. In fact, product quality improvement requires working on the manufacturing process. This will help for a better comprehension of the extrusion blow-moulding process.

To reach the final objective of reducing scraps and non-quality costs, a data-driven methodology will be proposed. The methodology is articulated around three principles consecutive stages, to best respond to the research objectives:

1. Proposal of a general framework to deal with "Predictive Quality" topics,
2. Application of the proposed methodology to the studied industrial context,
3. Proposal of a decision-making system.

Each of the four stages requires to overcome either some scientific or industrial issues or obstacles. In the scope of this thesis, we will only be able to work on the first two topics, although some elements concerning the last stage will be discussed in the conclusion of this research work.

Proposal of a general framework to deal with "Predictive Quality" During the first stage, we will leverage scientific literature to propose a general methodology to deal with all predictive quality topics. From an industrial point of view, this first stage requires identification of all necessary tools and methods needed to conduct a project from start to finish.

Application of the proposed framework to the industrial context The application of the proposed framework to the industrial use case requires the critical data needed to answer our research question to be identified. Then, a machine learning algorithm will be applied to infer the quality of a part given the set of input parameters identified in the previous stage. From a scientific point of view, this stage will require to identify or build an efficient and robust machine learning algorithm able to model the transfer function which relates the

input process data and the output quality data.

Proposal of a decision-making system The last stage will involve the implementation of a decision-making system able to assess the quality of a produced part and, if necessary, to reject it. The system should be able to communicate with the systems already in place, such as the *PES* system, to declare the part as non-compliant and to alert quality and production teams to trigger corrective actions.

1.5 Conclusion

In this fist chapter, we have described the context of our research project. The French automotive company Plastic Omnium, leader in the production of automotive plastic components, aims to take advantage of fast growing amount of data available in manufacturing plants to improve quality of fuel tanks produced through extrusion blow-moulding manufacturing process. This complex manufacturing process is composed of two sub-processes: *extrusion* and *blow-moulding*. Extrusion is a continuous-flow process where a plastic material feedstock is fed through a hopper onto a feeding transfer screw. The thermal energy provided by the heating clamps as well as the mechanical energy provided by the screw rotation allow the melting of the plastic material. The die will typically create a tubular extruded cross-section, round or oval depending upon the final shape of the finished blow-moulded part. Unlike extrusion, blow-moulding is a discontinuous-flow process. In extrusion blow moulding, the parison is vertically suspended in the air during which time two mould halves enclose it by the action of a pneumatic or hydraulic mechanism. Internally applied air pressure causes the parison to inflate and take the shape of the inner parts of the moulds. The fuel tank produced through this manufacturing process must respect some dimensional and geometrical constraints to comply with customer specifications. The thickness of the tank over the whole surface must be sufficient to ensure robustness of the part and therefore its safety, while avoiding an excessive and unnecessary weight of the finished product. The scientific literature, involving quality control and process monitoring in the extrusion blow-moulding process, identifies two different ways of working on the topic of improving the quality of a blow-moulded parts. The first approach mainly relies on expertise and expert systems to optimise the manufacturing process. The second method makes use of data and data-driven methods to try to explain the variability of final part quality given the input manufacturing process parameters. The limitations of

the approaches proposed in the literature, as well as the motivations that have oriented us towards the use of data-driven methods have been discussed. Finally, the main research objectives, as well as the research axes that will drive our research work, are presented. In the next Chapter, we will describe a general method to handle the predictive quality topics.

Chapter 2

Machine learning for quality control

Contents

2.1 Introduction

In this chapter, we describe a general framework to model the relationship between machine process data and product quality. The method described relies

on four consecutive stages: data acquisition, data processing, exploratory data analysis and supervised machine learning modelling. The objectives for the manufacturing company applying this method can be twofold: on the one hand, to determine the process parameters that have a significant influence on the quality of the finished parts. It is then possible to monitor them to ensure their stability over time. On the other hand, the trained model can be used to infer part quality from the process parameters. In this way, it is possible to provide a quality status to each part produced. This is particularly interesting when quality control cannot inspect all the parts produced, which is the case in most companies. By providing the quality status for all parts produced, it is possible to react faster to quality non-conformities and to avoid sending to the customer a part which is non compliant with the specifications. In the second part of this chapter, the definition of the core concepts and approaches used in machine learning are presented, thus serving as an introduction to the machine learning algorithms and techniques used in this thesis.

2.2 Towards data-driven quality control

In the manufacturing industry, product quality is an indicator of the production capacity of a company. Customers are increasingly demanding in terms of product quality, and providing them with a product that complies with the specifications is absolutely essential in an increasingly competitive market. The ideal solution would be to inspect in details all parts produced, but most companies cannot test every product. This may be due to the production rate being too high to allow inspection at a reasonable cost or within a reasonable time. Quality testing may also result in the destruction of the product or render it unfit for sale in some way. In our industrial context, quality control is a time-consuming operation that requires several minutes of work and that cannot be done online for each part. Only a subset of the produced parts can be inspected. One set of statistical tools for applying such a screening is acceptance sampling. Using such tools enables decision makers to determine what action to take on a batch of products. Decisions based on frequency testing, rather than on a thorough inspection, are more expedient and cost effective but they cannot guarantee the conformity of all parts of the population from which the sample was drawn (Fuchs and Kenett, 1998). Figure 2.1 describes acceptance sampling.

Although this method is widely applied and accepted in the manufacturing industry, it presents three major drawbacks:

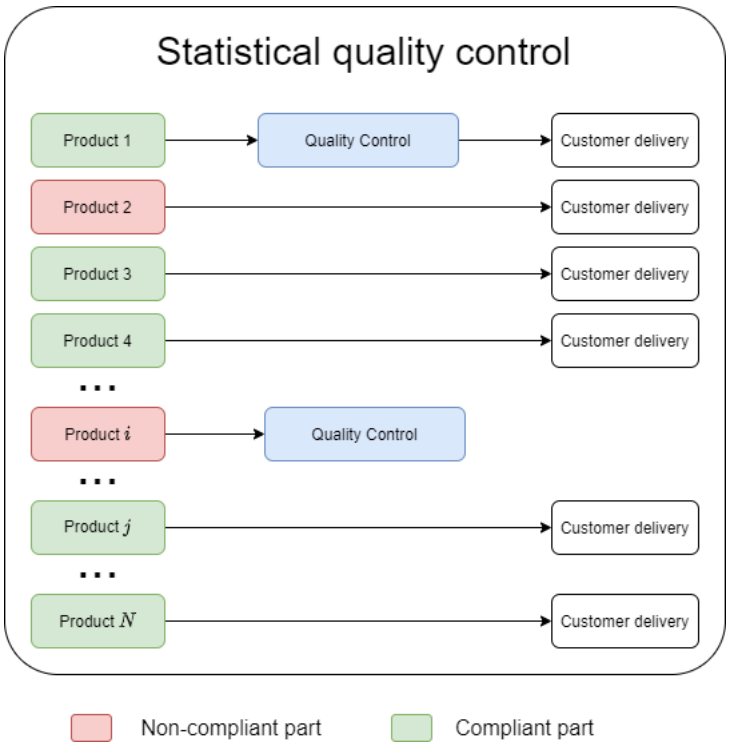

Figure 2.1: Statistical quality control

1. Acceptance sampling is able to track deviations in product quality, but it may happen that one or more non-compliant parts produced during a temporary malfunction are not detected. These non-compliant parts are then sent to the customer.

2. Acceptance sampling incurs delay in detecting process deviations. If the quality of the product starts to deviate, the manufacturer does not identify the problem until the next quality control, so the process is not corrected immediately. The parts produced in this time frame are potentially non-compliant and require extra time-consuming quality control to establish whether or not the parts can be sent to the customer.

3. Some quality controls require the destruction of the part or a modification that makes it unsuitable for sale. For instance, advanced quality control for assessing the material distribution of each thickness layer of a blow-moulded part requires the part to be cut into small samples, which are then analysed in the laboratory. Even if these tests are necessary to certify the part conformity, they increase the number of PPM (parts per million) waste.

In order to improve quality control, we seek to achieve a comprehensive quality control using a machine learning based approach, where the quality status of each part produced is inferred by a data-driven model. This ensures a control of all parts produced, which enables for a fast reaction to quality non-conformities (Figure 2.2 on the left side). In fact, by "virtually" measuring each part, we may eventually discard parts for which the model has provided a "Not-OK" result, or request the quality team to carry out more in-depth tests. Model-based quality measurement may be effectively used to detect those parts that turn out to be, from a statistical point of view, outliers. In this way, instead of randomly sampling the parts to be measured by the quality operators, the model may suggest parts that seem to be interesting. If the trained model is sufficiently accurate and robust, the real quality controls which destroy the parts or make them unusable could be reduced (Figure 2.2 on the right side). In such a case, not only the model-based control would be able to provide a thorough quality control, but it would also be able to reduce the scraps which account for an overall better production performance. Of course, real measurements cannot be completely replaced by model-based measurements: real measurements are the primary data sources to train and to validate the data-driven model.

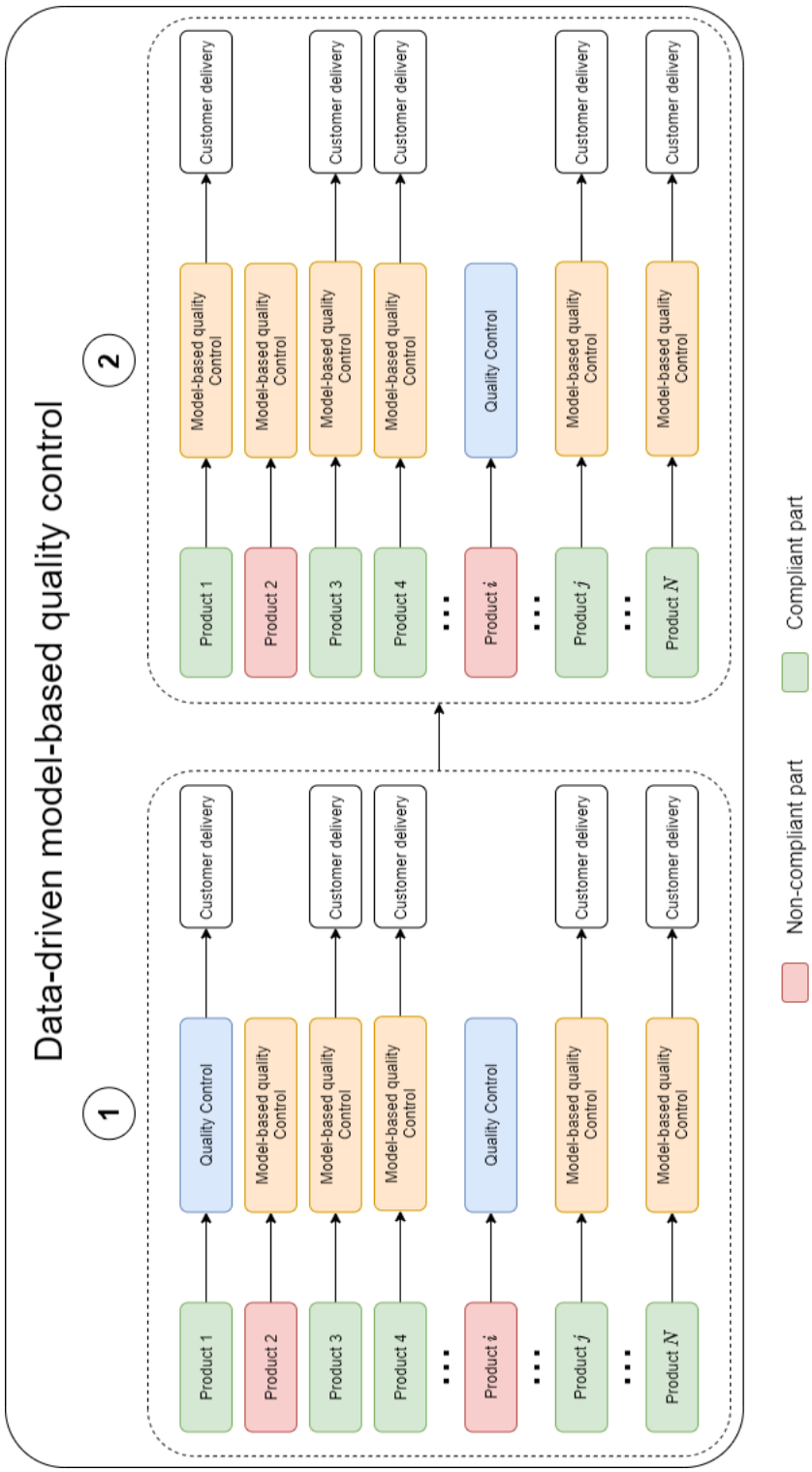

Figure 2.2: Data-driven model-based quality control

2.3 General method

In this section, we describe a general framework for dealing with predictive quality topics. We use supervised machine learning to discover patterns between process parameters and the quality of the parts that have been manufactured by that process. We proceed in four main stages:

1. *Data collection* consists in retrieving all the data needed to train the machine learning model It involves two main stages: data acquisition and data labelling (see Section 2.3.1).

2. *Data processing* covers the operations required to make the data suitable for the machine learning algorithm (see Section 2.3.2).

3. *Exploratory data analysis* is an ensemble of graphical and quantitative techniques that can be used to explore data and retrieve important information (see Section 2.3.3).

4. *Data modelling* corresponds to the statistical modelling of the relationship between the process data and the quality data by the machine learning algorithm (see Section 2.3.4)

The remaining part of the current section describes these four stages.

2.3.1 Data collection

In order to train a machine learning model, it is necessary to have samples that are representative of the entire operating range of the process. Iconic applications of machine learning, such as machine translation or object detection in images, rely on huge amounts of training data, but in new applications, such as in manufacturing, the training data is often limited.

Collecting data allows to capture a record of past events so that we can use data analysis to find recurring patterns. In the context of this research work, data collection is the task of retrieving the data that could be meaningful to explain the quality of parts given some process parameters. Among the many challenges in Industry 4.0, data collection is becoming one of the critical bottlenecks. It is known that the majority of the time for building end-to-end data-driven models is spent on preparing data, which includes collecting, cleaning, analysing, visualising, and feature engineering.

Two kind of data are required: input data, corresponding to the process data

and output data which is actually the measurement of the part quality. Data collection involves two different steps: *process data acquisition* and *quality data acquisition*.

2.3.1.1 Process data acquisition

We use here the term *process data* for any type of data belonging to the manufacturing process. For instance, for extrusion blow-moulding such data are extruder throughputs, extruder temperatures, or blowing air pressures. This process data pictures the process state at a given time.

Process data acquisition is a challenging task in Industry 4.0 due to different technologies, machines, sensors, IoT devices and communication networks. Sensors, actuators, and Programmable logic controllers (PLCs) are the main data generators in the automotive industry (Khan et al., 2017). In the last decade a new type of intelligent sensor, also called *smart sensors*, is more and more used in the manufacturing industry. Most of the data available in the manufacturing plants comes from PLCs, sensors and smart sensors. The three devices are further explained as below.

- *PLC* is a programmable unit that takes input from sensors, and controls actuators (Figure 2.3). A factory has a large number of PLCs which controls the machines. PLCs are manufactured by different suppliers and generate heterogeneous data, which is a big challenge for industrial big data.

- *Sensors* convert a physical state or activity into an electrical signal that is sent to the PLC for further processing. In manufacturing, sensors create a huge amount of data. Most machines and robots include sensors that collect data from their surroundings, such as temperatures of machine components or its environment.

- *Smart sensors* are devices that take information from a physical environment and use embedded microprocessors and wireless communication to monitor, examine and provide information about the proper functioning of the observed system. With the developments of IoT and machine learning, various types of smart sensors are nowadays available.

- *Actuators*, controlled by the PLC, produce a physical action. A basic example of an actuator connected to a PLC is the automatic starting of a motor.

Figure 2.3: Programmable logic controller *Siemens* S7-1500

All these devices produce a lot of data but they do not manage data storage. PLCs have a limited amount of storage space and cannot be used to store data permanently. The local machine data storage is most of the time handled by the SCADA (Supervisory Control And Data Acquisition) software. The term SCADA is used to identify any kind of software, installed on a personal computer or server, which allows the implementation, operation and management of supervisory, control and remote control systems without necessarily having to write code using programming languages. SCADA software have multiples functionalities which range from automation to alarm handling, logging, archiving and simple statistical analysis (Daneels and Salter, 1999). The data collected by the SCADA system is stored in a place where data is easily accessible, such as a cloud platform, whether internal to the manufacturing company or external. Cloud platforms, or *data lakes*, are designed to be highly scalable and provide a way to easily access data through Big data technology that facilitates and accelerates data analysis stages.

In order to be able to properly manage data acquisition, taking into account the heterogeneity of data coming from the different data sources, we propose to introduce a *gateway* system which constitutes an intermediary bridge between the shop floor PLCs, sensors, and the cloud platform where data is stored. Moreover, it can communicate with the *MES* system. The Manufacturing Execution System (MES) is a production management system serving as the information center in the enterprise to improve manufacturing transparency. It is the middle

layer connecting the manufacturing process on the shop floor and the business process on the Enterprise Resource Planning (ERP) (Chen and Voigt, 2020). By communicating with the MES system, it is possible to associate to a produced part the set of events that have enabled its production. The architecture of the overall data acquisition system is shown in Figure 2.4.

The gateway is a physical or virtualised server which acts as an intermediary between data acquisition systems available in the shop floor and the cloud platform where data is stored for data analysis. The gateway is connected to the shop floor network and it is able to interact directly with machine PLCs as well as smart sensors. The gateway has two main roles:

- It allows to centralise the data collection at plant level. It is in charge to retrieve data from all data sources, whether they are PLCs, smart sensors, SCADA software or the MES system. This process facilitates the subsequent sending of data to the cloud platform.
- This gateway is well suited to the eventual deployment on site of the machine learning models that have been trained.

The gateway should be equipped with different tools and software to allow the communication with machines through different communication protocols used in Industry 4.0. There exist a multitude of communication protocols. Among all these we can mention *OPC UA* and *MQTT*. OPC UA (Open Platform Communications Unified Architecture) is a service-oriented machine-to-machine communication protocol mainly used in industrial automation. Its main goals are to provide a cross-platform communication protocol while using an information model to describe data transfer. MQTT (Message Queuing Telemetry Transport) is an open message protocol which mainly focuses on a small code footprint and low network bandwidth usage, while handling high latency or bad network connections (Profanter et al., 2019). Further information regarding communication protocols used in Industry 4.0 are available in Profanter et al. (2019), Marcon et al. (2017) and Zezulka et al. (2018).

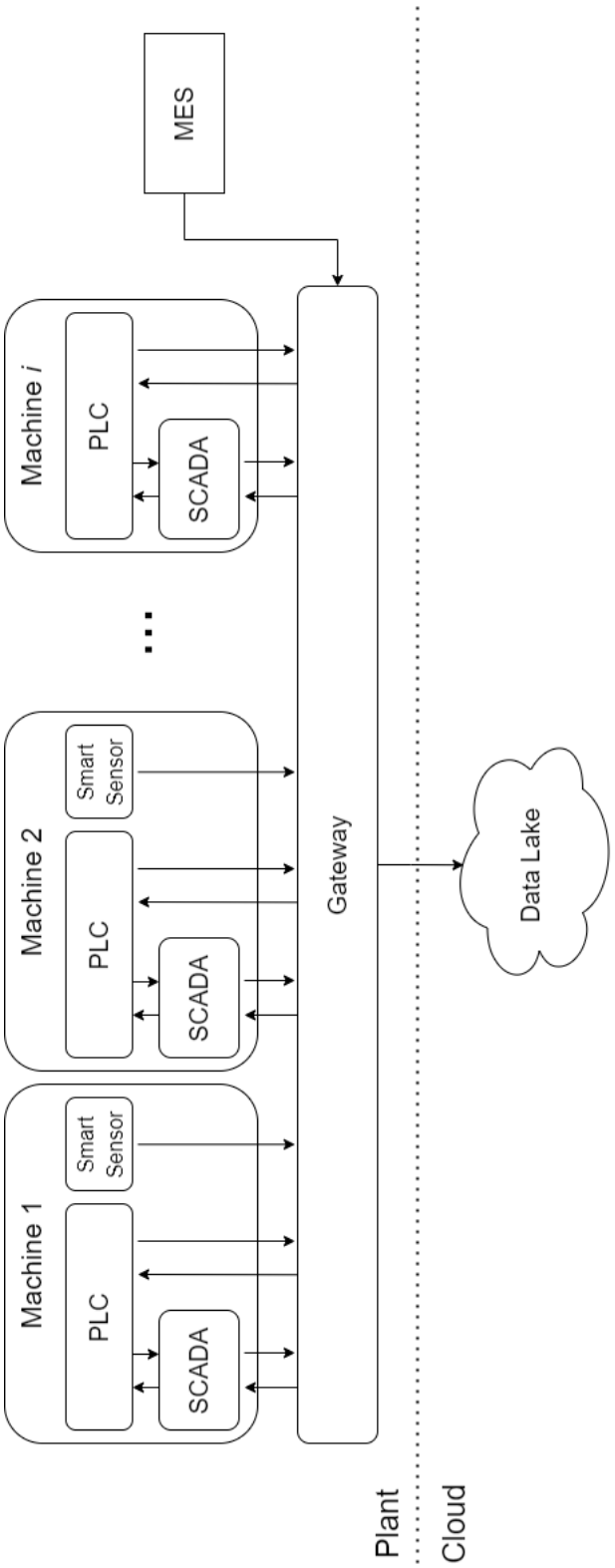

Figure 2.4: Data acquisition architecture

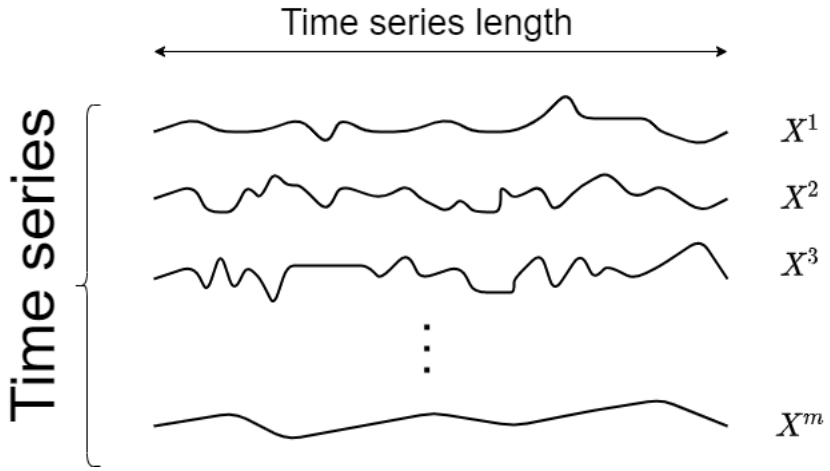

Figure 2.5: Time series

Process data types When dealing with process data, we distinguish two different types of data: *cyclical data* and *time series data*.

- *Cyclical data* are scalar values which provide information about a certain recurring event. Examples of cyclical data are the machine cycle time, or the time needed by the machine to perform an operation.
- *Time series* are sequential data that aggregate several sequential values of the same process parameters. For instance, the temperature of a machine component may be measured throughout the production cycle. The number of sequential values composing the time series depends on the sampling rate and may change according to the nature of the measurement.

2.3.1.2 Quality data acquisition

Quality data acquisition consist in collecting product quality data associated with all or some of the parts produced. The quality label can be continuous or discrete. For instance, the thickness of a manufactured part can be reported in the form of continuous values in meters, or can be directly associated with a class according to its compliance, or non-compliance, with the specifications. If the label is categorical, the modelling problem is a classification problem, if the label is continuous, it is a regression problem (see Section 2.4.1). Quality data acquisition can be done offline or online, as detailed in the following paragraphs.

Offline acquisition Offline acquisition, is the most common approach for the quality labelling of manufactured parts. In fact, for all non-visual product characteristics, it is extremely complicated to assess the quality of part in less then a minute. Most of quality controls on parts require specific equipment and the task of controlling can take several minutes. Moreover, effective control might result in the destruction of the product or render it unfit for sale. In such a case, the only option is to measure the part quality offline. Measuring offline has the advantage of allowing careful control of the manufactured parts, reducing the possibility of measurement error. However, as there are only a few parts measured, it may take a long time to build a dataset that is representative of all categories of part non-compliance. As quality is not measured on all parts produced, it is crucial to structure the data collection to facilitate access for future data analysis. In the database, the quality measurement must be related to a part number, or traceability number, in order to then associate it with the process data, which in turn must be tagged with the part number.

Online acquisition Performing online data acquisition, on the production line, eliminates two critical error risks:

1. The loss of the link between production measurements and off-line annotation.

2. The transformation of parts between their production and their annotation.

On the other hand, the time available for annotation is very limited. Most of the time, machine operators have time constraints to meet the production rate, so that their annotation must be done in a limited amount of time. This time may vary depending on the operator's charge. We estimate that the operator can allocate a maximum of one third of the production cycle time to this task, which means that for a process with a cycle time of 60 seconds the operator has at maximum 20 seconds to perform it. In addition to annotation time, the operator has to assist in the handling of the parts and related operations.

By automating the process data acquisition through the data collection presented above, and by ensuring the proper recording of the quality of parts, it is possible to permanently feed a dataset with new data automatically. Over time, it is hoped that enough data can be recovered to cover the entire distribution of all possible quality non-conformities.

2.3.2 Data processing

The heterogeneous data collected require processing to make them suitable to data analysis. In general, data processing comprises three major tasks: data cleaning, reduction, and scaling. In the remaining part of this section we will provide some additional elements regarding these three data processing tasks.

2.3.2.1 Data cleaning

Data cleaning is the result of two main operations: missing values imputations and outlier removals. For handling missing data, the first option is simply to reject data samples with missing values to allow the use of the numerous data mining algorithms that cannot handle missing data. This approach is only possible when the amount of missing values is small. The second technique is to impute the missing values, that is, to replace missing values with inferred ones. Mean imputation, forward or backward imputation, or moving average are examples of standard imputation procedures. In these techniques, missing values are inferred based solely on the data properties of that variable, and therefore are referred to as univariate techniques. The mean or median imputation method will replace missing values with the mean or median of that variable. The forward or backward method simply updates the missing value with the previous or next data measurement. More advanced techniques make use of multivariate predictors to obtain more accurate imputation results.

As regarding outlier removals, the most common techniques use statistical analysis to identify atypical data. Data outliers can be identified, for instance, if the data falls beyond a certain range constructed using conventional statistics such as standard deviations, means and quantiles. Identifying outliers is delicate because what at first sight might appear to be an outlier may turn out to be extremely interesting data. When dealing with manufacturing process data, the outlier can be representative of a process operating condition that is not normal and could therefore explain some product quality non-compliances. Physical knowledge of the production process is therefore indispensable in order to understand whether the outliers are due to a process malfunction or to a data acquisition error.

2.3.2.2 Data reduction

Data reduction aims to reduce data dimensions. Assuming that data are in a tabular format where the row represents the samples and columns the features,

or process parameters, data reduction may be conducted to reduce either the number of rows or the number of columns. There are three main methods of column-wise data variable reduction: The first is to use domain knowledge to directly select the variables of interest. The second is to use statistical feature selection methods to select important variables. The third is to adopt feature extraction methods to construct useful aggregated features. Human expertise plays a key role in the data acquisition process and, globally, in tasks of modelling by the relationship between process parameters and product characteristics. For complex processes, the number of available process parameters are huge, in the order of hundreds and sometimes thousands. The knowledge of human experts on the process may allow to pre-select a number of useful features that can be used to try predicting the target output.

As regarding feature selection techniques, we distinguish mainly three approaches: filter, wrapper and embedded methods. Filter methods are simple approaches where variables are ranked and selected based on specific univariate metrics. Pearson's correlation coefficient is a common filter technique for determining the direction and strength of a linear relationship between two variables.

Wrapper methods assess the usefulness of data variables for a given learning algorithm. Heuristic search methods, such as stepwise forward and backward selection methods, are commonly used. Compared to the filter approach, the wrapper methods take into consideration data variable correlations and interactions with learning algorithms. However, because it is generally performed via trial and error, the computing costs associated with wrappers is significantly higher.

Embedded approaches select features during the training of the model. A popular embedded method is L1 regularisation (based on the least absolute shrinkage and selection operator, Lasso).

Stability selection Stability selection (Meinshausen and Bühlmann, 2010) has been extensively used in this research work. This technique consuists in injecting noise into the original problem by generating bootstrap samples of the data (see Section 2.4.1.2), and to use a learning algorithm to find out which features are important in every bootstrap sample of the data. For a feature to be considered important, it has to be selected often among the perturbed versions of the original problem. This tends to filter out features that are only weakly related to the target variables, because the additional noise introduced by the

bootstrapping breaks that weak relationship. The algorithm takes as input a grid of values Λ for the regularisation parameter λ, and the number of bootstrap samples B to be generated. Stability selection returns a selection probability $\hat{\Pi}_k^\lambda$ for every value $\lambda \in \Lambda$ and for every feature k, and the set of stable features $\hat{S}^{stable} \subseteq \{1, \ldots, p\}$. The algorithm consists of two steps. In the sampling step the selection probabilities, or stability scores, are computed as follows. For each value $\lambda \in \Lambda$ do:

- For each i in $1, \ldots, B$, do:
  - Generate a boostrap sample of the original data (that is, a sample of size n formed by n independent draws with replacement of the original data).
  - Run the selection algorithm on the boostrap sample with regularisation parameter λ to get the selection set $\hat{S}_i^\lambda$.
- Given the selection operated on each boostrap sample, calculate the proportion of models selecting variable k:

$$\hat{\Pi}_k^\lambda = \frac{1}{B} \sum_{i=1}^{B} \mathbb{I}_{k \in \hat{S}_i^\lambda} \ .$$

In the scoring step, the set of stable features is defined as follows:

$$\hat{S}^{stable} = \{k : \max \hat{\Pi}_k^\lambda \geq \pi_{thr}\} \ ,$$

where π_{thr} is a predefined threshold. When the stability score for a variable exceeds the threshold π_{thr} for one value in Λ, it is deemed stable. In practice the Lasso penalisation (see section 2.4.1.1) is frequently applied to get the selection set $\hat{S}_i^\lambda$. This is due to the ability of Lasso to perform exact selection.

Unlike feature selection, which picks relevant features from existing variables, feature extraction seeks to create new aggregated features, built by linear or non-linear combinations of existing variables. Principal Component Analysis (PCA) is such a common linear feature extraction technique. The principal components are linear combinations of the original data variables that can be used as new features. PCA is particularly useful when there are collinearity problems. In practice, the number of principal components, or features extracted, is determined based on the proportion of total data variance explained; for instance, the principal components may explain at least 80 or 90% of the total data variance.

More advanced techniques, such as *auto-encoders* (Rumelhart et al., 1985) extract nonlinear features. When dealing with time-series data we can compress the entire information in a limited set of new features computed through the use of summarising statistics, such as the mean, peak, and standard deviation over a particular time span.

2.3.2.3 Data scaling

Data scaling aims to transform the original data into homogeneous ranges. It is necessary if the results are not to depend on the units of measurement chosen. The most used scaling techniques are *max-min normalisation* and *z-score standardisation*. Min-max normalisation is defined as follows:

$$x = \frac{x - x_{min}}{x_{max} - x_{min}} \ ,$$

where x_{min} and x_{max} refer to the minimum and maximum values of the generic feature x. The z-score standardisation is instead defined by the following equation:

$$x = \frac{x - \mu}{\sigma} \ ,$$

where μ is the mean and σ is the standard deviation of the feature x.

Z-score standardisation is well suited when data are normally distributed. The max-min normalisation, instead, is recommended when the data do not conform to a normal distribution and have no outliers.

2.3.3 Exploratory data analysis

Exploratory Data Analysis (EDA) is a set of data analysis techniques that may be applied to:

- uncover underlying structures,
- isolate important variables,
- detect outliers and other anomalies,
- suggest suitable models for conventional statistics.

EDA is usually the intermediate stage between data processing and machine learning modelling. By exploring the data, it is possible to discover interesting

patterns among data and drive the modelling phase depending on what have been observed. Moreover, EDA allows to fine-tune the data processing stage. In fact, by exploring the data we can identify useless features that cannot bring any added value and can therefore be discarded. The term "Exploratory Data Analysis" was introduced by Tukey et al. (1977) shows how simple graphical and quantitative techniques can be used to explore data.

Typical graphical techniques are:

- plotting the raw data (i.e. histograms, scatter plots),
- plotting simple statistics (i.e. box plots, residual plots),

Typical quantitative techniques are:

- interval estimation,
- measures of location or of scale,
- shapes of distributions.

A very convenient exploratory data analysis tool is principal component analysis (see Section 2.4.2). By projecting input data onto the first principal components, which capture the most variability in the data, it is possible to visualise some interesting global patterns within data, even if the number of features is large.

2.3.4 Supervised learning modelling

The objective of a machine learning algorithm is to *generalise from examples*. In supervised learning, this goal is formalised as finding a function $g(\cdot)$ that maps the features to the response variable. Generalising from examples means then to *accurately predict the value of the response from the function* $g(\cdot)$ *on future examples, based on a finite set of examples*, the training set. To formalise what future examples look like, it is assumed that they will be drawn from a distribution, and it is assumed that the training set has also been drawn from this distribution, so that it provides relevant clues for the learning objective. We therefore assume the existence of random variables for the features and the response; these random variables will be respectively denoted by X and Y. The (in)accuracy of predictions will be measured by a loss function $L(Y, g(X))$, also called cost function, which represents the loss incurred by predicting $g(X)$ instead

of Y. The generalisation goal is then to minimise the so-called *risk*:

$$R(g) = \mathbb{E}_{XY}[L(Y, g(X))] \ . \tag{2.1}$$

In practice, this objective cannot be achieved because the distribution of (X, Y) is not known, so the risk $R(g)$ cannot be computed. With the training set, it is however possible to compute a Monte-Carlo approximation of the risk, called the *empirical risk*, which is the average loss function on the training set:

$$R_{\text{emp}}(g) = \frac{1}{n} \sum_{i=1}^{n} L(y_i, g(x_i)) \ ,$$

where n is the number of training examples (x_i, y_i). Learning then consists in minimising $R_{\text{emp}}(g)$. This minimisation is performed on a fixed class of functions $\mathcal{G}$, for reasons that will not be explained here, but the theory states that this restriction is necessary (Vapnik, 2000). To convince oneself, it is sufficient to consider that all the functions that interpolate the training set are not necessarily interesting in terms of generalisation, since their behaviour between examples can be very different. Moreover, if the responses y_i are noisy, the function $g(\cdot)$ that minimises the risk (2.1) does not necessarily interpolate the training examples.

The training phase, also called the learning phase consists in the minimisation over $\mathcal{G}$ of the average loss computed on the training set:

$$\hat{g} = \underset{g \in \mathcal{G}}{\operatorname{argmin}} \, R_{\text{emp}}(g) \tag{2.2}$$

and the function $\hat{g}$ is thus the output of the learning algorithm. In practical applications, the available data is usually split in two parts: a training set and a test set. The training is used to estimate the function $\hat{g}$ by solving (2.2), and the test set is used to evaluate the risk $R(\hat{g})$ (2.1) unbiasedly. Indeed, for any function chosen independently from the training set, $R_{\text{emp}}(g)$ is an unbiased estimate of $R(g)$, but this is not true for $\hat{g}$, which minimises $R_{\text{emp}}(g)$: it is easy to show that $R_{\text{emp}}(\hat{g})$ is an optimist estimate of $R(\hat{g})$.

In the context of our research work, supervised learning modelling consists of using of machine learning algorithms to estimate the transfer function between input process data $X = (X_1, X_2, \ldots, X_p)$ and output quality data Y from experimental data. In the following section, we will provide more insights about

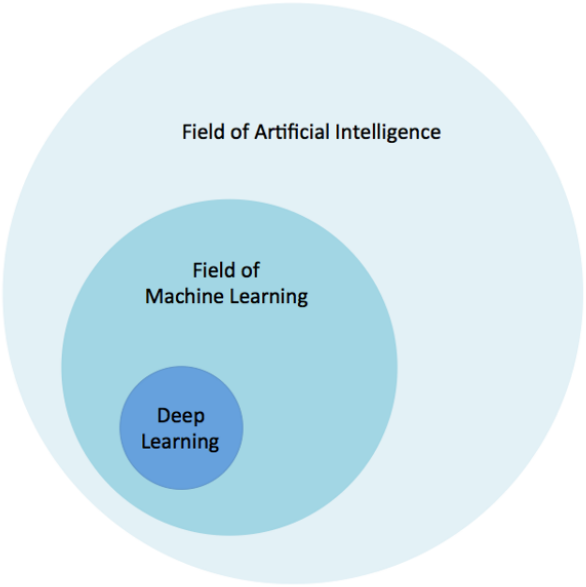

Figure 2.6: AI-Machine learning-Deep learning

the machine learning algorithms that we have applied throughout our research work.

2.4 Machine learning and deep learning

Machine learning (ML) is a field of computer science that aims to give computers the ability to learn and decide without being explicitly programmed. Instead of explicitly encoding knowledge by machine instructions, machine learning leverages data analysis, which involves building and fitting models, to allow machines to "learn" from experience. Over the years, a myriad of different kinds of machine learning algorithms and models have been devised for serving different situations and types of problems.

In everyday life, machine learning is often mistakenly confused as Artificial Intelligence (AI). Figure 2.6 locates machine learning, deep learning and artificial intelligence in relation to each other. Artificial intelligence is a technology which enables a machine to simulate human behaviour. AI may use, among other ones, one or several machine learning algorithms to learn from past data to solve complex tasks. Deep learning (DL) is a sub-field of ML focusing on deep neural network architectures. Deep-Learning models provide the top performances in numerous applications and challenges. Great improvements have been reached

in multiple domains: from web search to image recognition and classification through convolutional neural networks to natural language preprocessing with Recurrent neural networks and *self-attention* networks (Vaswani et al., 2017). The democratisation of the different models through open-source software libraries, specialised chip-set and highly scalable computing platforms has pushed companies to integrate these tools within their own production facilities.

In what follows, we review some concepts of machine learning in order to provide the reader with the elements to understand the following chapters. First, the key concepts, such as *supervised* and *unsupervised* learning are presented. Then, we describe the algorithms that have been applied throughout the study. This review is by no means exhaustive but aims to provide the elements necessary for understanding the approaches presented in Chapters 3 and 4. Exhaustive reviews of machine learning can be found in Bishop (2006); Friedman (2017). An exhaustive overview of DL architectures is beyond the scope of this research work: research in deep learning is advancing very fast and new architectures are being proposed every day. The end of this section will however describe three types of neural networks: feed-forward neural network, convolutional neural network and recurrent neural network, as well as some specific architectures that we will extensively use in Chapter 4. For further reading on this topic, Goodfellow et al. (2016) provides a comprehensive review of the main principles of deep learning.

2.4.1 Supervised learning

Supervised learning, the most widely used machine learning approach, aims at learning a function from examples of inputs and outputs pairs. The main objective is to accurately predict the responses for future observations (prediction), and a side goal may be to better understand the relationship between the response and the inputs. In general, supervised learning problems are formalised as optimisation problems, by looking for a function that minimises an error criterion that quantifies the discrepancy between predictions and the real value (or "ground-truth") associated. The cost function should be defined to reflect the cost associated with errors and should be optimisable. We will present below the most common costs in regression and classification.

Regression Regression corresponds to a training objective where training data and their corresponding outcome, a set of numerical continuous variables, are known and available for training. For instance, in a manufacturing context, a

regression model can be designed to predict the numerical value of some dimensional characteristic of a manufactured part, given a set of input process parameters. The loss functions used to train regression models are most often based on average distances. The most common and convenient loss function used for regression problems is squared error loss $L(Y, g(X)) = (Y - g(X))^2$, leading to:

$$R_{\text{emp}}(g) = \frac{1}{n} \sum_{i=1}^{n} (y_i - g(x_i))^2 \ .$$

where we recall that (x_i, y_i) are the input-output pairs from the training sample of size n.

Classification When the objective is to predict a categorical variable, or class, supervised learning takes the name of classification. In a manufacturing context, a classifier can be trained to recognise whether a part is compliant (OK), or not (NOK), to some quality specification. In classification problems we aim to reduce the misclassification error. In practice, this leads to a combinatorial problem that is hard to solve. For this reason, we generally estimate the probability that a certain sample belongs to a specific class c. In this context, the function g maps the features to the c class probabilities, and $g_j(x_i)$ refers to the estimate of the posterior probability $P(Y = j|X = x_j)$. The standard loss function in this case is the cross-entropy, which is also the neg-log-likelihood:

$$L(y_i, g(x_i)) = -\sum_{j=1}^{c} y_{ij} \log(g_j(x_i)) \ ,$$

where c is the number of classes, y_i is the indicator vector of class labels for features x_i, that is, $y_{ij} = 1$ if x_i belongs to class j and $y_{ij} = 0$ otherwise (all examples belong to a single class).

In *binary classification*, where there are only two classes, y_i can be encoded as a simple binary variable, and the loss function becomes:

$$L(y_i, g(x_i)) = -y_i \log(g(x_i)) + (1 - y_i) \log(1 - g(x_i)) \ .$$

Model evaluation For the evaluation and comparison of model performances, different performance metrics can be applied. For binary classification, several metrics can be calculated based on the entries of the confusion matrix shown in Table 2.1. The comparison of the predicted class with the true class allows

Table 2.1: Entries of a confusion matrix

	Actually Positive	Actually Negative
Predicted Positive	**True Positives (TPs)**	**False Positives (FPs)**
Predicted Negative	**False Negatives (FNs)**	**True Negatives (TNs)**

to distinguish between positive or negative examples correctly classified (true positive, true negative) and incorrectly classified examples (false positive, false negative). This is suitable to discriminate between conforming parts (OK) and non-conforming parts (NOK).

For regression, the most common metrics to compare model performance are mean squared error (MSE), the root mean squared error (RMSE) and the R^2. MSE corresponds to the average of the squared error loss. The root mean squared error (RMSE) is the square root of the MSE: RMSE = $\sqrt{\text{MSE}}$. RMSE has the advantage of being in the unit of measurement of the target variable. This provides a more direct interpretability of the final result. The coefficient of determination R^2 is the proportion of the variance in the dependent variable that is predictable from independent variables:

$$\text{R}^2 = 1 - \frac{\text{RSS}}{\text{TSS}} = 1 - \frac{\sum_{i=1}^{n}(y_i - g(x_i))^2}{\sum_{i=1}^{n}(y_i - \bar{y}_i)^2} \quad ,$$

where RSS is the residual sum of squares, TSS is the total sum of squares and $\bar{y}_i$ is the average value of y_i, $\bar{y}_i = \frac{1}{n}\sum_{i=1}^{n} y_i$. Values of the coefficient of determination range, normally, from zero (poor model) to one (perfect model), but can be negative when the model does not follow the trend of the data, leading to a worse fit than the average value of the target variable. As the a-priori selection of adequate algorithms is generally not feasible (Kotthoff, 2016), different models need to be evaluated and compared for each industrial application (Lee and Shin, 2020). A pre-selection can be made on the basis of criteria such as complexity, interpretability, and speed. From a practical point of view, the scoring time of the model should be short enough to adjust the production process in real-time. The required response time depends on the manufacturing process. The scoring time is affected not just by the algorithm employed, but also by the hardware and software on which it is implemented. However, in most situations, the scoring time does not constrain the model selection process.

2.4.1.1 Linear models

Linear models are of the form:

$$g(X) = \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p \ , \tag{2.3}$$

where β_j is the generic j-th coefficient, associated with the j-th feature.[1] We detail below three procedures to estimate the model coefficients. Note that, for ease of presentation, we present the case of scalar Y responses ($Y \in \mathbb{R}$), but the same principles apply to response vectors $Y \in \mathbb{R}^m$.

In ordinary linear regression, the model coefficients are estimated by the hyper-plane that minimises the residual sum of squares. Let $\mathbf{X} \in \mathbb{R}^{n \times p}$ and $\mathbf{y} \in \mathbb{R}^n$ denote the concatenation of training examples: $\mathbf{X} = (x_1 \ldots x_n)^T$ and $\mathbf{y} = (y_i \ldots y_n)^T$, where ^{T} is the transposition operator, the empirical risk is defined as follows:

$$R^{\text{ols}}_{\text{emp}}(g) = \sum_{i=1}^{n} (y_i - g(x_i))^2 = (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^T (\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) \ ,$$

where $\boldsymbol{\beta} = (\beta_1 \ldots \beta_p)^T$. Under the assumption that $\mathbf{X}$ has full column rank, we can differentiate the equation with respect of $\boldsymbol{\beta}$ to obtain the unique solution:

$$\hat{\boldsymbol{\beta}}^{\text{ols}} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y} \ .$$

One way to reduce the model variance is to apply a technique that constraints or regularises the coefficient estimates towards zero. The two best known methods are ridge regression (Hoerl and Kennard, 1970) and Lasso regression (Tibshirani, 1996).

In ridge regression a penalty term is added to the loss function, this penalty term is also called L2 regularisation. The penalised residual sum of squares can be written as follows:

$$\begin{aligned} R^{\text{ridge}}_{\text{emp}}(g)) &= \sum_{i=1}^{n} (y_i - g(x_i))^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \\ &= \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda \|\boldsymbol{\beta}\|_2^2 \ , \end{aligned}$$

where $\lambda \geq 0$ is a complexity hyper-parameter that controls the amount of shrinkage towards zero and $||\boldsymbol{\beta}||_2$ is the L2-norm (Euclidean norm). These parameters

[1] This expression allows for an intercept if a constant feature X_0 is added to X.

have to be determined separately, for example using cross-validation (Friedman, 2017). The ridge regression coefficient estimation is given by:

$$\hat{\boldsymbol{\beta}}^{\text{ridge}} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{y} \ ,$$

where $\mathbf{I}$ is the identity matrix in $\mathbb{R}^p$.

Lasso regression applies a similar shrinkage approach. In Lasso regression a penalty term (L1 regularisation), corresponding to an absolute value of magnitude, is applied to the residual sum of squares:

$$\begin{aligned} R_{\text{emp}}^{\text{lasso}}(g) &= \sum_{i=1}^{n}(y_i - g(x_i))^2 + \lambda\sum_{j=1}^{p}|\beta_j| \\ &= \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda||\boldsymbol{\beta}||_1 \ , \end{aligned}$$

where $\lambda \geq 0$ is a complexity hyper-parameter that can be estimated using cross-validation and $||\boldsymbol{\beta}||_1$ is the L1-norm (Manhattan norm). As with ridge regression, the Lasso shrinks the coefficient estimates towards zero. However, the Lasso penalty has the effect of forcing some of the coefficient estimates to be exactly equal to zero when λ is sufficiently large. Lasso yields sparse models that are generally much easier to interpret than those produced by ridge regression. With Lasso, the features that are not related to the dependent variable are decreased towards zero so that this method is quite useful to do feature selection. Unlike ridge penalty, however there is no closed form expression to solve the minimisation of the residual sum of squares. There are multiple algorithms for computing the entire path of solutions but their presentation is outside the scope of this paper.

Ordinary linear, Lasso and ridge regression are models for regression problems. When dealing with classification, though regression is an option, other approaches are preferable. One of the most standard *generalized linear model* in classification problems is *logistic regression* (Friedman, 2017). Logistic regression is a generalised linear model using the *logistic* function: $1/(1+\exp(-x))$. Linear regression models the relationship between the features and the response va-riable by a linear link (2.3). For binary classification, the log-ratio $log(P(Y = 1|X)/(1 - P(Y|X))$ is modeled by a linear relationship, leading to logistic function:

$$g(X) = P(Y = 1|X) = \frac{1}{1 + \exp -(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p)} \ .$$

As expected for a probability, the output takes values between 0 and 1.

2.4.1.2 Tree-based methods

Trees are simple and useful models for interpretation. These models use decision trees to determine which target value matches the observation. Trees split the feature space into multiple regions R_j and than fit a simple model in each one. For regression problems, for each observation that falls into the region R_j, the prediction is simply the average of response values for the training observations in R_j. For classification problems, for each observation that falls into the region R_j, the prediction corresponds to the class that has the major number of occurrences into the same region.

Unfortunately, it is computationally infeasible to consider every possible partition of the feature space into j regions. In order to overcome this issue, we use a greedy top-down approach. The most widely used method is the CART algorithm (Breiman et al., 2017). A CART Tree is a binary tree that is constructed by splitting a node into two child nodes repeatedly, beginning with the root node that contains the whole learning samples. The main idea is to grow the tree by choosing a split, among all possible splits, that minimise a defined splitting criterion. Usually, for regression problems, the splitting criterion is the mean squared error. For classification problems instead, cross entropy loss is often applied.

Even though these model are quite good for interpretability, they are not competitive with other machine learning techniques in terms of prediction. One possible way to improve the prediction capabilities is to use methods like bagging (Breiman, 1996), random forest (Breiman, 2001) and gradient boosting (Friedman, 2001). With *bagging* (bootstrap aggregating), several subsets of data are created from the training set and each of this subset is used to build a decision tree. By averaging the predictions of all the different decision trees we end up with more robust results and with the reduction of the variance of the estimated model. Given B different bootstrapped training sets, the final prediction can be written as follows:

$$g_{\text{bagging}}(x) = \frac{1}{B}\sum_{b=1}^{B} g_b(x) \ ,$$

where $g_b(x)$ is the prediction on the b-th bootstrapped training set for a point x. Random forests can be seen as an extension of bagging, with additional randomisation, for example by selecting a random subset of features. Once again, by averaging the results of the trees in the "forest" generated by this method, one

usually obtains a more accurate result compared to a single regression tree.

Gradient boosting is named after two different techniques: gradient Descent and boosting. In gradient boosting, the learning procedure consecutively fits new models to provide a more accurate estimate of the response variable. The principle idea behind this algorithm is to construct the new base-learner to be maximally correlated with the negative gradient of the loss function, associated with the whole ensemble. The loss functions applied can be arbitrary, but to give a better intuition, if the error function is the classic squared-error loss, the learning procedure would result in consecutive error-fitting (Natekin and Knoll, 2013).

2.4.1.3 Support Vector Machines

Boser et al. (1992) proposed a supervised algorithm for classification that has since evolved into what are now known as *Support Vector Machines* (SVMs): a class of algorithms for classification, regression and other applications. The SVMs were originally conceived for binary classification problems. In a given feature space, SVM learning aims to construct a hyper-plane to best separate training data with different class labels. The hyper-plane is derived on the basis of a limited number of training instances, so-called support vectors, to maximise a margin on each side of the plane. When the samples are not linearly separable, it is possible to perform a transformation, through a *kernel* in the original data space, corresponding to embedding the samples in a space where they are linearly separable. The most commonly used non-linear kernels are the polynomial and the *Radial Basis Function* (RBF) kernels.

Support vector regression (SVR) (Drucker et al., 1997), an extension of the SVM algorithm, was introduced for regression. In SVR, instead of generating a hyper plane for class label prediction, a linear predictor is built. As for SVMs, SVR can be kernelized to allow non-linear prediction.

2.4.1.4 Neural networks

A neural network is a model made up of a large number of simple, highly interconnected processing units (neurons). Feed-forward neural networks learn to map a fixed-size input to a fixed-size output. To go from one layer to the next, the units compute a weighted sum of their inputs from the previous layer and pass the result through a non-linear function (activation function). For a generic hidden layer H of a neural network the j-th unit computes the following

operation:

$$h_j^H = \sigma\left(\sum_{i \in H-1} W_{ij} x_i\right) \ ,$$

where W_{ij} is the incoming weight from the i-th unit of the previous layer to unit j, and σ is the activation function. Among all activation functions, the most popular nowadays is the $ReLU$ (Rectified Linear Unit) (Glorot et al., 2011), defined as follows:

$$ReLU(x) = \max(0, x) \ .$$

Without non-linear activation functions, the neural network would be a composition of linear functions and would not be able to model non-linear relationships between inputs and outputs. Units that are not in the input or output layers are conventionally called hidden units. By stacking multiple hidden layers it is possible to approximate complex non-linear functions. The back-propagation algorithm uses the derivative chain rule to calculate the gradient of the objective function with respect to each weight, so as to update the model. The gradient of the objective function with respect to the weights is computed by working backwards, layer by layer, from the gradient calculated with the respect to the outputs. For each weight, the corresponding gradient component indicates how the objective varies when that weight varies infinitesimally. Once the gradient is propagated throughout the network, it is used to upgrade all weights. In practice, the full gradient is rarely computed, and the most common optimisation algorithm is Stochastic Gradient Descent (SGD) and variants thereof. With SGD, the objective is not fully computed, it is estimated by a partial objective computed on a mini-batch of examples. In recent years, many variants of SGD have been proposed. Throughout this research work, we have mainly used Adam (Kingma and Ba, 2015), which has been empirically shown to be efficient in many situations.

2.4.1.5 Convolutional neural network

Convolutional Neural Networks (CNN) are neural networks that use convolutions in place of general matrix multiplications in at least one of their layers. The architecture of a typical ConvNet (Figure 2.7) is structured as an alternance of two types of layers: convolutional layers and pooling layers. The units of a convolutional layer are organised into feature maps, within which each unit is

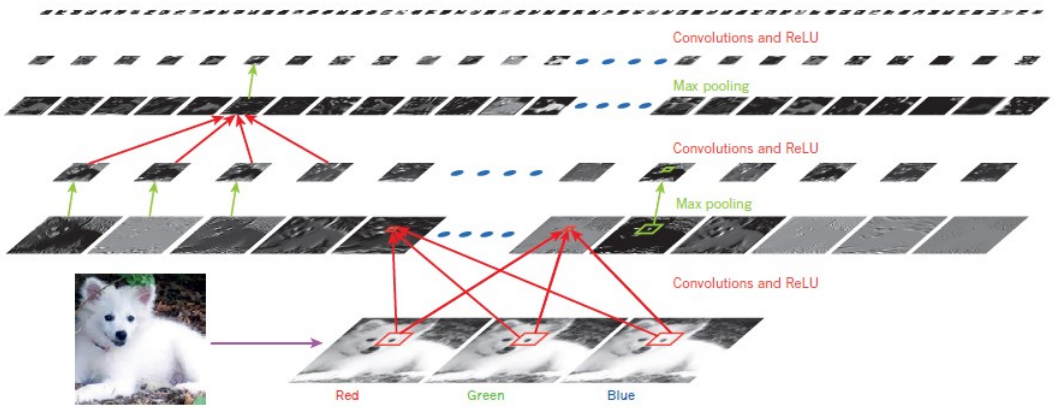

Figure 2.7: Convolutional network overview (LeCun et al., 2015)

connected to local patches of the previous layer's feature maps by a set of weights called a filter bank. The result of this local weighted sum is then passed through a non-linearity such as a ReLU. All units in a feature map share the same filter bank, and different feature maps in a layer use different filter banks.

CNN have three great properties which are well suited for processing data that has a known grid-like topology: "sparse interactions", "parameter sharing" and "equivariant representations" (Goodfellow et al., 2016). Compared to traditional fully connected layers where every unit of one layer interacts with every unit of the preceding layer, CNNs have sparse interactions: the size of the convolutional kernel (that defines the filter) being much lower than the size of the preceding layer, fewer parameters need to be stored and learned, which both reduces the memory requirements of the model and improves its statistical efficiency. Convolutions are also the basis for the equivariances that make CNN particularly suitable for processing images.

Among the many possible applications involving CNNs we recall *image classification*, *object detection*, *instance segmentation* and *semantic segmentation*. Image classification is a fundamental task that attempts to comprehend an entire image as a whole. The aim is to classify the image by providing it a label. Image classification often refers to images in which just one item appears and is analysed. Object detection, on the other hand, involves both classification and localisation tasks and is used to analyse more realistic scenarios in which numerous items may exist in an image. Advanced computer vision tasks, instance segmentation, are intended to achieve finer-grained object localisation in input images. The bounding boxes used in object detection find only coarse-grained

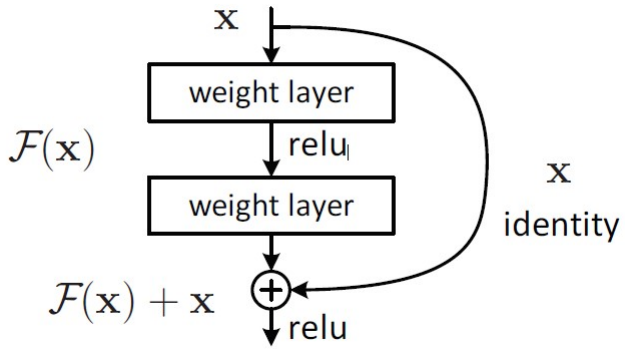

Figure 2.8: Shortcut in ResNet (from He et al., 2016)

object boundaries and include many pixels that do not belong to the object. In contrast, instance segmentation improves the object localisation accuracy by identifying each pixel that acts as part of a known object in the image. The semantic segmentation task involves associating each pixel in an image with a class label. In the following subsections we will review 3 different convolutional based architecture we have used in the course of our research work: *Residual networks* (for image classification), *Single Shot MultiBox Detector* (for object detection) and *U-Net* (for image segmentation).

Residual networks Most of the state-of-the-art Image classification methods use Residual networks, better known as *ResNet* (He et al., 2016). The ResNet architecture solves the vanishing gradient problem for very deep neural network architectures by applying the concept of residual learning. *Shortcut connections* (Figure 2.8) favour the propagation of gradients and allow for efficient training of very deep neural networks.

There is empirical evidence that Residual networks are easier to optimise, and can gain accuracy from considerably increased depth. By stacking multiple convolutional layers and by leveraging the concept of residual learning, Residual networks may be very depth with more than 100 convolutional layers. Depending on the number of convolutional layers, there exists multiples versions of these models. The most popular architectures are *ResNet18*, *ResNet34*, *ResNet50*, *ResNet101*, *ResNet152*. As shown in Figure 2.9, the generic ResNet$X$ is composed of 5 convolutional building blocks and a last fully connected layer which leverage the extracted features to produce the classification result. Depending on the depth of the architecture each convolutional building block is composed of a different number of convolutional layers.

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix} \times 3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix} \times 8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix} \times 23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix} \times 36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix} \times 3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Figure 2.9: ResNet architectures (from He et al., 2016)

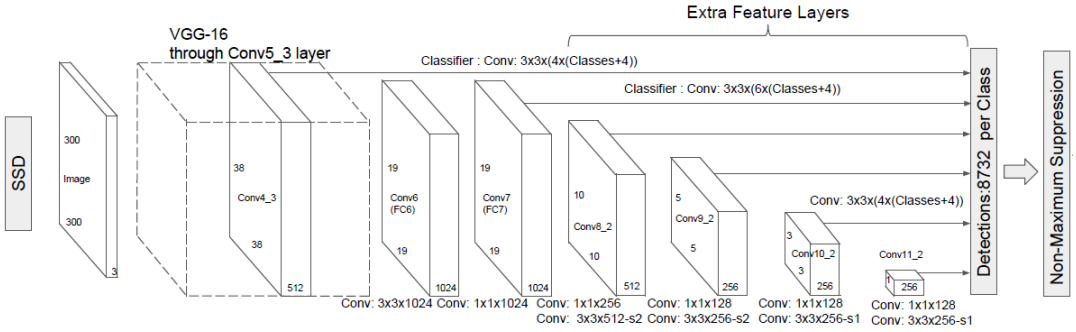

Figure 2.10: SSD architecture (from Liu et al., 2016)

Single Shot MultiBox Detector Single Shot MultiBox Detector (SSD) (Figure 2.10) is a single-stage object detection method that discretises the output space of bounding boxes into a set of default boxes over different aspect ratios and scales per feature map location (Liu et al., 2016). At prediction time, the network generates scores for the presence of each object category in each default box and produces adjustments to the box to better match the object shape. Additionally, the network combines predictions from multiple feature maps with different resolutions to handle objects of various sizes.

This model mainly consists of a base network followed by several multi-scale feature map blocks. The base network is for extracting features from the input image, so it can use a deep CNN. For example, the original single-shot multibox detection paper adopts a *VGG network* (Simonyan and Zisserman, 2014)

truncated before the classification layer. In recent years, other base network architectures have been combined with the multi-scale feature maps blocks. *MobileNet* has been developed to speed up the computation and lends itself well to being combined with the SSD model to perform Object Detection tasks in real-time.

MobileNet (Howard et al., 2017; Sandler et al., 2018; Howard et al., 2019) is a family of general purpose computer vision neural networks designed with mobile devices in mind to support classification, detection and more. The popularity of these architecture is motivated by the overall trade-off between the inference speed and the model performances. The main idea behind MobileNet models is based on the concept of *depth-wise separable convolutions*. Depth-wise separable convolutions is a form of factorised convolutions which factorise a standard convolution into a depth-wise convolution and a 1×1 convolution called a point-wise convolution. For MobileNets the depth-wise convolution applies a single filter to each input channel. The point-wise convolution then applies a 1×1 convolution to combine the outputs of the depth-wise convolution. The depth-wise separable convolution splits this into two layers, a layer for filtering and a separate layer for combining. This factorisation has the effect of drastically reducing computation and model size (Howard et al., 2017).

SSD-based architectures are usually trained by minimising the multi-box loss, which is a combination of a classification loss and a localisation loss. The classification loss measures the loss of making a class prediction. For every positive match prediction, we penalise the loss according to the confidence score of the corresponding class. The classification loss is mathematically defined as follows:

$$L_{\text{cls}} = -\sum_{i \in \text{pos}} \mathbb{1}_{ij}^{c} \log(\hat{y}_i^c) - \sum_{i \in \text{neg}} \log(\hat{y}_i^0) \ ,$$

where $\mathbb{1}_{ij}^{c} = \{1, 0\}$ is an indicator for matching the i-th default box to the j-th ground truth box of category c and y_i^c are the predicted scores for the same class. "pos" is the set of matched bounding boxes and "neg" is the set of bounding boxes without objects.

The localisation loss represents the mismatch between the ground truth box

and the predicted boundary box. The localisation loss is defined as follows:

$$L_{\text{loc}} = \sum_{i,j} \sum_{m \in \{\zeta,\xi,w,h\}} \mathbb{1}_{ij}^{\text{match}} L_1^{\text{smooth}}(d_m^i - t_m^j)^2$$

$$L_1^{\text{smooth}}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise} \end{cases}$$

$$t_\zeta^j = (g_\zeta^j - p_\zeta^i)/p_w^i$$
$$t_\xi^j = (g_\xi^j - p_\xi^i)/p_h^i$$
$$t_w^j = \log(g_w^j/p_w^i)$$
$$t_h^j = \log(g_h^j/p_h^i)$$

where $\mathbb{1}_{ij}^{\text{match}}$ indicates whether the i-th bounding box with coordinates $(p_\zeta^i, p_\xi^i, p_w^i, p_h^i)$ matched to the j-th ground truth box with coordinates $(g_\zeta^j, g_\xi^j, g_w^j, g_h^j)$ for any object and d_m^i are the predicted correction terms. The multi-box loss is then defined as linear combination of the classification and the location losses. It is defined as follows:

$$L = \frac{1}{N}(L_{cls} + \alpha L_{loc}) \ ,$$

where N is the number of matched bounding boxes and α balances the weights between two losses.

U-Net One popular approach for semantic segmentation models is to follow an encoder/decoder structure that first "downsamples" the spatial resolution of the input, for developing lower-resolution feature maps that are efficient at discriminating between classes. This representation is then "upsampled" to the resolution of the input to provide a full-resolution segmentation map. The encoder-decoder approach, as part of the semantic segmentation domain, was proposed for the first time by Long et al. (2015) with a fully convolutional network (FCN) architecture and it has been subsequently taken up by other research works (Ronneberger et al., 2015; Zhao et al., 2017; Chen et al., 2017, 2018; Badrinarayanan et al., 2017). The encoder, or contracting path, is, most of the time, a convolutional neural network whose task is to extract features of decreasing spatial resolutions while increasing the number of channels. The decoder, or expansive path, has the role of restoring the original spatial resolution by sequentially increasing the spatial dimension while reducing the number of channels. The decoder can be composed of one or several decoder blocks, in the same way the encoder can be more or less deep. Each decoder block first up-samples the feature maps using an interpolation method, then it applies a convolutional operation that halves the number of feature channels. Finally, a last convolutional block, sometimes

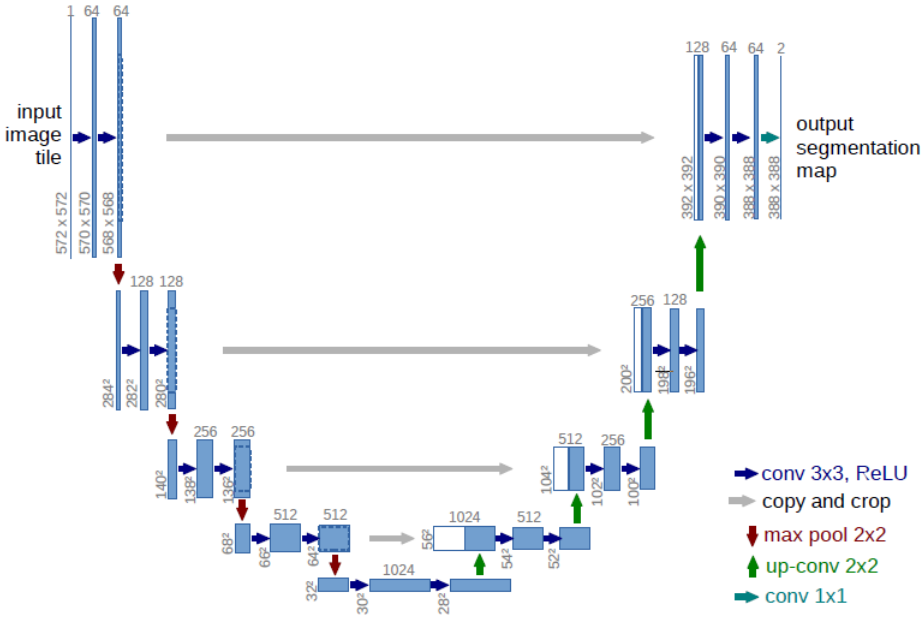

Figure 2.11: U-net architecture (Ronneberger et al., 2015)

called segmentation head, stacked right after the last decoder block, produces the segmentation mask.

The *U-net* architecture (Ronneberger et al., 2015) follows this approach (Figure 2.11). *U-net* improves the *FCN* architecture by proposing an expansive path which is more or less symmetric to the contracting path that yields a u-shaped architecture. Moreover, the expansive pathway combines the feature and spatial information through a sequence of up-convolutions and concatenations with high-resolution features from the contracting path. By introducing skip connections in the encoder-decoded architecture, fine-grained details can be recovered in the prediction. The contracting path follows the typical architecture of a convolutional network. It consists of repeated application of two 3x3 convolutions (unpadded convolutions), each followed by a rectified linear unit (ReLU) and a 2x2 max pooling operation with stride 2 for downsampling. At each downsampling step we double the number of feature channels. Every step in the expansive path consists of an upsampling of the feature map followed by a 2x2 convolution ("up-convolution") that halves the number of feature channels, a concatenation with the correspondingly cropped feature map from the contracting path, and two 3x3 convolutions, each followed by a ReLU. The cropping is necessary due to the loss of border pixels in every convolution. At the final layer a 1x1 convolution is used to map each 64-component feature vector to the desired number of classes.

In total the network has 23 convolutional layers. Although this architecture has been surpassed by more complex methods, itconstitutes a good trade-off between results accuracy and model complexity, allowing its use in contexts where the size of the training set is not particularly large.

2.4.1.6 Recurrent neural networks

Recurrent Neural Networks (RNN) (Rumelhart et al., 1986) are a family of neural networks with feed-back connections, which makes them suitable for dealing with sequential problems. The main advantage of RNN compared to other neural network architectures is their ability to process sequences of any length, while keeping historical information through their internal state. The most effective sequence models used in practice are *gated RNNs*, including the *LSTM* (Long-Short-Term-Memory) (Hochreiter and Schmidhuber, 1997) and *GRU* (Gated Recurrent Unit) (Cho et al., 2014). In a nutshell, gated RNNs create information flows through time that have derivatives that neither vanish nor explode (Goodfellow et al., 2016). The LSTM has been found extremely successful in many applications, such as speech recognition (Graves et al., 2013; Graves and Jaitly, 2014), machine translation (Sutskever et al., 2014) and image captioning (Kiros et al., 2014; Vinyals et al., 2015; Xu et al., 2015). Thanks to their ability to deal with sequential data, RNNs have also been applied reasonably to time series regression/classification tasks (Smirnov and Nguifo, 2018).

2.4.2 Unsupervised learning

Unsupervised learning (UL) relates to problems where there is no ground truth. In this situation, a standard objective is to group similar data points (i.e., clustering), another one is to look for a compressed representation of examples. In this type of learning, there is no "ground truth" answer, and it is thus difficult to compare methods objectively with an appropriate criterion.

Clustering Clustering focuses on finding common patterns in the data to find different groups within the input data. This can be used to summarise the data by prototypes which represent all possible patterns, or simply to uncover a hidden cluster structure in data.

Density Estimation Many UL objectives fit in a density estimation framework. A possible objectives of UL could is to learn the data distribution. The resulting model is able to produce new data coming from the learned distribu-

tion, hopefully very similar to the training data. This is in particular useful to create a model for detecting novelties, anomalies, or outliers in the situations where detecting deviations from the normal situation is important (e.g., IT security, dangerous situation detection). One of the common issue of these objectives is to be able to collect a representative dataset of both situations (i.e. normal and abnormal). Usually, abnormal events occur much less frequently, which inevitably results in an unbalanced distribution of data, often by several orders of magnitude (i.e., 99.5% normal data and 0.5% abnormal data).

Principal component analysis *Principal Component Analysis* (PCA) (Pearson, 1901; Hotelling, 1933), is the reference dimensional reduction method that relies on a factorisation of the matrix representing the input data. Given a generic input data $\mathbf{X} \in \mathbb{R}^{n \times p}$ the covariance matrix C can be computed as follows:

$$C = cov(X) = \frac{1}{n-1} \sum_{i=i}^{n} (X_i - \bar{X})(X_i - \bar{X})^T \ ,$$

where $\bar{X}$ is the sample mean for X. The covariance matrix is symmetric and so it can be diagonalised:

$$C = VLV^T \ ,$$

where V is the matrix of eigenevectors and L is the diagonal matrix with eigenvalues. The eigenvectors of the covariance matrix C take the name of *principal components* of X. The eigenvalues λ_k can than be used to order the eigenvectors in ascending order of the variance of the data expressed by each eigenvector. By selecting k principal components, with $k << n$ it is possible to account for most of the original dataset variability. Principal component can be used either as a method of reducing the size of the input data space and as a data exploratory tool. In fact, since the first principal components account for the most of the variability, it is sometimes possible to visualise most of input data variability by projecting the input sample on the first 2-3 principal components.

2.4.3 Model hyper-parameter fine-tuning

The algorithms presented in the previous section have one or several hyper-parameters, whose adjustment is crucial to obtain a satisfactory performance. This is called hyper-parameter selection or optimisation: the aim is to optimise the choice of the model for the task at hand. It is also necessary to select

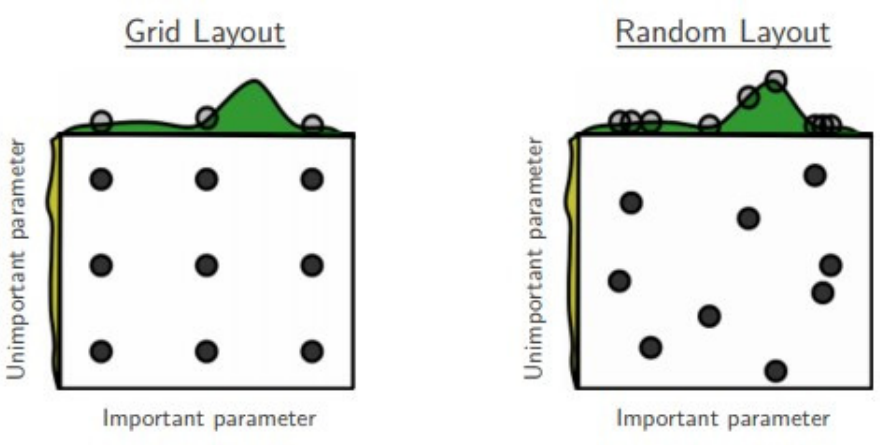

Figure 2.12: Grid and random search (from Bergstra and Bengio, 2012)

the right data preparation methods. This is particularly true for deep learning architectures where the number of hyper-parameters is important. Table 2.2 summarises the most critical hyper -parameters to for training a deep neural network.

Methods for automatic optimisation of these hyper-parameters have been proposed. There exists today many Open Source optimisation frameworks that allows to perform this optimisation tasks easily. Among them, we have used *Optuna* (Akiba et al., 2019) throughout our research work. It provides an easy interface to automate hyper-parameter selection, with two sampling algorithms: *Grid and random search* and *Tree-structured Parzen estimator*.

2.4.3.1 Grid and random search

In order to perform the optimisation, the performance of the method over the range of values of each of the hyper-parameters needs to be evaluated. It is possible to evaluate the whole value range exhaustively, by a regular test grid, which is computationally expensive. A less costly approach in terms of calculation time is the random drawing over the range of values, proposed by (Bergstra and Bengio, 2012). In Figure 2.12 the grid search and random search of nine trials are compared for optimising a generic function $f(x, y) = g(x) + h(x)$ where Above each square $g(x)$ is shown in green, and left of each square $h(y)$ is shown in yellow. With grid search, nine trials only test $g(x)$ in three distinct places. With random search, all nine trials explore distinct values of g. This failure of grid search is the rule rather than the exception in high dimensional hyper-parameter

Table 2.2: Most common hyper-parameters for training deep neural networks

Hyper-parameter	**Common values in scientific literature**	**Description**
Number of layers	{2,...,200}	Number of layers in the network
Layer type	Fully-connected, convolutional, recurrent, ...	Neural network layer family
Number of neurons per layer	{1,4000}	Number of neurons composing a layer
Activation function	ReLu, Softmax, Sigmoid, Tanh, ...	Function that defines the non-linarity of neurons
Cost function	Squared error loss, cross-entropy loss ...	Loss that measures the discrepancy between predictions and ground truth
Weight initialisation	Random initialisation, *Xavier* (Glorot and Bengio, 2010), *He* (He et al., 2015), ...	Method applied to initialise model weights
Learning rate	$\{10^{-10}, \ldots, 10^{-1}\}$	Amount that the weights are updated during training
Mini-batch	{2,256}	Size of the subsets of data seen before each weight updated

optimisation.

2.4.3.2 Iterative methods

Bergstra et al. (2011) carried out a state of the art of hyper-parameter optimisation methods for deep neural network models. This work shows the interest of iterative optimisation, based on the criterion of the Expected Improvement of the model performance, proposed by (Jones, 2001). The study introduces two optimisation methods. One method seeks to model the optimisation problem by *Gaussian stochastic processes* (GP) and the second TPE (*Tree-structured Parzen Estimator*) method proposes a kernel-based modelling. These methods are based on the construction of meta-models. The study highlighted the superiority of these two methods over the optimisation by random sampling.

In the context of this research work we applied solely the TPE algorithm. The Tree-structured Parzen Estimator (TPE) is a sequential model-based optimisation (*SMBO*) approach. SMBO methods sequentially construct models to approximate the performance of hyper-parameters based on historical measurements, and then subsequently choose new hyper-parameters to test based on this model. The TPE approach models $P(x|y)$ and $P(y)$ where x represents hyper-parameters and y the associated quality score. $P(x|y)$ is modelled by transforming the generative process of hyper-parameters, replacing distributions of the configuration prior with non-parametric densities.

In this subsection we have shown how hyper-parameters can be optimised. Whether it is carried out through a random sampling approach or through the use of iterative methods, hyper-parameter optimisation is an expensive task in terms of computation time. The cost of optimising these models is very high, due to the infinity of possible architectures and the many hyper-parameters, especially for neural networks. In the following subsection, we will present an approach that allows to reduce the overall computational time and which facilitates the convergence of the model, especially if the number of samples composing the training set is not particularly high.

2.4.3.3 Transfer learning

Transfer learning is biologically motivated by the way that humans apply learned knowledge to solve new problems, and consists in exploiting knowledge learned in one problem and searching a good protocol of transferring to a new problem. In practice, in transfer learning problems, a parametric model is trained

in the source problem and transferred to the target problem in a special way, like transferring parameters, or considering the relations between problems. This approach become particularly interesting when we deal with a dataset where the number of samples is small. There is no a well-defined rule to distinguish between a small and a large dataset. Moreover, the amount of data required to solve a machine learning problem depends on the task that we try to accomplish. In the context of this research project we consider as small, every dataset that have less than 1000 samples.

Convolutional networks are broadly applicable in the fields mentioned before. The success of transfer learning with convolutional networks relies on the generality of the learned representations that have been constructed from a large database like ImageNet (Deng et al., 2009). Yosinski et al. (2014) quantified the transferability of these pieces of information in different layers, e.g. the first layers learn general features, the middle layers learn high-level semantic features and the last layers learn the features that are very specific to a particular task. Zeiler and Fergus (2014) also visualised the features in the intermediate layers, demonstrating, with images, that convolutional networks learn features from general level to task-specific level. Overall, the learned representations can be conveyed to related but different domains and parameters in the network are reusable for different tasks. The intuition behind transfer learning for image-related tasks is that if a model is trained on a large and general enough dataset, this model will effectively serve as a generic model of the visual world. You can then take advantage of these learned feature maps without having to start from scratch by training a large model on a large dataset.

In practice, we distinguish two successive stages in the training of a neural network by transfer learning: the training of the new last layers, and then the specialisation of the whole network. The first stage is to guarantee the convergence of the classifier on the new task. We seek to obtain a satisfactory inference score. This is why in a first step, only the weights of the neurons of the new last layers are adjusted by back-propagation of the error gradient. Once the convergence of the last layers has been obtained, it is possible to fine-tune the whole network by performing an adjustment of all the weights of the layers in order to improve the classification score.

2.5 Conclusion

In the manufacturing industry, product quality is an indicator for evaluating the production capacity of a company. Customers are increasingly demanding in terms of product quality and providing the customer with a product that complies with the specifications is absolutely essential in a market that is becoming more and more competitive. The best possible solution to deliver 100% of compliant parts to the customer would be to inspect in details all parts produced. However, most companies cannot test every single product. There may simply be too high a volume or number of them to inspect at a reasonable cost or within a reasonable time frame. Or effective testing might result in the destruction of the product or render it unfit for sale in some way. Moreover, traditional process control approach does not take into account the relationship between process data and produced part quality. In this chapter we have described a general approach that can be applied every time that we want to take advantage of an historical set of data to improve manufactured parts quality. The presented data-driven approach requires four main stages: data acquisition, data processing, exploratory data analysis and machine learning modelling. Data acquisition involves the task of identifying all the input process data, as well as the output quality data, that are interesting to try to solve our quality improvement use-case. Data processing is the task of processing and filtering raw input data in order to reduce the noise within data and to make data suitable for machine learning modelling. Exploratory data analysis could be used for better understanding the correlation between data and to eventually fine tune the data processing stage. Finally, machine learning modelling allows to build a model which relate input process data and output quality data. In this way it is possible, for a new set of input data, to provide the prediction, or inference of the quality of the finished part. Moreover, using an interpretable model it is possible to identify which process parameters affect the most manufactured parts quality and eventually improve process monitoring. There exists a wide variety of machine learning algorithms. In the second section of this chapter we have presented the linear regression and the linear models with a penalisation term, Tree-Based methods, Support Vector Machines and a few Deep-learning architectures that we have applied throughout our research work. In the following chapters we will present an application of this method in the industrial context studied along this doctoral studies.

2.5.1 Industrial contribution

From the industrial point of view, this chapter describes a new way of monitoring processes and controlling quality. Instead of relying on acceptance sampling, the historical data can be used to infer a model able to provide real-time quality inspection without any direct measurement of the part. This leads to two major benefits:

- Faster detection of quality non-conformities. By providing a quality status for each part produced, it is possible to quickly react and adjust the production process to prevent occurrence of new non-conformities.
- Reduced destructive testing. If the model is reliable enough to provide accurate results, it is possible to reduce the number of destructive tests that are regularly performed to assess part quality.
- Better understanding of the production process. If the machine learning model is interpretable, it is possible to identify which process parameters that most affect part quality. In this way, it is possible to fine-tune the process by focusing more attention to the control of important parameters.

Chapter 3

From corrective to predictive process control

Contents

3.1 Introduction

This Chapter describes an application of the previously proposed method to our industrial context. Since out-of-tolerance tank weight is the primary cause of part non-conformity, we investigate process parameters that contribute the most to the variability of tank weight. The first step is to gather a data set representative of the phenomenon to be modelled. This historical data set is then used to model the relationship between process measurements and part quality. The results, as well as the difficulties encountered when applying such an approach in the industrial context studied will be discussed in detail. Finally, we will show how the results obtained in the framework of this research work have allowed us to identify some areas for improvement in our manufacturing process.

3.2 Motivation

Poor quality or "scrap" parts are very expensive for a company like Plastic Omnium Clean Energy Systems. The "Cost of Non-Quality" (CNQ) is one of the key indicators most used by the company. However, when a part is declared bad, it is first necessary to understand the origin of the problem, which can require a lot of time and energy. Historically, Plastic Omnium industrial process monitoring has been driven using a knowledge-based corrective approach (Figure 3.1). The quality measurements of each product is used to adjust the process and to maintain the process capability. Moreover, some of the process parameters, which are considered as critic for process safety, are kept under control through the use of uni-variate control charts. When a parameter falls outside the control limits, some warning messages are generated to alert the operators who have the task of regulating the machine so that the parameter can return in a safe zone.

Evidence has shown that the overall stability of the process ensures, in most cases, the stability of the product quality. However, it still remains unclear how the system parameters impact the variability of product quality. Quality prediction would allow better adjustment of system parameters at an early stage of production. In other words, anticipation of product quality could be used to adjust the process in real time rather than retrospectively (Figure 3.2). Such an approach would allow process failures to be anticipated and corrected just-in-time, with an overall reduction in the production of non-conforming parts.

In order to understand the correlation between process parameters and the

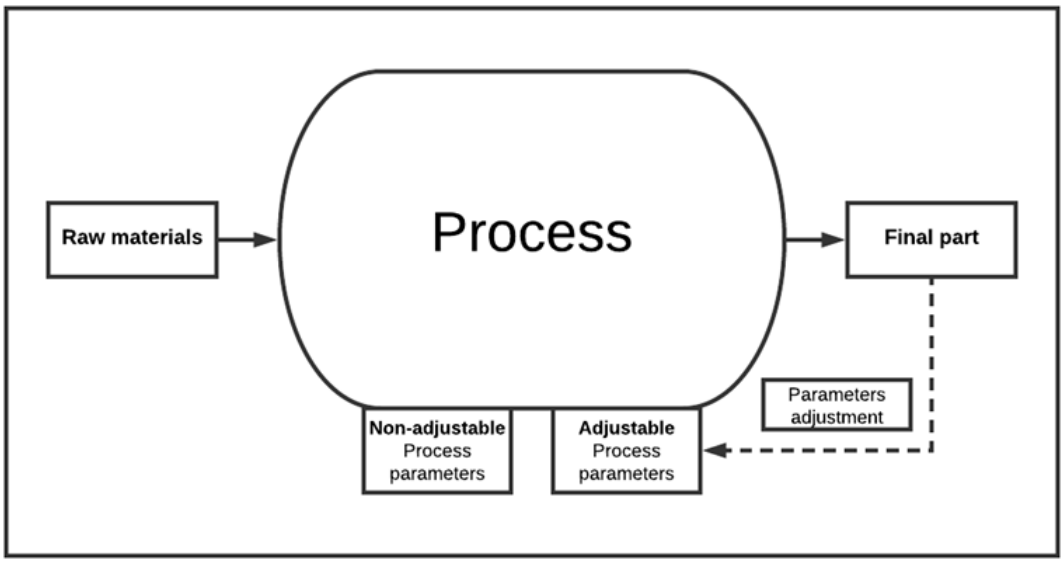

Figure 3.1: Corrective process control

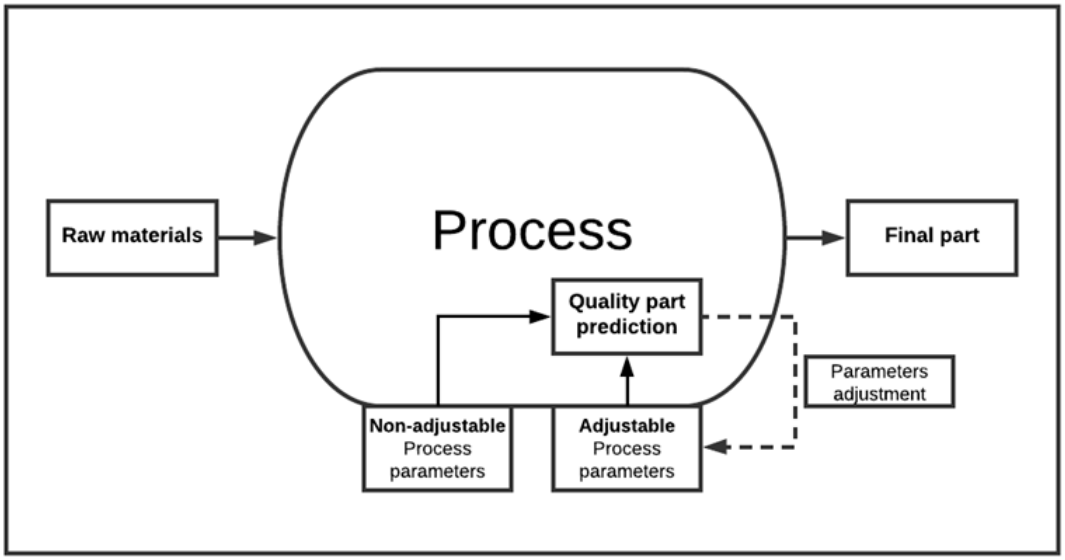

Figure 3.2: Predictive process control

quality of the final part, we have made the choice to use a supervised learning approach. We view our complex industrial process as a black box with multiple inputs and one output. Given p process parameters $(X_1, X_2, \ldots, X_p)$ and one product quality variable Y, we look for the function that better approximates the relationship between inputs and the output. Mathematically speaking, we look for the function g that approximates the relationship between the process variables and the quality result:

$$Y = \hat{g}(X_1, X_2, \ldots, X_p) + \epsilon \ , \tag{3.1}$$

where ϵ is defined as the part of Y that cannot be predicted from the input process parameters $(X_1, X_2, \ldots, X_p)$. By an automatic analysis of a set of examples (training set) of measured input-output behaviour of the process, learning algorithms can find out important correlations between process variables and construct classifiers for detecting dangerous or unwanted process states. This would allow for a better comprehension of the manufacturing process.

In this first experimentation we decided to try to explain what process parameters affect the most the weight of the final part. As presented in Section 1.2.2, the tank weight is historically considered as an important product characteristic as it provides an overall indication about the amount of material that composes the fuel tank. Moreover, the weight has the advantage of being a resultant variables. All the quality information is stored only in a single scalar values. By following the approach described in Section 2.3 of the previous Chapter, we aim to search for any hidden pattern or correlation within process data and quality data that could explain why some parts are not compliant in term of weight.

3.3 Data collection

3.3.1 Process parameters of the SCADA software

More than 5000 parameters are measured in real-time at each production cycle of our industrial process. Among these features, some are considered by the experts as critical to ensure the proper stability of the process (see Section 1.2.1). In addition to the critical process parameters there are timer and counter variables. A timer variable accounts for the time needed to execute a particular mechanical movement in the machine production cycle. The sum of all the mechanical times corresponds to the machine cycle time. A counter variable, instead, increases over

time because of a particular event. For example, the number of parts produced in a production day is recorded in a counter variable.

Process parameters are collected by the internally developed SCADA system and data are stored in multiple databases in accordance with the sources of each one. For instance, all the extrusion process data are stored in a database. The same is true for the blow-moulding data and for the weight of the tank that are stored in two separate databases.

Each process parameter measured in real-time during the production process needs to be associated to the scalar value corresponding to the quality measurement of the manufactured product, at the end of the production cycle. In order to do that, the SCADA software computes some aggregate data to summarise the information in a limited set of scalar features. For each variable belonging to a production cycle, the average, maximum and minimum values are calculated. Then, the SCADA software attaches the aggregates data belonging to a production cycle to a specific traceability serial number which can be used as key to link the different sources of data.

The extrusion blow-moulding process studied in this work has no system for measuring the parison length. As explained in Section 1.2, the parison length provides information about the material distribution. In the following subsection, we present the approach we have used to measure the parison length in real-time.

3.3.2 Parison length estimation by computer vision

The machine studied in this research work does not have any system to measure the parison length. As a consequence, it seemed necessary to equip the machine with a system capable of providing us with the parison length information. We looked for a measuring system respecting the following industrial constraints:

1. relatively low software and hardware costs,
2. requiring little expertise to adapt the model or process the data,
3. adaptable to changes (for example, adaptable to any blow moulding machine in Plastic Omnium's plants),
4. making analyses in real time, returning the result with low latency,

5. able to operate in hostile environments,

6. be robust to environmental variations (e.g. the system must operate day and night, regardless of lighting conditions).

To meet these industrial constraints, we opted for computer vision system. Our choice is motivated by the low cost of a camera, its ease of deployment, and the capabilities of deep learning-based models to detect objects in images. In Section 2.4.1.5 we shown how Convolutional Neural Networks reach state-of-the-art results in image classification, object detection and image segmentation tasks. We therefore chose this solution to detect the parison and to measure its length in real time.

Proposed method The length measurement involves two main stages:

- Parison detection: the parison should be detected inside the field of view of the camera. A CNN is trained to detect a (tight) bounding box containing the parison.

- Length measurement: once the parison is detected, its length corresponds to the height of the bounding box containing the parison object.

The CNN architecture we chose is *SSD MobileNet-V2* (see Section 2.4.1.5), which presents an interesting trade-off between inference speed and model performances in the wild.

To limit the burden of data collection and annotation, we made the choice of using transfer learning (see Section 2.4.3.3), with the chosen architecture initialised with the pre-trained coefficients of the *COCO* dataset (Lin et al., 2014). The last linear layer of the model was replaced in order to be consistent with the number of classes that we want to detect. Since we are only interested in detecting the parison class, our last layer returns a result in $\mathbb{R}$.

Data collection and training In order to train such a model, 200 images of parisons were collected using a camera of HD resolution (1280×720). By taking parison images during the extrusion, we built a dataset representative of all possible parison lengths. All images were then manually annotated using the open-source software *Labelme* (Wada, 2016). Firstly, the mask of the parison is manually drawn using polylines. Subsequently, it is possible to retrieve the bounding box containing the parison (Figure 3.3).

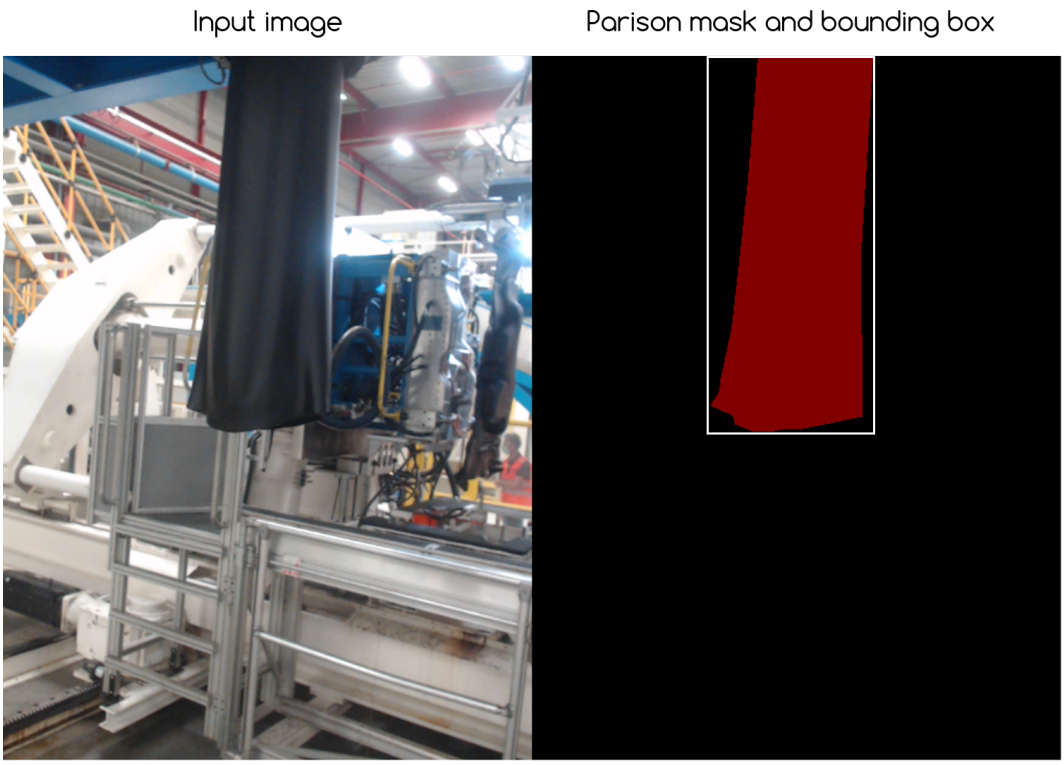

Figure 3.3: Input image and parison mask

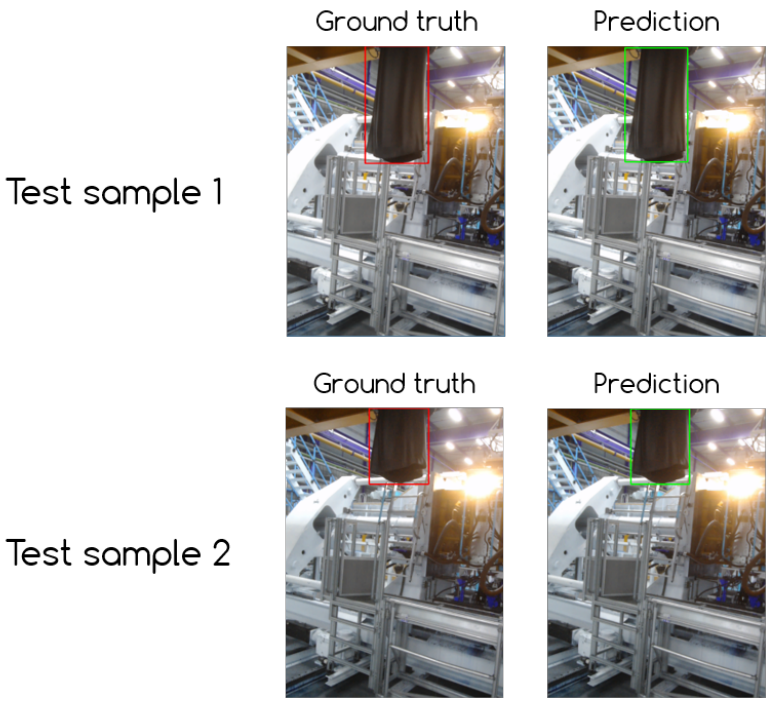

Figure 3.4: Two parison length inference examples (from the test set)

The *SSD MobileNet-V2* has been trained by minimising the multi-box loss with $\alpha = 1$ (see Section 2.4.1.5). Data was split into three different subsets: a training set (70%), a validation set (10%) and a test set (20%). *Adam* optimiser (Kingma and Ba, 2015) with the default parameter values ($\beta_1 = 0.9$, $\beta_2 = 0.98$ and $\epsilon = 10^{-9}$) was used to minimise the multi-box loss function. The validation loss is monitored during the training in order to prevent the model overfitting.

Results SSD MobileNet-V2 has been proven to be able to provide accurate results in a limited amount of time. Figure 3.4 shows two ground truth boundary boxes from the test set and their corresponding prediction. The computation time is below 200 millisecond on a *Nvidia* Jetson Nano (a small computer equipped with a cheap GPU designed for embedded applications and deep learning based IoT). The proposed computer vision system has proven to be robust and able to meet the identified industrial constraints, making accurate and reliable predictions in real time. A simple RGB camera and a small computer offer a cheap, non-intrusive solution that does not require any modification of the machine.

Once the parison length acquisition was running online, we built a dataset with the SCADA software variables as well as the final parison length just before the moulds close to blow-mould the final part. We collected data from 5 different batches, corresponding to as many production days, for a total of 5597 samples and more than 5000 features.

3.4 Data processing

Before moving to the quality modelling given our input process data, we need to preprocess the data to keep only the interesting variables for our use-case. Moreover, the number of input features is too large compared to the number of samples available in the dataset. This can lead to overfitting issues. In order to reduce the variable space dimension, we employed two different procedures: an expert-based procedure and a statistical-based procedure.

Expert-based data procedure We relied on expert knowledge of the process to discard all features that are not relevant to explain the weight variability. For instance, all counter variables collected by the SCADA software do not bring any interesting information and can be removed. Also, many timer variables, representing the time needed to execute a particular mechanical operation, are redundant and provide no added value. Therefore, most of the timer variables have been removed from the dataset.

Statistical-based procedure In order to further reduce the number of features, three different statistical feature selection approaches have been used, based on:

- correlation between features,
- feature variance,
- *Stability selection* (see Section 2.3.2.2).

Removing highly correlated variables eliminates redundant features and reduces collinearity between features that can cause stability problems when fitting the regression model. For each pair of features with a correlation value greater than 0.90, one of the two features was removed. Features with very low variance (constant or with no more than 3 different values) were removed. To further reduce the number of features, we applied stability selection. By generating bootstrap

samples of the data, and by leveraging the ability of the LASSO penalty to estimate which features are important in each sampled version of the data, we are able to select only those features that have been selected for many perturbed versions of the original problem. Data were finally normalised to have zero-mean and unit-variance (see Section 2.3.2.3). Normalising features is not only important if we are comparing measurements that have different units, but it is also a general requirement for many machine learning algorithms. Figure 3.5 resumes the data processing flow.

3.5 Exploratory data analysis

3.5.1 Weight versus parison length

This was the first time that data on parison length was available. The relationship between the parison length just before the blowing phase and the weight of the blown part is illustrated by a scatter plot in Figure 3.6. The plot shows a weak negative correlation between the parison length and the weight. The Pearson correlation is equal to -0.32. The physical explanation is simple: the longer the parison, the less material remains inside the mould during the blow-moulding. Moreover, the upper part of the parison is often thicker to prevent the parison from breaking under its own weight.

3.5.2 Low dimensional representation

The large number of features makes it difficult to detect possible hidden patterns in our data by a direct visual representation. We use Principal Component Analysis (see Section 2.4.2) to produce the low-dimensional representation that best preserves the distance between examples. Figure 3.7 represents the projection of the samples from 3 batches on the first 2 principal components of PCA. This representation shows two major patterns:

- data are organised in clusters,
- for each cluster of data there exist a subset of points moving apart from the cluster.

Each cluster corresponds to a specific production day or batch. This means that, for the same tank reference, the process parameters differ more between each batch than within a batch. We will call this phenomenon the "batch effect". Moreover, the samples are at the periphery of the centre of each clus-

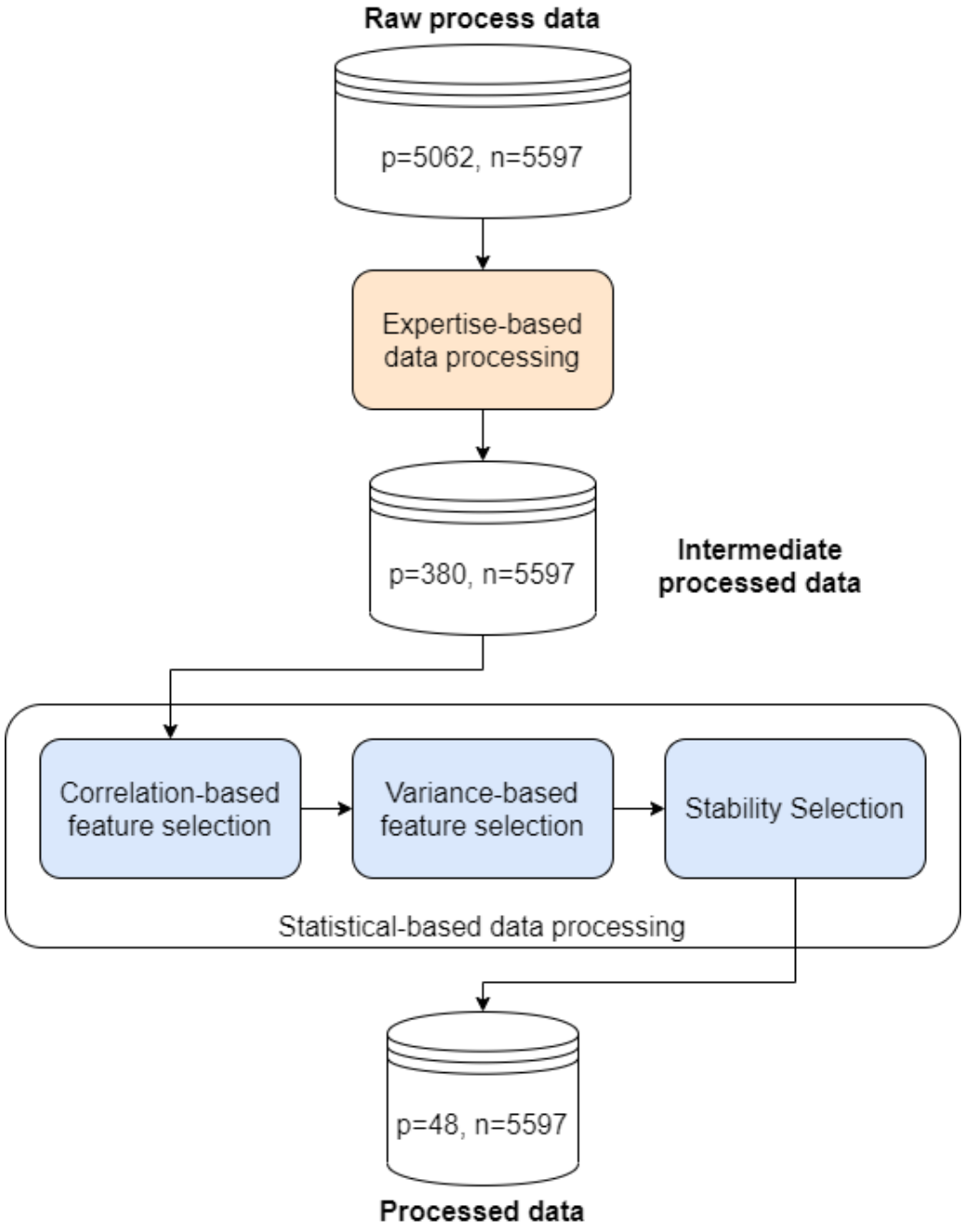

Figure 3.5: Data processing flow

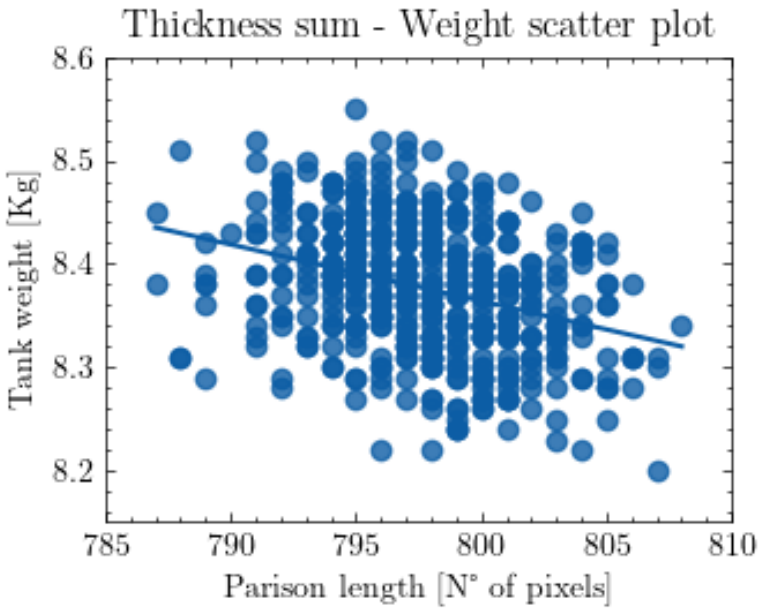

Figure 3.6: Parison length - Weight scatter plot

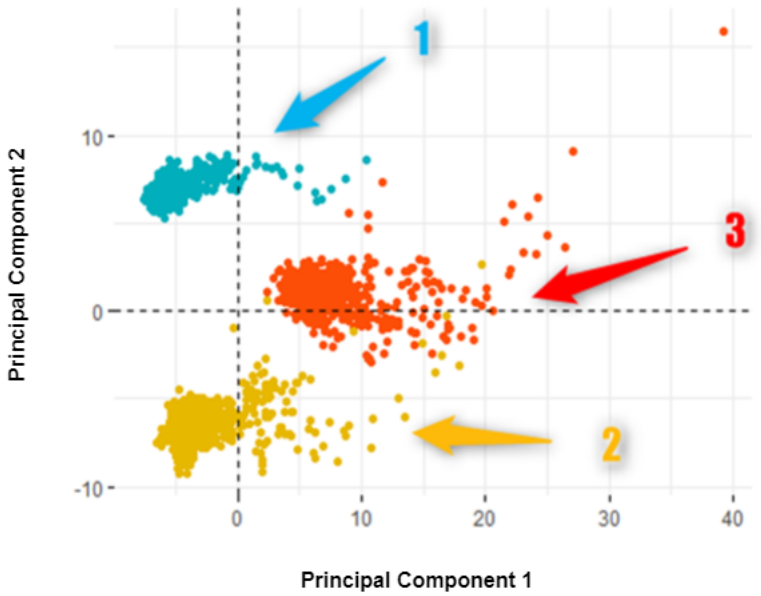

Figure 3.7: Sample projection on the two first axes of Principal Components Analysis

ter correspond to tanks produced in the first two hours after the machine was started. This allows to identify two operating regimes: the *transient* and the *stable* regimes. During the transient regime, the process parameters are slightly different, particularly in terms of temperature. During the transient regime the rate of non-conformity rate in terms of weight is considerably higher (about 300%) compared to the stable regime.

3.6 Supervised learning modelling

Given our input processed data X composed of p process parameters and n samples and the output vector $Y \in \mathbb{R}^n$ of the tank, the role of the supervised learning modelling is to find the function $\hat{g}$ such that:

$$\hat{g} = \underset{g \in \mathcal{G}}{\operatorname{argmin}} \sum_{i=1}^{n} (Y_i - g(X_i))^2 \quad . \tag{3.2}$$

We made the choice to treat our problem as a regression one. Instead of predicting only if the part is compliant, or not, (classification) we look for predicting the continuous weight value. In this way we hope to be able to explain even the smallest weight variations. Since we are interested in understanding what process parameters affect the most the weight of the manufactured tank, we privilege for this task easily machine learning algorithms such as linear models (see Section 2.4.1.1) and tree-based methods (see Section 2.4.1.2). The choice of these algorithms is motivated by:

- *Interpretability*: Since we aim to understand which parameters most affect the weight of the blow-moulded tank, we are interested in applying interpretable models. Linear models and tree-based methods are considered to be among the most easily interepretable models. We claims that deep learning based methods are not well suited for this task as a consequence of their "black-box" nature.

- *Performance*: These methods work quite well with tabular data. When dealing with tabular data, deep learning hardly surpass traditional machine learning algorithms (Shwartz-Ziv and Armon, 2021).

In order to train and evaluate the predictive accuracy of our models, we used the following training strategies. Firstly, the dataset was split into three different subsets: the train set (70%), the validation set (10%) and the test set (20%). For each algorithm, the *Tree-structured Parzen Estimator* algorithm (see

Table 3.1: Hyper-parameter search space

Model	**Hyper-parameter**	**Search space**
Lasso	λ	LogSpace(10^{-5}, 1)
Ridge	λ	LogSpace(10^{-5}, 1)
Gradient Boosting	Number of predictors	[5, 300]
	Learning rate	LogSpace(10^{-5}, 10^{-1})
	Maximum tree depth	[4, 50]
	Minimum samples leaf	[1,60]
Random Forest	Number of predictors	[5, 300]
	Maximum tree depth	[4, 50]
	Minimum samples leaf	[1,60]

Section 2.4.3.2) is used to select the hyper-parameters that minimise the *Mean Squared Error* (MSE) on the validation set. The exhaustive list of model hyper-parameters and their search space is summarised in Table 3.1.

Finally, the model with the set of hyper-parameters which minimise the MSE, for each algorithm, is used to evaluate the performance on previously unseen data (test set). The R^2 metric is used to evaluate the models performance. We have decided to use this metric because we are primarily interested in understanding if our input data are able to explain the weight variability.

3.7 Results and discussion

Results are resumed in table 3.2. All models return negative R^2 values. This means that taking the average of the weight of the train samples as the prediction for each test sample would have provided better results. Our input process parameters do not allow to explain the variability of the tank weight. Results highlight how all the approaches tend to over-fit but struggle in generalise what has been learned on the train set to unseen new samples. This is especially true for tree-based methods that in general are more prone to over-fit.

A further analysis was conducted to try to explain and motivate these results.

Table 3.2: Supervised learning modelling results

Algorithm	R^2 **train**	R^2 **validation**	R^2 **validation**
Linear regression	0.80	-0.25	-1.34
Lasso regression	0.72	-0.34	-0.45
Ridge regression	0.78	-0.26	-0.39
Random forest	0.94	0.05	-0.12
Gradient boosting	0.95	-0.11	-0.35

We have identified four possible reasons for these negative results:

1. *Non-stationarity of data*
2. *Lack of relevant data to explain weight variability*
3. *Low variability in product quality*
4. *Reliability of the input data*

The following paragraphs provide more details about each of the possible reasons.

Non-stationarity of data Results obtained show that our models do not generalise among different batches. Actually, if we look at distributions of our input features we can see how they change considerably among different batches (Figure 3.8). This is also deductible by looking at the PCA plot in Figure 3.7. A "Two-sample Kolmogorov-Smirnov" test was applied to all pairs of batches. According to these tests, only 35% of the input features share the same probability distribution over all batches (at a 0.95 confidence level). Standard machine learning models rely on stationarity hypothesis to generalise: a trained model expects that the test distribution follows the distribution of data used to train the model.

Lack of relevant data to explain weight variability The change in data distribution may also be due to some external events or factors that we do not control and do not take into account within our own input process data. For instance, we claim that the "batch effect" we observe in the data could be a consequence of certain changes in the rheological properties of raw materials. In

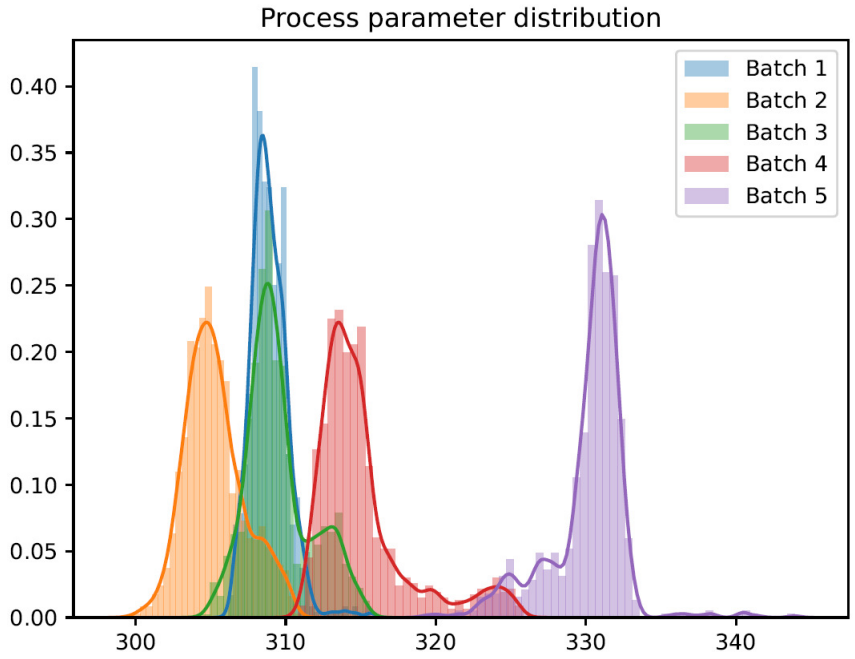

Figure 3.8: Probability distributions across batches for the pressure of one of the 6 extruders

fact, the final part is the result of the transformation of raw material through our complex process. Unfortunately, to this date, these data are not available and they cannot be integrated in our dataset. Further studies have been carried out in order to see if it is possible to measure some rheological properties of the material in real-time. There are industrial on-line rheological systems that provide continuous measurements of the melt flow rate or apparent viscosity directly on the manufacturing process. Unfortunately, this solution is not economically viable, especially as such a system would have to be installed for each of the 6 screws.

Low variability in product quality The manufacturing process is already relatively reliable and stable, with low variability in product quality. The scrap rate is under 3% and the tolerance limits set to evaluate the compliance of blow-moulded tanks are quite strict. In general, a weight variability of about 300 grams is sought, which for an average tank weight of 8.5 kilograms corresponds to about 3.5% of the total weight. The phenomenon modelled would have been more pronounced, and the problem would have been simpler with a larger weight variation.

Reliability of the input data As we look for small variations of weight, it is important that the input data is accurate enough. In an industrial environment, such as a production plant, the collected data are most of the time noisy. The maintenance of the sensors cannot be done regularly. As a consequence, some sensors may provide erroneous values. Moreover, the SCADA software computes some aggregate operation on the input time series-data, which can lead to a loss of information. Finally, it is quite complex to attach extrusion data to the traceability serial number of a tank since extrusion is a continuous process and there are no precise triggers to define what data belongs to a given part. The extrusion data may be misaligned with the manufactured part.

These results highlights the difficulties we can encounter when dealing with manufacturing process data. However, in the following section we will show how the work presented in this Chapter has made it possible to start a new project to improve the manufacturing process.

3.8 SmartBMM: towards smarter machines

The data analysis results presented all along this Chapter have shown the inability to explain the tank weight variability given the blow-moulding process data that are considered as critical by process experts. The possible reasons have been discussed in detail in the previous section. What the analysis has also highlighted is that the most scrap occurs just after the machine start-ups (see Section 3.5.2). As shown previously, right after the machine start-up, the extrusion blow-moulding process is not completely stable which increases the overall scrap rate of the blow-moulded parts. Moreover, an interview of different extrusion blow-moulding experts has highlighted that there are no common and shared best practices to start the machine. As a consequence, there is a lot of variability between startups.

Figure 3.9 illustrates this variability among 27 machine start-ups performed in a Plastic Omnium plant. On the left bar-chart, we can see how the time needed to start the machine changes from a start-up to another. Sometimes the start-up is done in 10 minutes, other times a full one may take around 15-20 minutes. There are also three occurrences for which the start-up took more than 20 minutes. The right bar-chart reports the scrap rate in the first 60 minutes. Most of the time, the scrap rate does not exceed 5%, but there are some starting for which the scrap rate is above 10%. These observations call for some efforts to improve the way the extrusion of blow-moulded machines is started. By automating and by

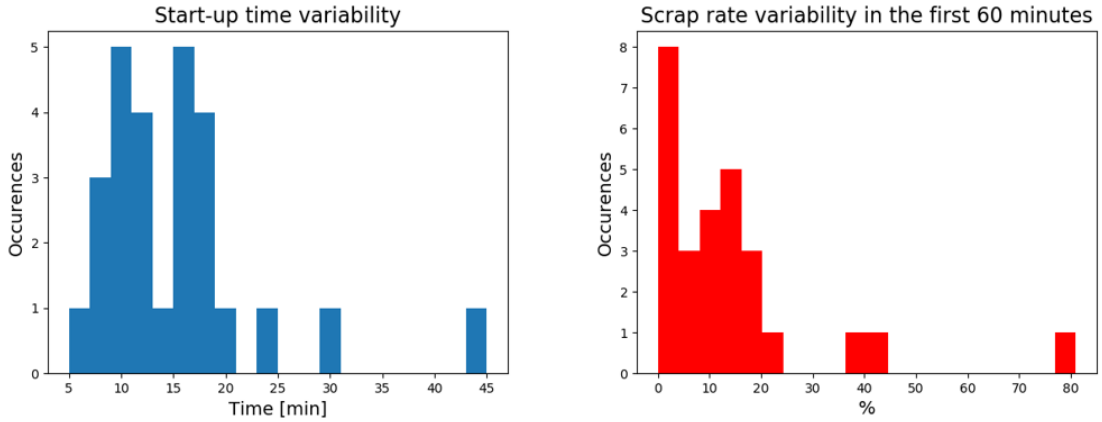

Figure 3.9: Time and scrap rate variability for 27 machine start-ups in a Plastic Omnium plant

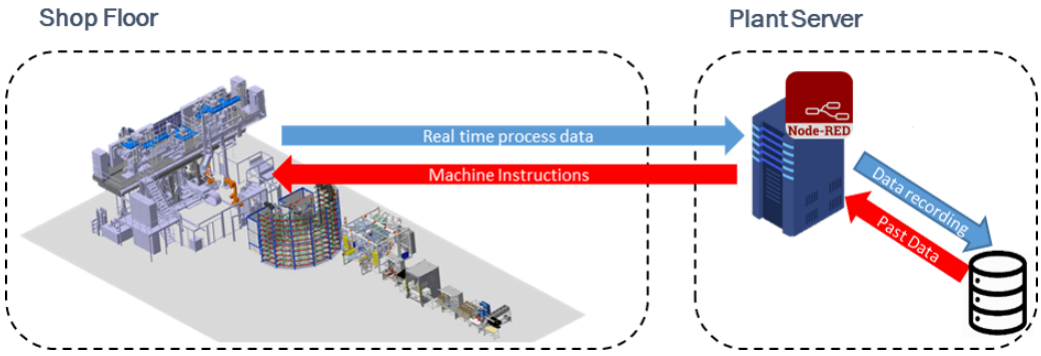

Figure 3.10: *SmartBMM* software

optimising the machine start-up we should reduce the uncertainty introduced by manual starts. This would allow for a faster convergence towards the stable regime of the machine and, as a consequence, to a smaller number of part non-conformities.

The project was initially conceived to handle just the machine starting phase but later on, it was extended to also cover the purge cycles of the machine. Indeed, ensuring good purge cycles reduces the risk of contamination/inclusion problems (see Section 1.2.2). In this context, we have developed the *SmartBMM* solution. *SmartBMM* is a software which leverages the real-time data collected directly from the Programmable Logic Controller (PLC) of the machine and the past data to follow the best instructions to get the machine started without any manual intervention of the operators.

Figure 3.10 shows broadly how the *SmartBMM* works. The *SmartBMM* software collects real time data directly from the PLC and stores it in a database for later retrieval. When the *SmartBMM* software is started the data collection continues, but, this time, the machine starts to write information to the PLC to execute a set of operations. The software takes the real time incoming data and the past data to elaborate the machine instructions to bring the machine to production conditions. The stored data are used to compute production extruder speeds which, accordingly to past data, minimise the non-conformity rate. Looking at previous production runs we are able to retrieve the process conditions which lead to a better performance and a lower scrap rate. The software is developed using the *Node-RED* (OpenJS Foundation, 2013) programming tool. Node-red is a low-code programming for event-driven applications which was specifically designed to work with IoT and that allows easy interfacing with machines through different communication protocols.

Instead of manually start the machine by pressing simultaneously multiple buttons on the HMI (Human Machine Interface), machine setters and operators have to press only one button to start a cycle, whether it is a *starting* cycle or a *purge* cycle.

- The starting functionalities leverages the real-time data collected directly from the PLC of the machine and the past data to elaborate the best instructions to get the machine started without any manual intervention of the machine operators. The set of consecutive instructions provided to the machine are fairly standard. The extruders are started, then the material is fed into the screw, then the extruder speed is raised and so on. However, our system does not rely on timers to trigger the machine instructions. In real time the process status is controlled and the next machine instruction is triggered only if the process meets all requirements. Two starting functionalities are available: *full* and *downtime*. The first one executes a starting phase when the machine is completely stopped; the second one returns to the production condition in which the machine was temporarily idled.

- The purge functionalities allow to improve purge cycles of the machine. During purge cycles we want to ensure that enough material transits into the extruders at high pressure to clean them of residual production material. Instead of relying on fixed speed values or timers, we have developed a *PID controller* that regulates the extruder speeds to ensure that they are constantly above the pressure targets. Moreover, the amount of material

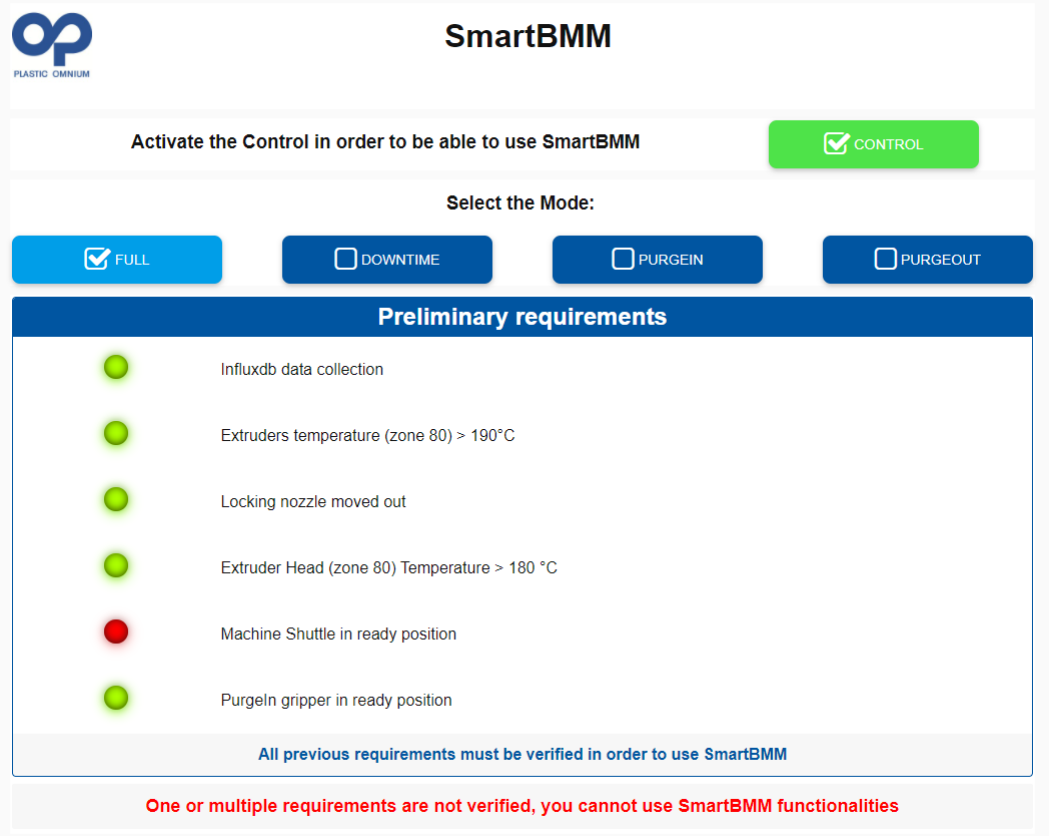

Figure 3.11: *SmartBMM* GUI

transiting through the screws is controlled in real-time. This allows to finish the purge cycle only when the correct amount of material has transited through the screws. The software also handles the machine stopping by ensuring the correct emptying of the extruders.

The starting and purge functionalities reduce the "weight not OK" and the "contamination" scraps, which account for 2/3 of the total amount of non-conformities. The functionality is chosen by the operator through a graphical user interface (GUI) specifically developed to allow an easy interaction with the software (Figure 3.11). Historically, all manufacturing processes, including extrusion blow-moulding, rely on PLC programs to execute the set of instructions to allow the manufacturing machine to work correctly and to allow for the transformation of raw materials into finished products. In this project, we decided to pilot the machine outside of the machine PLC for the following reasons:

- PLCs are very robust and safe, but they lack flexibility. PLCs are conceived

to execute a set of logical instructions but they do not lend themselves well to be used concurrently with other systems such as databases. Developing a more complex logic which involves data storage and communication with databases is by far more easy with traditional software development tools.

- Implementing the logic on an external system such as a physical or a virtual server reduces the number of PLC modifications of the machine. Our software is non-intrusive and can be installed remotely on a server without any direct modification of the PLC program. It is something that can be plugged to the machine to introduce new functionalities. This reduces costs considerably, since it does not require the intervention of an external consultant.

This strategy has, however, a main drawback. Our tools completely rely on plant network to communicate instructions to the PLC. Which means that a bad network could be a bottleneck for the correct functioning of our software. Security features have been added on the software side to interrupt the communication with the machine if any network failures prevents to communicate with the machine. In our opinion this is an example of a cyber-physical system. *SmartBMM* leverages the sensor networks with data processing to monitor and control physical environment, with feedback loops able to elaborate the best set of machine instructions given the different process conditions.

3.9 Conclusion

In this Chapter, we have presented an empirical evaluation of our approach in the industrial context. The features currently collected by home made *SCADA* system, even when enriched by parison length measurements, cannot predict the tank weights. These results show the difficulty of applying statistical models in batch manufacturing industries, where it is not always possible to have knowledge of all the elements that contribute to the variability of the part quality. Possible explanations for these results were discussed. Nevertheless, this research work has allowed the identification of avenues for improvement of the blow-moulding process. In particular, the exploratory data analysis has shown how most of the part non-conformities occur just after machine start-up, when the process is not yet stable. This led to the launch the *SmartBMM* project.

3.9.1 Scientific contribution

The results presented in this Chapter question the effectiveness of data-driven methods in certain manufacturing contexts. The end-to-end data-driven methods have proven to be effective in many applications, but for them to work well, informative data on a stationary phenomenon is required. In other situations, it may be better to decompose the original problem into several sub-problems with properly controlled data quality. The following Chapter will present such an approach.

3.9.2 Industrial contribution

This Chapter presents three main industrial contributions.

- Firstly, the work challenges certain beliefs about the operation of the blow-moulding process. The process parameters considered critical to ensure proper functioning of the process do not explain the variability in tank weights. The control limits previously set for the critical parameters of the process to ensure the correct functioning of the production process were found to be insufficient for providing such an explanation.

- The parison length measurement has opened new research perspectives. By measuring in real-time the length of the parison, we will eventually be able to control the distribution of material over the entire length of the parison to improve the quality of the manufactured parts.

- Finally, the *SmartBMM* software that was developed starting from the results obtained trough the data analysis process has been used to improve the machine start-up phases which could lead to a higher scrap rate. By ensuring a better start-up, we reduce the transient phase and thus the percentage of parts that do not meet quality standards. Future works will make use of the parison length measured by the camera to add new functionalities to *SmartBMM*.

Chapter 4

Thickness inference using thermal imaging

Contents

4.1 Introduction

In the previous chapter, we showed the difficulty of inferring the weight of the tank given the process parameters as they are collected today. The possible explanations have been addressed in section 3.7. The weight is a quality indicator that summarises the information about the overall amount of material which composes the fuel tank. However, as we saw earlier, the weight does not guarantee the correct distribution of the material over the whole surface, as different thickness distributions may lead to the same weight. As a consequence, we decided to focus directly on thicknesses, which provide more accurate information about the distribution of the material over the surface. Of course, measuring the tank thicknesses using the only the available process parameters could be challenging as we have no information about the geometry of the tank. It is therefore necessary to collect new data that can provide us with spatial information.

In this chapter, we propose a new approach to perform a real-time, non-destructive quality control to measure thicknesses of blow-moulded parts. The proposed approach makes use of deep learning data-driven methods to leverage the thermal inertia of the manufactured plastic part, captured through thermal imaging, to infer the thicknesses of the part surface without any direct measurement. Compared to traditional quality inspection approaches, which aims to detect visual defects of manufactured products, our approach leverages thermal information to perform a non-visual quality control. The first experimental results on real industrial data are very promising and demonstrate that the proposed method could achieve satisfactory performance in industrial conditions.

4.2 Weight and thicknesses

Before digging into the details of the proposed approach, it is interesting to explore the relationship between the thickness values and the tank weight. In order to asses if there exist a measurable statistical relationship between the weight and the tank thickness a simple correlation analysis has been conducted. By taking advantage of 300 measured tanks, for which the weight, as well as the thickness of a limited number of critical points, the correlation between each thickness-weight pair is computed. Figure 4.1 shows the coordinates of the six points for which the thickness was measured. Figure 4.2 shows the scatter plots drawn by plotting each thickness-weight pair, with one variable on each axis. For each plot, the regression line is drawn to visualise the trend. As visible in the Figure, there exist a minor correlation between some of the thickness points

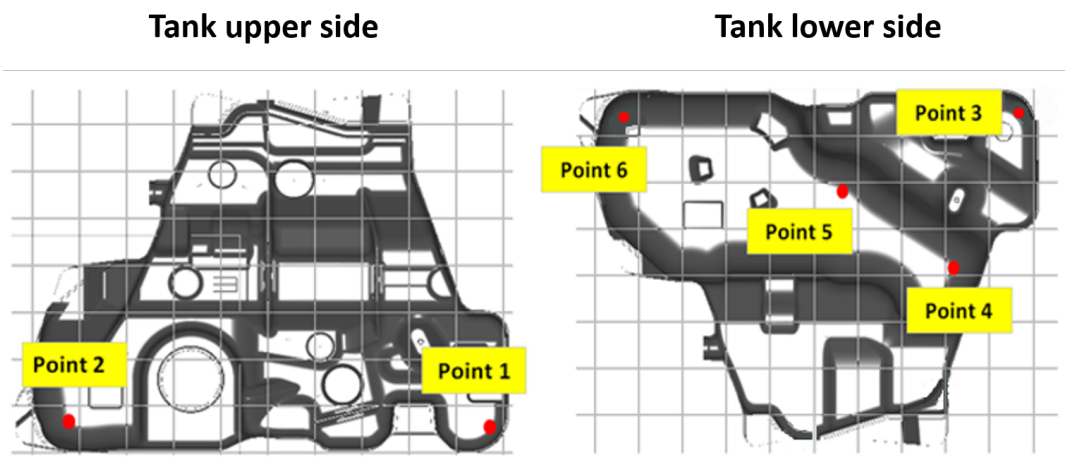

Figure 4.1: Thickness points

and the tank weight. The highest correlation, between the thickness at point 5 and the weight, is 0.40: there is a lot of dispersion within data, and a higher weight does not mean a higher thickness. This confirms our assumption that it is impossible to ensure the correct distribution of material over the surface of the tank by monitoring solely the weight. The weight and the sum of the 6 measured thicknesses are slightly more correlated (Figure 4.3) as expected since the weight is proportional to the amount of material composing the fuel tank. Results presented in this section show that the weight of the tank is not sufficient to ensure the correct distribution of the material along the surface. In order to improve quality control and ensure that thicknesses meet customer specifications it is advisable to focus our research work directly towards the inference of thicknesses.

4.3 Motivation

Our work is motivated by the empirical observation of the cooling of blow-moulded parts in the first minutes after blowing. Areas of the parts have different cooling behaviours depending on their thickness.

Areas with smaller thicknesses cool down faster than those with higher thicknesses. For the thicker zones, the surface temperature even starts to increase before decreasing (Figure 4.4). This phenomenon is due to the release of energy from the innermost plastic layer that has not be in direct contact with the mould surfaces. As presented in Section 1.2.2, this surface temperature decay, easily

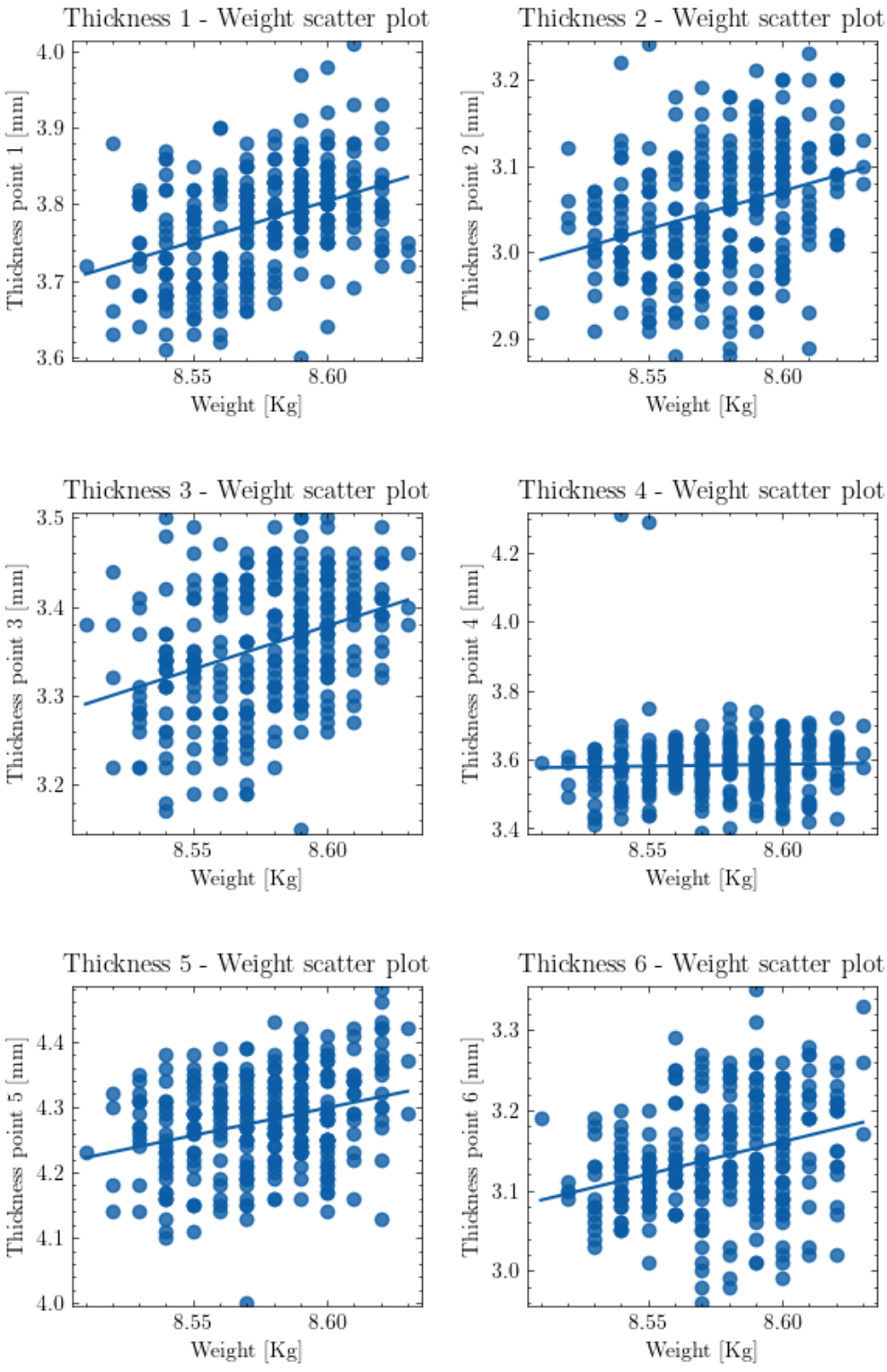

Figure 4.2: Thicknesses - weight scatter plots

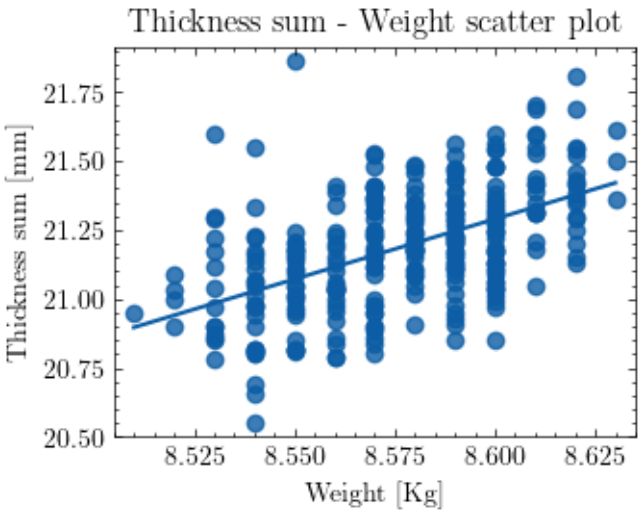

Figure 4.3: Thickness sum - Weight correlation

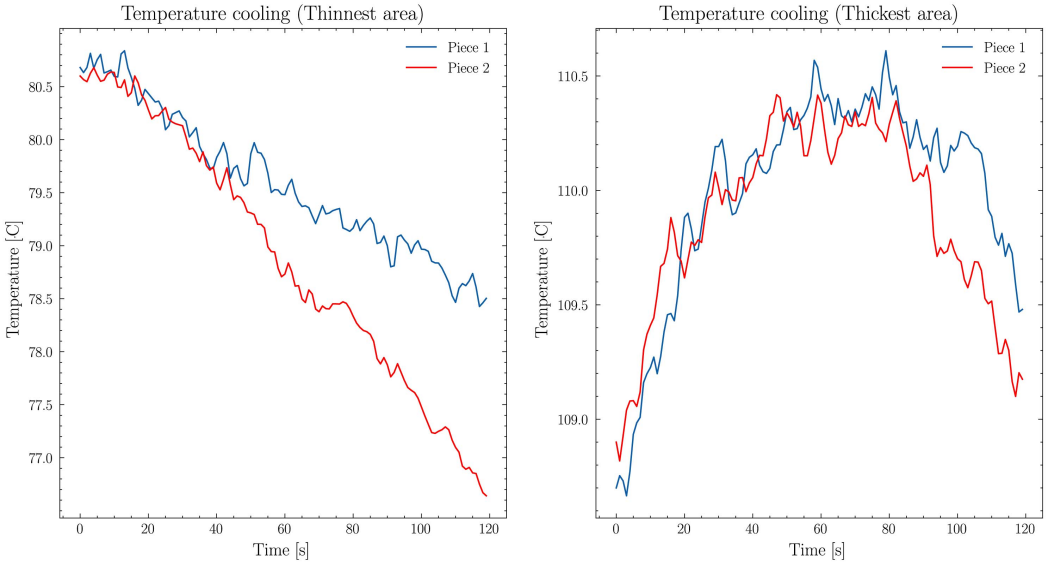

Figure 4.4: Surface temperature cooling profiles at the thinnest (left) and thickest (right) areas of the blow-moulded part

measured by thermal imaging, could be leveraged to infer the thickness of the part. In particular, we make two assumptions:

- The cooling conditions of the environment where the part is monitored are the same for all parts produced. Therefore, we consider that the variations of the temperature of the production area are negligible.
- The physical and chemical properties of the material are assumed to be constant over the entire surface of the part. In particular, the thermal and infrared transmittances of the material should be approximately constant.

Based on these assumptions, we designed three data-driven methods to model the relationship between the surface temperature variations of critical areas of the part and the corresponding thickness values. The first approach, by one-dimensional time series, exploits the cooling dynamics, one point of the part at a time, whereas the second method processes the part globally, taking into account unity of part (points belonging to the same part are processed simultaneously) and spatial information (points are positioned on the part surface).

4.4 Proposed methods

In this section, we present new approaches to predict the thickness of blow-moulded parts. We propose a non-intrusive real-time thickness inference exploiting the variations of surface temperatures over time on different areas of the blow-moulded part. Unlike traditional thickness quality control methods, our system is able to predict thickness values at critical points of the part within minutes, allowing real-time operation.

Three different approaches to model the temperature-thickness relationship are proposed:

- *Parametric temporal approach*: The Parametric temporal approach involves the use of a parametric function to approximate the pointwise temperature surface decay. The function parameters, retrieved through curve fitting, may then be used as input features for a machine learning regressor.
- *Flexible temporal approach*: The flexible temporal approach, as the parametric temporal one, takes advantage of the pixel-wise temperature decay. Instead of compressing the information through the use of a parametric function, the flexible temporal approach leverages the ability of deep learn-

ing to extract meaningful features from raw signals.

- *Spatio-temporal approach*: The spatio-temporal approach leverages not only the temporal temperature information, but also the spatial one. Instead of extracting the temperature time series for each critical point of the blow-moulded parts we can design an *end-to-end* deep learning architecture able to directly handle the input thermal video. In such a way, we should be able to take into account the tank unit intrinsic information which is completely lost in the previous approaches.

These three approaches are presented in more detail below.

4.4.1 Parametric temporal approach

The first approach consists of three phases: time series extraction, time series approximation by parametric curve fitting and thickness prediction using the parameters of the approximated temperature surface decay.

Time series extraction: The time series extraction phase aims to process the input thermal video in order to retrieve the temperature time series of a limited number of critical points, for which the thickness value is known. These time series constitute the input data of our data-driven model. Given K critical points of each part for which the thickness values are known, and given the thermal video X_i, we are interested in extracting the time series $x_{ik} = (X_i(\xi_k, \zeta_k, 1), \ldots, X_i(\xi_k, \zeta_k, T)) \in \mathbb{R}^T$, with T time steps, where $(\xi_k, \zeta_k) \in \{1, \ldots, h\} \times \{1, \ldots, w\}$ indexes the pixel matched to key point $k \in \{1, \ldots, K\}$. For a set of n input thermal videos, this first phase produces a dataset

$$D = \left\{ \{x_{ik}, y_{ik}\}_{k=1}^{K} \right\}_{i=1}^{n} \tag{4.1}$$

of $K \times n$ pairs (x_{ik}, y_{ik}) where $x_{ik} \in \mathbb{R}^T$ is the time series corresponding to key point k in thermal video i and $y_{ik} = Y_i(\xi_k, \zeta_k) \in \mathbb{R}$ is the corresponding thickness value.

Time series approximation: In order to reduce the number of input features we approximate the surface-temperature decay time series by a parametric expansion, to compress the temporal information into a limited number of new features, corresponding to the parameters of the function. Temperature-surface time series have a fairly simple shape (Figure 4.4). The predominantly parabolic shape of the time series lends itself well to be approximated with simple functions, with few parameters. We compared the following three parametric models:

- Power Law: $y = ax^b$,
- Polynomial (2nd degree): $y = ax^2 + bx + c$,
- Logarithmic: $y = a + b\log(x)$.

In such a way, the different cooling behaviour, observable in different areas of the part could be expressed through a limited number of new parameters. Moreover, the functional approximation smooths the time signal, thereby filtering the measurement noise. The new features may be then used as the input data for a regression model. Mathematically speaking, the time series approximation produces a new dataset,

$$D = \left\{ \{z_{ik}, y_{ik}\}_{k=1}^{K} \right\}_{i=1}^{n} \quad , \tag{4.2}$$

composed of $K \times n$ pairs (z_{ik}, y_{ik}) collection pairs where $z_{ik} \in \mathbb{R}^P$ are the P parameters obtained by fitting the parametric function to the time series x_{ik}.

Time series regression: Any machine learning method can then be applied to predict the thickness value of a given key point based on the parameters computed by fitting the surface-temperature time series. The role of the machine learning algorithm is to estimate a function $\hat{g}$ such that

$$\hat{g} = \underset{g \in \mathcal{G}}{\operatorname{argmin}} \sum_{i=1}^{n} \sum_{k=1}^{K} (y_{ik} - g(z_{ik}))^2 \quad . \tag{4.3}$$

4.4.2 Flexible temporal approach

As illustrated in Figure 4.6, the proposed method consists of two phases: extraction of the time series and thickness value regression through a recurrent neural network (RNN). The time series extraction is carried out exactly in the same way as for the parametric temporal approach (see Section 4.4.1).

Time series regression: Given the extracted time series and the corresponding thickness values, our problem can be formulated as a supervised machine learning problem, more specifically, as a univariate time series regression problem. With univariate time series regression we mean the task of predicting the real number value of the dependent variable, the thickness, given a single dependent variable corresponding to a sequence of discrete-time data. Formally, we look for function $\hat{g}$ such that

$$\hat{g} = \underset{g \in \mathcal{G}}{\operatorname{argmin}} \sum_{i=1}^{n} \sum_{k=1}^{K} (y_{ik} - g(x_{ik}))^2 \quad . \tag{4.4}$$

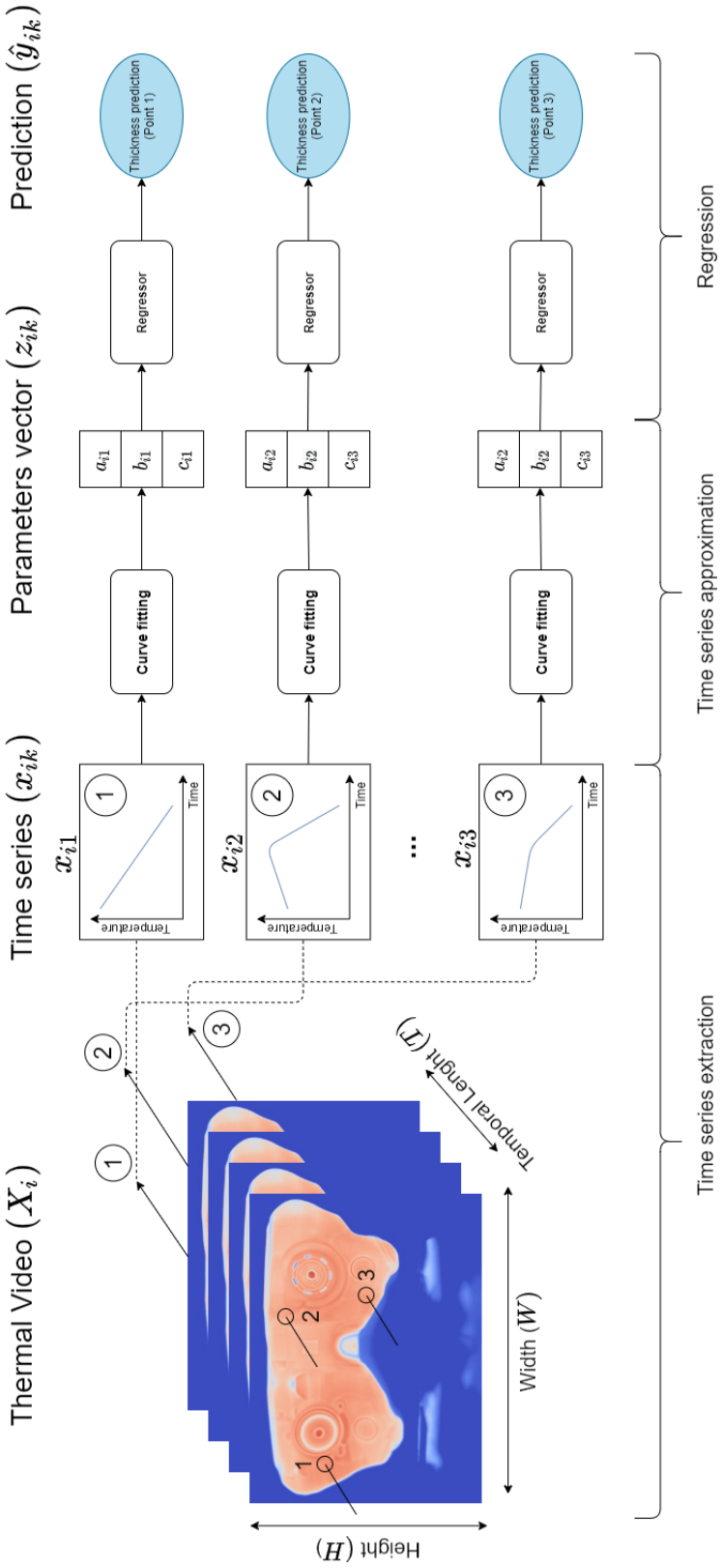

Figure 4.5: Parametric temporal approach

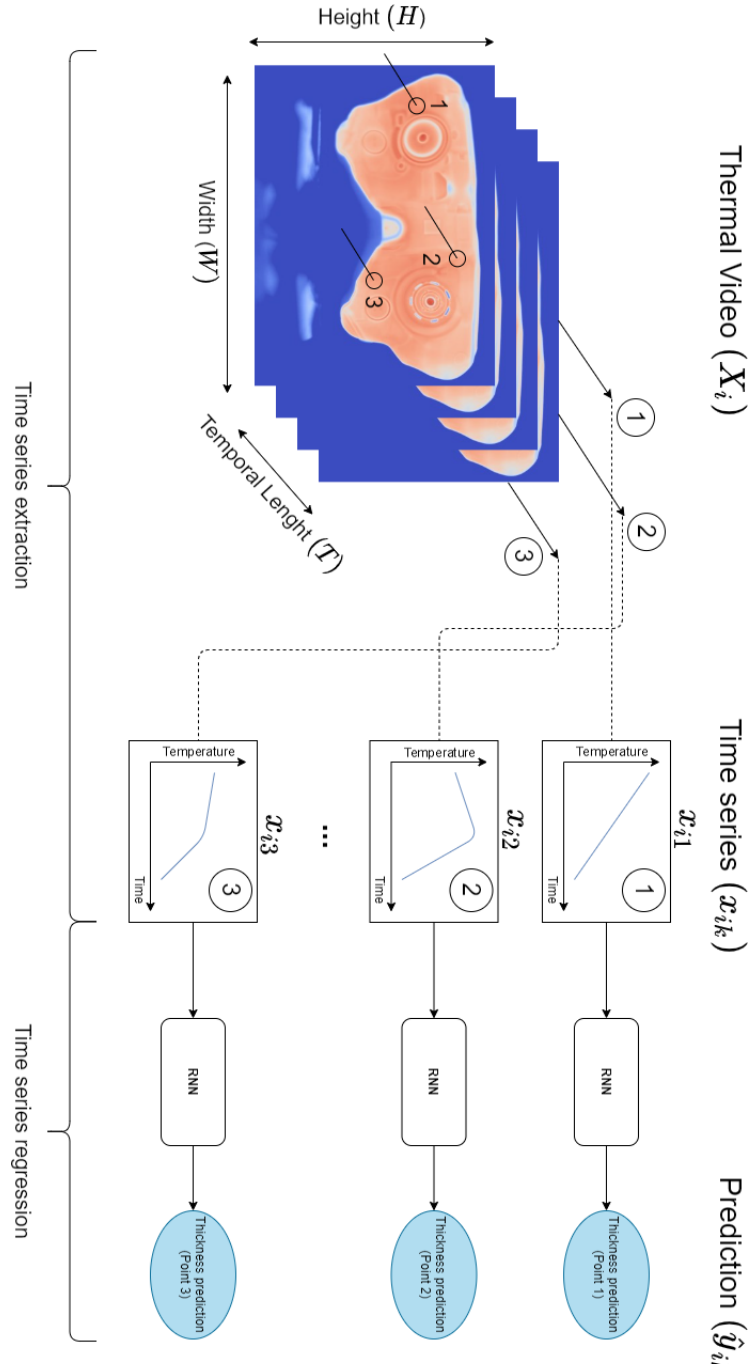

Figure 4.6: Flexible temporal approach

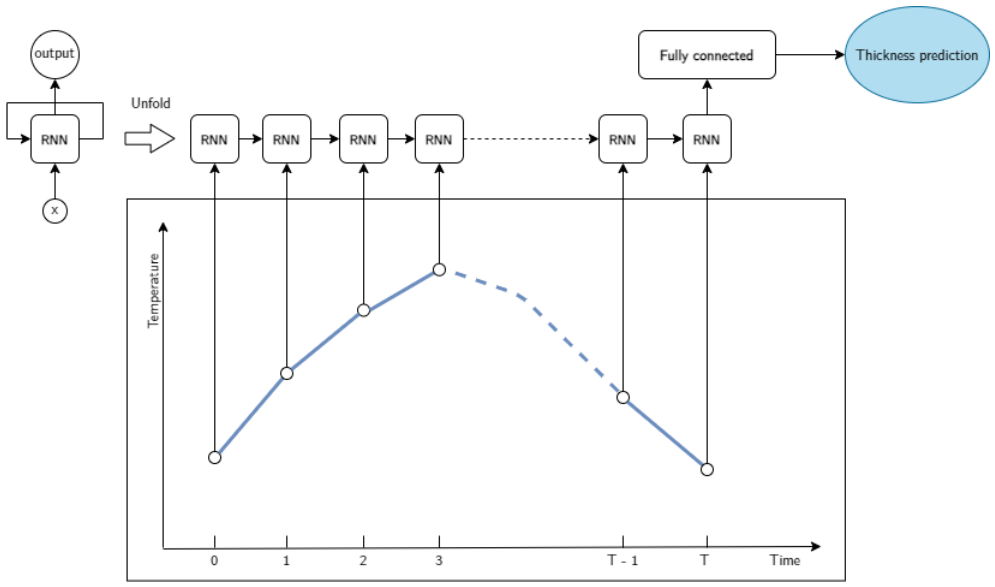

Figure 4.7: RNN-based model

As our hypothesis is that temporal information plays a key role in thickness discrimination, we chose to use a recurrent neural network (see Section 2.4.1.6) to model the dependency. Inspired by the results achieved by RNNs in the domain of sequential data, we designed a simple RNN to address our problem (Figure 4.7). The main idea behind our pipeline is to sequentially process the temperature at each time step t by taking into account information from prior inputs to predict the current input and output. The last output computed by the RNN model is then passed through a fully connected layer to produce a scalar output aiming at approximating the thickness of the part area for which the time series has been extracted.

4.4.3 Spatio-temporal approach

Compared to the previous approaches, the method proposed in this section aims to leverage not only the temporal temperature information, but also the intrinsic information of the tank unit. Instead of extracting the temperature time series for each critical point of the blow-moulded parts we design an *end-to-end* deep learning architecture capable of directly processing thermal video as input data. In this setup, the dataset

$$D = \{(X_i, Y_i)\}_{i=1}^{n}, \tag{4.5}$$

is a collection of pairs (X_i, Y_i) where $X_i \in \mathbb{R}^{h \times w \times T}$ is the input thermal video made of T frames of height h and width w, and $Y_i \in \mathbb{R}^{h \times w}$ is the corresponding

thickness image, that is, the 2D array of thicknesses on each of the input pixels. The role of the spatio-temporal approach is to estimate a function $\hat{g}$ such that

$$\hat{g} = \underset{g \in \mathcal{G}}{\operatorname{argmin}} \sum_{i=1}^{n} \left(Y_i - g(X_i)\right)^2 \quad , \tag{4.6}$$

or more precisely such that

$$\hat{g} = \underset{g \in \mathcal{G}}{\operatorname{argmin}} \sum_{i=1}^{n} \sum_{k=1}^{K} \left(Y_i(\xi_k, \zeta_k) - g(X_i(\xi_k, \zeta_k)\right)^2 \quad , \tag{4.7}$$

where $(\xi_k, \zeta_k) \in \{1, \ldots, h\} \times \{1, \ldots, w\}$ indexes the pixel matched to the key point k.

The work presented in this paragraph is inspired by the computer vision domain of *semantic segmentation*, whose objective is to classify the pixels of an image that belong to the same object class. Basically for an input image, semantic segmentation produces a segmentation mask which has the same spatial dimension of the input image where each pixel value correspond to the predicted class. As for other computer vision tasks, state-of-the-art results are obtained by convolutional neural networks (see Section 2.4.1.5). Here, instead of predicting a segmentation mask per image and per class, we propose to use a encoder-decoder architecture to learn thicknesses from a thermal video sequence. As before, thicknesses are only measured on the key points of the part. The end-to-end pipeline that infers the thicknesses of the key points from the thermal video sequence is depicted in Figure 4.8. The proposed pipeline is largely inspired by the *Unet* (see Section 2.4.1.5), and consists of three main blocks: the spatial encoder, the temporal encoder and the spatial decoder.

- *Spatial Encoder:* The spatial encoder is a CNN which aims to extract the spatial feature map for each frame of the video. We use the encoder of a traditional encoder-decoder architecture. Compared to the original *Unet* architecture, we replace the original contracting path proposed in the paper with a Residual Network (ResNet, see Section 2.4.1.5). The ResNet architecture is composed of five main building blocks. Each block sequentially produces higher-level and lower-resolution features which will later be used by the decoder to predict thicknesses. The general input thermal video is a four-dimensional tensor of shape (h, w, c, t) where h, w, c and t are respectively the height of the image, the width, the number of channels and the number of frames of the video sequence. The spatial encoder processes each frame (h, w, c) of the input thermal video and returns a feature

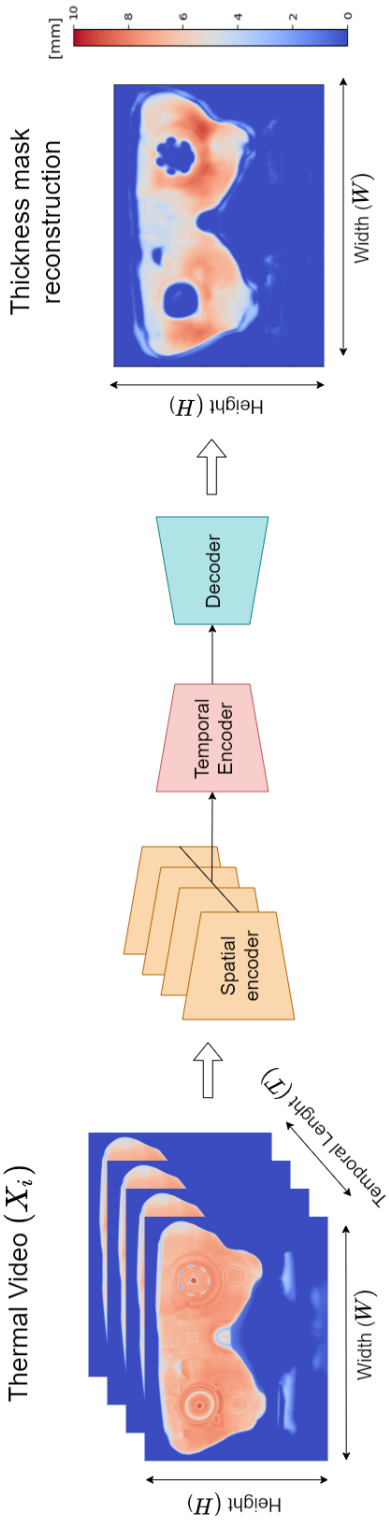

Figure 4.8: Spatio-temporal architecture

map of size (h', w', c', t) where $h' << h$, $w' << w$ and $c' >> c$ and a set of intermediate feature maps.

- *Temporal Encoder:* The temporal encoder aims to encode the temporal information of the spatial feature maps produced by the spatial encoder, which encodes each frame independently, without leveraging any kind of temporal information. In order to process the temporal information, we use a 3D convolutional layer with a kernel size of $(1 \times 1 \times 1)$, stacked right after the spatial encoder, which produces a linear projection of the stack of frame-wise feature maps. This projection acts as a dimensional reduction along the temporal axis. In such a way we are able to produce a new feature map which encodes both the spatial and the temporal information. Given the spatial feature map of size (h', w', c', t), the temporal decoder compress the temporal dimension and produces a new *spatio-temporal feature map* with size $(h', w', c', 1)$. The same convolutional operation, with the same parameters, is applied to compress the temporal information of all the intermediate features maps produced by the spatial encoder.

- *Decoder:* The decoder projects the discriminant spatio-temporal feature map back to the original input spatial dimension. In the same way as for the *Unet* architecture, intermediate high-level low-resolution features from the encoder path are combined and reused with the upsampled output of each decoder block to help the model reconstructing the prediction mask. The decoder is composed on five decoder blocks and each decoder block applies an upsample operation using the nearest neighbour algorithm followed by two 2D convolutional layers with a kernel size of 3×3, batch normalisation and *ReLu* activation function. A final convolutional layer produces the thickness map from which the prediction of the thickness of the critical points is extracted. Given the encoded pattern of size $(h', w', c', 1)$ the decoder projects the encoded pattern to the original spatial dimension producing a 1-channel map reconstruction $(h, w, 1)$. Given the pixel coordinates of the n critical points, their corresponding thickness predictions can be easily retrieved.

Training such an architecture is more challenging compared to that of the flexible temporal approach because of the number of learnable parameters. Furthermore, we have two main shortcomings:

- The size of the training sample is low compared to what is normally used to train this kind of network. For semantic segmentation tasks, a dataset

of 1000 images is considered small and we could hardly have more than a few hundred input samples.

- The thickness ground-truth map is not known for all the points of the tanks. In the training phase, semantic segmentation requires the output mask of the input data. For our problem, we do not have access to the full thickness map of the tanks, but only to the thickness of key points.

The first shortcoming may be alleviated by applying two different machine learning techniques: *transfer learning* (see Section 2.4.3.3) and *data augmentation*. Instead of training a model from scratch, a network with pre-trained parameters may be used as a starting point. Most of the state-of-the-art semantic segmentation approaches make use of pre-trained encoders to start with, while the remaining part of the network is trained from scratch. However, we deal with thermal images. Unlike traditional colour images, which are commonly represented using a 3-channel matrix (RGB) with 256 different integer values, thermal images are 1-channel matrices where each pixel takes a continuous value corresponding to a temperature in degrees Celsius. Because pre-trained networks have been trained on RGB images composed of unsigned integers ranging from 0 to 255, to leverage transfer learning is necessary to map the continuous temperatures values to the 0-255 range.

Data augmentation is a second option to deal with small datasets. Data augmentation in data analysis is a set of techniques used to increase the amount of data by adding slightly modified copies of already existing data or newly created synthetic data from existing data. Popular image augmentation techniques involves geometric transformations such as image flipping, rotation or translation, or colour space transformations. More advanced techniques make use of generative models to create artificial training sample. Regarding our problem, the only augmentation techniques that can be readily applied are the ones that involve geometric transformations.

As image colour is temperature related, any augmentation techniques that alter the colour space risk corrupting the input data. While the lack of a large volume of input data can be handled by transfer learning and data augmentation, the lack of labelled data in the training dataset can be a little more challenging. In fact, since the full thickness map of the part is not available, we cannot train the model in the traditional pixel-wise loss between the ground truth map and its reconstruction computed by the network. The computed loss is then back-propagated in the network to adjust the learnable parameters of the architecture.

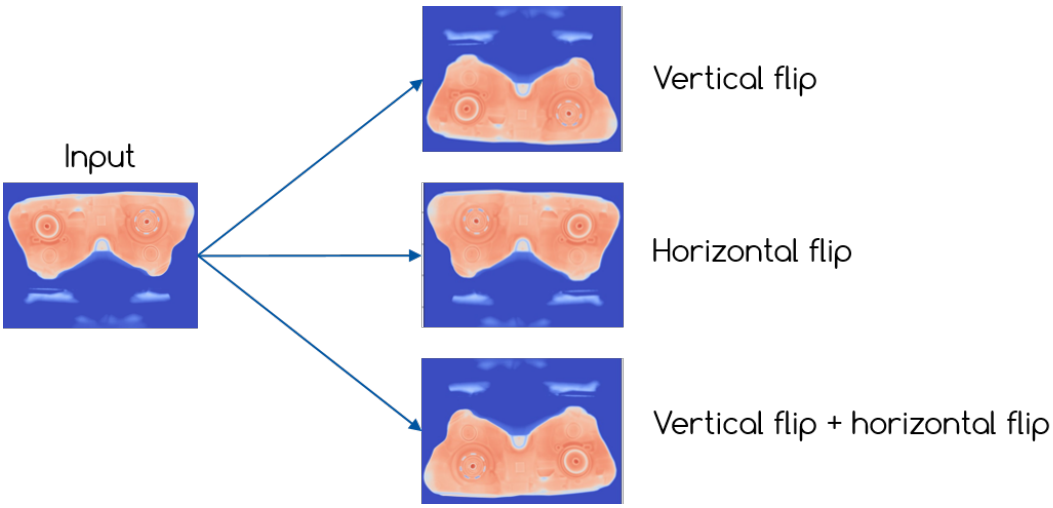

Figure 4.9: Image augmentation

In our particular case, the full thickness ground truth of the part is not known, which means that we cannot compute the pixel-wise loss everywhere. However, it is possible to train the network by computing the pixel-wise loss between the map and the reconstruction for the pixels whose thickness is known.

4.5 Experimental validation

This section presents the results from a real-world implementation in an industrial setting. First, the empirical context is described as well as the data processing and the training pipeline. Finally, we provide the results of this first experimentation.

4.5.1 Data collection

To evaluate the performance of the proposed data-driven model-based quality control, a data acquisition campaign in an industrial environment has been organised. In order to train our models to learn to infer the thickness, given the cooling profile, we need two different types of data: the temperature of the part surface and the thickness measurement of the corresponding part. The temperature data has been acquired with an industrial thermal camera OPTRIS PI 640 with a good resolution (640×480), previously calibrated to correctly measure the temperature of the plastic material. Unlike traditional colour images, which are commonly represented using a 3-channel matrix (RGB) with 256 different integer values, thermal images collected through the OPTRIS 640 PI are 1-channel

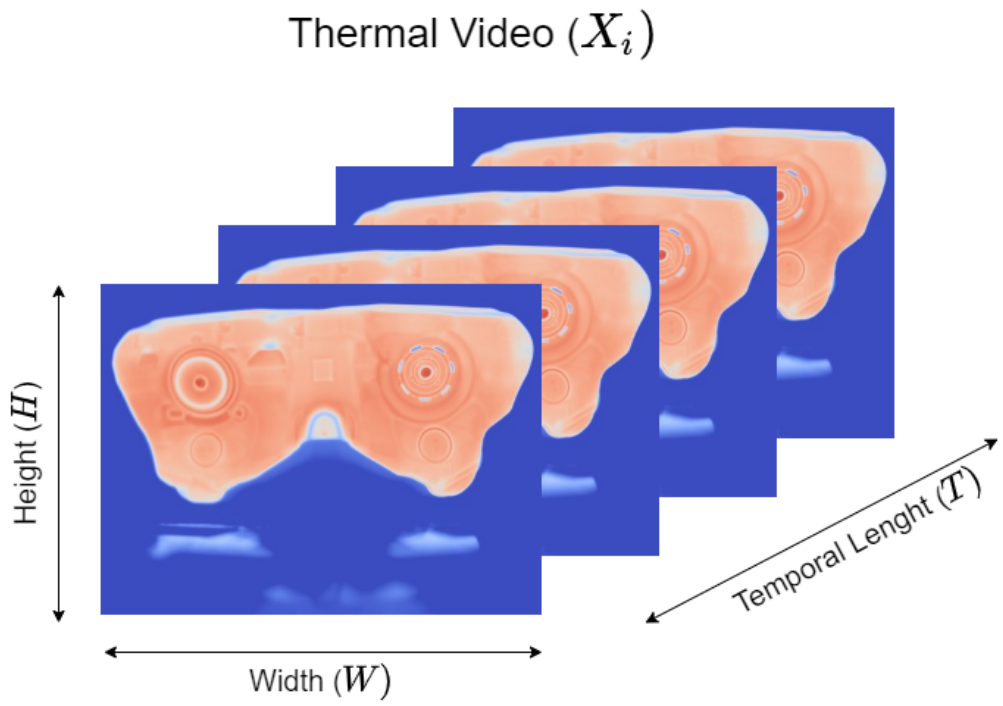

Figure 4.10: Thermal video X_i

matrices where each pixel takes a continuous value corresponding to a temperature in degrees Celsius. Unlike other temperature measuring instruments, the thermal camera has the advantage to evaluate, in a single shot, the temperature of the whole visible surface. Several consecutive shots of the same part make it possible to follow the evolution of the temperature over time (Figure 4.10).

In order to gather comparable data, the image acquisition of each part produced needs to be synchronised with the blowing process in order to synchronise the acquisition of images on the same relative time. Repeatable mechanical movements of the machine have been used to "trigger" data acquisition. The trigger starts a one minute countdown after which the acquisition of thermal images begins. This time is mandatory to give the machine the time to unload the part and the machine operator time to place the part in the measuring area. For our experimental measurement campaign, we collected thermal videos on 50 different parts. For each part we recorded 120 consecutive thermal images equally spaced with a one second interval between images, for a total of two minutes of data acquisition.

The thickness measurement has been carried out on key points through the use of an ultrasonic measurement instrument routinely used for sampling control.

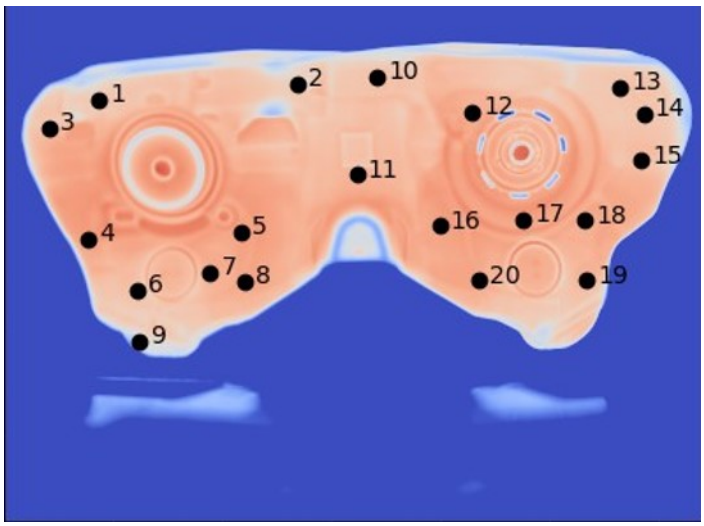

Figure 4.11: Critical points distribution on the tank surface

The thicknesses of the part selected for our experimentation have typical values ranging between 3 and 10 millimetres. For each thermal video, and therefore for each part, we measured 20 critical thickness points evenly distributed on the tank surface (Figure 4.11).

4.5.2 Data processing

Once data have been collected, some processing operations are needed to prepare the data for training. Raw thermal frames are transformed in RGB images using a colormap which maps low temperature value to blue colours and high temperature value to red colours. In order to have comparable colours among the different frames, the colormap is computed on values ranging from 40°C (dark blue) and 135°C (bright red). Moreover, we need to associate the 20 key points whose thickness is measured to their corresponding pixel coordinates on the thermal video. This was done manually, each measured point of the part is identified on a thermal image by its (ξ_k, ζ_k) pixel coordinate. Hopefully, since the position of the part does not change during the image acquisition, it is sufficient to find the pixel coordinates for one frame per sequence.

Although the part does not move during the acquisition, each part is not precisely positioned with respect to the field of view of the thermal camera. The (ξ_k, ζ_k) coordinates on different thermal videos may therefore refer to different surface areas of the part. In order to align pixel coordinates along parts, a transformation is applied, to each frame, to project the part onto a reference

position. Before describing the transformation in more detail, we present the *pinhole camera model*.

The pinhole camera model The pinhole camera model describes the mathematical relationship between the coordinates of a point in a three-dimensional space and its projection onto a two-dimensional pixel coordinate system. This mathematical relationship depends on the extrinsic and intrinsic parameters of the camera. The extrinsic parameters refer to the rotation matrix and translation vector of the camera coordinate system with regard to the world coordinate system. Given a point $(x, y, z, 1)^T$ in the world coordinate system, we can form its pixel coordinates $(\xi, \zeta, 1)^T$ as follows:

$$\begin{bmatrix} \xi \\ \zeta \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} , \tag{4.8}$$

That is:

$$\begin{bmatrix} \xi \\ \zeta \\ 1 \end{bmatrix} = K[R|T] \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} , \tag{4.9}$$

where:

- K is the 3×3 intrinsic camera matrix.
- f is the focal length.
- (c_x, c_y) are the coordinates of the principal point at the center of the image plane.
- R is the 3×3 rotation matrix.
- T is the translation vector.

Given this mathematical description, a homography H can be defined as a transformation matrix that maps points from one plane to another. It can be

derived by the equation 4.8:

$$\begin{bmatrix} \xi \\ \zeta \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix} \begin{bmatrix} \xi' \\ \zeta' \\ 0 \\ 1 \end{bmatrix} \tag{4.10}$$

$$\begin{bmatrix} \xi \\ \zeta \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & t_x \\ r_{21} & r_{22} & t_y \\ r_{31} & r_{32} & t_z \end{bmatrix} \begin{bmatrix} \xi' \\ \zeta' \\ 1 \end{bmatrix} \tag{4.11}$$

$$\begin{bmatrix} \xi \\ \zeta \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} \xi' \\ \zeta' \\ 1 \end{bmatrix} \tag{4.12}$$

$$P = HP' \tag{4.13}$$

where:

- P and P' are two points on different planes.
- H is a 3×3 matrix composed by intrinsic and extrinsic parameters that relate the two planes together.

The mathematical framework presented above can be applied to project all the thermal frames composing the 50 thermal video collected over a reference image. By estimating the homography between the two planes, that of the reference image and that of the image to be projected, each point P of the input image can be projected to the reference image.

The first frame of each thermal video sequence is used to estimate the homography matrix, through the use of the ORB (Oriented FAST and Rotated BRIEF) algorithm (Rublee et al., 2011). In a nutshell, ORB is a feature matching algorithm that allows to automatically find some key points in an image. The ORB algorithm takes as inputs two images, a reference and the one we want to project on this reference, and features that can be matched in the two images are used

Table 4.1: Curve fitting reconstruction error

Parametric function	**RMSE (train)**
Power Law	0.29
Logarithmic	0.27
Polynomial (2nd degree)	0.03

to estimate the homography matrix. Once the homography is computed, it is applied to map all the pixels of the second image to the pixel of the reference image. The same transformation can then be applied to all the frames in the video sequence, provided that both the camera and the tank are motionless. Then, depending on the used method, extra processing is needed to prepare data for training.

4.5.2.1 Parametric temporal approach

The parametric temporal approach makes use of the temperature time series extracted from the thermal video sequences to infer the corresponding thickness. Given the coordinates (ξ_k, ζ_k) of the 20 critical thickness values, it is simple to retrieve the temperature time series by simply collecting the temperature of the coordinate for each sequential frame of the thermal video. By extracting all temperature time series for each critical point of the 50 parts considered, we can build a new time series dataset composed of 1000 (50 x 20) time series and the corresponding thickness values. The time series are then approximated by their parametric expansion. As explained in Section 4.4.1, three different parametric functions have been identified as good candidates for approximating the pixel-wise temperature surface decay. In order to identify the parametric expansion that best fits the input time series, each expansion was applied to the time-series of the training set. The root mean square error (RMSE) of the overall reconstruction is used to select the best expansion.

Table 4.1 shows the average RMSE reconstruction for the three parametric expansions considered. The polynomial expansion, minimises the reconstruction error and is thus used to summarise the raw time series. For each time series, the three parameters defining the second-order polynomial approximation constitute the predictor of the machine learning model.

4.5.2.2 Flexible temporal approach

As for the parametric temporal approach, the flexible temporal approach makes use of the temperature time series extracted from the thermal video sequences to infer the corresponding thickness. Given the coordinates (ξ_k, ζ_k) of the 20 critical thickness values, it is simple to retrieve the temperature time series by simply collecting the temperature of the coordinate for each sequential frame of the thermal video. Unlike the previous approach, no extra data processing is required because the raw time series constitute the input data of the RNN model.

4.5.2.3 Spatio-temporal approach

Compared to the previous approaches, the spatio-temporal approach does not require to rearrange the input thermal video in another format, but some processing operations are still needed to allow the usage of a pre-trained spatial encoder. In order to be consistent with the data of the *ImageNet* dataset (Deng et al., 2009) that was used to pre-train the network, each thermal frame is converted to a 3-channel RGB image. The maximum and minimum temperatures among all frames are retrieved and the same colormap is applied on all thermal frames of each video sequence. The colormap is the one applied to visualise the thermal frames in the current Chapter. All values are then rescaled in $[0, 1]$ and then normalised using the default mean and standard deviation value of the *ImageNet* dataset.

4.5.3 Training

In this section our training pipeline is presented. For each approach, the data was divided into three sets: the training set (data from 38 parts), the validation set (data from 8 parts) and the test set (data from 5 parts). The training is used to fit the models, the validation set is used to select the model hyper-parameters and the test set is used to evaluate the ability of the model to generalise on new data. For each proposed approach, a Bayesian optimization of hyper-parameters with the *Tree-structured Parzen Estimator* (TPE) (Bergstra et al., 2011) algorithm was used to select the hyper-parameters that minimise the *Mean Squared Error* (MSE) on the validation set.

Table 4.2: Hyper-parameter search space for the parametric temporal models

Model	**Hyper-parameter**	**Search space**
Lasso	λ	LogSpace(10^{-5}, 1)
SVM	Kernel	{Linear, Polynomial, RBF}
	C	LogSpace(10^{-6}, 1)
	Polynomial degree (Polynomial)	[2, 4]
	γ ({Polynomial, RBF})	LogSpace(10^{-3}, 10^{3})
Random Forest	Number of predictors	[50, 500]
	Maximum tree depth	[4, 50]
	Minimum samples leaf	[1,60]

4.5.3.1 Parametric temporal approach

As explained previously, a second-degree polynomial is fitted over the time series to compress the input data into a few features corresponding to the coefficients of the polynomial expansion of the thermal signal. Three machine learning algorithms were compared to model the relationship between the polynomial coefficients and the corresponding thickness values: *Lasso* (linear) regression, *random forest* and *Support Vector Machine* (linear or kernelised) regression. The exhaustive list of model hyper-parameters and their search space is summarised in Table 4.2.

4.5.3.2 Flexible temporal approach

We compared the performances of three RNN-based models: a vanilla RNN, an LSTM and finally a GRU network. The number of hidden units of each computational cell, as well as the number of stacked layers are hyper-parameters that are optimised. A comprehensive summary of the search space for the hyper-parameters is available in Table 4.3.

Each model has been trained to minimise the mean squared error metric using the *Adam* optimiser (Kingma and Ba, 2015) with $\beta_1 = 0.9$, $\beta_2 = 0.98$ and $\epsilon = 10^{-9}$ (default values) and a learning rate sampled by the TPE algorithm on a logarithmic scale, from 10^{-7} to 10^{-2}.

Table 4.3: Hyper-parameter search space for the flexible temporal models

Hyper-parameter	**Search space**
RNN Cell type	{RNN, LSTM, GRU}
N°hidden unit	{8, 16, 32}
N°stacked layers	{1, 2}

4.5.3.3 Spatio-temporal approach

As for the previous approach, we compared different model configurations on two pre-trained networks: a ResNet18 (18 layers) and a ResNet34 (34 layers). Deeper architectures exist, but we made the choice to limit the search space to the two smaller ResNet architectures because of the limited number of samples in our training set. Deeper architectures would allow to extract richer data representations, but at the expense of more parameters and increased computation time. We believe that ResNet18 and ResNet34 are powerful enough to produce a relevant representation of thermal images. Another hyper-parameter of the presented architecture is the number of encoder/decoder blocks. As stated in Section 2.4.1.5, all ResNet architectures, independently on their depths, have 5 main building blocks. Actually, it is possible to slightly change the architecture in order to stop the encoding computation before the last block. This allows to reduce the complexity of the architecture and thus prevent possible over-fitting problems. For instance, we could take into account only the first 3 convolutional building blocks of the ResNet in such a way that the output of the third convolutional building block would be the spatial feature map and the output of the first and second blocks would be the intermediate encoded features. Since the encoder and decoder are symmetrical, reducing the number of encoder blocks also implies a reduction in the number of decoder blocks.

Table 4.4: Hyper-parameter search space for the spatio-temporal models

Hyper-parameter	**Search space**
Encoder	{ResNet18, ResNet34}
N°of blocks	{3, 4, 5}

As for the flexible temporal approach, each model configuration has been trained to minimise the MSE metric using *Adam* optimiser with $\beta_1 = 0.9$, $\beta_2 =$

0.98 and $\epsilon = 10^{-9}$ and a learning rate sampled by the TPE algorithm on a logarithmic scale from 10^{-10} to 10^{-2}.

4.5.4 Results

In this section the results obtained with the three approaches are presented and compared. Performances are measured using the Root Mean Squared Error (RMSE), which has the benefit of penalising large errors and whose value is easy to interpret because it has the same unit as the dependent variable: all the scores presented below represent the average thickness reconstruction error in millimetres.

For the parametric temporal approach, random forest performs better in both fit on the training set and generalisation on the test set (Table 4.5). The error

Table 4.5: RMSE for the parametric temporal models

Model	**Train**	**Validation**	**Test**
Lasso	0.95	0.91	0.89
Support Vector Regressor	0.80	0.73	0.76
Random Forest Regressor	0.19	0.54	0.48

is about 0.5 mm, which is not accurate enough from the industrial point of view. However, this first experiment allowed us to demonstrate that there is a relationship between the temperature evolution of a zone of the part and the corresponding thickness, as assumed in Section 4.3.

Figure 4.12 shows the scatter plots of the predicted thickness and ground-truth thickness for the training and test sets. The plots confirm the ability of the model to distinguish between large and small thicknesses. Moreover, they show that the model is more accurate for thin points (close to 3 mm), which is positive because the thinnest points are also the most critical for the part to meet the customer's specifications.

Among all flexible temporal trained model, the configuration that minimises the validation error is a GRU model with 8 hidden units per cell and one layer. The results obtained are summarised in Table 4.6. These results are similar to those obtained with random forest, but random forest has a slightly lower error than GRU. Moreover, the computation time needed to train the random forest

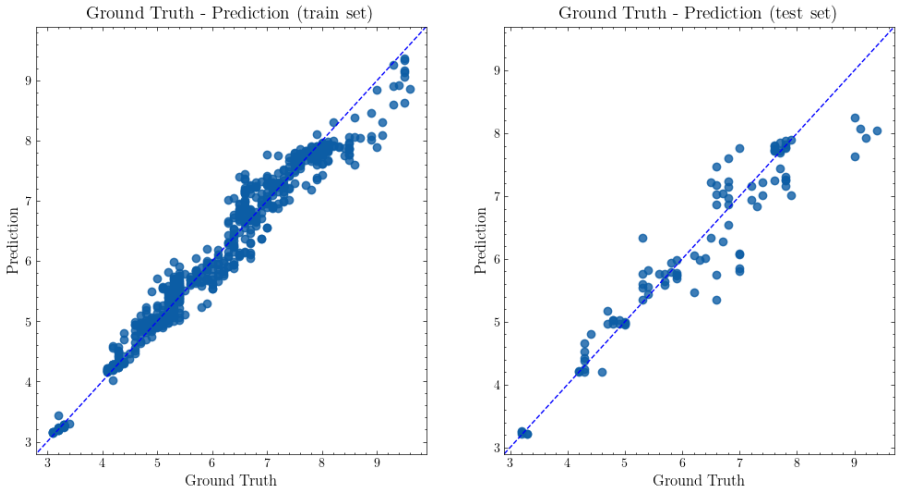

Figure 4.12: Prediction *versus* ground truth scatter plots in train (left) and test (right) for the parametric temporal approach based on random forest regression

Table 4.6: RMSE for the flexible temporal models

Model	**Train**	**Validation**	**Test**
RNN	0.55	0.58	0.58
LSTM	0.54	0.57	0.56
GRU	0.54	0.56	0.54

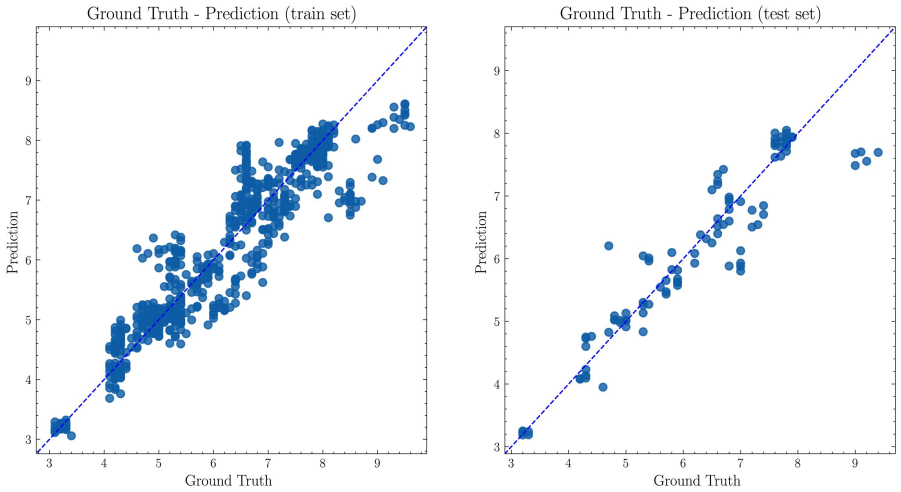

Figure 4.13: Prediction *versus* ground truth scatter plots in train (left) and test (right) for the flexible temporal approach based on GRU

Table 4.7: RMSE for the spatio-temporal model

Model	**Train**	**Validation**	**Test**
ResNet34-5 blocks	**0.14**	**0.18**	**0.16**

is significantly lower and it does not rely on dedicated hardware (GPU). The parametric temporal approach therefore seems preferable in all respects. This is confirmed in Figure 4.13, where the parametric temporal approach outperforms the flexible temporal approach over all thickness ranges.

The third approach (Table 4.7) completely outperforms the previous ones. The best results are obtained using a pre-trained ResNet34 encoder with 5 encoder-decoder blocks, which achieves an RMSE of 0.16 mm on the test set, which is about one-third of the error of the previous approaches. A precision of 0.2mm is considered extremely interesting in our industrial context. Figure 4.14 shows that the greatest benefits over the parametric temporal approach are at higher thicknesses, but the improvement is already visible at only 4 mm.

4.5.5 Model performance on unseen data point

The results presented in the previous section show the ability of the spatio-temporal model to correctly estimate the thicknesses at the critical points of the part. The question is whether the model is able to generalise the relationship

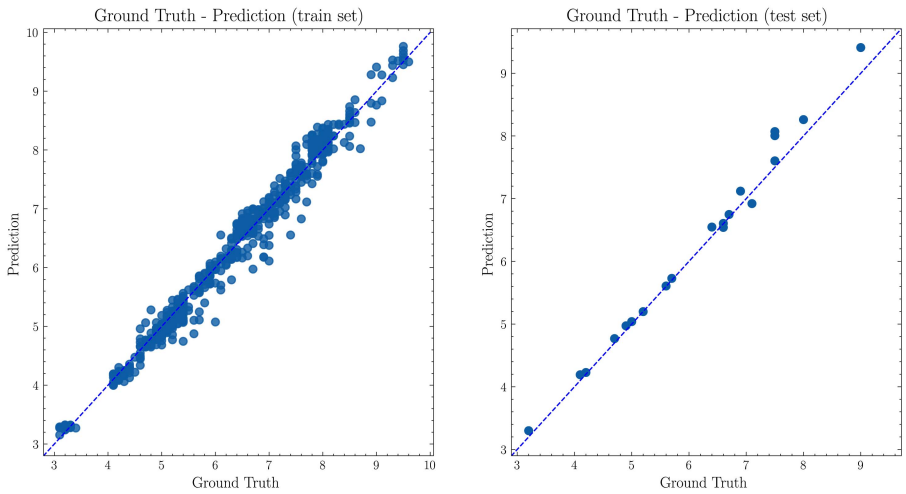

Figure 4.14: Prediction *versus* ground truth scatter plots in train (left) and test (right) for the spatio-temporal approach based on ResNet34

learned at a limited set of critical points to the entire surface of the part. The spatio-temporal architecture computes a thickness map for the entire surface of the tank. An example of the thickness map produced in this way is shown in Figure 4.15 for a part in the test set. In this figure, the colour represents the predicted thickness, not the tank temperature. The figure highlights the ability of the model to produce a thickness reconstruction beyond the critical points, but some areas, especially those close to the edge of the tank, are more difficult to predict. The prediction of thicknesses outside the critical points is unreliable.

We modified the evaluation strategy to test accurately the accuracy of predictions outside of the measured critical points. Since the entire thickness map is not available, we used only a subset of the 20 critical points to train the model, reserving the others to compute an out-of-critical-point prediction error. We removed 4 out of 20 critical points of the training set to adjust the model on the remaining points. The 4 points removed are then used to evaluate the ability of the model to predict the thickness outside critical points on test parts. This was repeated 20 times by randomly selecting the 4 removed points, to evaluate the results on a larger set, composed of 4×20 samples. The results are presented in Figure 4.16, which shows a positive correlation between ground truth and prediction. However, the prediction of the thickness of unseen points is nowhere near as good as the prediction on these critical points, and our approach does improve on the manual inspection in this respect: our model is reliable on the

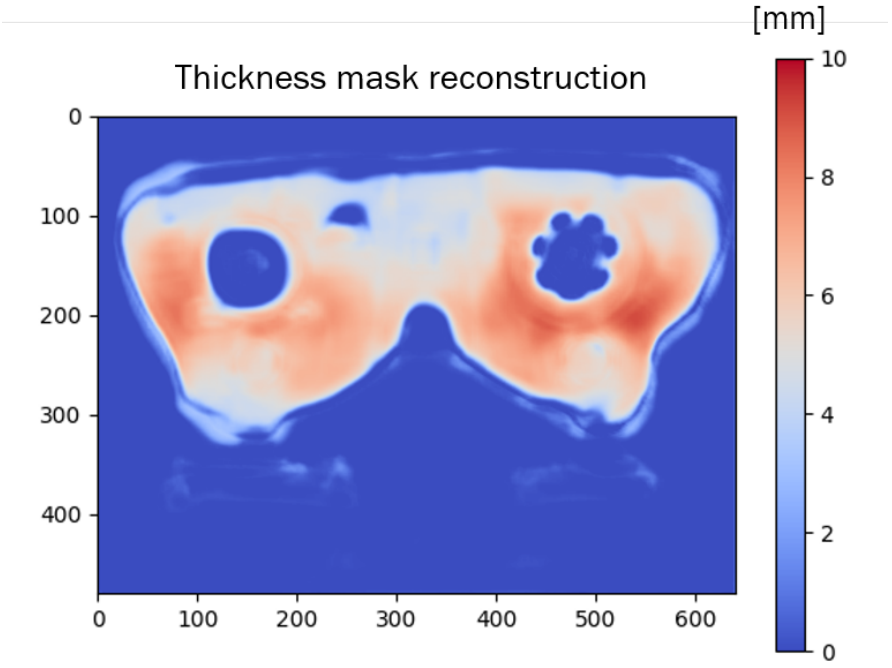

Figure 4.15: Thickness mask reconstruction example

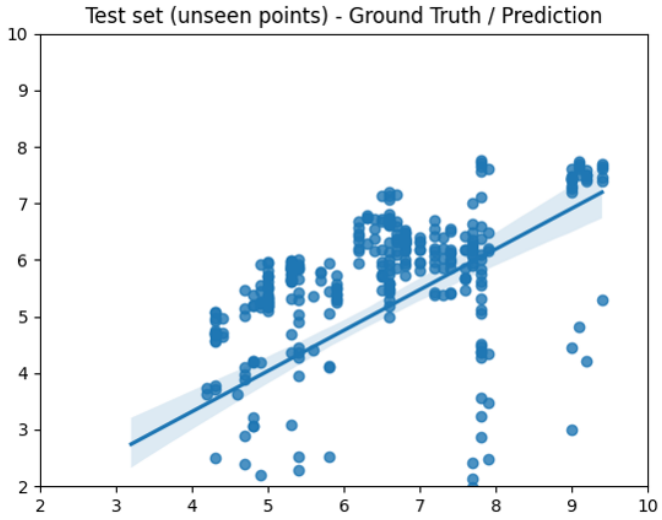

Figure 4.16: Prediction *versus* ground truth scatter plot for the spatio-temporal approach based on ResNet34 on critical points not seen in training

critical points usually measured but fails to predict the thickness of the entire part.

4.6 Conclusion

Traditional quality control methods for measuring blow-moulded part thickness involves time-consuming operations that cannot be applied online for a 100% quality inspection on all parts. This chapter proposes a new data-driven approach for measuring in real-time the part thickness by leveraging the surface-temperature decay over time. Three different pipelines have been designed to model the relationship between the cooling behaviour of a part area, captured through the use of a thermal camera, and the corresponding thickness. The procedure was validated in a real-world manufacturing setting. These results have shown the ability of our method to provide accurate results in predicting thickness values of a set of critical points. Among the three pipelines presented above, the spatio-temporal approach is the one that achieve the best performance in reconstructing the thickness values of the test set data. Future research works aim to generalise predictions outside of the critical points, in order to reconstruct the full thickness map of the whole visible part surface.

4.6.1 Scientific contribution

Thermography for measuring thickness is not a new idea, however, to our knowledge, this is the first time that thermography has been proven to be effective at measuring the thickness of a part in an industrial environment. Certainly the first time it has been used to infer the thickness of a blow-moulded part. By leveraging the surface-decay temperature due to the part cooling at the room temperature, we can infer the thickness of a tank in a non-contact manner.

Compared to previous proposed approaches using thermography, which make use of physics-based approaches to compute thickness given the surface decay temperature, we have decided to leverage the ability of data-drive methods to discover hidden patterns within data. The idea of using a data-driven approach is dictated by the following two motivations:

- The tank is composed of multiple layers of different plastic material which complicates physical modelling. The physic model should take into consideration the different physical properties of each layer constituting the thickness of the fuel tank.

- The thermal inertia of the material, which causes an initial rise in temperature in the first few seconds after the tank has been blow-moulded, have to be taken into account. In our opinion, it is simpler to model this phenomenon through a data-driven approach.

Transfer learning has been proven to be effective in a context where the number of data was very limited. Although the proposed method was validated in a single case study, we claim it should generalise to other industrial contexts. The presented empirical setting was designed to respond to a specific need of a manufacturer, but we think it should apply to others manufacturer dealing with blow-moulding and others manufacturing process involving plastic processing. Whenever it is possible to observe a different cooling behaviour between different areas of the part surface, our approach may be potentially applied.

4.6.2 Industrial contribution

From the industrial point of view, this chapter introduces a new idea of quality control. The traditional Acceptance Sampling approach, used in the industrial context studied, may be enhanced by introducing data-driven model quality control. In this chapter we have shown that by adding extra sensors, or cameras, it could be possible to infer the quality of the part and therefore provide a quality status in real-time without any time-consuming or destructive measurement. This provides to the manufacturing company a dual benefit:

- It ensures a quality control on all parts produced which enable for a fast reaction to quality non-conformities. In fact, by "virtually" measuring each part, we are able to eventually discard parts for which the model has provided a "Not-OK" result, or request the quality team to carry out more in-depth tests. Model-based quality measurement may be effectively used to detect those parts that turn out to be, from a statistical point of view, outliers. In this way, instead of randomly sampling the parts to be measured by the quality operators, the model is able to suggest the parts that seem to be interesting. By discarding all non-compliant parts, this approach indirectly reduces product recalls and thus the whole series of requests to return, exchange or replace a product that has been found to be defective, and which could impair performance, harm consumers or cause legal problems for producers.

- Such an approach could be applied to reduce the real quality controls which destroy the parts or makes them unusable. In such a case, not only the data

driven model-based control would be able to provide a thorough quality control, but it would also be able to reduce the scraps which account for an overall better production performance. Of course, real part measurements cannot be completely replaced by model-based measurements. In fact, real measurements are the primary data source for training the data-driven model.

As regarding the proposed approach to infer the thickness, results are considered accurate enough to move towards the industrialisation of the proposed system. In order to industrialise the proposed system, we should be able to take into account extra information such us the temperature of the environment where the machine is located, as well as the temperature of the moulds. In fact, the surface temperature of the blow-moulded part depends on the ability of the moulds to absorb the heat. If the heat absorption capacity changes, as a result of the change in temperature of the moulds coolant, the tank surface temperature will be different and the data-driven model accuracy would drop. In the same way, the plant temperature may influence a bit the cooling behaviour of the blow-moulded part. However, the impact of the temperature of the industrial environment on the surface temperature decay should be minimal due to the insulating properties of the plastic material (PEHD).

Conclusion

The future direction of the manufacturing industry is to transform industrial processes and products, moving from the reliance on experience-based decision making towards data-centric or evidence-based decision making. It is through this process that machine learning will play a major role to advance the digitisation of traditional industrial systems. In this research work, we have demonstrated how machine learning can contribute to a continuous quality improvement process. Internally collected plant data, as well as new sources of data coming from new sensors may be leveraged through machine learning to model the relationship between input process data and output quality data. We claim that the improvement of the overall quality of a production line can be obtained by working either on process or on quality control. This, for us, definitely makes sense as the production of a part compliant with the specifications is the result of a careful work in optimising the production process, as well as the ability to quickly identify a deviation in quality of the finished part. By identifying a quality problem early on, it is possible to react faster and adjust the process, thus limiting the production of non-conforming parts.

Our dissertation is developed as follow. In Chapter 1, we have emphasised the importance of the ever-growing amount of data available in manufacturing plants. Subsequently, the industrial context of the extrusion blow-moulding is presented. Extrusion blow-moulding is a complex manufacturing process which consists of two sub-processes: the extrusion and the blow-moulding. The large number of degrees of freedom of this manufacturing process is captured by a large number of process parameters that allows to define the process status at a given instant in time. Fuel tanks produced through this manufacturing process must respect some dimensional and geometrical constraints to ensure the integrity of the part and to comply with the costumer specifications. In order to assess the conformity of the part, thickness measurement are routinely performed. Measuring thickness in real-time is a time-consuming task that cannot be done online. For that reason,

only certain parts of the entire production batch are measured. Historically, the weight of the tank is used as an alternative of the thickness to assess that the plastic shell is made of sufficient material to ensure its robustness. Compared to thickness, tank weight is easily measurable online. A literature review targeting the quality improvement domain in extrusion blow-moulding has been conducted to evaluate previous works and to define the starting point of our research work. Due to the complexity of the manufacturing process, data-driven methods has proven to be preferable compared to expertise-based methods.

Chapter 2 describes a general data-driven framework that can be used to leverage past manufacturing data to try to improve the quality of parts produced. This approach is composed of four main stages: data collection, data processing, exploratory data analysis and supervised machine learning modelling. Unsupervised learning could be used in other industrial problems, for example to detect unstable operating conditions in a production process, but for our research question, we need supervised learning to infer the relationship between input process data and part quality.

Linear models, penalised or not, tree-based methods and Support Vector Machines are well known supervised learning algorithm that lend themselves very well to be used with structured data. Deep learning, a sub-field of machine learning, however, provides the best results when input data is unstructured. For instance, when dealing with images, videos or time series, state-of-the-art result often rely on deep learning. However, deep learning architectures require a large amount of data to converge towards a stable solution. Transfer learning can partially mitigate this limitation.

In Chapter 3, we presented a first application of the proposed method to our industrial case. We chose to work on the prediction of the weight of the tank from the process parameters. A model capable of inferring the weight of the vessel could improve our understanding of the industrial process. The length of the parison, which is considered in the literature as one of the key parameters to explain the dimensional variability of the part, was not available. A system based on a camera and a deep learning algorithm for the detection of the parison of objects was implemented to estimate its length in real time. The estimated length, together with other process parameters, constitute the input to the model. The experiments revealed a number of difficulties. In particular, we were unable to find a usable predictor of the weight of the products produced from the process parameters historically monitored to control the process. Possible explanations have been discussed: the non-stationarity of our process and, above all, the lack

of data about raw material. Nonetheless, this research work has enabled us to identify some areas for improvement in the production process. The *SmartBMM* project started from the observation that most of scraps are produced in transient regimes, when the machine state is not yet stable. As a consequence, we fully automated the way the extrusion blow-moulding machines are started, in order to reduce the duration of the transient phase. The project was then extended to automate and optimise other machine phases, such as the purge phase.

Chapter 4 presents another problem, for which we have found a workable a solution. It is to infer the thickness of the part instead of relying on the tank weight. Firstly, a statistical analysis has shown that the correlation between the weight and the thicknesses of the tank is quite low. This calls into question the relevance of using the weight to decide on the conformity of the part. We have therefore proposed a method of inferring the thickness of the tank, in real time, based on thermal imaging and machine learning. Depending on their thickness, the zones of the part cool differently: cooling is faster in thin zones than in thick ones. In thicker areas, the surface temperature even starts to increase before decreasing. This phenomenon is due to the release of energy from the innermost plastic layer that has not be in direct contact with the mould surfaces. The surface temperature is easily measured by thermal imaging. Three different pipelines have been presented. The first leverages parametric expansions to compress information of the temperature time series extracted from each critical point. The parameters of the parametric expansions are then fed into a machine learning algorithm. The second pipeline directly exploits the extracted time series using a recurrent neural network. The third pipeline, which completely outperforms the two previous ones, directly processes the input video without extracting the temperature time series. Finally, we proposed some directions that appear as promising to extend the thickness reconstruction to others points outside the critical ones. In particular, we introduced an approach that could be applied in order to reconstruct the entire thickness mask of the part.

Scientific contributions

This research work presents three main scientific contributions.

We propose new industrial applications of machine learning and deep learning for quality improvement. Our work highlights not only the benefits that machine learning approaches can bring to the manufacturing production line, but also the limits of these methods in industrial contexts where the distribution of examples

varies.

We provide a new approach for measuring the thickness of hollow parts, in real-time and without contact. Our approach is based on the exploitation of thermal imaging by deep learning to infer the thickness of a blow-moulded fuel tank. Thermal imaging has been used to infer the thickness of an object before, but to our knowledge, this is the first time a data-driven method has been used to infer thickness based on the decay of surface temperature. We believe that such an approach could be extended to other manufacturing production processes where the manufactured parts are subject to a cooling process.

We have shown how deep learning can be exploited in the manufacturing industry, even when the amount of data available is limited. Transfer learning and data augmentation have proven to be effective techniques to address the quality data scarcity issue.

Industrial contributions

This research work presents five main industrial contributions:

We provided the company with a new data-driven culture based on machine learning. This approach opens up new possibilities in terms of quality control and process improvement. This thesis focuses on the application to extrusion blow-moulding, but the same approach could be applied to other production processes. For instance, such a framework is currently used in finishing centres of the production line to assess the quality of welded parts.

The results presented throughout this research work have helped to revise certain beliefs about the functioning of the extrusion blow-moulding process. The critical process parameters, as collected today, are not sufficient to explain the part weight variability. Moreover, it has been found that the weight is not sufficient to ensure correct material distribution.

The *SmartBMM* software is currently in deployment in all the manufacturing plants of the company. Daily use in the pilot plants, where the solution has been developed, has shown a significant reduction in purge cycle time and starting phases. This reduction in cycle time results in a reduction in part non-conformities.

The parison length provides interesting information about the process. Initia-

tives are currently in progress to exploit the real-time measurement of this length to adjust the die-gap opening according to the length of the parison. This would allow a more regular distribution of material along the parison length.

We propose an approach to infer the thickness of blow-moulded parts using thermal imaging and a deep learning without any direct measurement of the part thickness. The results are considered sufficiently accurate to move towards the industrialisation of the proposed system. Such an approach would allow the thorough inspection of all parts.

Perspectives

Finally, we would like to draw attention to some research directions. In our opinion, there are three major challenges for the manufacturing industry to properly apply machine learning models in production: the machine learning model accuracy management, the machine learning life-cycle management and the model generalisation across different machines. These topics need to be addressed in order to answer the third project objective presented in the Section 1.4.

If we want to employ machine learning to monitor the operation of a production process or to assess the quality of a manufactured part, we must be extremely careful with mission-critical tasks, which necessitate models with a very high accuracy. For instance, for a manufacturing company with low percentage of quality scraps, a machine learning model with a 80% accuracy in prediction will lead to many incorrect alerts. Not detecting bad parts, or declaring a part as non-compliant when in fact it is fully compliant, would be problematic for the company. The experiments conducted during this research work have highlighted how extremely hard it is to use machine learning in the context of our manufacturing process. We claim that a 100% model accuracy is not only utopian in our industrial context, but in general in the manufacturing industry. Therefore, we think the in most cases machine learning models could not be able to completely replace the quality inspection performed by a human operator. These methods will rather be used as a tool to help diagnose quality problems. Future works will focus on how a model with a limited accuracy could provide value to a company.

Another topic related to decision making is the life-cycle management of machine learning models. Results presented in Chapter 3 of this PhD dissertation have highlighted that the extrusion blow-moulding process is non-stationary. In

the manufacturing industry, multiple non controllable factors can lead to a change in the input data distribution. If the input data distribution changes over time, a model will not be able to provide the same accuracy over time. Therefore, it is crucial to correctly monitor this data distribution change, possibly re-training the model. In order to re-train the model, new quality control data are needed to fit the model. Moreover, future research works could take advantage of *domain adaptation*, which is a particular case of transfer learning that utilises labelled data in one or more relevant source domains to execute new tasks in a target domain. The common algorithms for shallow domain adaptation can mainly be categorised into two classes: instance-based (Bruzzone and Marconcini, 2009; Chu et al., 2013) and feature-based (Gheisari and Baghshah, 2015; Gong et al., 2013; Pan et al., 2010). The first class reduces the discrepancy between domains by re-weighting the source samples, and it trains on the weighted source samples. For the second class, a common shared space is generally learned in which the distributions of the two datasets are matched. These types of approaches could be used to limit the amount of new samples needed to re-train the model over time.

One last topic that should be addressed in the context of the presented research work is the generalisation across different machines or productions. For a company like Plastic Omnium, it would be interesting to deploy a single machine learning model in different production plants. Unfortunately, a machine learning model trained on a specific machine will not necessarily work for another machine. Some external factors that are not controlled may change. For instance, the input raw material may have slightly different properties, or the room temperature may change across different plants. This means that a lot of time and energy is needed to get training data in each plant of the company. We think that transfer learning could partially solve this problem. Instead of training a model from scratch, we can use a pre-trained model on data from another machine or production. In this way, the model should achieve an accuracy that makes it usable with a limited number of training samples.

Bibliography

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Akiba, T., Sano, S., Yanase, T., Ohta, T., and Koyama, M. (2019). Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Arabacı, H. and Bilgin, O. (2010). Automatic detection and classification of rotor cage faults in squirrel cage induction motor. *Neural Computing and Applications*, 19(5):713–723.

Attar, A., Bhuiyan, N., and Thomson, V. (2008). Manufacturing in blow molding: Time reduction and part quality improvement. *Journal of Materials Processing Technology*, 204(1-3):284–289.

Badrinarayanan, V., Kendall, A., and Cipolla, R. (2017). Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on Pattern Analysis and Machine Intelligence*, 39(12):2481–2495.

Benítez, H. D., Ibarra-Castanedo, C., Bendada, A., Maldague, X., Loaiza, H., and Caicedo, E. (2008). Definition of a new thermal contrast and pulse correction for defect quantification in pulsed thermography. *Infrared Physics & Technology*, 51(3):160–167.

Bergstra, J., Bardenet, R., Bengio, Y., and Kégl, B. (2011). Algorithms for hyper-parameter optimization. In *25th annual conference on Neural Information Processing Systems (NIPS 2011)*, volume 24. Neural Information Processing Systems Foundation.

Bergstra, J. and Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(2).

Bishop, C. M. (2006). Pattern recognition. *Machine learning*, 128(9).

Blumberg, R. and Atre, S. (2003). The problem with unstructured data. *DM Review*, 13(42-49):62.

Boser, B. E., Guyon, I. M., and Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152.

Breiman, L. (1996). Bagging predictors. *Machine learning*, 24(2):123–140.

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

Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. (2017). *Classification and regression trees*. Routledge.

Bruzzone, L. and Marconcini, M. (2009). Domain adaptation problems: A dasvm classification technique and a circular validation strategy. *IEEE transactions on pattern analysis and machine intelligence*, 32(5):770–787.

Buneman, P., Davidson, S., Fernandez, M., and Suciu, D. (1997). Adding structure to unstructured data. In *International Conference on Database Theory*, pages 336–350. Springer.

Chen, L.-C., Papandreou, G., Schroff, F., and Adam, H. (2017). Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*.

Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., and Adam, H. (2018). Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 801–818.

Chen, W.-C., Tai, P.-H., Wang, M.-W., Deng, W.-J., and Chen, C.-T. (2008). A neural network-based approach for dynamic quality prediction in a plastic injection molding process. *Expert Systems with Applications*, 35(3):843–849.

Chen, X. and Voigt, T. (2020). Implementation of the manufacturing execution system in the food and beverage industry. *Journal of Food Engineering*, 278:109932.

Chen, X., Zhou, J., Xiao, H., Wang, E., Xiao, J., and Zhang, H. (2014). Fault diagnosis based on comprehensive geometric characteristic and probability neural network. *Applied Mathematics and Computation*, 230:542–554.

Cheng, W. (2017). Thickness measurement of metal plates using swept-frequency eddy current testing and impedance normalization. *IEEE Sensors Journal*, 17(14):4558–4569.

Cho, K., Van Merriënboer, B., Bahdanau, D., and Bengio, Y. (2014). On the properties of neural machine translation: Encoder-decoder approaches. In *Proceedings of SSST@EMNLP*, pages 103–111. Association for Computational Linguistics.

Choi, M., Kang, K., Park, J., Kim, W., and Kim, K. (2008). Quantitative determination of a subsurface defect of reference specimen by lock-in infrared thermography. *NDT & E International*, 41(2):119–124.

Chollet, F. et al. (2015). Keras. https://keras.io.

Chu, W.-S., De la Torre, F., and Cohn, J. F. (2013). Selective transfer machine for personalized facial action unit detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3515–3522.

Daneels, A. and Salter, W. (1999). What is SCADA? In *International Conference on Accelerator and Large Experimental Physics Control Systems*.

De Chiffre, L., Carmignato, S., Kruth, J.-P., Schmitt, R., and Weckenmann, A. (2014). Industrial applications of computed tomoboography. *CIRP Annals*, 63(2):655–677.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on Computer Vision and Pattern Recognition*, pages 248–255. IEEE.

Diraddo, R. and Garcia-Rejon, A. (1993a). Modeling of membrane inflation in blow molding: neural network prediction of initial dimensions from final part specifications. *Advances in Polymer Technology: Journal of the Polymer Processing Institute*, 12(1):3–24.

Diraddo, R. and Garcia-Rejon, A. (1993b). On-line prediction of final part dimensions in blow molding: A neural network computing approach. *Polymer Engineering & Science*, 33(11):653–664.

Diraddo, R. and García-Rejón, A. (1993). Profile optimization for the prediction of initial parison dimensions from final blow moulded part specifications. *Computers & Chemical Engineering*, 17(8):751–764.

Drucker, H., Burges, C. J., Kaufman, L., Smola, A., Vapnik, V., et al. (1997). Support vector regression machines. *Advances in Neural Information Processing Systems*, 9:155–161.

Einabadi, B., Baboli, A., and Ebrahimi, M. (2019). Dynamic predictive maintenance in industry 4.0 based on real time information: Case study in automotive industries. *IFAC-PapersOnLine*, 52(13):1069–1074.

Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of Statistics*, pages 1189–1232.

Friedman, J. H. (2017). *The elements of statistical learning: Data mining, inference, and prediction*. springer open.

Fuchs, C. and Kenett, R. S. (1998). *Multivariate quality control: theory and applications*. CRC Press.

Gheisari, M. and Baghshah, M. S. (2015). Unsupervised domain adaptation via representation learning and adaptive classifier learning. *Neurocomputing*, 165:300–311.

Glorot, X. and Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256. JMLR Workshop and Conference Proceedings.

Glorot, X., Bordes, A., and Bengio, Y. (2011). Deep sparse rectifier neural networks. In *AISTATS 2011*.

Gomes, M., Silva, F., Ferraz, F., Silva, A., Analide, C., and Novais, P. (2016). Developing an ambient intelligent-based decision support system for production and control planning. In *International Conference on Intelligent Systems Design and Applications*, pages 984–994. Springer.

Gong, B., Grauman, K., and Sha, F. (2013). Connecting the dots with landmarks: Discriminatively learning domain-invariant features for unsupervised domain adaptation. In *International Conference on Machine Learning*, pages 222–230. PMLR.

Goodfellow, I., Bengio, Y., Courville, A., and Bengio, Y. (2016). *Deep learning*. MIT Press.

Goodship, V. D., Middleton, B., and Cherrington, R. (2015). *Design and manufacture of plastic components for multifunctionality: structural composites, injection molding, and 3D printing*. William Andrew.

Graves, A. and Jaitly, N. (2014). Towards end-to-end speech recognition with recurrent neural networks. In *International Conference on Machine Learning*, pages 1764–1772. PMLR.

Graves, A., Jaitly, N., and Mohamed, A.-r. (2013). Hybrid speech recognition with deep bidirectional LSTM. In *2013 IEEE workshop on Automatic Speech Recognition and Understanding*, pages 273–278. IEEE.

Haeussler, J. and Wortberg, J. (1996). Quality control in injection molding with an adaptive process model based on neural networks. In *technical papers of the annual technical conference-society of plastics engineers incorporated*, pages 537–543. Society of Plastics Engineers INC.

He, K., Zhang, X., Ren, S., and Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1026–1034.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 770–778.

He, Y., Pan, M., Chen, D., Tian, G., and Zhang, H. (2013). Eddy current step heating thermography for quantitatively evaluation. *Applied Physics Letters*, 103(19):194101.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Hoerl, A. E. and Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67.

Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24(6):417.

Howard, A., Sandler, M., Chu, G., Chen, L.-C., Chen, B., Tan, M., Wang, W., Zhu, Y., Pang, R., Vasudevan, V., et al. (2019). Searching for mobilenetv3. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1314–1324.

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., and Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*.

Huang, H.-X. and Liao, C.-M. (2002). Prediction of parison swell in plastics extrusion blow molding using a neural network method. *Polymer Testing*, 21(7):745–749.

ISO (1992). *ISO 9000: International standards for quality management*.

Jiang, D., Lin, W., and Raghavan, N. (2020). A novel framework for semiconductor manufacturing final test yield classification using machine learning techniques. *IEEE Access*, 8:197885–197895.

Jones, D. R. (2001). A taxonomy of global optimization methods based on response surfaces. *Journal of Global Optimization*, 21(4):345–383.

Khan, M., Wu, X., Xu, X., and Dou, W. (2017). Big data challenges and opportunities in the hype of industry 4.0. In *IEEE International Conference on Communications (ICC)*, pages 1–6. IEEE.

Khayat, R. E., Derdorri, A., and García-Rejón, A. (1992). Inflation of an elastic cylindrical membrane: non-linear deformation and instability. *International Journal of Solids and Structures*, 29(1):69–87.

Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR*.

Kiros, R., Salakhutdinov, R., and Zemel, R. S. (2014). Unifying visual-semantic embeddings with multimodal neural language models. *arXiv preprint arXiv:1411.2539*.

Kotthoff, L. (2016). Algorithm selection for combinatorial search problems: A survey. In *Data Mining and Constraint Programming*, pages 149–190. Springer.

Krimi, S., Klier, J., Jonuscheit, J., von Freymann, G., Urbansky, R., and Beigang, R. (2016). Highly accurate thickness measurement of multi-layered automotive paints using terahertz technology. *Applied Physics Letters*, 109(2):021105.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, volume 25, pages 1097–1105.

LeCun, Y., Bengio, Y., and Hinton, G. E. (2015). Deep learning. *Nature*, 521(7553):436–444.

Lee, D. and Soh, S. (1996). Prediction of optimal preform thickness distribution in blow molding. *Polymer Engineering & Science*, 36(11):1513–1520.

Lee, I. and Shin, Y. J. (2020). Machine learning for enterprises: Applications, algorithm selection, and challenges. *Business Horizons*, 63(2):157–170.

Lee, Y. O., Jo, J., and Hwang, J. (2017). Application of deep neural network and generative adversarial network to industrial maintenance: A case study of induction motor fault detection. In *2017 IEEE International Conference on Big Data (Big Data)*, pages 3248–3253. IEEE.

Lenz, B. and Barak, B. (2013). Data mining and support vector regression machine learning in semiconductor manufacturing to improve virtual metrology. In *2013 46th Hawaii International Conference on System Sciences*, pages 3447–3456. IEEE.

Li, F., Wu, J., Dong, F., Lin, J., Sun, G., Chen, H., and Shen, J. (2018a). Ensemble machine learning systems for the estimation of steel quality control. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 2245–2252. IEEE.

Li, X., Tao, N., Sun, J., Zhang, C., and Zhao, Y. (2018b). Thickness measurement by two-sided step-heating thermal imaging. *Review of Scientific Instruments*, 89(1):014902.

Li, Z., Wang, Y., and Wang, K.-S. (2017). Intelligent predictive maintenance for fault diagnosis and prognosis in machine centers: Industry 4.0 scenario. *Advances in Manufacturing*, 5(4):377–387.

Lieber, D., Stolpe, M., Konrad, B., Deuse, J., and Morik, K. (2013). Quality prediction in interlinked manufacturing processes based on supervised & unsupervised machine learning. *Procedia Cirp*, 7:193–198.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014). Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer.

Liu, J. and Zio, E. (2016). Prediction of peak values in time series data for prognostics of critical components in nuclear power plants. *IFAC-PapersOnLine*, 49(28):174–178.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., and Berg, A. C. (2016). Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer.

Lolli, F., Balugani, E., Ishizaka, A., Gamberini, R., Rimini, B., and Regattieri, A. (2019). Machine learning for multi-criteria inventory classification applied to intermittent demand. *Production Planning & Control*, 30(1):76–89.

Long, J., Shelhamer, E., and Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 3431–3440.

Lorenz, M., Kuepper, D., Ruessmann, M., Heidemann, A., and Bause, A. (2016). Time to accelerate in the race toward industry 4.0. *Boston: The Boston Consulting Group. Recuperado a partir de https://www. bcgperspectives. com/Images/BCG-Time-to-Accelerate-in-the-Race-Toward-Industry-4.0-May-2016_ tcm80-211060. pdf.*

Malik, H. and Mishra, S. (2017). Artificial neural network and empirical mode decomposition based imbalance fault diagnosis of wind turbine using turbsim, fast and simulink. *IET Renewable Power Generation*, 11(6):889–902.

Mao, X. and Lei, Y. (2016). Thickness measurement of metal pipe using swept-frequency eddy current testing. *NDT & E International*, 78:10–19.

Marcon, P., Zezulka, F., Vesely, I., Szabo, Z., Roubal, Z., Sajdl, O., Gescheidtova, E., and Dohnal, P. (2017). Communication technology for industry 4.0. In *2017 Progress In Electromagnetics Research Symposium - Spring (PIERS)*, pages 1694–1697.

May, R. K., Evans, M. J., Zhong, S., Warr, I., Gladden, L. F., Shen, Y., and Zeitler, J. A. (2011). Terahertz in-line sensor for direct coating thickness measurement of individual tablets during film coating in real-time. *Journal of Pharmaceutical Sciences*, 100(4):1535–1544.

Meinshausen, N. and Bühlmann, P. (2010). Stability selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(4):417–473.

Melhem, M., Ananou, B., Ouladsine, M., and Pinaton, J. (2016). Regression methods for predicting the product's quality in the semiconductor manufacturing process. *IFAC-PapersOnLine*, 49(12):83–88.

Morariu, C., Morariu, O., Răileanu, S., and Borangiu, T. (2020). Machine learning for predictive scheduling and resource allocation in large scale manufacturing systems. *Computers in Industry*, 120:103244.

Nagorny, P., Lacombe, T., Favreliere, H., Pillet, M., Pairel, E., Le Goff, R., Wali, M., Loureaux, J., and Kiener, P. (2018). Generative adverserial networks for geometric surfaces prediction in injection molding: Performance analysis with discrete modal decomposition. In *2018 IEEE International Conference on Industrial Technology (ICIT)*, pages 1514–1519. IEEE.

Nagorny, P., Pillet, M., Pairel, E., Le Goff, R., Loureaux, J., Wali, M., and Kiener, P. (2017). Quality prediction in injection molding. In *Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA), 2017 IEEE International Conference on*, pages 141–146. IEEE.

Natekin, A. and Knoll, A. (2013). Gradient boosting machines, a tutorial. *Frontiers in Neurorobotics*, 7:21.

Nguyen, K. T. and Medjaher, K. (2019). A new dynamic predictive maintenance framework using deep learning for failure prognostics. *Reliability Engineering & System Safety*, 188:251–262.

OpenJS Foundation, C. (2013). Node-red.

Pan, S. J., Tsang, I. W., Kwok, J. T., and Yang, Q. (2010). Domain adaptation via transfer component analysis. *IEEE transactions on neural networks*, 22(2):199–210.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. (2019). Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, 32:8026–8037.

Pearson, K. (1901). Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Poli, C. (2001). *Design for manufacturing: a structured approach*. Butterworth-Heinemann.

Poslinski, A. J. and Tsamopoulos, J. A. (1990). Nonisothermal parison inflation in blow molding. *AIChE journal*, 36(12):1837–1850.

Profanter, S., Tekat, A., Dorofeev, K., Rickert, M., and Knoll, A. (2019). Opc ua versus ros, dds, and mqtt: performance evaluation of industry 4.0 protocols. In *2019 IEEE International Conference on Industrial Technology (ICIT)*, pages 955–962. IEEE.

Ramana, E. and Reddy, P. R. (2013). Data mining based knowledge discovery for quality prediction and control of extrusion blow molding process. *International Journal of Advances in Engineering & Technology*, 6(2):703.

Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-assisted Intervention*, pages 234–241. Springer.

Rublee, E., Rabaud, V., Konolige, K., and Bradski, G. (2011). Orb: An efficient alternative to sift or surf. In *2011 International Conference on Computer Vision*, pages 2564–2571. IEEE.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1985). Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning representations by back-propagating errors. *nature*, 323(6088):533–536.

Rüßmann, M., Lorenz, M., Gerbert, P., Waldner, M., Justus, J., Engel, P., and Harnisch, M. (2015). Industry 4.0: The future of productivity and growth in manufacturing industries. *Boston Consulting Group*, 9.

Ryan, M. and Dutta, A. (1982). The dynamics of parison free inflation in extrusion blow molding. *Polymer Engineering & Science*, 22(9):569–577.

Sagiroglu, S. and Sinanc, D. (2013). Big data: A review. In *2013 international conference on collaboration technologies and systems (CTS)*, pages 42–47. IEEE.

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L.-C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 4510–4520.

Schläpfer, R. C., Koch, M., and Merkhofer, P. (2015). Industry 4.0 challenges and solutions for the digital transformation and use of exponential technologies. *Deloitte, Zurique*.

Schwab, K. (2016). The 4th industrial revolution. In *World Economic Forum. New York: Crown Business*.

Sharma, G., Rao, R. U., and Rao, P. S. (2017). A taguchi approach on optimal process control parameters for hdpe pipe extrusion process. *Journal of Industrial Engineering International*, 13(2):215–228.

Shwartz-Ziv, R. and Armon, A. (2021). Tabular data: Deep learning is not all you need. *arXiv preprint arXiv:2106.03253*.

Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Smirnov, D. and Nguifo, E. M. (2018). Time series classification with recurrent neural networks. *Advanced Analytics and Learning on Temporal Data*, 8.

Su, K., Shen, Y.-C., and Zeitler, J. A. (2014). Terahertz sensor for non-contact thickness and quality measurement of automobile paints of varying complexity. *IEEE Transactions on Terahertz Science and Technology*, 4(4):432–439.

Sun, J. (2003). Method for determining defect depth using thermal imaging. US Patent 6,542,849.

Sun, J. (2006). Analysis of pulsed thermography methods for defect depth prediction. *Journal of Heat Transfer*, 128(4):329–338.

Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. *arXiv preprint arXiv:1409.3215*.

Tao, F., Qi, Q., Liu, A., and Kusiak, A. (2018). Data-driven smart manufacturing. *Journal of Manufacturing Systems*, 48:157–169.

Tellaeche, A. and Arana, R. (2013). Machine learning algorithms for quality control in plastic molding industry. In *2013 IEEE 18th Conference on Emerging Technologies & Factory Automation (ETFA)*, pages 1–4. IEEE.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288.

Toma, R. N., Prosvirin, A. E., and Kim, J.-M. (2020). Bearing fault diagnosis of induction motors using a genetic algorithm and machine learning classifiers. *Sensors*, 20(7):1884.

Tukey, J. W. et al. (1977). *Exploratory data analysis*, volume 2. Reading, Mass.

Vapnik, V. (2000). *The Nature of Statistical Learning Theory*. Springer.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Vinyals, O., Toshev, A., Bengio, S., and Erhan, D. (2015). Show and tell: A neural image caption generator. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 3156–3164.

Wada, K. (2016). Labelme: Image polygonal annotation with python. *GitHub repository*.

Wang, H., Li, W., and Feng, Z. (2015). Noncontact thickness measurement of metal films using eddy-current sensors immune to distance variation. *IEEE Transactions on Instrumentation and Measurement*, 64(9):2557–2564.

Wong, M., Jack, L., and Nandi, A. (2006). Modified self-organising map for automated novelty detection applied to vibration signal monitoring. *Mechanical Systems and Signal Processing*, 20(3):593–610.

Woschank, M., Rauch, E., and Zsifkovits, H. (2020). A review of further directions for artificial intelligence, machine learning, and deep learning in smart logistics. *Sustainability*, 12(9):3760.

Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., Zemel, R., and Bengio, Y. (2015). Show, attend and tell: Neural image caption generation with visual attention. In *International Conference on Machine Learning*, pages 2048–2057. PMLR.

Yin, W. and Peyton, A. (2007). Thickness measurement of non-magnetic plates using multi-frequency eddy current sensors. *NDT & E International*, 40(1):43–48.

Yosinski, J., Clune, J., Bengio, Y., and Lipson, H. (2014). How transferable are features in deep neural networks? *arXiv preprint arXiv:1411.1792*.

Zeiler, M. D. and Fergus, R. (2014). Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer.

Zeng, Z., Zhou, J., Tao, N., Feng, L., and Zhang, C. (2012). Absolute peak slope time based thickness measurement using pulsed thermography. *Infrared Physics & Technology*, 55(2-3):200–204.

Zezulka, F., Marcon, P., Bradac, Z., Arm, J., Benesl, T., and Vesely, I. (2018). Communication systems for industry 4.0 and the iiot. *IFAC-PapersOnLine*, 51(6):150–155.

Zhang, D., Mishra, S., Brynjolfsson, E., Etchemendy, J., Ganguli, D., Grosz, B., Lyons, T., Manyika, J., Niebles, J. C., Sellitto, M., et al. (2021). The ai index 2021 annual report. *arXiv preprint arXiv:2103.06312*.

Zhang, J., Ding, G., Zou, Y., Qin, S., and Fu, J. (2019). Review of job shop scheduling research and its new perspectives under industry 4.0. *Journal of Intelligent Manufacturing*, 30(4):1809–1830.

Zhao, H., Shi, J., Qi, X., Wang, X., and Jia, J. (2017). Pyramid scene parsing network. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2881–2890.

Zhong, R. Y., Xu, X., Klotz, E., and Newman, S. T. (2017). Intelligent manufacturing in the context of industry 4.0: a review. *Engineering*, 3(5):616–630.

Plastic Omnium Clean Energy System

The Clean Energy Systems (CES) is one of the three divisions of the Plastic Omnium company, specialised in plastic fuel tanks systems, and depollution systems, mostly for private and commercial vehicles. In 2018, more than 22M fuel systems have been delivered, representing 1 out of 4 commercialised vehicles equipped with a fuel system coming from the CES division. The material used for producing the fuel tanks is HDPE (High-Density Polyethylene). There are several reasons why they are made of plastic and not in metal as they used to be (for cars):

- Plastic is lighter than metal (about 30%), which allows a reduction of fuel consumption.
- The raw material is less expensive.
- A plastic tank cannot explode: it will melt, and the fuel will be spilled on the floor.

However, one issue is the permeability: as plastic is a porous material, fuel will eventually end up going through it and that leads to two major issues: the consumer will lose some of his gas, and this one will go into the air and pollute the atmosphere. That is why a fuel tank is not a simple container of fuel: it's a real part composed of complex technologies, from the production processes to the material used, and also the parts attached to the fuel system: filling system, pump gauge module, ventilation. These are used to make the fuel system (Figure 17) less permeable to cope with the different regulations. A fuel system is composed of a fuel tank (Figure 18) and a filler pipe (Figure 19, the latter is the only part visible of the system by the end user, to refill the tank at the station.

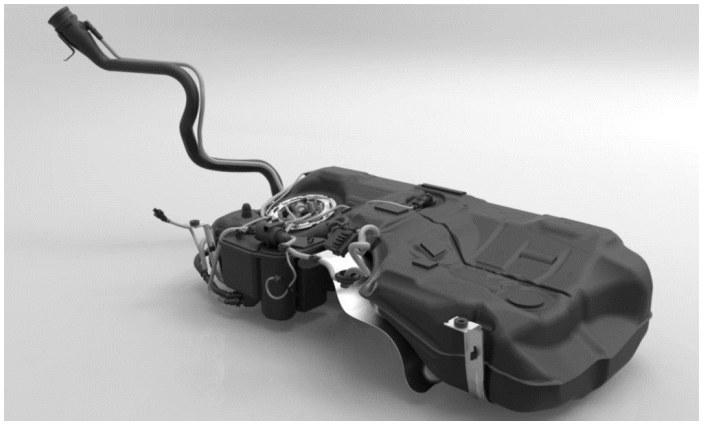

Figure 17: Fuel System

Figure 18: Plastic Fuel Tank

Figure 19: Filler Pipe

The SCR technology (Selective Catalytic Reduction) is an effective response to the regulatory requirements limiting emissions of nitrogen oxides (NO_x) from diesel vehicles. Combining a tank with a pump and gauge module, this system injects vaporised *AdBlue®* into the hot exhaust gases, causing a chemical reaction that transforms NO_x into water vapor. Plastic Omnium CES has developed a range of SCR systems to meet the needs of all types of vehicles, from the smallest European city car to the largest American pickup truck.

Fuel system production process

The manufacturing of a complete fuel system is the result of multiple production stages:

- Material Supply: plastic material is stored in dedicated silos into the plant. The material supply equipment is directly connected to the extrusion blow-moulding machine in order to continuously feed the extruders with new raw material.
- Extrusion blow-moulding: it is the core manufacturing process of the entire production line. During this stage raw material is transformed in a hollow fuel tank.
- Post Cooling: once the fuel tank has been blown, it enters a phase of post-mould cooling, where its temperature is lowered a second mould. This operation allows for the stabilisation of the tank dimensions.
- Finishing: during the finishing stage the hollow tank is cut in different areas and extra components are welded on it.
- Assembly: finally, some parts such as the filler pipe are assembled on the tank and every tank is checked to detect possible leaks.

The full process is outlined in Figure 20.

Figure 20: Full production process