

Politecnico di Torino

ScuDo
Scuola di Dottorato ~ Doctoral School
WHAT YOU ARE, TAKES YOU FAR

Doctoral Dissertation
Doctoral Program in Computer and Control Engineering (35^{th}cycle)

A management framework for smart manufacturing applications in Industry 4.0

By

Davide Cannizzaro

******

Supervisor(s):
Prof. Massimo Poncino, Supervisor
Prof. Edoardo Patti, Co-Supervisor

Doctoral Examination Committee:
Prof. Donkyu Baek, Chungbuk National University
Prof. Elisa Ficarra, Università di Modena e Reggio Emilia
Prof. Haroon Mahmood, FAST National University of Computer and Emerging Sciences, Islamabad Campus
Prof. Davide Quaglia, Università di Verona

Politecnico di Torino
2023

Declaration

I hereby declare that, the contents and organization of this dissertation constitute my own original work and does not compromise in any way the rights of third parties, including those relating to the security of personal data.

Davide Cannizzaro
2023

* This dissertation is presented in partial fulfillment of the requirements for **Ph.D. degree** in the Graduate School of Politecnico di Torino (ScuDo).

I suppose it will all make sense when we grow up.
Calvin & Hobbes

Acknowledgements

I want to express my deepest gratitude to my supervisor, Prof. Massimo Poncino, and the EDA group that gave me the opportunity to live this Ph.D. experience and to expand my knowledge in the world of Machine Learning.

I want to extend my gratitude to Prof. Enrico Macii for his support and guidance in this exciting experience.

My deepest gratitude goes to Prof. Santa Di Cataldo and Edoardo Patti for guiding me in my research activities and publications. I am deeply grateful for their support and valuable flair, which were instrumental in encouraging me to think creatively and objectively and to consider multiple perspectives during my Ph.D.

Last but not least, I would like to express my gratitude to all my colleagues and professors I met during my journey. Thank you for working with me on a daily basis, spending time and sharing knowledge together, and allowing me to improve myself.

To my friends, past and present, that inspired me and, in particular, to Tommaso, he too embarked on the doctoral adventure, and he supported and endured me down the rabbit hole.

Finally, my family has a special place in these thanks, my mother, father, brother, uncle, and beloved grandmother. Without them, I would not be the same person.

Abstract

The fourth industrial revolution, or Industry 4.0, began in 2011 at Hannover Fair with a focus on computerizing manufacturing. The combination of Internet of Things (IoT) and Machine Learning (ML) facilitated this revolution by enabling plants to connect to cloud services, collect previously unavailable data, and enhance production. The paradigm shift has transformed the manufacturing industry and has the potential to revolutionize other industries as well.
The emergence of Smart Manufacturing systems like Additive Manufacturing (AM) created new customer experiences and business models. However, Industry 4.0 requires appropriate management frameworks, new types of devices, and services, suitable communication, and energy efficiency. In the past years, huge amounts of data began to be produced by companies from industrial plants. However, in most cases, these data are only used for basic checks or disregarded entirely because a suitable platform to collect and analyze them is missing. In this context, researchers and companies concentrated on developing ad-hoc solutions for data collection and analysis based on the industrial use cases and machines they needed to analyze. Moreover, due to the complexity of industrial infrastructures, they mainly focused on a fast and simple collection of data without developing structured data management frameworks, as their main goal was only the analysis of data. By analyzing the current state-of-the-art, it is possible to observe that a management platform for heterogeneous industrial contexts is missing.
To address the problem of the lack of a general platform that could be applied in different industrial scenarios, my research activities focused on the development of an innovative microservices management framework to collect, manage, process, and analyze the data available in different smart manufacturing contexts, from Automotive to Aviation fields. The framework consists of several services with the objective to collect and manage the data provided by users and machines (Backend), visualize the data stored (dashboard and frontend), and provide an easy way to analyze them

in order to improve the overall production line by exploiting AI and ML algorithms (e.g., defect detection algorithms, profile monitoring). I designed and validated the platform, IoT devices, and the data analysis methodologies proposed in different Industry 4.0 scenarios: i) Smart Industry, ii) Smart Manufacturing, iii) Smart Sensors integration, and iv) Smart Energy.
In the industrial context, in order to store the data available, I integrated the platform developed in an AM production line. Then, I focused on the manufacturing process by analyzing the data collected, and I proposed several services for the framework to address most of the AM problems. I designed an in-situ monitoring system for the analysis of Laser Powder Bed Fusion (L-PBF) process, where I implemented ML algorithms based on images to detect typical defects on the powder bed and monitor the profile of the object during the printing process. Due to the small number of data available with defects, I also integrated a Generative Adversarial Network (GAN) to create synthetic images with defects from the images available.
Next, I moved towards Electron Beam Powder Bed Fusion (EB-PBF), another AM process, where I proposed a methodology for anomaly detection in order to predict in advance the success or failure of a printing job. Finally, I focused my research on the analysis of quality inspection procedures utilized for aircraft engine components with the aim of reducing the whole inspection time, ensuring higher quality, and guaranteeing standardization and process control.
In the context of smart sensors, I analyze the performances of several Artificial Intelligence (AI) algorithms and methods for the localization of industrial vehicles with IoT devices inside and outside a warehouse.
Regarding the smart energy context, I proposed an ML methodology for Global Horizontal Irradiance (GHI) predictions in short-, mid-, and long-term starting from exogenous inputs (i.e., physical factors related to the weather). The GHI is particularly important in the industrial context to forecast photovoltaic production and develop business models for energy efficiency.
In this thesis, a properly structured management framework suitable for data collection and analysis in different industrial contexts (e.g., automotive and aerospace production lines) is proposed and implemented. Besides, a set of ML and AI algorithms are presented and integrated into the platform in order to analyze the data collected from the different scenarios and provide insights to improve the overall processes (e.g., smart manufacturing, industrial localization, energy forecasting).

Contents

List of Figures **xi**

List of Tables **xv**

1 Introduction **1**

1.1 Industry 4.0 3

1.1.1 Cloud Computing 4

1.1.2 Internet of Things and Cyber-Physical Systems 8

1.1.3 Additive Manufacturing 11

1.1.4 Artificial Intelligence and Machine learning for smart manufacturing 17

1.2 Contributions 20

1.2.1 Smart Industry: Digital platform for production line in smart manufacturing 21

1.2.2 Smart Manufacturing: Online monitoring and post-production inspection in additive manufacturing processes . 22

1.2.3 Smart Sensors integration: Industrial indoor localization . . 23

1.2.4 Smart Energy: Energy efficiency in Industry 4.0 24

2 Digital platform for production line in smart manufacturing **25**

2.1 Introduction 25

2.2 Background 28
2.3 Data structure 30
2.4 Digital platform structure 33
2.5 Discussion and remarks 36

3 Online monitoring of L-PBF additive manufacturing 38
3.1 Introduction 38
3.2 State-of-the-art 39
3.3 Case study 42
3.3.1 Data acquisition and processing system 42
3.3.2 Defects detection algorithms 43
3.3.3 Part profile monitoring algorithm 45
3.3.4 Dataset augmentation with GAN 47
3.4 Experimental results 50
3.5 Discussion and remarks 55

4 Machine anomaly detection in EB-PBF process 57
4.1 Introduction 57
4.2 Multivariate time-series anomaly detection: current scenario 58
4.3 EB-PBF: Process and Hardware 61
4.4 Dataset description 64
4.5 Methodologies 69
4.5.1 Problem statement and EB-PBF-AE architecture 69
4.5.2 Experimental setup 72
4.6 Results 74
4.7 Discussion and Remarks 75

5 Automation of post-production inspection in Additive Manufacturing 77

5.1 Introduction . . . 77
5.2 Case study analysis and design concept . . . 79
5.3 Automated inspection system . . . 82
5.3.1 Robotic setup . . . 83
5.3.2 Camera setup . . . 84
5.3.3 Disparity Map software . . . 85
5.3.4 Optical fiber sensor . . . 86
5.3.5 Artificial Intelligence framework . . . 88
5.4 Experimental inspection process . . . 89
5.4.1 1st phase inspection . . . 89
5.4.2 Inspection results . . . 91
5.4.3 1st phase validation . . . 93
5.4.4 2nd phase inspection . . . 95
5.5 Discussion and Remarks . . . 96

6 Industrial indoor localization 98

6.1 Introduction . . . 98
6.2 Background and Related Work . . . 101
6.3 Localization Algorithms . . . 106
6.4 Experimental Setup . . . 110
6.5 Trilateration results . . . 112
6.6 Fingerprint Methods results . . . 113
6.6.1 Error Maps . . . 114
6.6.2 Impact of Beacons' Density . . . 116
6.6.3 Impact of Fingerprint Grid Density . . . 117
6.6.4 Impact of Environmental Perturbations . . . 118
6.7 Discussion and Remarks . . . 120

7 Solar radiation forecasting for energy efficiency in Industry 4.0 122

7.1 Introduction . . . 122

7.2 Related works . . . 125

7.3 Theoretical Background . . . 128

7.3.1 Variational Mode Decomposition . . . 128

7.3.2 Convolutional Neural Network . . . 130

7.3.3 Random Forest regressor . . . 131

7.3.4 Long Short-Term Memory Neural Network . . . 131

7.4 Performance Indicators . . . 132

7.5 Proposed model . . . 133

7.5.1 Dataset and pre-processing . . . 133

7.5.2 CNN-IMF and CNN-RES . . . 137

7.5.3 RF and LSTM . . . 138

7.5.4 Post-processing . . . 140

7.5.5 Creation and utilization of the proposed model . . . 141

7.6 Experimental results and analysis . . . 142

7.6.1 Performance assessments of the M15 model . . . 143

7.6.2 Performance assessments of the M60 model . . . 149

7.6.3 Comparison with NAR-based model in literature . . . 154

7.7 Discussion and Remarks . . . 155

References 161

List of Figures

1.1 The four stages of the industrial revolution 2

1.2 Industry 4.0 pillars 3

1.3 Cloud computing infrastructure models 7

1.4 IoT connected devices worldwide 8

1.5 Industrial IoT 10

1.6 AM process flow: from CAD to final object 13

1.7 AM technologies 13

1.8 Laser Powder Bed Fusion setup 15

1.9 Electron Beam Melting setup 16

1.10 Data types structure 18

1.11 Common structure 21

2.1 Digital platform: proposed architecture with the base infrastructure (in blue) and the customization applied (in orange) 33

2.2 User interface dashboard 35

3.1 In-situ monitoring system 43

3.2 (a)-(e) Examples of powder bed defects targeted by the system (f) Profile monitoring example 44

3.3 Example of spattering defects detection pipeline 46

3.4 U-Net architecture 47

3.5 Generative Adversarial Network architecture 48
3.6 ConSinGAN architecture [86] 49
3.7 Example of image harmonization [86] 50
3.8 Example of image generation 50
3.9 Profile monitoring: validation example 52
3.10 Comparison between a training image and a synthetic one with defects (holes) 54

4.1 EB-PBF hardware scheme 61
4.2 Pulse Length signals diverging from the optimal value (dashed line) leading the process to a non-recoverable situation 68
4.3 Example of complete and partial data 69
4.4 Architecture overview 70

5.1 Honeycomb structure and defects: a) missing cells, b) damaged cells, c) opened cells, d) excessive bond material, and e) not bonded cells . 78
5.2 Scheme of the inspection process: robotic tools 81
5.3 Inspection setup 83
5.4 Multi-view image of a honeycomb 85
5.5 Disparity map 86
5.6 Comparison between the sensor 3D CAD design and the first manufactured prototype 87
5.7 Fiber holder with emitting fibers (lighted) and sensor fibers (not lighted) 87
5.8 Image acquisition 90
5.9 Honeycomb image with different perspectives 90
5.10 Disparity map of honeycomb in a real plant 91
5.11 Honeycomb cells segmentation (a) and honeycomb defects analysis (b) 92
5.12 Honeycomb mask 93

5.13 Interface with the microcontroller output: (Left side) output for healthy cells, (Right side) output for heavily damaged cells 95

5.14 Glass fiber sliding inside the honeycomb cells 96

6.1 Trilateration algorithm for object localization using three beacons B_1, B_2, and B_3 107

6.2 Multi-Layer Perceptron (MLP) schematic model 108

6.3 Support-Vector Regressor (SVR) example for a two-dimensional problem, using a linear kernel 109

6.4 Error box plots for the trilateration algorithm. The x-axis reports the number and the corresponding density of the beacons in the environment 112

6.5 Error map of the three fingerprint-based methods in Environment A 115

6.6 Error map of the three fingerprint-based methods in Environment B 115

6.7 Error box plots for the three fingerprinting algorithms with different beacon densities. Results for a fingerprint grid with one measurement every $0.5m$ 116

6.8 Error box plots for the three fingerprinting algorithms with different fingerprint grid densities in Environment A 117

6.9 Error box plots for the three fingerprinting algorithms with different fingerprint grid densities in Environment B 118

6.10 Error box plots for the three fingerprinting algorithms as a function of the standard deviation of the Gaussian noise in Environment A . . 119

6.11 Error box plots for the three fingerprinting algorithms as a function of the standard deviation of the Gaussian noise in Environment B . . 119

7.1 Schema of the proposed methodology 123

7.2 Schema to create the input data (15*min* time-resolution) 135

7.3 VMD results for the first day in December 2009 (15*min* time-resolution) 136

7.4 Detailed schema of the proposed methodology 142

7.5 Flow chart of our model to forecast GHI 143
7.6 Overall performance trends for M15 146
7.7 GHI forecast trends for M15 at next 15*min* 147
7.8 GHI forecast trends for M15 at next 24*h* 148
7.9 Overall performance trends for M60 151
7.10 GHI forecast trends for M60 at next 1*h* 152
7.11 GHI forecast trends for M60 at next 24*h* 153

List of Tables

2.1 Main data in the Additive Manufacturing process 31

3.1 Defects detection algorithms validation 51
3.2 Mean execution time of the algorithms 53
3.3 ConSinGAN results 54
3.4 Defects detection results on artificially generated images 54

4.1 Data categories 65
4.2 List of signals analyzed 67
4.3 LSTM-AE parameters 73
4.4 LSTM-AE results 74
4.5 Example of failed job prediction 75

5.1 Experimental results 94

6.1 Summary table including relevant works on indoor localization based on Bluetooth low-energy (BLE) received signal strength indication (RSSI) and the corresponding feature extraction methods 105
6.2 Median error and interquartile range (IQR) of fingerprint-based methods in Environment A and Environment B using four beacons and a fingerprint grid with one measurement every $0.5m$ 114
6.3 Percentage of testing points lower than minimum half-distance between beacons 115

6.4 Error median and IQR for the three fingerprinting-based algorithms in Environment A after adding a large object not present during the collection of the RFM 120

7.1 Correlations between decomposed components and weather data . . 137

7.2 Comparisons of CNNs with different pooling layers 138

7.3 Structure of CNN-IMF 139

7.4 Structure of CNN-RES 140

7.5 Structures for both RF and LSTM 140

7.6 Performance indicator of M15 (i.e. 15*min* time-resolution) at different time-horizons 145

7.7 Performance indicator of M60 (i.e. 60*min* time-resolution) at different time-horizons 150

7.8 Comparison of our methodology with [6] at different time-horizons for both 15min and 60min time-resolutions 155

Chapter 1

Introduction

Over the last few decades, the progress of technology had a profound impact on humanity, leading to several industrial revolutions:

- The *First Industrial Revolution*, which took place in the late 18th and early 19th centuries, was characterized by the mechanization of manufacturing processes using steam power and water power. This led to the widespread use of machines in industry and the growth of factories.
- The *Second Industrial Revolution*, which occurred in the late 19th and early 20th centuries, was marked by the widespread adoption of electricity and the development of new technologies such as the telegraph and the telephone. This revolution led to the growth of large-scale industries and mass production.
- The *Third Industrial Revolution*, also known as the Digital Revolution, began in the late 20th century and continues to the present day. This period is characterized by the development of computers and the Internet, which have transformed nearly every aspect of society and industry.
- The *Fourth Industrial Revolution*, also known as Industry 4.0, is the current trend towards automation and data exchange in manufacturing technologies with the integration of Internet of Things (IoT) systems that connect and communicate with other devices and systems over the Internet. This revolution is still in progress and is expected to have a significant impact on industry and society in the coming years.

In Figure 1.1, a summary of the four stages of the industrial revolution is reported.

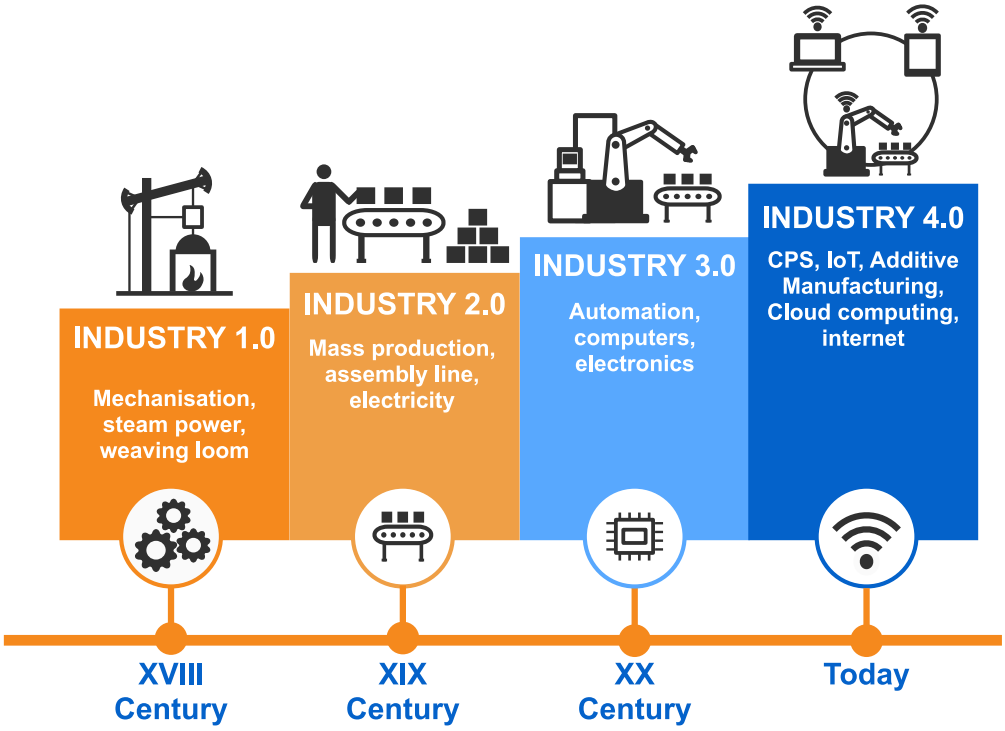


Figure 1.1 The four stages of the industrial revolution

Nowadays, with the development of Industry 4.0, a large number of connected IoT devices and machines that transmit a huge amount of data are made available in production lines.

In this context, a management framework for Industry 4.0 is required in order to provide a structure to effectively integrate, analyze and utilize all the different technologies involved in a production line. This framework would address key issues such as data management, interoperability, and human-machine collaboration.

Moreover, with the integration of Artificial Intelligence (AI) and Machine Learning (ML) networks as services of the platform, the data collected can be exploited to improve the production plant and develop new technologies, providing enormous economic advantages and development benefits.

Finally, according to the company's needs, the framework can be implemented on local or cloud computing, allowing different possible scenarios. By adopting a comprehensive management approach, companies can unlock the full potential of Industry 4.0 and gain a competitive advantage in the marketplace.

1.1 Industry 4.0

According to the World Economic Forum (WEF) [63], one of the key concepts of Industry 4.0 is based on the integration of information and communication technologies as well as industrial technologies like Cloud Computing, Internet of Things (IoT), Cyber-Physical Systems (CPS), Additive manufacturing (AM), Artificial Intelligence (AI) and Machine learning (ML) (Figure 1.2). All these technologies lead to increased efficiency, productivity, and sustainability [13].

For example, sensors and other smart IoT devices can be used to monitor production in real-time in order to study the power consumption of the plant and minimize energy waste, allowing manufacturers to quickly identify and address any potential issues. Moreover, the collection and organization of the data available allow to improve the decision-making process by company management and better organize the development of the plant production [206].

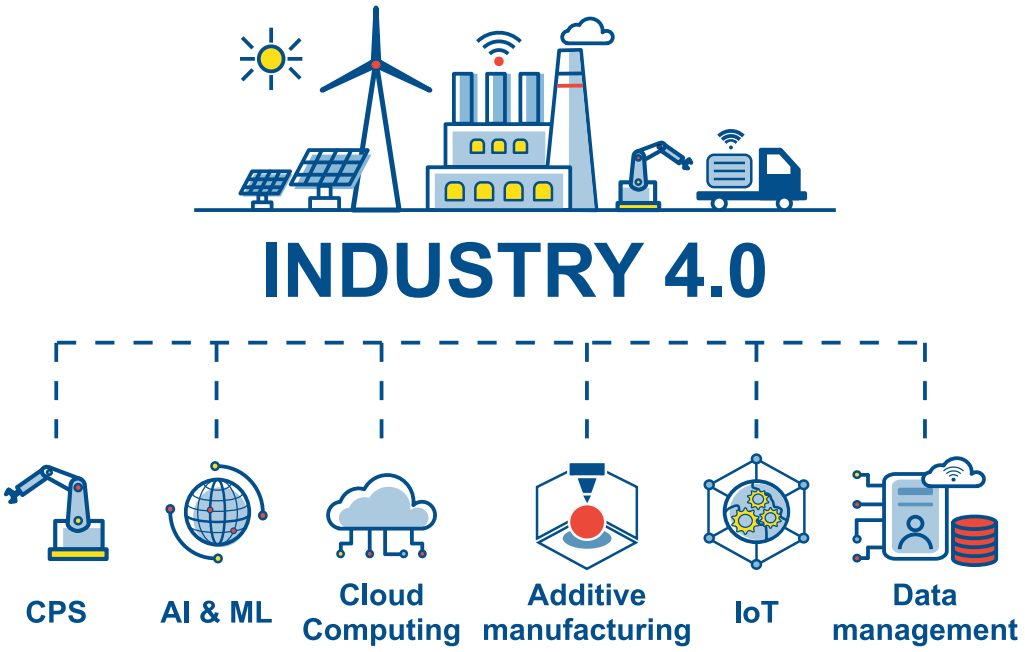


Figure 1.2 Industry 4.0 pillars

Thus, creating a more connected and integrated industrial ecosystem is one of the most critical aspects of Industry 4.0.

The first step for a company to implement the concepts of Industry 4.0 involves the development of new IoT devices in order to make "smart" and retrofit the previous generation production machines still working [80, 129].

Accordingly, the next step requires the development of a management framework to integrate the different CPS and technologies across the entire value chain, from product design and development to production, distribution, and customer service [78] and connect all the data available and systems through cloud computing [24]. Indeed, by connecting these different stages of the production process, manufacturers can gain a more comprehensive view of their operations and make more informed decisions, especially by integrating AI and ML services in order to analyze the processes from different points of view.

In literature, several authors reported the recent advances in industrial AI that showcased the potential of this technology in tackling the challenges associated with Industry 4.0 through its data-driven predictive analytics and capacity to assist decision-making in highly complex, non-linear and often multistage environments [162, 52, 98].

Overall, Industry 4.0 is expected to bring about significant changes to the way that companies operate and do business, with many industries already seeing the impact of these technologies on their operations [96, 203].

In this thesis, I will report the development and integration in an automotive industrial context of a management framework able to collect and manage data from CPS and IoT devices through cloud computing, together with the implementation of ML and AI algorithms to analyze the data provided by machines, sensors, and users in order to make the industrial environment smart, improve the production plant and reduce time, costs and energy consumption.

1.1.1 Cloud Computing

With the advent of the Internet, several companies have started to create new models of business that require the development of services and infrastructure around the globe and the interconnection between them to share information and develop products. In this context, cloud computing can play an important role because it allows users to take benefits from all the technologies developed during the Industry

4.0 era without the requirement of extensive understanding or proficiency in each individual topic [163].

All the hardware and services are managed by cloud providers that maintain and manage distributed resources on-demand based on users' needs. Cloud computing exhibits several advantages for the companies [14]:

- *Cost reduction*: Typically, large companies require huge capital expenditures in order to buy and maintain the infrastructure (e.g., servers, network infrastructures, etc.) and hire people with the required skills to manage them (e.g., system engineers, IT departments, etc.). In this context, cloud providers propose usage-based complete infrastructure solutions and human expertise in order to implement projects that use large computing resources [140].

- *Device and location independence*: Enable users to access systems through the Internet using, for example, a browser for web applications regardless of their location or the device type (e.g., PC, mobile phone, etc.). As infrastructure is off-site and accessed via the Internet, users can connect to it from anywhere.

- *Availability and scalability*: Cloud computing allows to have multiple redundant services that improve resource availability and makes it suitable for business continuity and disaster recovery. Moreover, scalability provides via on-demand provisioning of resources on a fine-grained, self-service basis in near real-time, without users having to engineer for peak loads [37]. This gives the ability to scale up the services based on the users' needs, with an increase in the resources when required or down if resources are not being used. The time-efficient benefit of cloud scalability also means faster time to market, more business flexibility, and adaptability, as adding new resources takes less time than it used to.

- *Productivity*: Multiple users can work and access the same data and services simultaneously around the globe, making sharing easier and avoiding delays between information transmission and updates. Time may be saved, allowing companies to split the workload and save money [7].

- *Performances and security*: Typically, cloud providers set up the latest hardware available that is monitored by IT experts in order to guarantee high computer performance and security. Moreover, cloud providers are able to

devote resources to solving security issues that many customers cannot afford to tackle or lack the technical skills to address. However, the complexity of the infrastructure is greatly increased when data is distributed over a wider area, as well as in multi-tenant systems shared by unrelated users. In some cases, concerns can arise about the loss of control over certain sensitive data and private cloud installations are in part motivated by users' desire to retain control over the infrastructure and avoid losing control of information security [36].

- *Maintenance*: The maintenance of the infrastructure is offloaded to the cloud provider where the data are hosted without the need for the company to invest resources in data center hardware. Also, the software maintenance of cloud computing is typically managed and updated by the cloud provider, which further reduces costs compared with on-premises data centers.

As already mentioned, cloud companies provide different service models that offer customers several levels of support and flexibility based on their needs.

The National Institute of Standards and Technology (NIST) defined three standard model architectures [130]: Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS).

These models offer increasing levels of abstraction and are thus often portrayed as layers in a stack, as shown in Figure 1.3. However, they are not necessarily required together. For example, it is possible to provide SaaS implemented on physical machines without using underlying PaaS or IaaS layers. Conversely, it is possible to run a program on IaaS without wrapping it up as SaaS.

These cloud computing services were developed following the recommendations of the NIST [143] in order to address different user requirements:

- *Infrastructure as a Service (IaaS)*: IaaS is defined as the provision of basic computing resources (e.g., processing, storage, and networking) that can be accessed and used by the consumer to run arbitrary software, such as operating systems and applications. The consumer does not have control over the underlying infrastructure but can manage the operating systems, storage, and deployed applications.

- *Platform as a Service (PaaS)*: Based on the recommendations, PaaS is the ability for the consumer to deploy their own applications onto the provider's

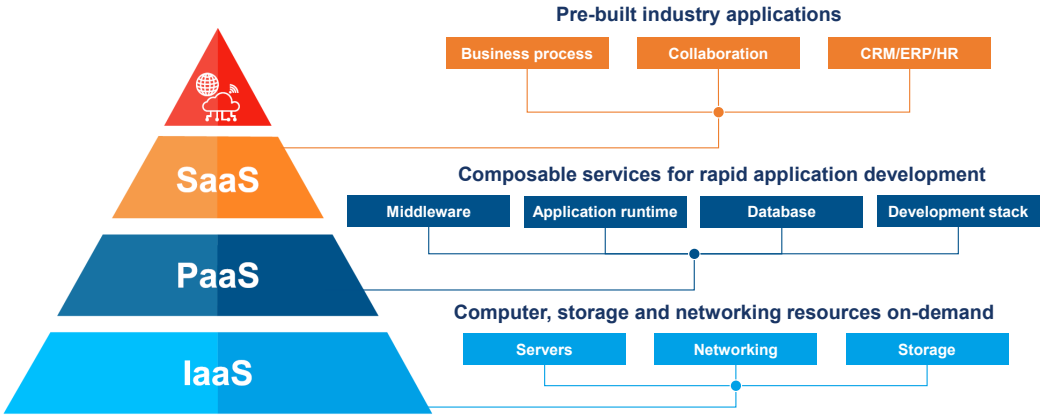


Figure 1.3 Cloud computing infrastructure models

cloud infrastructure. The consumer can use supported programming languages, libraries, services, and tools to create these applications. The consumer does not have control over the underlying infrastructure but has control over the deployed applications and possibly the configuration settings for the application-hosting environment.

- *Software as a Service (SaaS)*: According to the NIST, SaaS is defined as the ability for consumers to use the provider's applications that are running on a cloud infrastructure. These applications can be accessed from various client devices through a web browser or a program interface. The consumer does not have control over the underlying infrastructure and may only have limited control over user-specific application configuration settings.

In this context, a management framework for Industry 4.0 can be offered to companies as PaaS and developed in cloud systems. The solution can be alternatively provided as a public, hybrid, or private cloud, thus, reducing the data exchange over the Internet and increasing privacy and data management. Thanks to the features provided by cloud computing, different services can be developed to be either executed remotely on the cloud or locally on edge and connected together.

1.1.2 Internet of Things and Cyber-Physical Systems

The Internet of Things (IoT) term describes physical objects with sensors, processing ability, software, and other technologies that connect and exchange data with other devices and systems over the Internet or other communications networks [126].

In this context, a huge impact on the widespread of IoT devices was given by the advancement of connectivity that allowed for a wider range of devices to be connected, including those in remote or less accessible locations. This has been made possible through the use of power-efficient sensors and various connectivity options such as low-power wide-area networks, 4G cellular, and the new 5G cellular standard, which is specifically designed for the IoT [125].

IoT devices range from ordinary household objects to complex industrial tools, with more than 13 billion connected IoT devices estimated today. As can be seen in Figure 1.4, the number of IoT devices is expected to grow exponentially in the future.

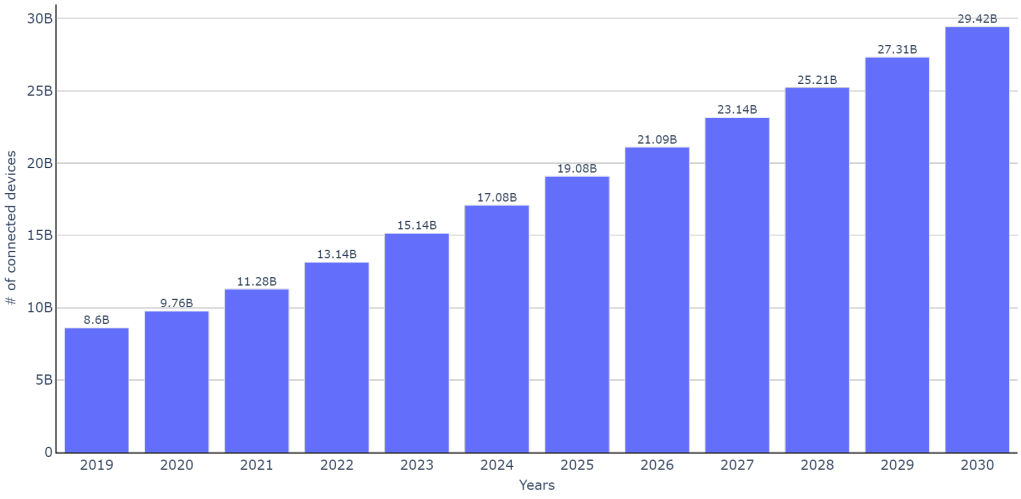


Figure 1.4 IoT connected devices worldwide

This is particularly true for industrial fields like manufacturing, automotive, and aerospace, where IoT has grown to empower the advantages of machinery and smart systems.

The cloud-based technology of remote data collection, intelligent machine interconnectivity, and sensor monitoring provides the opportunity for pattern modification

across all manufacturing divisions, including the latest and rapidly growing technology of Additive Manufacturing (AM) [81].

IoT can be used to improve efficiency, reduce costs, and increase productivity in several ways in an industrial plant. For example, in Industry 4.0, IoT-enabled devices and sensors can be used to monitor and control industrial equipment and machinery, to detect and predict failures, and to optimize the use of resources [109]. IoT can also be used to automate and streamline supply chain and logistics processes, improve energy efficiency and reduce waste, and enhance the maintenance and repair of industrial equipment [160].

This development of new IoT devices in order to make smart industrial plants for Industry 4.0 led to the birth of Cyber-Physical Systems (CPS). CPS is a subclass of IoT systems that involve the integration of computing, networking, and physical processes. CPS systems are characterized by the tight coupling between the cyber (digital) and physical (analog) components of the system, as well as by the real-time and dynamic nature of their operation [16].

The integration of IoT technologies into industry has enabled the development of advanced industrial systems and applications that combine innovative capabilities with the connection of physical operations to computing and communication infrastructures. This foundation has allowed the creation of highly functional and interconnected systems that can improve various industrial processes and operations [132].

However, introducing IoT technologies in existing production plants to build CPS systems could be a complex task that requires knowledge of the machinery in order to change the technological level of outdated physical industrial systems. This could be achieved with IoT by retrofitting all old equipment into new equipment. The retrofitting concept emerges as a rapid and low-cost solution aimed at reusing existing equipment, with the addition of new technologies in order to upgrade legacy industrial equipment to CPS [114]. Besides the installation of new hardware, smart retrofitting involves updating installed equipment with networking capability in order to enable the adoption of data-driven processes (Figure 1.5).

Thus, with this huge number of IoT devices and CPS increasing exponentially, in order to manage the data transmitted, several different protocols were integrated that are commonly used in IoT systems nowadays to facilitate communication between devices, based on the data transmission requirements [151]:

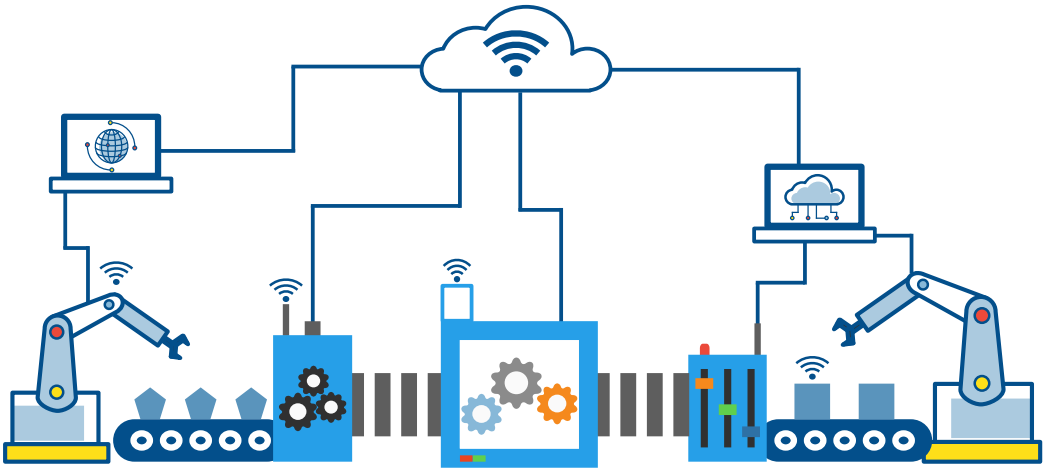


Figure 1.5 Industrial IoT

- MQTT (Message Queuing Telemetry Transport) [85]: A lightweight messaging protocol designed for resource-constrained devices and low-bandwidth networks. It is commonly used in IoT systems to transmit data from sensors and other devices to a central server or cloud platform. MQTT is based on a publish-subscribe model, in which devices can publish data to a central broker, and other devices can subscribe to receive data from the broker. This allows devices to communicate with each other and with other systems in a flexible and scalable way. MQTT is designed to be simple and lightweight, making it well-suited for resource-constrained devices and low-bandwidth networks. It uses a small number of bytes to transmit data, and it has a low overhead, which makes it efficient to use.

- HTTP (HyperText Transfer Protocol) [195]: A widely used protocol for exchanging information over the Internet. HTTP is based on a request-response model, in which a client (such as a web browser) sends a request to a server, and the server replies with a response. The request and response are transmitted using a specific format defined by the HTTP protocol (GET, POST, PUT, DELETE). HTTP is also used in other contexts, such as in APIs (Application Programming Interfaces), to enable communication between systems.

- CoAP (Constrained Application Protocol) [166]: A protocol that is designed for use in resource-constrained devices and low-power networks. It is often used in IoT systems to enable communication between devices using a simple,

RESTful (representational state transfer) style. CoAP is based on a request-response model, similar to HTTP, but it is designed to be simpler and more lightweight, making it well-suited for resource-constrained devices. It uses a small number of bytes to transmit data like MQTT.

- AMQP (Advanced Message Queuing Protocol) [23]: An open standard for messaging middleware that enables the exchange of messages between systems in a reliable and flexible way. It is designed to support a wide range of messaging scenarios, including point-to-point, publish-subscribe, and request-response. AMQP is based on the concept of messaging channels, which are used to send and receive messages between systems. It defines a set of rules for how messages should be formatted and transmitted, as well as how systems should authenticate and authorize each other. AMQP is often used in distributed systems where there is a need to exchange large volumes of data in a reliable and scalable way. It is commonly used in financial, healthcare, and other industries where the ability to exchange data between systems in a reliable and secure manner is critical.

By using these protocols, the management framework for Industry 4.0 presented in this thesis is able to exchange data between devices and systems in a reliable and scalable way, enabling the automation of processes and the adoption of advanced technologies in order to obtain smart CPS.

1.1.3 Additive Manufacturing

As described in the previous sections, Industry 4.0 offers Cyber-Physical Systems, cloud computing, and IoT to cooperate profitably, aiming to build smart factories by redefining the role of humans. While these technologies enable the effective utilization of data in smart manufacturing systems, the physical aspect of smart factories is currently limited by the capabilities of current manufacturing systems.

In this context, the fourth industrial revolution has recently moved towards the integration of modern manufacturing skills in order to make more effective use of data and address the demands of the market [50].

One of these solutions is called Additive Manufacturing (AM), also known as 3D printing [72]. AM is the process of creating objects by adding materials layer

by layer from 3D model data. This process allows better control with respect to subtractive manufacturing methods that have been used so far.

Indeed, unlike traditional machining techniques that involve either shaping or removing material to create a final product, AM allows to create finished parts directly. This process eliminates the need for intermediate steps like mold or die creation and does not require any tools, making it virtually unrestricted by geometric limitations. As a result, AM is seen as an important technology in the future for achieving the mass customization and personalization goals of Industry 4.0.

In this context, AM began to spread in the later part of the XX century. However, in the beginning, 3D printing techniques were considered suitable only for rapid prototyping and the production of functional or aesthetic prototypes. Nevertheless, in the last years, the precision, repeatability, and material range of 3D printing have increased to the point that some AM processes are considered viable as an industrial-production technology [217].

The standard process for creating a finished part using AM typically involves the following steps (shown in Figure 1.6):

1. A computer-aided design (CAD) software is used to design the object, including its geometry and characteristics.

2. The CAD design is converted into an STL (Stereo Lithography interface or Standard Triangulation Language) file, which represents the design as a triangular mesh.

3. The STL file is used to guide the additive manufacturing process, which builds up the object layer by layer according to the triangular mesh representation.

4. The end part is refined, polished, and cleaned to obtain the final object.

In the early days of 3D printing, the focus was mainly on using thermoplastics as the layering material, particularly for rapid prototyping. However, the range of materials that can be used in 3D printing has expanded significantly to include ceramics, composites, glass, bio-inks for creating artificial organs and tissues, and even edible substances. Among these, metals are the most widely used materials in industrial applications after polymers due to their favorable mechanical properties. The metals that are most commonly employed in 3D printing include titanium, steel,

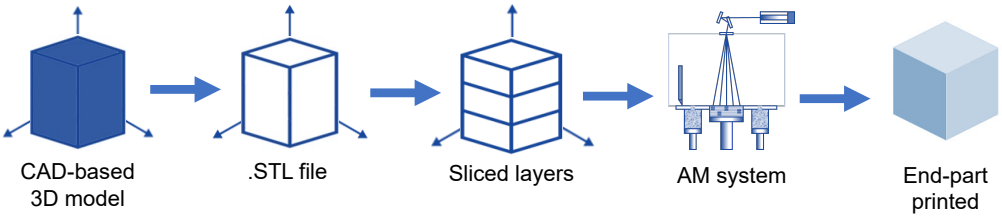


Figure 1.6 AM process flow: from CAD to final object

stainless steel, aluminum, copper, cobalt chrome, titanium, and nickel-based alloys, as well as precious metals like gold, platinum, palladium, and silver [124].

Based on the material used, AM can be distinguished into different categories. Figure 1.7 reports the main 3D printing technologies and the materials that are typically used with each. Stereolithography, Multi Jet Fusion, and Selective Laser Melting are primarily intended for plastic materials, while Fused Deposition Modeling and Laminated Object Manufacturing can be used with both composite and plastic materials. Laser Powder Bed Fusion and Electron Beam Powder Bed Fusion, on the other hand, are only suitable for creating metal products.

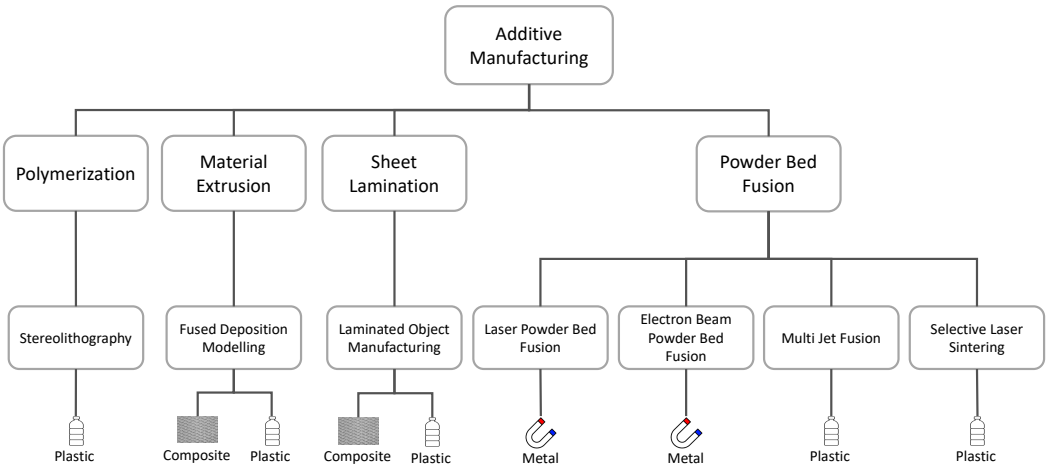


Figure 1.7 AM technologies

The *polymerization* process involves using ultraviolet light to transform a liquid plastic polymer into a solid form.

- *Stereolithography* is a 3D printing process that uses ultraviolet light or an electron beam to initiate a chemical reaction in a layer of photosensitive polymer [92]. It is often necessary to subject the printed parts to a post-processing treatment such as heating or photo-curing in order to achieve the desired mechanical properties. This process is known for producing high-quality parts with a fine resolution as low as $10\mu m$ [209]. However, stereolithography is relatively slow and costly and is only suitable for a limited range of materials, such as resins that undergo structural changes after intense exposure to ultraviolet light.

Material Extrusion is a process that involves using a heated nozzle to extrude a continuous stream of material from a spool, which is used to build a 3D object layer by layer.

- *Fused Deposition Modeling* (FDM) is a 3D printing process that uses a continuous filament of thermoplastic polymer to create an object. The process involves extruding a heated thermoplastic filament through a nozzle tip, which deposits the material on the platform to build the part layer by layer [42]. One advantage of FDM is that it does not require any chemical post-processing or curing, which helps to reduce costs. However, the process is not capable of achieving high resolutions ($< 0.25mm$) compared to other AM processes.

Sheet lamination is an AM methodology that involves bonding thin sheets of material together to form a single piece, which is then cut into the desired 3D shape.

- *Laminated Object Manufacturing* (LOM) is one of the first commercial 3D printing methods to be developed. It involves cutting and bonding sheets or rolls of material layer by layer. This process is useful for reducing the cost of tooling and manufacturing time and is often used to create larger structures [158]. However, LOM has some drawbacks, including lower surface quality (without post-processing) and lower dimensional accuracy compared to powder bed methods.

In *Powder Bed Fusion* (PBF), a layer of powder material is spread over the previous layers using a recoater while a reservoir supplies new material as needed. A heat source, such as a laser or electron beam, is used to selectively melt the powder layers together [22].

- *Laser Powder Bed Fusion* (L-PBF), also known as Direct Metal Laser Sintering (DMLS), is a PBF 3D printing technology that is widely used in AM. It allows for the creation of metallic parts with a density of up to 95% without requiring any additional post-build sintering [104]. L-PBF was developed by EOS GmbH and was first introduced in the EOS M250 machine in 1995. The process involves exposing metal powder in a liquid state to a laser in order to sinter it. In Figure 1.8 is reported a scheme of the L-PBF machine. The chamber where the process takes place is filled with an inert gas to prevent the powder from oxidizing. The powder bed is heated to reach the melting point of the material and is controlled by a piston that is lowered by the same amount as the layer thickness each time a layer is completed. The powder is spread over the bed using a recoater while, at the same time, excess powder is removed. The laser then fuses the powder at specific locations based on the design of each layer. L-PBF is currently the most widely used technique for creating metal parts due to its balance of printing accuracy and cost [216].

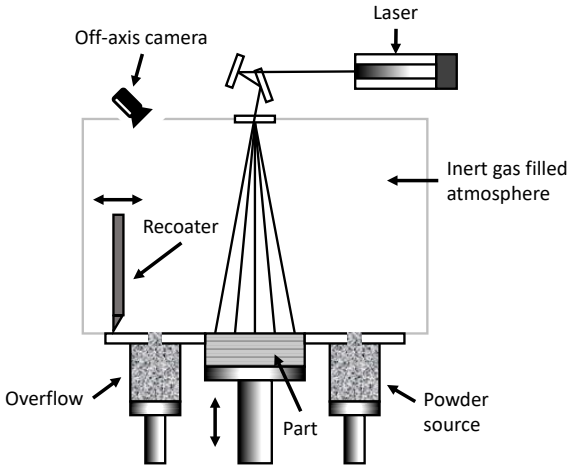


Figure 1.8 Laser Powder Bed Fusion setup

- *Electron Beam Powder Bed Fusion* (EB-PBF) is a relatively novel 3D printing process that uses a high-voltage electron laser beam to melt powder [108]. The process is similar to L-PBF, but the part is printed in a high vacuum chamber to prevent oxidation. The powder is fused using an electron beam guided by some lens in order to build the part, as shown in Figure 1.9. The high power of the laser beam used in EB-PBF results in greater penetration depth in the powder bed compared to L-PBF. However, the high power can also lead to the

formation of cracks on the surface of the material, reducing the stability of the process.

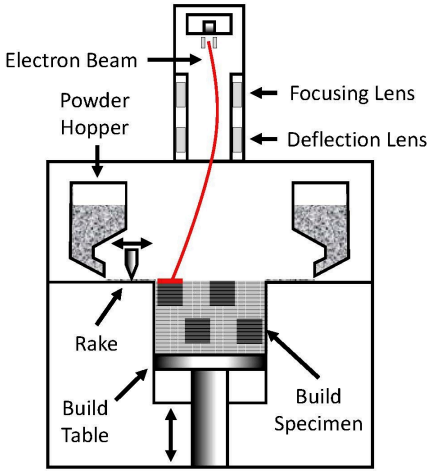


Figure 1.9 Electron Beam Melting setup

- *Multi Jet Fusion* is an additive manufacturing method developed by Hewlett-Packard that uses a multi-agent printing process to create parts [186]. It is particularly useful for creating unique plastic parts with a good surface finish. However, the method is only compatible with a limited number of plastic materials.

- *Selective Laser Sintering* (SLS) is a 3D printing process that is similar to L-PBF, in which a laser beam is used to sinter or fuse a powder material. While L-PBF is limited to metal powders, SLS can be used with various polymers. One disadvantage of SLS compared to L-PBF is that the accuracy is limited by the size of the particles in the material. However, SLS allows for fine resolution and high-quality printing of complex structures [1].

This thesis will focus on two AM technologies: L-PBF and EB-PBF. I will show the possibilities provided by the collection and analysis of data from smart manufacturing systems in the Industry 4.0 context.

Overall, the use of AM in the industry is expected to lead to greater efficiency, productivity, and flexibility, as well as reduced waste and costs. The opportunities provided by these technologies, together with the huge amount of data available, will allow to revolutionize the market, provided that companies will manage, collect and

analyze the data correctly with a suitable framework, machine learning, and artificial intelligence systems.

All the benefits provided by the integration of these systems (AM, IoT, cloud computing, and CPS systems) align with the principles of Industry 4.0. They are likely to play a significant role in the future of companies.

1.1.4 Artificial Intelligence and Machine learning for smart manufacturing

As of today, many industrial systems do not have the ability to evaluate the quality of the products they create without performing costly and time-consuming post-processing analysis. To address this issue, some companies started to develop Artificial Intelligence (AI) and Machine learning (ML) software for real-time visualization and monitoring of various processes in Industry 4.0 to reduce the overall time and cost of production [212].

Although data processing capabilities are still limited, with the retrofitting of previous machines and the improvement to the new ones, many of the current systems on the market have the ability to track the history and behavior of key machine parameters throughout the entire production process. For example, in the AM field, current systems can be equipped with advanced hardware such as high-speed and infrared cameras, thermocouples, pyrometers, and photo-detectors, which enable the monitoring of various process parameters that can affect the quality and integrity of the finished product [49]. By utilizing these sensors, it is possible to gather different types of data from the Industry 4.0 factory.

In this context, the data can be classified into four major categories (Figure 1.10):

- *Structured*: these data are organized and formatted in a specific way, usually in a table or database, and are easy to search and filter. They are well-suited for use with machine learning algorithms. Examples of structured data include records of production runs with information about materials and process parameters organized in a table or spreadsheet regarding post-process analysis.
- *Semi-structured*: they are data that have some structure, but not as much as fully structured data. They typically include both structured and unstructured

elements and may be more difficult to process and analyze than fully structured data. In the context of additive manufacturing, semi-structured data could include sensor readings that are not organized in a strict table or database structure or inspection reports that contain both structured fields (such as pass/fail results) and unstructured text or images.

- *Quasi-structured*: these types of data appear to have some structure, but that structure is not consistently applied, and it is not easily recognizable, actually making them more difficult to process and analyze than fully structured or semi-structured data as it may not fit neatly into a table or database structure. Examples of quasi-structured data include web pages, PDF documents, and other types of documents that contain a mix of structured and unstructured elements but where the structure is not consistently applied.

- *Unstructured*: These data are not organized in a specific format and do not fit neatly into a table or database structure. They are more difficult to process and analyze than the other types of data, as they do not have a consistent or easily recognizable structure. Examples of unstructured data include process images, notes, and analysis. In general, unstructured data may require more specialized processing and analysis techniques in order to extract useful information.

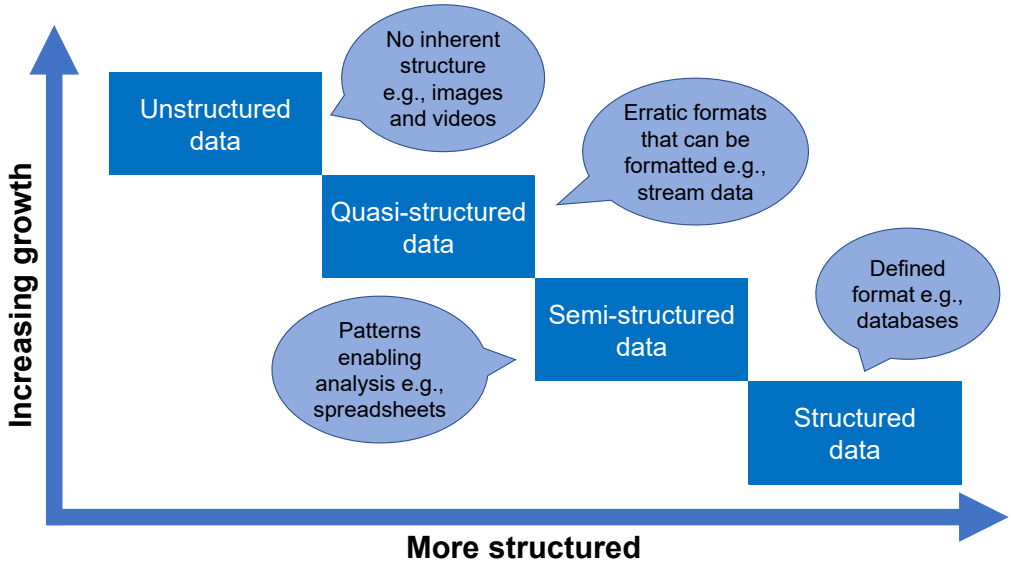


Figure 1.10 Data types structure

The analysis of all these data often involves combining offline ML techniques with other algorithms, depending on the type analyzed [205, 144, 19].

For example, by using optical sensors to capture images during AM processes, it is possible to apply machine learning techniques in combination with computer vision algorithms to perform defect analysis in additive manufacturing [28]. In other cases, researchers tried more complex algorithms, they combined several data from the process like laser system information, powder bed, and recoating phase, and they analyzed them using a Support Vector Data Description model to identify significant deviations from the expected pattern, such as cracks or holes on the surface [77]. However, one downside of these methods is that they often involve making invasive modifications to the machine.

Recent literature demonstrates that data-driven techniques are quite promising in order to improve smart manufacturing processes. Nonetheless, the application to real-world industrial scenarios is often limited; indeed, ML techniques, particularly deep learning, typically require a large amount of annotated data for training. For example, in image-based manufacturing systems, a large number of images with different defects and conditions are needed. However, it is not practical in the industry to intentionally produce defective parts for training purposes, as it is costly and time-consuming.

In this context, ML is particularly useful in addressing this issue. It can be used to generate synthetic and realistic images that mimic those obtained during the printing of a defective part [5].

One promising approach for generating synthetic data is the use of Generative Adversarial Networks (GANs). GANs are generative models that consist of two neural networks, a generator, and a discriminator, who are trained to compete with each other in order to generate realistic data [53]. There are several types of GANs that have been proposed in recent literature, and among these, Conditional GANs (CGANs) have been found to be particularly effective in many data augmentation tasks [73]. Nevertheless, they still require a large amount of data for training. To address this issue, another type of architecture called ConSinGAN represents a promising alternative that is able to learn generative models from a single training image [86]. This allows for the creation of synthetic data without the need of a large dataset.

Overall, there is significant potential for ML and AI to improve and revolutionize the field of additive manufacturing and, in general, smart manufacturing and Industry 4.0.

In the next Chapters, this thesis presents how ML and AI could be implemented in different industrial scenarios to improve the overall production processes. Several ML and AI services were integrated together with the management framework in order to analyze the data available from automotive and aviation smart manufacturing production lines. The integrated algorithms showed improvements to the process and the reduction of the number of defects in the parts produced. Moreover, a ConSinGAN network was applied in order to produce synthetic data and increase the available dataset.

1.2 Contributions

In recent years, research about smart manufacturing and Industry 4.0 has gained strong momentum, also pushed by European and national funding initiatives. The primary purpose of this great effort is to develop new smart processes in order to improve the production plants, increase the performances and reduce human operation and subjectivity, overall energy consumption, and production costs.

In this scenario, monitoring frameworks, Cloud Computing, Internet of Things (IoT), Cyber-Physical Systems (CPS), Additive Manufacturing (AM), Artificial Intelligence (AI), and Machine Learning (ML) all together play a crucial role in improving the overall Industry 4.0 and moving forward to a more sustainable and profitable smart factory.

My research focuses on developing a platform monitoring framework through cloud computing that allows to collect and manage all the data provided by IoT devices connected to industrial and CPS systems like AM machines, even if scattered around the world. Furthermore, all the data collected could be analyzed by AI and ML algorithms integrated as platform services to discover hidden relationships and improve energy efficiency and production processes.

Consequently, I applied my knowledge and developed several systems for four different Industry 4.0 scenarios: i) Smart Industry, ii) Smart Manufacturing, iii) Smart Sensors integration, and iv) Smart Energy.

After the development of a common platform for the collection of data, all the scenarios were conducted following a base-line structure with a bottom-up approach starting from the analysis of the plant or process environment, a setup of IoT devices for the collection of data, and an appropriate pre-processing of them, to finally arrive at the design of models suitable for the type of dataset and its characteristics. The common structure is shown in Figure 1.11.

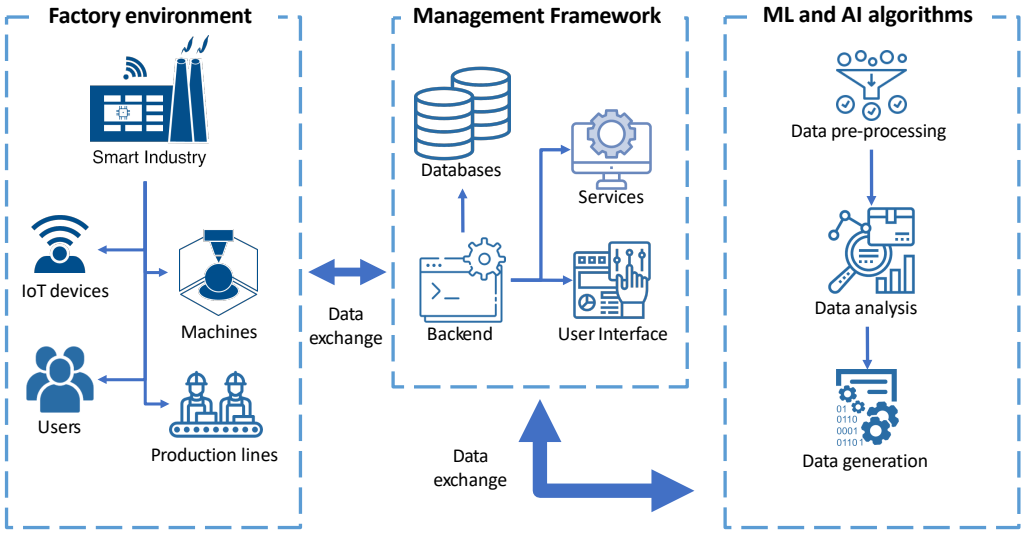


Figure 1.11 Common structure

1.2.1 Smart Industry: Digital platform for production line in smart manufacturing

Focusing on a production line in the automotive industry, I developed a digital platform for the collection and management of data in a smart manufacturing context.

In detail, I analyzed the machines available, the type of data, and the exchange protocols, together with the user experiences and requirements, in order to select and deploy the most suitable infrastructure for the use case scenario, as deeply illustrated in Chapter 2.

The platform was built using state-of-the-art open-source software and deployed both in a research and industrial environment. To retrofit the machines available,

some IoT devices were deployed and connected to the machines in order to exchange data with the platform by means of popular protocols like HTTP and MQTT.

The results showed that the adoption of the platform could significantly increase user performance and improve the management and analysis of data in a production plant.

1.2.2 Smart Manufacturing: Online monitoring and post-production inspection in additive manufacturing processes

In the context of Smart Manufacturing, I implemented an innovative system for the in-situ monitoring of the additive process and the subsequent analysis of the data and quality inspection. I integrated in a Laser Powder Bed Fusion (L-PBF) printer, an IoT device connected to the camera inside the machine chamber, in order to monitor the powder bed and collect images of each layer printed before and after the printing is done.

The aim is to take images of the powder bed during the process and collect them in the management framework in order to be analyzed. After that, I applied several algorithms that analyze the data available to detect possible defects during the printing process and create synthetic data with GANs starting from the real ones available. These algorithms were integrated as services in the framework developed in the Smart Industry scenario. The methodology can be easily applied to other types of machines and requires only the equipment of an IoT device for the collection of the data.

After the L-PBF process, I started collecting data from another additive manufacturing process called Electron Beam Powder Bed Fusion (EB-PBF). In this case, the data come from all the sensors installed in the machine, like process parameters, printing phase information, and so on. All these data can be collected and stored in the management framework using an IoT device connected to the machine. Then, I analyzed all the information and integrated a set of ML algorithms to predict the future status (failure or success) of a printing job in real-time. A detailed explanation of the methodologies applied to L-PBF and EB-PBF is reported in Chapter 3 and Chapter 4, respectively.

The results showed that new functionalities could be easily added to previous-generation equipment through the integration of IoT devices in the production plant, besides the algorithms implemented demonstrate the potential of the methodologies and showed promising results in detecting different categories of defects during the powder bed fusion process and predict the status of a printing job by means of machine learning, artificial intelligence, and computer vision algorithms.

Lastly, seeing as the ultimate goal of Industry 4.0 is the automation of processes. I also focused my research on the analysis of quality inspection procedures utilized for aircraft engine components built with AM technology in order to reduce the whole inspection time, ensure higher quality, and guarantee standardization and process control.

The proposed solution is a two-phase process fully controlled by a robot, leveraging a camera as well as a purposely designed optic fibers sensor coupled with AI algorithms for the detection of different types of defects. To assess the functionality and validity of the proposed solution, a fully functioning prototype was developed. A detailed explanation of the challenges faced during the development and the advantages provided by the automation of the quality inspection are deepened in Chapter 5.

1.2.3 Smart Sensors integration: Industrial indoor localization

In the industrial context, the localization of industrial equipment inside the plant is an important aspect for the correct management of the assets available, especially with the advent of Industry 4.0 and an increase in connected objects.

In this context, I evaluated the effectiveness of several state-of-the-art algorithms for the localization in industrial environments of Bluetooth Low Energy (BLE) IoT devices under varying experimental parameters.

The main goal is to assess the feasibility of indoor localization using commercial off-the-shelf BLE transmitters and receivers due to their low cost, ready availability, and ease of installation and usage. The localization algorithms tested are based either on trilateration or environment fingerprinting combined with machine learning-based regressor (k-Nearest Neighbors, Support Vector Machines, or Multi-Layer Perceptron) in different types of industrial environments.

The results showed that fingerprinting-based methods reduce the positioning error compared to distance-based methods, even in an industrial setting. However, it is not possible to identify a clear winner among the fingerprinting-based algorithms considered in the experiments, as they all perform similarly both in terms of positioning error and computational complexity. A detailed explanation of the analysis performed and the results are reported in Chapter 6.

1.2.4 Smart Energy: Energy efficiency in Industry 4.0

Lastly, I applied machine learning also in the renewable energy production field, which represents a key topic for energy efficiency in Industry 4.0.

In detail, I proposed a methodology to perform short-, mid-, and long-term Global Horizontal Irradiation (GHI) forecasting useful for photovoltaic energy predictions, as deeply illustrated in Chapter 7.

The experiments were conducted based on weather data (GHI, temperature, pressure, etc.) measured by a meteorological station located at Politecnico di Torino. They consist of a Variational Mode Decomposition (VMD) to pre-process the data and two Convolutional Neural Networks (CNN) together with Random Forest (RF) or Long Short Term Memory (LSTM) to predict the GHI.

The results showed the potential of the methodology developed. It allows to predict GHI in short- mid-, and long-term and provides insight regarding the management of PV energy production. In the industrial context, this is of particular importance to know the amount and type of energy available in order to establish adequate energy policies and correctly manage production loads and machines available.

Chapter 2

Digital platform for production line in smart manufacturing

2.1 Introduction

As introduced in Chapter 1.1.3, Additive manufacturing (AM) is the industrial name of 3D printing, referring to a large family of computer-directed technologies that grow an object by joining materials layer upon layer starting from a 3D model. This process is typically considered one of the pillars of the Industry 4.0 revolution, and it is pushing forward innovative designs and applications in highly competitive fields like aerospace, automotive, and medical. In aerospace and automotive, AM is helping suppliers and companies to develop consolidated, lightweight components that lead to more efficient vehicles. In the medical industry, manufacturers are taking advantage of a wide range of high-strength and biocompatible 3D printing materials to customize designs and create functional prototypes, true-to-life anatomical models, and surgical grade components [177].

While the advantages compared to traditional technologies are making AM more and more popular, there are still considerable improvements to be made in terms of process repeatability and part qualification before AM can be applied to mass production. Due to the complex interaction of many machine and process parameters at different scales, the layering process generates a variety of defects that may compromise the geometrical and mechanical properties of the fabricated

object, making expensive and time-consuming post-process qualification analysis a necessary step for most applications.

To overcome this problem, in the last few years, there are growing efforts towards the design of *monitoring systems* to detect or even correct the arising defects at an early stage during the layering. The process information can be monitored in-situ by a multiplicity of heterogeneous sensors, IoT devices, and cameras embedded into the AM machine, providing a very large stream of data that can be effectively used to characterize the fabrication quality [142, 148, 170]. The collection and interpretation of these data open the way not only to the early detection of many defects but also to fine control of the process to avoid the generation of defects altogether. In this context, data analytics approaches, mostly combining Machine Learning (ML) methods with other approaches targeting specific types of sensors (e.g., Computer Vision and signal processing), are playing a very important role [144].

Nonetheless, while the possibility to generate large amounts of sensing data is opening new unforeseen possibilities, to this date, the application of data-driven monitoring approaches in real industrial settings is still limited by major issues related to the efficient retrieval, storage, visualization, and systematic analysis of the data [49]. In this context, several challenges can be highlighted:

1. *AM system HW/SW monitoring equipment*: To this date, most industrial AM systems do not provide the hardware and software equipment to assess the quality of the fabricated products during the layering. In the standard cases, even though multiple sensors are present in the machines, data interpretation is very limited (for example, based on fixed empirical thresholds on individual sensors triggering warning or error messages), and the raw sensed data is not made available to the users. In recent years, more companies are providing commercial software for real-time visualization and monitoring of several process parameters. Nonetheless, these solutions are generally limited in their scope, as they do not develop a fully automated defect detection strategy, failing to detect even minor defects in the printed part that could be easily corrected before another layer is built [35].

2. *Heterogeneous data integration and storage*: A single AM build can potentially generate huge amounts of heterogeneous data (in the order of terabytes for a standard PBF process). To fully characterize a process, useful information may

include not only the output of heterogeneous sensors and cameras mounted in the machine but also different sources like machine parameters and settings, powder characteristics, design information such as CAD models and STL files, as well as post-processing test outcomes. These data may be either structured (e.g., images acquired by in-situ cameras) or unstructured (e.g., reports of post-processing analysis) and have different formats and granularity. This, added to the lack of a common reference data structure and of Application Programming Interfaces (APIs) for accessing and handling the information from different sources, poses huge challenges to data collection, storage, synchronization, and correlation [185].

3. *Training data collection and annotation*: Recent literature shows the many potentials of using data-driven approaches (either standard Machine Learning, Artificial Neural Networks, or Deep Learning) to the monitoring and control of different types of AM processes, including PBF. Nonetheless, ML typically requires the availability of a large amount of training data, which is often unfeasible in a real industrial setting due to difficulties and costs of storing and appropriately annotating the training samples [101]. On these premises, even though they show interesting results, most of the available literature approaches are typically no more than proofs-of-concept, very difficult to replicate in a real industrial setting, and with very limited scope. Moreover, the lack of public benchmarks and/or datasets makes it very difficult to effectively assess the accuracy and generalization capabilities of the proposed solutions.

In this scenario, we deal with the problem of machine retrofitting and data collection for in-situ defect monitoring of L-PBF process, taking all the challenges mentioned above into account.

We start by analyzing the AM fabrication process from design to post-processing to provide a clear overview of the data involved and of their corresponding structure and characteristics. Then, we propose a scheme of a distributed AM monitoring platform tackling all these data. To ensure the flexibility and scalability of the system, our platform is designed with a microservice software design pattern, with discrete services implementing specific functions which can be built, deployed, and scaled independently. Each microservice communicates with the others through standardized APIs and protocols, enabling the integration of different technologies and the addition of new components in a plug-and-play fashion.

On these premises, we deal with the specific problem of in-situ PBF defect detection as a service component of our distributed microservices platform, entailing all the phases of the process. More specifically, we propose a set of fully-automated Computer Vision and Machine Learning techniques to detect five different categories of powder bed defects that are hard to spot by visual inspection and to monitor the profile of the fabricated objects in real-time. To address the problem of the lack of annotated images, we add to the framework an additional synthetic image generation module that uses ML, and more specifically, Generative Adversarial Network (GAN), to generate realistic images with defects similar to the ones occurring during a typical PBF build. These services are explained in detail in Chapter 3.

The rest of this Chapter is structured as follows. Section 2.2 provides an overview of the state-of-the-art of monitoring systems for AM, with specific regards to AM data collection/generation. Section 2.3 describes the structures and types of the data involved in the process, Section 2.4 reports our proposed solution of a digital platform for production lines in smart manufacturing, and Section 2.5 discusses the concluding remarks. Some of the work described in this Chapter was also previously published in [31].

2.2 Background

With the growth of industry 4.0 and the increasing role of AM across multiple sectors, future developments are expected to lead to a closer interrelationship between AM and the other Industry 4.0 pillars, with special regard to Internet of Things (IoT), cloud computing, and big data analytics [26]. In a production plant, connected machines and sensors generate a large volume of data during manufacturing processes. To turn these data into useful information and knowledge to pose the basis for smart manufacturing, they need to be cleaned, stored, and processed [193]. However, the large amount of data generated makes it difficult for users to process it.

In this context, the development of cloud-based platforms for AM process monitoring, with IoT devices and machines communicating with each other, becomes fundamental to fulfill the potential for mass production and to store the data available for further analysis [180]. This is indeed a very hot problem that is destined to attract ever-increasing attention from the research community. In [13], the authors analyze the benefits of integrating IoT devices with the AM process, coming to the

conclusion that cloud computing and IoT can improve the efficiency of the AM processes and reduce manufacturing waste while fulfilling customer specifications. In [210], they highlight the limitations of the current cloud-based platforms, which fail to support the customers throughout the AM process and propose a solution supporting remote printing as well as design and process planning.

With this regard, another problem that arises concerns the retrofit of machines and how to collect and analyze different types of data to monitor the AM process and improve the overall quality of the system. Thus, also due to the strong motivation to meet industrial requirements in terms of product quality and reduced post-processing analysis, even the specific problem of in-situ AM monitoring, which is poorly addressed by commercial solutions [57, 194], is attracting more and more attention in the literature. In the last few years, researchers applied more and more data-driven approaches, with specific regard to ML methods, to different types of sensed data to obtain automated detection of a number of defects in AM fabricated objects [18, 229, 127]. Nonetheless, the problem of in-situ monitoring is mostly addressed as if it was independent from the one of AM heterogeneous data collection/integration, which considerably limits the scope and real applicability of the proposed solutions. Extensive research is still needed to integrate the in-situ monitoring techniques within a standardized *digital thread* for AM, where ML algorithms can directly access and create a meaningful correlation between design, process parameters, and post-process analysis.

Among the proposed monitoring solutions available in the literature, visual and camera-based methods have been extensively used to collect data and identify PBF processing errors, such as powder bed discontinuities and geometrical defects [226]. Nonetheless, a typical problem is the lack of generalization, as the proposed solutions are ad-hoc for specific machines and experimental setups. Other solutions exploit images acquired by high-speed cameras to analyze the characteristics of the by-products and of the laser-induced vapor plume of Laser Powder Bed Fusion (L-PBF) during the powder melting, based on the hypothesis that these characteristics can be correlated with micro-structural and mechanical defects of the fabricated objects [76, 226]. Nonetheless, these solutions are majorly affected by the disturbance caused by vapor plume ejection, and they require expensive equipment that is typically not available in a standard industrial setting. Finally, a lower number of solutions exploit low-cost thermal imaging, relying on relative temperature measurements to

detect unwanted process variability, even though the proposed techniques are not real-time [8].

In general, as already mentioned in Section 2.1, a common limitation of all the data-driven techniques is that they typically require to be trained on a large amount of annotated data (e.g., for image-based systems, a large number of images of the powder bed per each possible defect and condition). As the intentional generation of defective parts is not viable in industry, as it is costly and time-consuming, the availability of suitable datasets for training and validation purposes becomes a critical issue. A promising and yet mostly unexplored solution to this problem, at least for camera-based monitoring systems, is the use of a Generative Adversarial Network (GAN [75]), a generative model characterized by training a pair of neural networks (a generator and a discriminator) in competition with each other to create realistic synthetic images. While GANs are becoming very popular in many Computer Vision tasks, their application to the AM sector is still extremely limited. In [113], the authors analyze the benefits and challenges of GANs to generate new data for ML algorithms. In [73], the authors apply a Conditional GAN (CGAN), an extended form of GAN [146], to produce synthetic data using images from a near-infrared high-resolution optical camera, obtaining quite promising results. In our work, proposed in the next Chapter, a similar approach is applied to a more complex task involving a large number of PBF defects, using a standard low-cost visible-range camera for image acquisition. For this purpose, our solution exploits a ConSinGAN [86], an improved architecture that requires a single real powder bed image for the training.

2.3 Data structure

For the development of a digital platform for AM monitoring, it is important to understand the source, type, and structure of the data to be stored and their role in the AM process. To do so, a typical AM pipeline was analyzed from CAD generation to post-processing analysis, with special regard to L-PBF processes. All the involved data are divided into five categories based on the AM life-cycle process: i) design parameters, ii) process parameters, iii) process characteristics, iv) post-processing treatments, and v) product analysis. Table 2.1 reports a list of data in each category.

Table 2.1 Main data in the Additive Manufacturing process

Category	Type of data
Design parameters	Design features (CAD, Part material, Roughness, etc.) Build features (Supports, build position and orientation, etc.)
Process parameters	Laser parameters (Type, Operational mode, Frequency, Width, Power, Scan settings, etc.) Material properties (Type, Conductivity, Melting temperature, Particle structure, etc.) Powder bed properties (Powder density, Thermal conductivity, Heat capacity, Temperature, etc.) Recoater properties (Type, Speed, Powder diffusion, Layer thickness, etc.) Build environment properties (Gas type, level and characteristics, pressure, temperature, etc.)
Process characteristics	Melt pool (Temperature and geometry, etc.) By-products (Plume, Spatter, etc.) Laser track (Width, Continuity, Depth, Defects, etc.) Layer (Surface topography, Temperature, Defects, etc.)
Post-processing treatments	Heat treatment (Technique, Temperature, Atmosphere, Heating and cooling time, etc.) Post Machining (Machining operation, Velocity, Feed rate, Tool characteristics, Environment, etc.) Powder remove treatment (Vibration settings, Time, etc.) Other processes (Painting, Engraving, Process characteristics, etc.)
Product quality analysis	Geometry and dimension analysis (Accuracy, Precision, etc.) Surface quality (Roughness, Deformation, Defects, etc.) Physical properties (Density, Porosity, Microstructure, etc.) Mechanical properties (Yield and Tensile strength, Elongation, Fatigue, Hardness, etc.)

1. *Design parameters*: Design parameters refer to all the parameters defined during the design of the product to be printed in the first phase of the AM process. These parameters are of particular importance for the correct build of the object and its quality. In this step, the data range from the design features, including the CAD file and material of the produced part, to design constraints like the required surface roughness or dimensional tolerances. In this context, also the position of the supports plays a fundamental role, and the data regarding the type of supports (geometry and shape), the build position, and the build orientation need to be stored.

2. *Process parameters*: Process parameters involve the input settings selected during the setup of the machine. They are machine and technology dependent, and they are mainly set at the start of the process and typically remain constant throughout the build. The process parameters can be related to the material used, such as the material type, composition, melting point, heat capacity, etc., or to the powder bed composition, like powder density, thermal conductivity, temperature, etc.. Other process parameters may include the machine settings like the type of laser, its operational mode or frequency, speed and power, the build environment gas type, level, and characteristics, or the recoater properties such as speed, layer thickness, powder diffusion, etc..

3. *Process characteristics*: They refer to the data that can be used to characterize the printing process during the build, either acquired as-is or derived by sensors mounted on the machine. They are highly influenced by the process parameters, and their number/nature depends on the specific sensing equipment of the AM system. For an L-PBF process, process parameters can be obtained at different levels of observation, from the melt pool (the small pool of metal liquid that is generated when the laser heats the powder) up to the whole powder bed layer. Data include temperature measures and geometrical characteristics obtained with thermal cameras, by-product measurements obtained from IR imaging like spattering and plume, laser track characteristics, as well as whole layer properties, including surface topography, temperature, etc. Process characteristics at this level also involve visible-range images acquired to characterize the morphological and geometrical characteristics of the powder bed as well as of the printed object.

4. *Post-processing treatments*: Post-processing treatments refers to the post-build processes required to obtain the final product characteristics. The post-processes could change based on the specific design requirements and treatment type. For example, the post-process heat treatment data used to achieve the desired hardening or softening of the material involve technique, temperature, atmosphere, heating, and cooling time utilized. Other possible treatments are post-machining (involving the type of operation, velocity, tool material, and geometry, etc.) and powder remove treatments characterized by vibration techniques, time, etc..

5. *Product analysis*: Product analysis refers to the final quality tests of the AM fabricated parts, performed to verify that all the characteristics of the object obtained after the post-processing treatments comply with the design requirements as well as to the industry standards. The outcomes of these tests define or characterize the success of the build. Hence, they are of fundamental importance for the future improvement of product quality, for design optimization as well as for the training of supervised ML models. In this context, data include dimensional and geometrical accuracy, surface roughness, porosity, mechanical properties, chemical properties, etc..

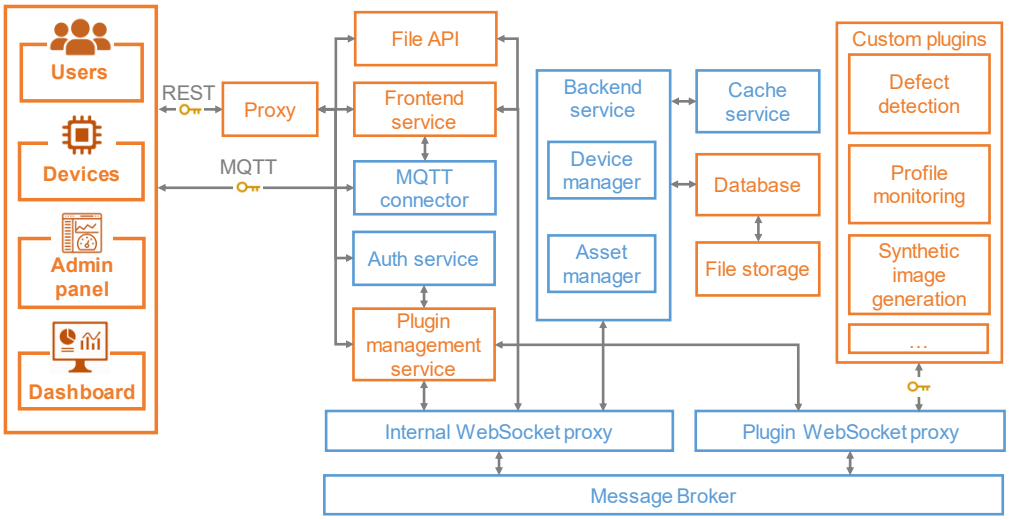


Figure 2.1 Digital platform: proposed architecture with the base infrastructure (in blue) and the customization applied (in orange)

2.4 Digital platform structure

All the categories described in the previous Section involve different types of data (textual, numeric, CAD, images, reports, etc.) that need to be monitored, exchanged between different assets, and finally stored. To do so, we propose the AM distributed monitoring platform scheme reported in Figure 2.1, which is an extension and customization of a general-purpose, open-source industrial framework called DeviceHive [48].

As mentioned previously in Section 2.1, in order to ensure the flexibility, modularity, and scalability of the platform, the infrastructure is deployed as a PaaS service using an open-source software called Docker. This is done on two separate servers: one located locally in the production plant of an automotive company and the other remotely, serving as a fallback solution on the Politecnico di Torino High-Performance Computing (HPC) infrastructure. The platform is composed of several microservices components that allow to: i) manage different users and devices, ii) receive, monitor, and store the data available, iii) provide API for internal and external communication, even with third-party software and iv) integrate new functionalities (e.g., in-situ defects detection, predictive maintenance, etc.) through plug-and-play components.

We implemented and customized several modules of the platform (highlighted in orange in Figure 2.1) in order to develop a framework suitable for smart manufacturing use cases.

First of all, we used the default *Message Broker* together with the *Internal* and *Plugin WebSocket proxy* that are in charge of managing all the communication between the platform components, using several protocols like Message Queuing Telemetry Transport (MQTT) and Representational State Transfer (REST) web services. MQTT is a communication protocol that implements the publish/subscribe paradigm [149], while REST [62] is a software architectural style implementing request/response. REST is commonly used to create interactive applications that use Web services. The role of the WebSockets proxy in the platform is to provide a reliable and secure connection between the client application and the server, even in situations where the network infrastructure restricts or blocks the WebSockets traffic. The proxy is responsible for handling WebSocket connections, routing messages, and managing WebSocket sessions. By using the WebSockets proxy, clients can establish a persistent, bi-directional communication channel with the server, enabling real-time data exchange, event notification, and command execution.

The *Proxy* and *MQTT connector* manage the communication between the platform and the external assets. The role of the proxy module is to act as a reverse proxy, routing client REST requests to the appropriate assets and caching frequently requested resources to improve performances, while the MQTT connector allows communication between the platform and the devices using the MQTT protocol. They both are connected to the *Frontend service* that provides access to the resources through a dashboard, shown in Figure 2.2, written with a state-of-the-art JavaScript library called React and allows the management of the users using an *Admin panel*. The *Auth service* is in charge of managing the authentication of users and devices in order to secure the data transferred and stored.

The *File API* module is developed from scratch and contains all the API required to manage the files stored in the databases (e.g. images, PDF, CSV) and retrieve information from them.

The *Backend service* is used to internally manage the databases and devices available using two modules: the *Asset Manager* and the *Device Manager*. The *Asset Manager* is a software module that handles all the information regarding users and devices (e.g. the user authorization, the process data, and the device location).

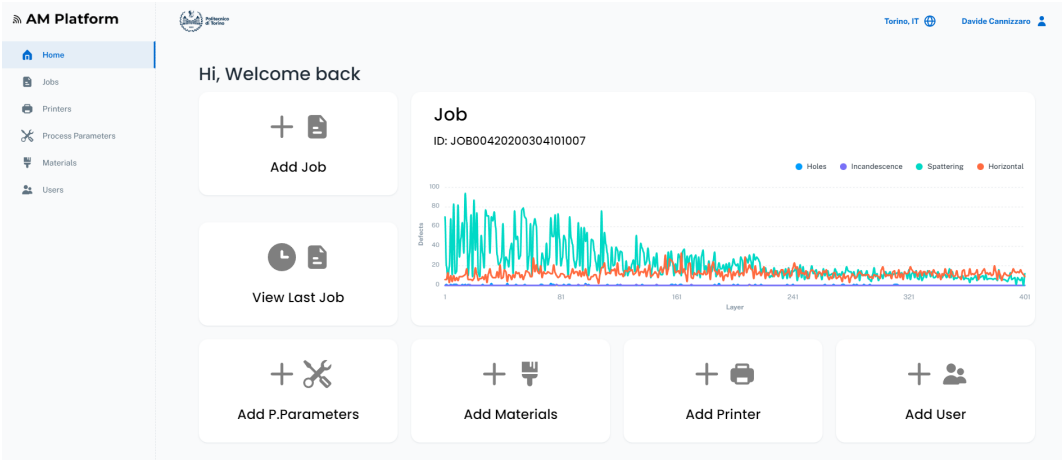


Figure 2.2 User interface dashboard

The *Device Manager* provides the connection between the assets, the databases, and the plugins that require to store or retrieve the data available. The Backend is also supported by the *Cache service* that stores in memory the information about the most requested data to make them available quickly.

The *Database* together with the *File Storage* service integrates different databases based on the type of data involved. A relational database, called PostgreSQL, is used to store non-time series data, while a non-relational database, called Cassandra, is employed to store time-series data. The platform could be easily extended to integrate other types of databases if required.

The *Device module* provides a crucial component called *Device Connector* to integrate heterogeneous IoT devices into the platform, providing functionalities to: i) provision new devices exploiting the Asset Manager; ii) enable a secure and trusted communication through the Auth Service module; iii) allow a bidirectional communication in MQTT and REST thought the MQTT Connector and Proxy service, respectively; iv) send and store newly sampled data in the remote database via the Device Manager; v) execute plugins on edge and manage their communications with the *Plugin management service*.

Finally, the *Custom Plugins* are the additional components that can be added to the system in a plug-and-play fashion. They are integrated into the platform using the *Plugin management service* that allows communication with both internal and external resources besides providing the data required. The plugins can implement different functionalities that are typical of an Industry 4.0 setting: for example,

anomaly detection, predictive maintenance, in-situ defect detection, part profile monitoring, etc.

As previously mentioned, the proposed AM distributed monitoring infrastructure can be developed in cloud systems and offered to companies as Platform as a Service (PaaS) [118]. PaaS is a category of cloud computing services that provide customers with a platform to build, run, and manage applications without the need to build and maintain the infrastructure. PaaS solutions can be alternatively provided as a public, hybrid, or private cloud, thus, reducing the data exchange over the Internet and increasing privacy and data management. Thanks to these features provided by the distributed platform, plugins can be developed to be either executed remotely on the cloud or locally on edge.

In the next Chapter, we will deepen AM manufacturing process and analyze in detail specific ML applications related to the platform and the in-situ monitoring of an L-PBF machine. These include a description of the hardware and setup used to retrofit the machine and three functionalities integrated (and corresponding custom plugins): the automated detection of the powder bed defects, the real-time monitoring of the object profile during the layering as well as the automated generation of synthetic powder bed images.

2.5 Discussion and remarks

In this Chapter, we proposed a digital platform for production lines in smart manufacturing that allows the collection and analysis of data. In the context of Industry 4.0, this type of framework is required in order to provide a structure to effectively integrate, analyze and utilize all the different technologies involved in a production line. For the development of the platform, we first analyze the source, type, and structure of the data to be stored and their role in the AM process. We discussed together with the partners involved the user requirements and analyzed the solutions already available in the literature in order to understand the trends, advantages, and issues of this type of platform.

Several state-of-the-art software were used, such as Docker, for the deployment and distribution of the platform, React for the development of the frontend and user interface and a customization of DeviceHive for the backend. The platform also

allows the integration of heterogeneous technologies and devices, as well as the extension of new industry 4.0 functionalities in a plug-and-play fashion.

Finally, to verify the robustness of the platform, we run some basic tests using online benchmark tools to simulate the workload of users. On the fallback platform developed at Politecnico di Torino HPC, we simulated 20 users for 20 minutes that access different resources of the platform at the same time with an average of 100 requests per second, and we obtained a maximum response time of 196ms, 90% response time of 188ms and latency of 132ms. During the tests, the docker containers that run the platform used a maximum of 20GB of memory and 6 cores, mainly for the execution of AI services, the databases and the frontend interface. As a first prototype, the platform performs well, and it can be further optimized in the future by reducing resource consumption and improving network stability.

However, a general-purpose platform, like the one developed, for data collection, management, and analysis of data from smart manufacturing use cases offers several advantages over state-of-the-art solutions. These advantages include greater flexibility, ease of use, scalability, and interoperability. A general-purpose platform can be customized to meet the specific needs of each manufacturing use case, and it provides a simplified user interface and easy-to-use tools for non-technical users. The platform is also designed to handle large volumes of data from multiple sources, and it can integrate with other systems to provide a more comprehensive view of manufacturing operations. This makes it a more efficient and effective solution for managing and analyzing data in a manufacturing environment.

In conclusion, the framework is currently running both in a research environment at Politecnico di Torino and in a real industrial automotive scenario, where it collects and analyzes the data provided by users and machines in the production line, increasing the overall productivity. In the future, the platform can be further improved by introducing new services (e.g., Predictive maintenance, Deep Learning analysis), data (e.g., Acoustic and thermal information), and protocols (e.g., Advanced Message Queuing Protocol, Bluetooth Low-Energy).

Chapter 3

Online monitoring of L-PBF additive manufacturing

3.1 Introduction

As mentioned in previous Chapters, one of the major barriers to the widespread of AM is the lack of quality assurance and repeatability. In particular, due to the complexity of additive processes and the parts to be produced, some defects might occur during the printing process. To guarantee the quality standards required by industries (especially in critical sectors like aerospace, automotive, and healthcare), expensive and difficult post-process inspection are necessary. In order to address possible defects in an effective way, monitoring and control systems can be adopted for supervising the AM process during the layering procedure, enabled by sensors integrated with the 3D printing machine. The real-time analysis of those sensor data can be useful for the early detection of defective parts or even to correct the process in order to proactively address defects from occurring. In this regard, Machine Learning (ML) and Computer Vision (CV) analytics can significantly contribute.

Empowered by the aforementioned, the main focus of this Chapter is in-situ defects monitoring in one of the most diffused metal AM technology, Laser Powder Bed Fusion (L-PBF). In the following sections, a brief introduction to the state-of-the-art in the field of monitoring systems is provided (Section 3.2). Then, follows a near real-time AM monitoring system applied in an industrial use case, using visible-range cameras and computer vision algorithms (Section 3.3). Finally, the

experimental results are discussed (Section 3.4) to conclude with Section 3.5. Some of the work described in this Chapter was also previously published in [32].

3.2 State-of-the-art

To this date, many AM systems do not have the capability to assess the quality of the products that they produce unless with time-consuming and expensive post-process analysis. This majorly affects the overall time and cost of production. To address this issue, many AM companies are providing software for near real-time visualization and monitoring of several process parameters. Nonetheless, these commercial solutions are generally limited in their scope: they do not develop a fully automated quality control strategy, and they fail to detect minor defects in the printed part [35]. Moreover, they are either not available on older machine models that are currently used by manufacturers or typically require an expensive and complex setup process.

Even though the data processing capabilities are still limited, most of the machines that are now available on the market allow to keep track of the history and behavior of the most important parameters of the machine for the entire job process. These parameters are related to either basic process conditions like laser power, build platform temperature, process chamber ambient atmosphere, recoater speed, etc., or machine conditions like cooling system status, electrical power levels, powder source level, etc. [139]. Up-to-date machines can also be equipped with increasingly complex hardware like high-speed and infrared cameras, thermocouples, pyrometers, and photo-detectors, which allow to monitor directly or indirectly many process parameters that are involved in the generation of defects.

Taking advantage of such advanced sensors, researchers are using an increasing amount of data analytics approaches to address the problem of AM monitoring and control, mostly combining off-line ML methods with other types of algorithms, depending on the nature of the sensor data [18, 229, 127].

With the use of optical sensors (either in the visible or in the Infra-Red range) to acquire images of the build chamber, many approaches combine ML together with Computer Vision algorithms to perform AM defect analysis. For example, in [179], the authors propose a Convolutional Neural Network (CNN) for autonomous

detection and classification of different types of anomalies. They extract the relevant features from the images and train the algorithm to detect possible part damages, incomplete spreading of powder, and recoating errors. In [9], the authors propose an online monitoring algorithm that uses computer vision to detect defect formation in every layer of the PBF process. They collect a set of images with a high-resolution camera and analyze the images to detect layers with low-quality fusion or defects. In [227], the authors utilized a high-speed camera to acquire the powder bed images. Then, they combined a Support Vector Machines with a CNN to analyze the images and detect possible defects during the printing process.

Other researches focus on more unconventional types of data sources. For example, in [182, 221] PBF defects are detected based on acoustic emissions acquired by either microphone of optical fibers, using different types of neural networks like Deep Belief Networks or Spectral Convolutional Neural Networks for the classification task. Finally, in [77], heterogeneous data from a combination of different sensors are collected from the laser system, powder bed, and recoating process in an Electron Beam Powder Bed Fusion system and then fed into a Support vector data description (SVDD) model to find major deviations from the expected pattern (e.g., cracks and holes on the surface). A disadvantage of the latter approaches is that they generally require invasive modification on the machine and expensive hardware additions for the data collection.

Recent literature demonstrates data-driven techniques being promising to monitor and improve AM processes. Nonetheless, the application to real-world industrial scenarios is often limited by a number of factors.

First, by the need of complex and expensive hardware. Second, by current limitations in the efficient collection, storage, annotation, and integration of the sensed data.

Moreover, ML techniques, and especially deep learning, typically require to be trained on a large amount of annotated data (e.g., for image-based systems, a large number of images of the powder bed per each possible defect and condition). Considering that the intentional production of defective parts is not viable in industry, as it is costly and time-consuming, the generation of a significant training set becomes an issue.

To address this problem, ML can be put at work even to create synthetic and realistic images resembling the ones obtained during the print of a defective part [5].

In this regard, one of the most promising approaches is the Generative Adversarial Network (GAN), a generative model characterized by training a pair of neural networks (a generator and a discriminator) in competition with each other [75]. Among the many families of GANs recently proposed by literature, Conditional GAN (CGAN) uses an additional input in the generator network to direct the generation process [146], obtaining very good results in many data augmentation tasks. While GANs are becoming very popular in many Computer Vision tasks, their application to the AM sector is still limited. In [73], the authors developed a CGAN to produce synthetic data using as input the layer-by-layer images captured with a near-infrared high-resolution camera. As conditional images, the authors use a synthetic picture of the shape of the built artifact. Then they generate the corresponding near-infrared image, which mimics the characteristics of the captured ones. Ideally, a similar approach may be used for more complex tasks involving a large number of defects, like the one addressed in this Chapter. More specifically, starting from an original image without flaws, CGAN can generate synthetic images of defects, augmenting the training set for defect classification. Unfortunately, the amount of data needed to train a standard CGAN for this purpose would not be much lesser than the data required to train a classifier from scratch. To overcome this problem, in the present work, a very promising architecture called ConSinGAN [86] is employed, which is able to learn generative models from a single training image (more details will follow in the next Sections).

At present, the use of real-time monitoring systems in AM, coupled with ML models and, eventually, data augmentation strategies, is still mainly applied to the early detection of layer defects for part qualification purposes. Given the lack of maturity of AM for mass production, little attention has been devoted so far to monitor the status and health of the AM machine for predictive maintenance purposes. Nonetheless, the extensive undergoing research in the field of sensing technology and ML models for AM defect analysis and process characterization is creating a solid background for future developments, where similar concepts may be easily applied to the early detection of machine anomalies and failure prediction. This is expected to enable significant advantages in terms of reduced unnecessary post-processing analysis, equipment replacement, and increased process safety, availability, and efficiency.

3.3 Case study

In this section, a fully-automated framework for layer-wise defect monitoring in L-PBF AM is presented. The system is based upon an off-axis low-cost optical sensor for image acquisition of the powder bed and the manufactured object during the layering process. Then, a set of fully-automated tools based on CV and ML allows the detection of possible defects of the part that are hard to spot by visual inspection. The prototype was designed in a real-world industrial scenario and developed on top of a fully operative L-PBF machine in an automotive company. By allowing the early stopping/correction of faulty artifacts, this system is expected to improve process repeatability and majorly reduce human intervention, with major positive impacts on production costs.

Together with the data collection and the automated defects detection methods, a first prototype of synthetic image generation leveraging ConSinGAN is proposed, which starts from the images acquired with the proposed fully-automated framework to produce synthetic images addressing two different types of powder bed defects.

3.3.1 Data acquisition and processing system

The implemented monitoring system is built upon low-cost hardware and a camera on top of an EOS M290 L-PBF printer in an automotive company. As shown in Figure 3.1, it includes:

- an Arduino Uno computing platform directly connected with the 3D printer used to manage the system, trigger the camera, and take images of the powder bed.
- an IDS UI-1540-SE 1.31Mpix camera (1280×1024 resolution). The camera is triggered through the Application Programming Interface (API) made available by the manufacturer. The acquisition is off-axis with respect to the optical path of the laser.
- A standard PC running Linux to collect images in the Portable Network Graphics (PNG) format with a resolution of 1280×1024 and run the image analytics algorithms.

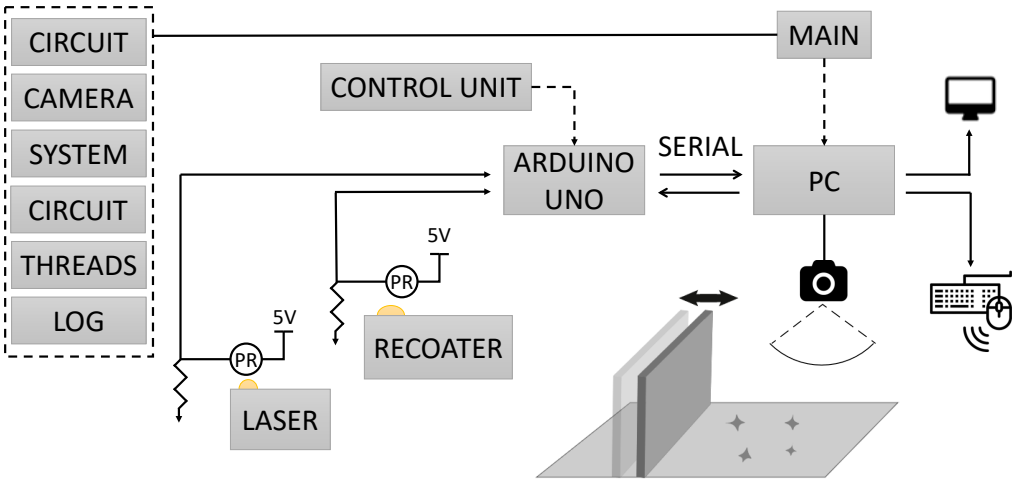


Figure 3.1 In-situ monitoring system

The image acquisition is automatically triggered by 3D printer states, exploiting the signals emitted respectively by the action of the laser and of the recoater by means of photo-resistors. By doing so, the system is able to acquire images of the powder bed before and after each layer is printed without requiring any user interaction.

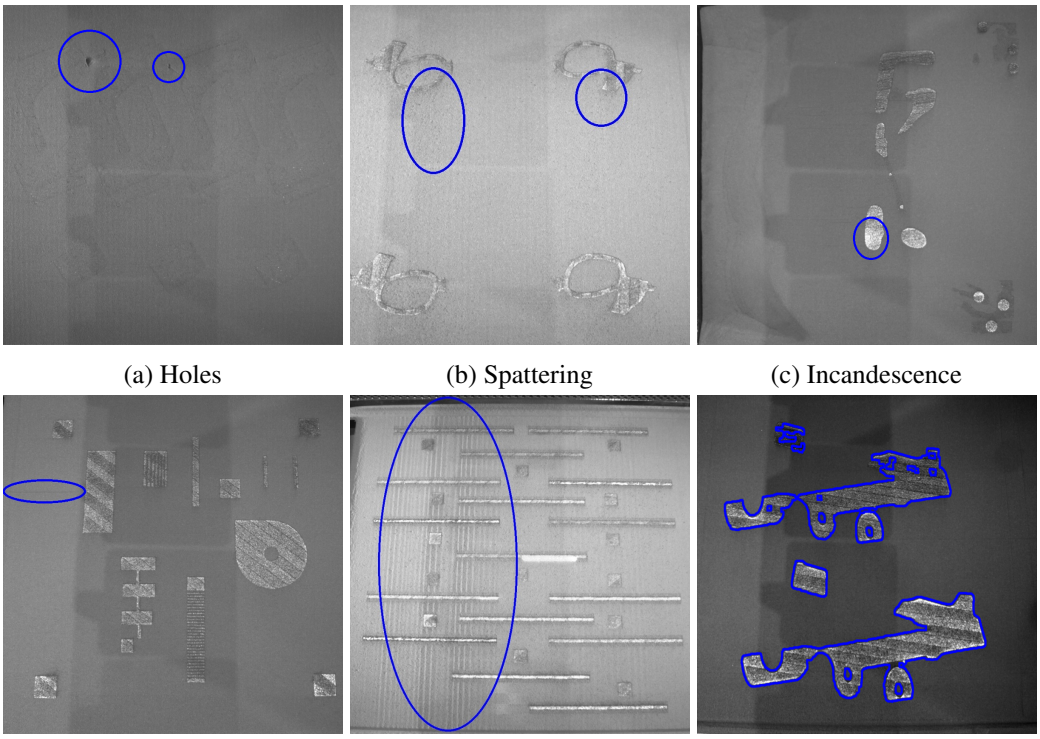
3.3.2 Defects detection algorithms

The proposed solution includes a set of near real-time image analytic algorithms that allow the detection of five different powder bed defects, as well as continuous monitoring of the profile of the object that is being printed. The algorithms are based on image processing and ML and were developed in Python using OpenCV and Keras standard libraries.

Figure 3.2 (a)-(e) show five main categories of defects targeted by our system:

- *Holes* (Figure 3.2 (a)): localized lacks of metallic powder that create small dark areas in the powder bed image. The origin is a lack of powder due to bad regulation of the dosing factor.
- *Spattering* (Figure 3.2 (b)): droplets of melted metal ejected from the melt pool and landed in the surroundings.

- *Incandescence* (Figure 3.2 (c)): high-intensity areas in the completed layer image, resulting from an excess of laser energy density and a consequent inability by the melt pool to cool down correctly.
- *Horizontal* defects (Figure 3.2 (d)): dark horizontal lines in the powder bed caused by the incorrect spreading of the powder, possibly because of the geometric imperfection of the piece or of the metallic powder.
- *Vertical* defects (Figure 3.2 (e)): vertical undulation of the powder bed, consisting of alternated dark and light lines along the direction of the recoater's path. The origin is either a mechanical interference between the part and the recoater or a mechanical defect of the recoater's surface.



(a) Holes (b) Spattering (c) Incandescence

(d) Horizontal (e) Vertical (f) Profile monitoring

Figure 3.2 (a)-(e) Examples of powder bed defects targeted by the system (f) Profile monitoring example

Each of these powder bed defects is known to cause either porosities or microstructural alterations in the printed parts, as well as lower mechanical characteristics.

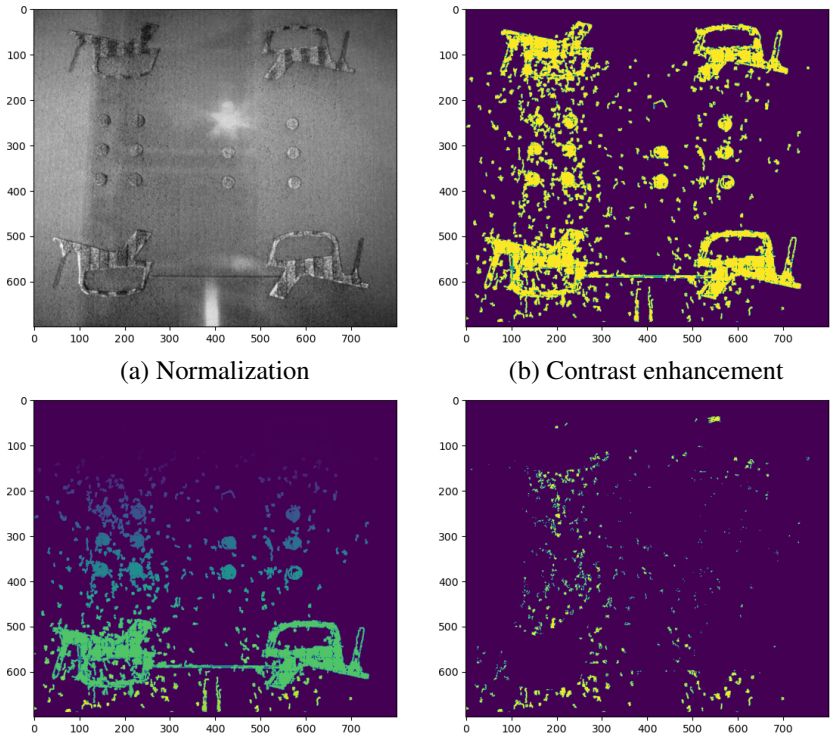
The pipeline for run-time defects detection consists of several image-processing steps:

- *Normalization* (Figure 3.3 (a)): the images are first normalized against a common reference frame in order to correct uneven illumination problems. To do so, an image of the powder bed is acquired before the start of the layering process and used as a reference throughout.
- *Contrast enhancement* (Figure 3.3 (b)): a standard background subtraction algorithm is applied to make the objects more distinguishable from each other, as well as from the background [150].
- *Objects identification* (Figure 3.3 (c)): intensity discontinuities are identified by means of automated intensity thresholding algorithm. This provides a rough identification of the different objects in an image.
- *Morphological filtering* (Figure 3.3 (d)): specific objects are recognized based on their shape, exploiting morphological algorithms. More in detail, Watersheds and Hough transform, followed by standard morphological regularization (i.e., opening, closing, holes filling), are respectively applied to identify round-shaped and horizontal/vertical lines. Based on the specific shape and number of detected objects, the software identifies a specific category of defect.

As an example, in Figure 3.3, the pipeline applied to detect a spattering defect is presented, showing each intermediate step's outcome. Spattering is indeed one of the defects that most frequently happen during powder bed fusion: it involves tiny particles of liquid metal being ejected from the laser's path, which may contaminate the powder bed and create issues such as porosity, roughness, and lack of adhesion in the finished parts. At the end of the last step in Figure 3.3 (i.e., morphological filtering), the most relevant spatters are identified.

3.3.3 Part profile monitoring algorithm

Besides powder bed defects detection, the system includes a fully automated *profile monitoring* suite that is able to monitor the profile of the build on a layer-by-layer basis (see an example in Figure 3.2 (f)). This task has additional algorithmic and



(a) Normalization (b) Contrast enhancement

(c) Objects identification (d) Morphological filtering

Figure 3.3 Example of spattering defects detection pipeline

computational challenges compared to basic powder bed defects because the printed parts may have very different shapes and dimensions.

In our solution, profile monitoring is addressed as a semantic segmentation problem. Semantic segmentation aims to cluster parts of an image together, which belong to the same object, using a pixel-level prediction to classify each pixel in an image according to a category. In other words, image segmentation becomes a binary classification task, where each pixel needs to be labeled as belonging to the object of interest (in our case, the printed part) or the powder bed.

To achieve this purpose, a U-Net architecture is employed [176], a deep learning algorithm that was initially designed for biomedical image segmentation and then successfully applied to many different Computer Vision applications. The network implements an end-to-end Fully Convolutional Network (FCN) composed of convolutional and pooling layers without any dense layer, which makes it suitable for any image size. As shown in Figure 3.4, the architecture is composed of two paths. The first path (encoding) is the contraction path or encoder used to capture the context in the image. It consists of a stack of various convolutional and max-pooling layers

with gradually decreasing feature map dimensions. The second path (decoding) is a symmetric expanding path or decoder used to enable precise localization using transposed convolutions. In the presented approach, the U-Net follows the implementation suggested by [176]. The encoding path consists of the repeated application of two 3×3 convolutions, with Rectified Linear Unit (ReLU) as the activation function and a 2×2 max pooling operation. The decoding path consists of an upsampling of the feature map (up-conv) followed by a 2×2 convolution with ReLU as the activation function, a concatenation with the correspondingly cropped feature map from the contraction path, and two 3×3 convolutions, each followed by a ReLU. The copy and crop are necessary due to the loss of border pixels in every convolution. At the final layer, a 1×1 convolution is used to map each feature vector to the desired number of classes. In total, the network has 23 convolutional layers.

In this application, the two classes to be classified are the foreground (i.e., the printed part) and the background (i.e., the powder bed). First, it was used as input images with a size of 360×480. Then, the input size was fine-tuned on a validation set consisting of images of a representative PBF layer and finally set the optimal size to 320×320.

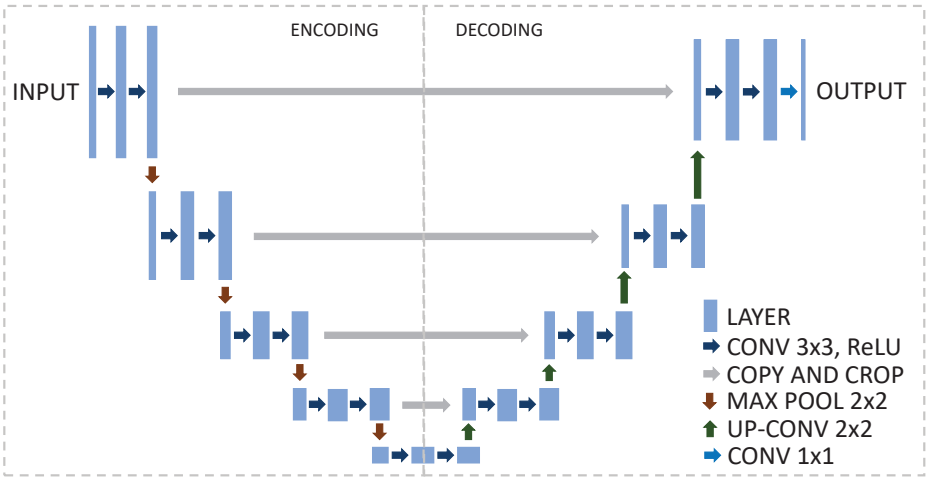


Figure 3.4 U-Net architecture

3.3.4 Dataset augmentation with GAN

Unfortunately, training ML algorithms to AM defect classification requires a large number of training images, with and without defects. As anticipated in Section 3.2,

this is difficult to obtain in a real-world industrial setting, especially for the defect categories. To overcome this issue and increase the number of defective images available for the training, we presented a solution that adopts an extended version of Conditional Generative Adversarial Network, ConSinGAN [86]. While traditional GANs still require large datasets for the training, ConSinGAN is able to generate synthetic data starting from a single image. This characteristic is of major importance, given the very limited number of defective images.

As shown in Figure 3.5, a classic GAN architecture is composed of two Neural Networks, the Generator, and the Discriminator, that compete to produce a generative model. The generator network produces synthetic data starting from the input, with the aim of obtaining data that resembles the real and available samples. At the same time, the discriminator network is a binary classifier that is trained to distinguish the synthetic samples produced by the generator from the real ones. The training phase consists of an adversarial process between the two networks, in which the generator tries to increase the error rate of the discriminator while the latter tries to distinguish real and synthetic data. Hence, the core idea is to implement an indirect training of the generative model through the discriminator: the generator has no direct access to real data and interacts only with the discriminator [11].

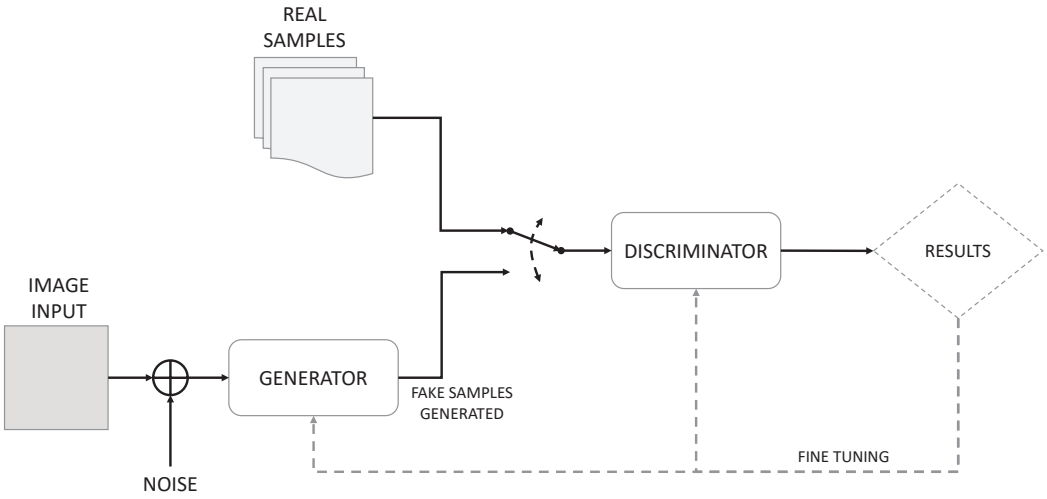


Figure 3.5 Generative Adversarial Network architecture

The ConSinGAN is structured in a multi-stage and multi-resolution approach, as shown in Figure 3.6. First, the ConSinGAN uses, as input for few iterations, an image with a coarse resolution to learn mapping a random noise vector z to a low-resolution

image (See *Generator Phase 0* in Figure 3.6). After the initial training has converged, the size of the generator is increased, adding three additional convolutional layers. Each stage takes as input the raw features from the previous stage, and a residual connection is used in each stage to feedforward the last convolutional layer (See *Stage 1* in Figure 3.6). The process is repeated N times until the desired output resolution is reached. The synthetic image created by the generator is given as input to the discriminator, together with a real image from the dataset. The discriminator is trained to discern if the images given are real or synthetic, comparing them.

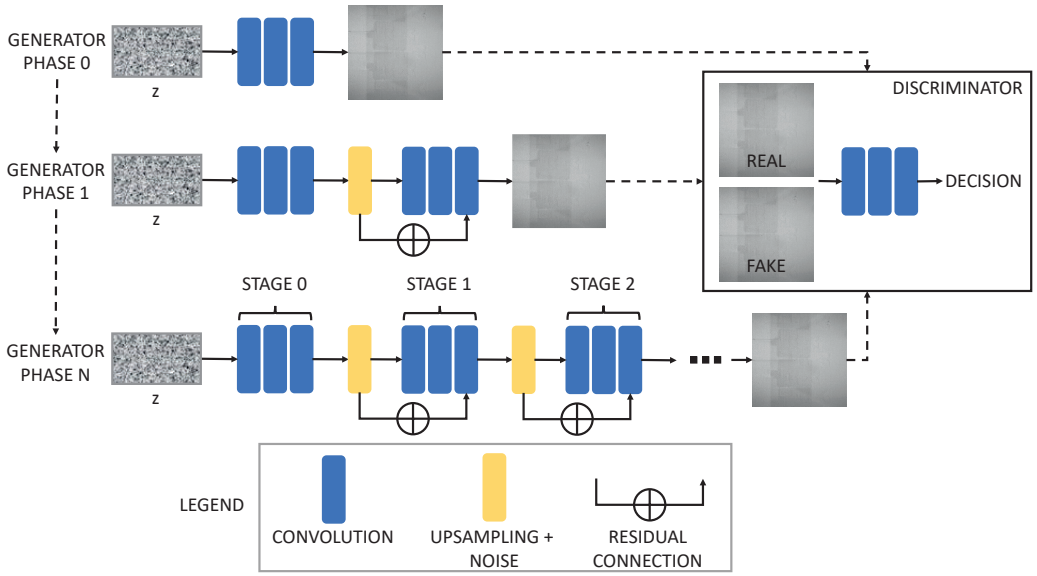


Figure 3.6 ConSinGAN architecture [86]

The most critical part of the synthetic image generation is image harmonization (see an example in Figure 3.7). It consists in transforming a composite image, called naive (Figure 3.7 (b)), into a realistic image by applying to the changes made the same style and appearance of the original image, called training (Figure 3.7 (a)). The output of this process is a synthetic image similar to the original one (Figure 3.7 (c)). To perform this process, the network is first trained to learn a generative model from the original image. Then, given the naive image as input, it tries to transform it into an image that should resemble the original learned distribution. Once the model is trained on a specific image, various synthetic images can be generated by varying only the naive picture.

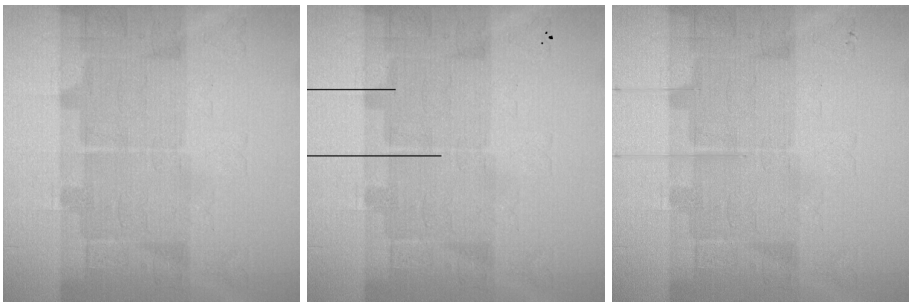
Figure 3.8 shows an example of AM image generation with the ConSinGAN. To generate new images presenting defects, the image harmonization process is carried



(a) Training image (b) Naive image (c) Harmonized image

Figure 3.7 Example of image harmonization [86]

out as follows: i) the generator is trained using a real image captured without defects (Figure 3.8 (a)); ii) a naive image is created applying modifications to the training image by using a photo editing software (Figure 3.8 (b)); iii) the naive image is harmonized to resemble the training image characteristics (Figure 3.8 (c)). In this way, the synthetic defects are assimilated into the original image.



(a) Training image (b) Naive image (c) Synthetic image

Figure 3.8 Example of image generation

The data generated can be used to increase the dataset available to both the train and validation phases of defect detection algorithms, breaking down the costs of generating a huge set of real defect data.

3.4 Experimental results

To validate the defect detection algorithms, a set of pre-annotated images was used with all the defects targeted by the proposed system. For the five main categories of defects, we run a statistical validation by analyzing whether the algorithm identified the defect or not, using metrics that are widely accepted in classification tasks:

- Accuracy: it represents the number of correct classifications with respect to the total cases.
$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$$
- Precision: it is the fraction of relevant cases among the retrieved instances.
$$Precision = \frac{TP}{TP+FP}$$
- Recall: it is the fraction of the total amount of relevant instances that were retrieved.
$$Recall = \frac{TP}{TP+FN}$$

In this work, True Positives (TP) represent the instances when the algorithms were able to detect a defect that was really present. True Negatives (TN) represent the instances when a given defect was not present, and the algorithm was right in not detecting it. False Positive (FP) and False Negative (FN) represent the possible errors of the algorithms, respectively, in detecting a defect that was not present or not being able to identify a defect that was present. Table 3.1 reports the results obtained on a test set of 24 images with different powder bed conditions and defects.

Table 3.1 Defects detection algorithms validation

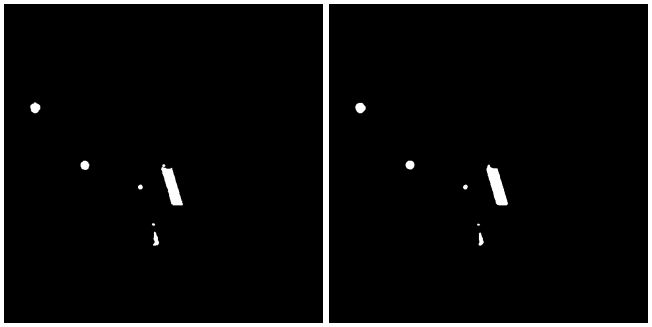
	Holes	**Spatt.**	**Incand.**	**Horizontal**	**Vertical**
TP	14	22	15	11	6
TN	8	1	4	11	16
FP	2	1	5	1	0
FN	0	0	0	1	2
Accuracy	91.3%	95.8%	79.2%	91.6%	91.6%
Precision	87.5%	95.6%	75.0%	91.6%	100%
Recall	100%	100%	100%	91.6%	75.0%

For all the five defects, the results of the metrics considered are $\geq 75\%$, with the worst results for the Incandescence defects (Precision: 75%) and Vertical defects (Recall: 75%). According to our tests, Incandescence proved to be the most challenging defect to be recognized, probably due to the high pixel luminosity variation. On the other hand, Spattering defects are the easiest due to the high amount of spatters generated.

For the profile monitoring task, the validation exploits the Sørensen–Dice coefficient (DSC) to compare the profile segmentation obtained with our algorithm against a manually obtained ground truth. This metric is used to gauge a 0 to 1 similarity of two binary images, as follows:

$$DSC = \frac{2|X \cap Y|}{|X| + |Y|}$$

where $|X|$ and $|Y|$ are the number of pixels of the two images (in our case, the automatic segmentation and the ground truth), and $|X \cap Y|$ is the number of pixels that are common to both images. Figure 3.9 shows an example of this procedure, with (a) the binary mask obtained by manual segmentation, used as the ground truth, and (b) the binary mask obtained by our profile monitoring suite.



(a) Ground Truth (b) Automatic algorithm

Figure 3.9 Profile monitoring: validation example

In the conducted tests, a very good similarity between automated segmentation and manual ground truth was obtained, with a mean DSC value equal to 0.878 when computed on each single-segmented object and to 0.911, when computed on each layer image taken as a whole.

Finally, Table 3.2 reports the execution time of all the tested algorithms. As can be seen from the reported values, execution times are all below 2.5s, which is well below the time elapsing between two subsequent layers. Profile monitoring is the algorithm taking the longest time (2.461s) because it involves running a deep neural network. The other algorithms, which exploit standard image processing operations, are all below 1s of execution.

As the final step, the results of the synthetic data generation process were validated. In the implemented prototype, the image augmentation was preliminarily

Table 3.2 Mean execution time of the algorithms

Operation	**Time [s]**	**Operation**	**Time [s]**
Holes	0.791	**Horizontal**	0.593
Spattering	0.574	**Vertical**	0.932
Incandescence	0.821	**Profile monitoring**	2.461

carried out on two types of defects: holes and horizontal. We assessed the results obtained by the ConSinGAN through a most widely used GAN evaluation metric, that is, the Fréchet Inception Distance (FID) [84]:

$$FID = ||\mu_X - \mu_Y||^2 + Tr(\Sigma_X + \Sigma_Y - 2\sqrt{\Sigma_X \Sigma_Y})$$

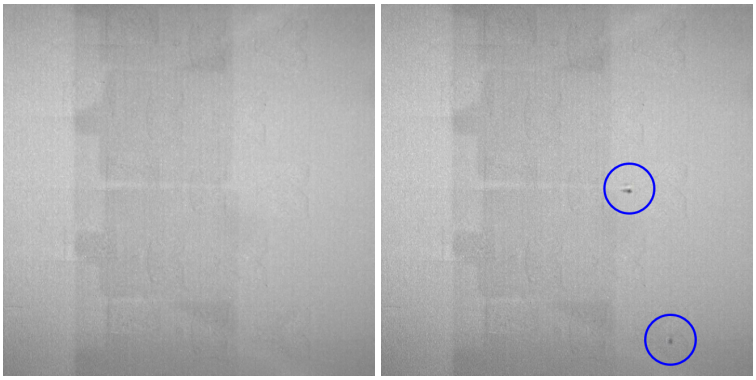
Where X is the set of real images, Y is the set of synthetic images; μ_X and μ_Y are the feature-wise mean of the real and generated images, respectively; Σ_X and Σ_Y are the covariance matrix for the real and generated feature vectors, respectively; Tr refers to the trace operation in matrix form, which is the sum of the diagonal elements; $||\mu_X - \mu_Y||^2$ refers to sum squared difference between the two mean vectors.

The FID value ranges between 0 and plus infinite: a value near 0 means that the two images are the same, while a very high value means that the two images are completely different.

Figure 3.10 shows an example of a real image without defects (a) and a synthetic image (b) with holes defects, as generated by our approach. As can be seen, the two images are quite similar, corresponding to a FID value of 85.74.

Table 3.3 reports the results obtained after 2000 steps, on a test set of 20 generated images with holes and horizontal defects. As expected, the FID shows similar results for all the generated images. This is due to the fact that the synthetic images were generated starting from the same training set and that the added defects are quite similar. The FID values of the synthetic images are also quite small, demonstrating a good similarity to real ones with the same defects.

As a final experiment, the synthetic images were tested with the defects detection algorithms. Table 3.4 reports the results obtained on a test set of 24 synthetic images with holes and horizontal defects generated with the ConSinGAN.



(a) Real image (b) Synthetic image

Figure 3.10 Comparison between a training image and a synthetic one with defects (holes)

Table 3.3 ConSinGAN results

Synthetic image	1	2	3	4	5	6	7	8	9	10
FID	131.07	156.45	93.66	85.74	97.74	107.27	103.71	119.70	93.49	99.81
Synthetic image	11	12	13	14	15	16	17	18	19	20
FID	110.82	90.14	102.64	117.35	127.00	101.58	112.89	98.61	111.51	103.21

Table 3.4 Defects detection results on artificially generated images

	Holes	**Horizontal**
TP	8	8
TN	14	12
FP	1	2
FN	1	2
Accuracy	91.7%	83.3%
Precision	88.9%	80.0%
Recall	88.9%	80.0%

For both the defects, the results of the considered metrics (Accuracy, Precision, and Recall) are $\geq 80\%$ with the worst results for the Horizontal defects (Precision and Recall: 80%). Hence, according to the experiments conducted, the defect detection results are similar to the results obtained with real images. The holes detection algorithm has better Accuracy and Precision with synthetic images (Accuracy: 91.6%, Precision: 88.9%) than with real images (Accuracy: 91.3%, Precision: 87.5%), while the Recall is better with the real images (Recall with synthetic images: 88.9%, Recall with real images: 100%). Instead, the horizontal detection algorithm has accuracy, precision, and recall better with real images (Accuracy: 91.6%, Precision:

91.6%, Recall: 91.6%) than with synthetic images (Accuracy: 83.3%, Precision: 80.0%, Recall: 80.0%) Sure, indeed, holes are easier to detect compared to horizontal defects, which can be easily confused with the powder bed background. However, in the test case with synthetic images, the metrics results for the horizontal defect were similar to those with real images, with a small reduction of performance $\leq 12\%$ for the worst-case Recall.

3.5 Discussion and remarks

Computer Vision and ML have proven to be promising approaches for the in-situ monitoring of the AM process. Nonetheless, the actual use of these approaches in a real industrial scenario is still limited due to a number of challenges. First of all, the lack of effective data collection infrastructure specifically devoted to AM. Second, the necessity to train the models on large annotated datasets, which are typically costly and difficult to obtain in industrial environments.

This Chapter presented a low-cost camera-based in-situ defect monitoring system for metal PBF and data generation. The preliminary prototype, integrated and tested in a real industrial scenario of an automotive company, includes: i) a set of run-time Computer Vision and ML algorithms to detect five different categories of powder bed defects, as well as the layer-wise monitoring of the profile of a printed part and ii) a synthetic image generation model, for data augmentation purposes. Experimental results suggest that the algorithms have a good performance in terms of defect detection accuracy and profile segmentation, and they are suitable for near real-time execution with low-cost hardware. The framework is currently being extended to provide layer-by-layer comparisons between the profile of the printed part (as returned by the profile monitoring suite) and the desired profile as defined by the slicer. This will allow a near real-time automated detection of any profile alterations during a build.

In its current form, the presented framework only allows the continuous monitoring of the AM part and the timely detection of macroscopic defects based on real-time comparison with a baseline. Nonetheless, the continuous growth of collected images is creating a solid ground for future extensions in the direction of predictive analytics. In this regard, it is planned to integrate the available image dataset with process parameters as well as with post-process information (i.e., results

of quality and mechanical tests on the finished parts, as well as historical information about part breakdowns) in order to train models able to predict and possibly correct the failure of a part, applying process optimization strategies.

Chapter 4

Machine anomaly detection in EB-PBF process

4.1 Introduction

Nowadays, with the advent of new technologies and sensors, the raw data produced by AM machines have increased exponentially, making available information about the process that was previously inaccessible. The analysis of those unprocessed datasets could allow to provide crucial insights about the process, such as the job's actual state and its stability. However, processing such a large amount of data with a complex meaning requires developing specifically designed computational and analytical tools. The adoption of these techniques for industrial production demands an efficient and robust monitoring of the process that could guarantee the component quality.

Thus, monitoring and control of AM processes with internal information from the machine are still in the early stage. In this regard, several researchers devoted their efforts to utilize external sensors, such as infrared cameras (IR), to monitor some physical process features, such as melt pool geometry and its dynamics [135, 198], temperature distribution [174, 181], spattered particles [220], melting defects [65, 61]. The collected data are then analyzed using state-of-the-art Artificial Intelligence (AI) algorithms [179, 227, 197, 219, 167].

However, this type of approach is challenging, especially when working with complex processes and machines, such in the case of the Electron Beam Powder Bed Fusion (EB-PBF), where high and rapid temperature variations occur [51]. To overcome this problem, an alternative solution is to use the Backscattered Electrons (BSE) techniques to create images of the powder bed in a way comparable to scanning electron microscopy [12]. This approach has significant advantages over classical image systems; however, the signals are still complex to be analyzed in order to obtain definitive and satisfying results, not to mention that the machine needs additional equipment and software, which are very expensive.

In this context, manufacturers are improving their machines in order to directly provide process information. Recent EB-PBF machines allow to control process stability using internal diagnostic sensors. The collected data are stored in a log file and made accessible for the user but without any further closed-loop control during the process [57]. For example, the temperature of key machine components is continuously monitored, and a threshold logic is applied; if the measured temperature is above a certain limit, the process is stopped. This rule is defined empirically by the machine manufacturer to reduce, e.g., the risk of machine damage when process drift occurs.

In this regard, by analyzing those raw datasets could be possible to improve the overall process.

The remaining part of the Chapter is organized as follows: Section 4.2 reviews literature solution on time-series anomaly detection; Section 4.3 introduces the EB-PBF system; Section 4.4 describes the signals analyzed; Section 4.5 explains the theoretical background behind our solution and the experimental. Then, Section 4.6 discusses the results obtained in the detection of anomalies in jobs of smart manufacturing EB-PBF systems. Finally, Section 4.7 discusses the concluding remarks.

4.2 Multivariate time-series anomaly detection: current scenario

Nowadays, the early detection of anomalous behaviors of production lines is of paramount importance. On the one hand, the diagnosis of the anomalous state of an industrial system ensures its normal operation, avoiding severe deterioration

in productivity. On the other hand, the consequences of system failure may have serious social impacts, for instance, in terms of dangerous accidents. To prevent such failures, collect and proficiently process a large amount of time-series data is an essential task that, moreover, allows manufacturing systems to be programmable, efficient, and accurate as well as to require less human intervention.

On top of these considerations, time-series anomaly detection (TAD) has recently become a primary research topic, attracting a large body of studies over the last decades [178]. Early examples of research on TAD mainly include clustering (e.g., k-NN, k-means), statistical solution (e.g., auto-regressive moving-average models), and outlier detection methods based on the dataset density [196]. Nonetheless, such *traditional* methodologies demonstrate poor performance when employed in high-dimensional multivariate contexts, where the complex structures in the data hamper the construction of the boundary between abnormal and normal events [121].

Conversely, many machine learning methodologies have proved to be effective in tackling intricate and high-dimensional anomalies. In this regard, the vast heterogeneity among distinct anomaly types and time series properties has led to the spread of a multitude of different anomaly detection algorithms, about which diverse surveys exist [121, 4, 178].

As an example, [152] proposed a traditional and one-class support vector machine for classifying robotic arm executions as benign or anomalous. The authors demonstrated the efficacy of the proposed supervised solution, although dependent on anomaly annotations, which are cumbersome to be collected, especially in the early life of the production line when the equipment is still behaving correctly most of the time. More recently, [164] faced hydrological time series anomaly pattern detection with isolation forest, showing good classification accuracy, yet again with a supervised solution.

To specifically reduce the impact of the anomaly annotations on the learning-based methodologies, various late studies have investigated the use of deep learning technologies for TAD, outlining three main spheres of research:

i) *Forecasting-based models*, which detect anomalies based on prediction errors. The prediction model is semi-supervised in the sense that it takes a time-series sequence as input and learns to predict future observation without requiring explicit labels on given time intervals. Once the prediction model, trained

by nominal executions, learns to predict the next observation for the input variables, an error vector at time t can be derived by computing the discrepancy between the actual future signal and the model prediction.

Downstream to computing the prediction error, a validation set is typically exploited to set a threshold or to learn a probabilistic model able to distinguish nominal and abnormal behavior. For example, [94] introduced a non-parametric thresholding strategy to interpret predictions generated by an LSTM network. More recently, [218] developed a probabilistic regression model able to categorize anomalous versus nominal behaviors, exactly relying on the LSTMs prediction error.

ii) *Reconstruction-based models*, which learn a latent representation of the normal time series on top of which the original input is realistically reconstructed. A threshold-based anomaly detector, typically optimized on an independent validation set [121], is then devoted to detect the anomalous samples presenting larger reconstruction errors [196]. Under the hood, most of the methodologies in this family are made up of an auto-encoder (AE) network able to proficiently reconstruct the input signal. For instance, [222] leveraged a one-dimensional convolutional auto-encoder to learn features from complex vibration signals and perform gearbox fault diagnosis. Overall, recent evidence suggests that the AE-based TAD is the most promising strategy, being able to capture underlying complexity that often exists in industrial multivariate datasets.

iii) *Hybrid models*, featuring both forecasting and reconstruction-based strategies. A remarkable solution is the one proposed by [228], where the authors jointly optimize a forecasting-based model and a reconstruction-based one.

Despite the importance of TAD and the vast production of studies focused on anomaly detection algorithms, there remains a lack of evidence on the feasibility of such methodologies in a real-world industrial case study. In this work, we introduce *Electron Beam Powder Bed Fusion - Autoencoder* (EB-PBF-AE), a novel reconstruction-based anomaly detector able to effectively categorize *Successful* versus *Failed* EB-PBF processes. To our knowledge, this study is the first to exploit AE-based TAD on the EB-PBF real-world industrial scenario, where the inherent complexity of the industrial procedures delineates an extremely challenging, as well intriguing, real-world case study.

4.3 EB-PBF: Process and Hardware

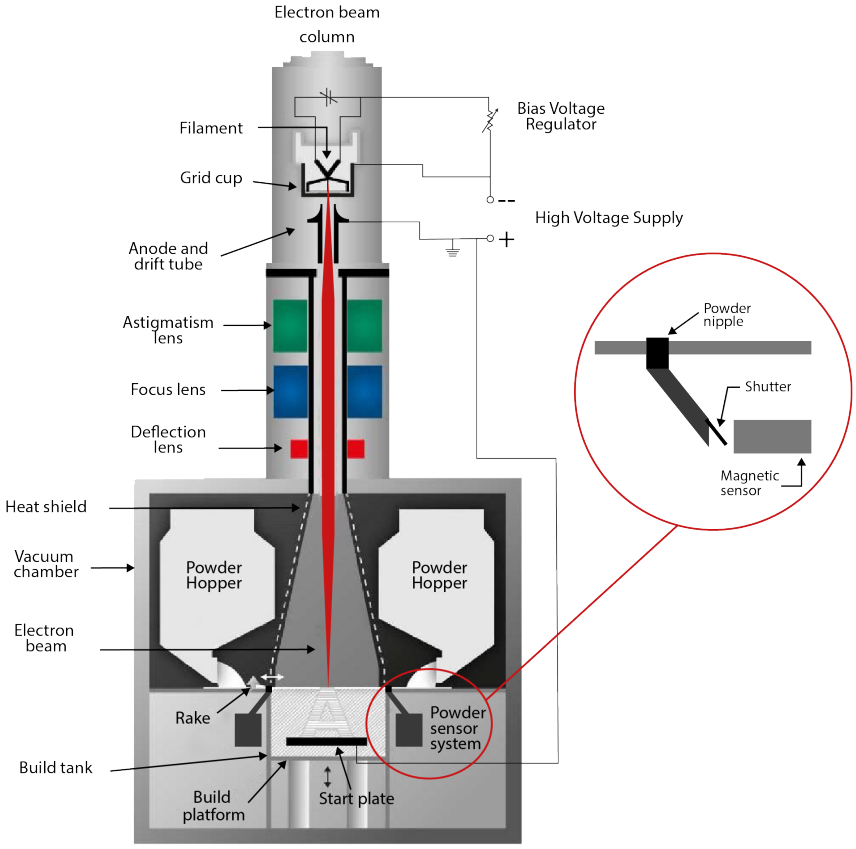


Figure 4.1 EB-PBF hardware scheme

The EB-PBF process uses electrons at high speed to selectively melt a bed of metallic powder particles [47]. The EB-PBF process, therefore, exploits the heat generated when the beam impacts the powder bed [66]. Thus the interaction between the electron beam and the powder bed is one of the most critical phases of the process [66]. During the impact, the particles are rapidly charged negatively. The charge accumulation leads to a diffusion of particles in the chamber due to the generated repulsion forces between the particles [41], provoking the so-called "smoke" phenomenon [145, 184]. To avoid or limit this issue, usually, before the component melting phase, the powder bed is heated up with a preheating step [66, 145]. This step sinters the particles through the formation of necks, increasing the cohesion of the particles, enhancing the thermal and electrical conductivity [214] and the charge dispersion.

All the EB-PBF systems have a similar architecture. In this work, more details were provided for the Arcam A2X, in which the presented methodology was applied. In Figure 4.1, a brief scheme of the principal hardware components of the Arcam A2X is reported. From the hardware point of view, each EB-PBF system can be divided into two parts: the top column and the build chamber.

The top column includes the electron gun and the magnetic lenses. The electron gun contains the cathode for the emission of electrons. The cathode can be a LaB crystal or a Tungsten filament, as in the case of Arcam A2X, which works with a potential of 60 kV. The emitted electrons are accelerated up to a fraction of the speed of light (40%) [184] by an anodic potential and then collimated by the magnetic lenses. These lenses are therefore responsible for the beam focusing and deflection [108].

Another central component of the electron beam gun is the grid cup. This element is positioned between the cathode and the anode and controls the current flow of the electrons reaching the anode. The voltage applied on the grid works as a gate. As an example, a negative voltage will push the electrons back toward the cathode. The grid cup is also, therefore, an important filter among the different process phases since it avoids electron flow when it is not required.

The build chamber contains the build steel tank, the powder storing (hopper), and the powder distribution system (rake). The steel build tank contains the build platform, which is moved along the build direction by means of an electric motor. In an Arcam A2X, there are two hoppers, one for each side of the build chamber. In this case, the powder is fed on the build platform by gravity. The rake system then collects and distributes the powder. This phase is called fetch. The rake consists of a series of thin and flexible stainless-steel teethes [66]. The amount of distributed powder is monitored by the powder sensors, a measuring system that deploys two magnetic sensors. In this case, using closed-loop control, if the amount of powder differs from the calibrated amount, the positions of fetch are adjusted accordingly. To explain how this closed loop works, it should be considered that the amount of powder distributed on the build platform always exceeds the quantity needed to form the powder layer. The correct excess of powder that indicates a proper powder layer is calibrated empirically as a function of the required layer thickness and the powder features (i.e., granulometry). Therefore, the control is realized on this excess of powder with the following sequence. After the distribution, the excess powder falls

in a small channel consisting of two elements: the first element is parallel to the build direction (powder nipples), while the second element is tilted to a certain degree. At the end of this second element, the powder flows through a small shutter that is opened mechanically by the powder flow. Then, two magnetic sensors measure the time that the shutter remains open (pulse length), corresponding to the time needed to empty the channel. The machine control compares the pulse length with the calibrated reference value, called optimal pulse, and adjusts the fetch position accordingly. If the measured value is too low, the powder distribution step is repeated. The powder sensor on the right side of the chamber is deployed for controlling the left fetch position (left regulator), while the right one controls the right fetch position (right regulator) [38, 10].

A high vacuum level is ensured both in the beam column and in the build chamber. The Arcam A2X is equipped with four vacuum pumps: one rough pump, two turbo pumps for the chamber, and one for the column, respectively. The pressure is typically maintained at 10^{-3} Pa in the build chamber and around 10^{-5} Pa in the column [10]. In addition, inert gas (typically Helium at a pressure of 10^{-1} Pa) is insufflated during the process to avoid charge accumulation, guarantee high thermal stability, and accelerate the cooling phase [71].

Due to the high temperatures generated during the process, several thermocouples are installed in different areas of the machines to monitor the temperature evolution of some fundamental elements. To measure the temperature in the build job, a thermocouple is positioned under the start plate while another one monitors the temperature of the top column. If the top column thermocouple detects a value over a certain threshold, the process is shut down to avoid damage in the electron gun.

Overall, an EB-PBF process occurs as follows. At the beginning of the process, the start plate is heated using the electron beam and up to a target temperature that depends on the processed material [137]. After this step, a sequence of phases is repeated until the build job is completed [108]:

1. Table lowering: the start plate is lowered to a quantity equal to layer thickness.
2. Powder spreading: the rake fetches the powder from one side and spreads the new powder bed.
3. Preheating 1: during this phase, a defocused electron beam heats a portion of the new powder layer equal to the maximum area that contains all the

components [172]. As mentioned above, this step produces the sintering of the particles.

4. Preheating 2: the electron beam further heats only the portions of powder containing the parts to be built for an area corresponding equal to an offset of the section that must be melted [172].

5. Melting: the electron beam selectively melts the section of the component with the selected strategy (hatching and contouring [67, 192]) and process parameters.

6. Post-heating: based on the amount of energy (beam current) provided in the two preheating and melting phases, it is possible to use the electron beam to further heat the section or let it cool down through Helium flow. This is done to accomplish the energy equilibrium and the average beam current to be achieved over the layer [68].

Once the build has been completed, the build chamber is filled with helium, and the job is cooled down until room temperature. Then the build job is removed from the chamber, and the parts are cleaned from the sintered powder with a blasting operation. The unmelted powder is collected and reused for the next jobs.

4.4 Dataset description

The employed dataset used to predict the future status of a job is extracted directly from the Arcam EBM A2X available in our research laboratory and made fully accessible by the producer. As mentioned above, the machine is equipped with different types of sensors (current, voltage, temperature, etc.) that monitor all the components and processes during the activity of the machine. The system is also equipped with a set of internal thresholding methods that analyze the sensors available in order to raise alarms to the user during the process and, in the worst case, to automatically stop the process in critical situations in order to prevent severe damage to the machine.

The machine writes for each printing job all the data produced, from the input parameters to the sensor's output, in an internal log file that is updated every time

a new sensor value is detected. This makes the pre-processing and analysis of the dataset rather complicated as there is no fixed sampling time to use, and the amount of data available varies according to the part manufactured and the process phases.

To facilitate the analysis of the data, the producer provides internal software to extract and export in a PLG log file format all the machine information. The data contained in this file is organized into six categories (See Table 4.1). However, the most useful information is collected in the Alarms, OPC, and Process categories.

Table 4.1 Data categories

Category	Description
Analyze	contains calculated value for each layer of the process
Alarms	contains the status of all the alarms available in the machine
Builds	contains information about the loaded job, i.e., current height, max layer time, and machine serial number
Machine	contains machine-specific and threshold values, i.e. rake method, rake calibration, maximum column temperature, information about the focus and astigmatism
OPC	contains signals sent by various hardware to the control system of the machine, i.e., rake motor, turbo pumps, and thermo elements
Process	contains signals that are sent from the software to the hardware, i.e., power supply, rake, and temperature control
Themes	contains all the values for the different themes. Each theme can be a set of process parameters for the melt, the preheating, the start plate heating, filament, and material themes

The Alarms category contains information about all the alarms set during the process and about the status of a job. Alarms can be divided into warning and error. A warning can affect the process but does not lead to a process shutdown, which occurs in case of error. From the data extracted during the EB-PBF processes, the most frequent alarms in the failed jobs are related to model clipping and arc trip.

The model clipping alarm is a warning that occurs when a model of the components to be built is clipped because the component is placed close to the edges of the building envelope. The arc trip alarm is set during the building process whenever an electric high-voltage arc occurs. This alarm notifies that the high voltage of the power supply instantly drops from 60 kV to 0 kV, involving the shutdown of the beam. In this case, the voltage is ramped up again automatically to the initial value, and the process continues.

Arc trips could happen because of overheating and overloading, but the main cause is "smoke" events, where powder particles reach the top column. The arc trip alarm is, therefore, a recoverable error. However, if arc trip occurrences are high in a short time, an "Arc Trip Error" alarm appears in the Alarms category, and the process is stopped.

The other alarms typically arise only in critical situations shortly before the machine stops the process. However, all these signals are required in order to correctly identify the status of a job and create the dataset that will be used for the training and testing of the methodology.

From the Builds category, the main information extracted regards the height of the platform that allows to understand the percentage of completion of a job. Indeed the printing process is completed when the platform reaches the height set in the process parameters. In the OPC category, it is possible to find all the information related to the internal sensors of the machine, like the power supply, the vacuum chamber, and the temperature of the components. Lastly, the Process category collects all the information sent from the software to the machine hardware, like the powder regulators, and the current phase of the process with the expected completion time.

From these categories, the alarm signals were analyzed in order to split the jobs between two classes: *Successful* and *Failed*. The *Successful* class is composed of jobs that show few alarms but most of all that have the *Alarms.BuildDone* set at the end of the process. On the contrary, the *Failed* class contains the job without the *Alarms.BuildDone* arisen or with the *Alarms.BuildFailed* set. From the other categories, we selected 18 signals that are the most meaningful parameters to be analyzed in order to correctly predict the future status of a job and analyze the performances of the algorithms utilized. A detailed explanation of the signals analyzed is reported in Table 4.2. As a matter of fact, these signals show particular trends before the failure of a printing job and, in non-recoverable critical situations, tend to diverge (See Figure 4.2).

All the parameters selected from the Builds, OPC, and Process categories are given in input to the methodology developed except the *Process.ProcessManager.Task* and *Process.TimeApproximation.BuildDoneTime* parameters.
The *Process.TimeApproximation.BuildDoneTime* stores the date and time when the job is expected to end. Theoretically, this information should be the same from the

Table 4.2 List of signals analyzed

Category	Signal	Description
Alarms	• ArcTripWarning	Occurrence of an electric high voltage arc. High voltage power supply is restarted.
	• BuildDone	Set when the printing job is completed.
	• BuildFailed	Set in case a critical error occurs and the job fails.
	• FilamentShutOffAlarm	The filament is shut off since the beam has been turned off for too long time. For several reasons (e.g., after an ArcTripError).
	• GridLowWithoutBeamOnError	The build is aborted due to the high voltage of the grid cup being too low.
	• LeakageCurrentWarning	A small beam current was detected when not demanded, but the machine is not in a critical condition yet.
	• ModelClipping	Set when the model is clipped because it is too close to the building envelope.
	• RakeFailureWarning	Set when the rake fails to move in the demanded position. Possibly because of high friction (e.g., a component is deformed above the powder bed, and the motor of the rake must apply a higher current to allow the rake to reach the final position).
	• RegulationFailed	It warns about the possible failure of the rake control system in determining the new positions for powder spreading. This can happen when the hoopers are empty or the control is not able to regulate the fetch position to obtain a pulse comparable to the optimal one.
Builds	• State.CurrentBuild.CurrentHeight	It shows the current height of the platform during the printing process.
OPC	• PowerSupply.BeamCurrent	It stores the values of the current provided to the electron beam.
	• PowerSupply.Filament.CurrentFB	It allows to monitor the current flowing in the tungsten filament.
	• PowerSupply.Focus.FocusFB	It shows the current values of the magnetic lenses to focus or defocus the electron beam.
	• PowerSupply.Rake.CurrentFeedback	It reports the feedback current of the rake in the machine.
	• Temperature.BottomTemperature	It collects the temperature measures from the thermocouple placed under the platform.
	• Temperature.ColumnTemperature	It collects the temperature measures from the beam column thermocouple.
	• Vacuum.BackingVacuumGaugeFB	It monitors the vacuum level at the backing pump.
	• Vacuum.ChamberVacuumGaugeFB	It monitors the vacuum level inside the build chamber.
	• Vacuum.ColumnVacuumGaugeFB	It monitors the vacuum level inside the beam column.
	• Vacuum.TurboPump[0].Current	It reports the demanded current from the chamber turbo pump.
	• Vacuum.TurboPump[1].Current	It reports the demanded current from the column turbo pump.
Process	• LeftRegulator.NextPosition	It contains the values of the next left fetch position of the rake.
	• LeftRegulator.PulseLenght	It represents a measure of the amount of powder spread into the build area from the left fetch position in terms of pulse length.
	• ProcessManager.Task	It reports the current phase of the printing process during a job.
	• RightRegulator.NextPosition	It contains the values of the next right fetch position of the rake.
	• RightRegulator.PulseLenght	It represents a measure of the amount of powder spread into the build area from the right fetch position in terms of pulse length.
	• TimeApproximation.BuildDoneTime	It provides the expected time when the printing process will end.

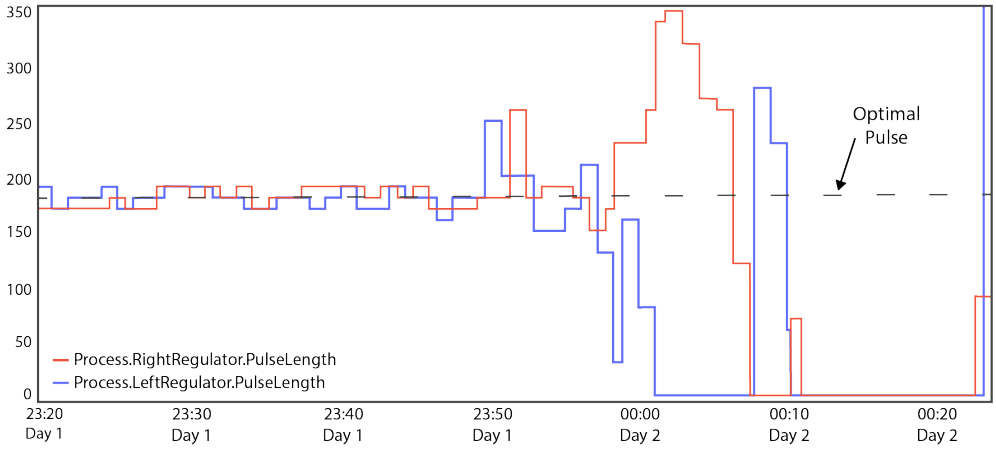


Figure 4.2 Pulse Length signals diverging from the optimal value (dashed line) leading the process to a non-recoverable situation

start of the process until the end, and it is based on the characteristics of the part to be produced and the input parameters. However, during the process, this information is updated and could be used in order to analyze the performances of the methodology developed and how much time is saved by predicting the failure of a job. Instead, the *Process.ProcessManager.Task* signal is used to split the data of each job in order to apply a sliding window approach for the methodology described in the next section.

In the starting phase and in the ending phase of the printing process, there is a transient behavior that is to be expected, in as much as these phases are done to warm up and cool down the system, respectively, before and after the print of the part is done. Heating could last between one hour to two hours while cooling up to ten hours, and a lot of data are produced due to the constant increase or decrease of the values measured by the internal sensors, such as temperature.

In order to avoid training the algorithm with misleading and unnecessary data, we choose to neglect the data produced during these two periods, and we consider only the data of the actual job build. With this in mind, from the complete data, a partial set of data was extracted and used as input to the methodology developed.

Furthermore, the partial data shows a transient at the beginning and end of the build process, which varies between 40 minutes to 1 hour. This behavior is related to the part produced, and it is present in all jobs regardless of the status. Thus, we decided to analyze only the data at steady state that are in between 1 hour from the start of the build and 1 hour before the end of the process (See Figure 4.3).

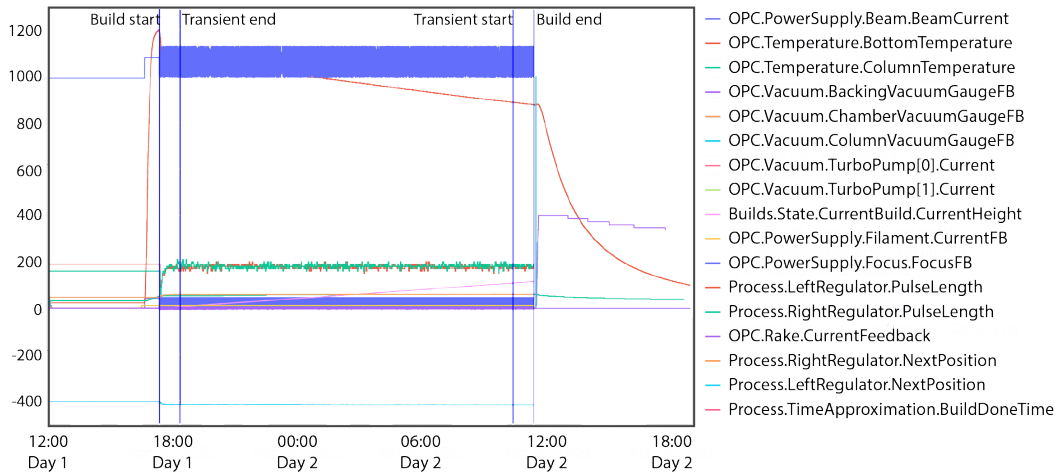


Figure 4.3 Example of complete and partial data

Finally, the data were split based on the standard phases of the EB-PBF process, described previously, in order to generate the input data for our methodology. Each build job was divided into sub-sequences through a sliding window approach, with a window size equal to 64. This is a necessary choice since there is no fixed sampling time for the signals. However, we verified empirically that a sliding window of size 64 is a good compromise because the signal phases repeat on average after 64 values. Thus, this approach allows to consider together all the dynamics of the 16 signals that could occur during the printing of a layer.

4.5 Methodologies

4.5.1 Problem statement and EB-PBF-AE architecture

Our work pursues the promising success of reconstruction-based methodologies for anomaly detection in smart manufacturing [121], specifically focusing on the EB-PBF systems. In this context, detecting anomalous behaviors requires the analysis of raw sensor signals covering an entire production cycle of a given component that might either result in a success (expected) or in a failure (rare but costly). In the latter case, the related time series should exhibit a long-term abnormal behavior significantly different from those observed in correspondence of the physiological conduct of the EB-PBF system.

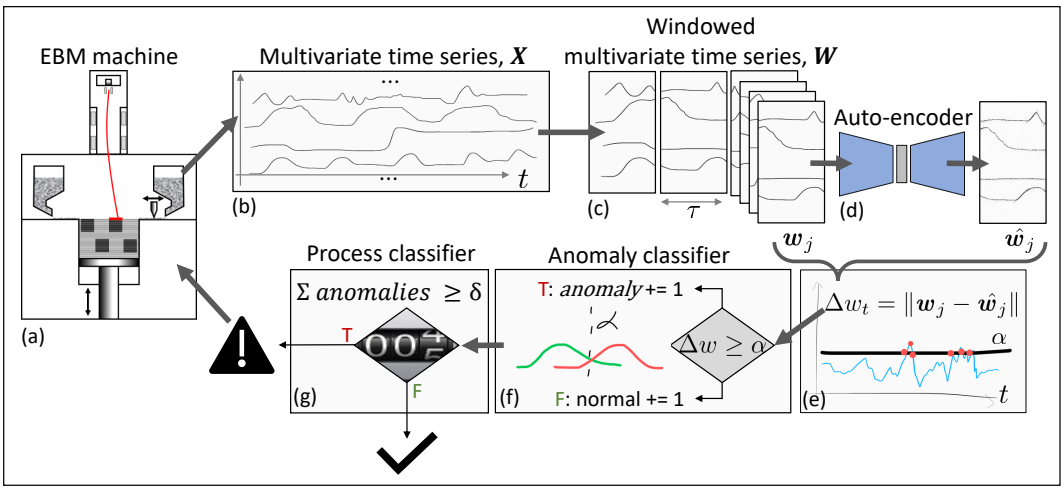


Figure 4.4 Architecture overview

According to the comprehensive review by [4], long-term anomalous behavior falls in the so-called *collective anomalies* category, where the discrimination between the abnormal time series and the rest of the involved data is particularly cumbersome, at least due to a two-fold reason. First, the computational cost to analyze the time series increases significantly with the scale of the dataset. Second, time series are only coupled with a high-level label corresponding to the final state of the process under sensing, namely failed or successful. As a result, any useful information or events concerning time points or interval is completely missing.

This aspect is furthermore aggravated by the fact that the failure can not even be directly associated with a specific time series or sensor.

Lastly, as above-mentioned in Section 4.4, our real-world dataset results are highly unbalanced, the majority of the sample being EB-PBF industrial processes successfully completed.

Formally, given an EB-PBF process p^* of length T, we can denote the associated multivariate time-series observed from our $M = 16$ sensors as $\boldsymbol{X}_{p^*} = \{\boldsymbol{x}_{1,p^*}, ..., \boldsymbol{x}_{T,p^*}\}$ with $\boldsymbol{x}_{t,p^*} \in \mathbb{R}^M$, and $t \leq T$ (see Figure 4.4(b)). For simplicity, and with abuse of notation, we hereafter omit the explicit reference to the process p^*.

During the pre-processing phase, $\boldsymbol{X}$ is transformed into sub-sequences through a sliding window approach, with window size τ equal to 64, empirically set. Thus, as shown in Figure 4.4(c), we can now represent our process p^* as a sequence

of non overlapping observations of length τ: $\boldsymbol{W} = \{\boldsymbol{w}_1, ..., \boldsymbol{w}_{T-\tau+1}\}$, where $\boldsymbol{w}_j = \{\boldsymbol{x}_t, ..., \boldsymbol{x}_{t+\tau-1}\} \in \mathbb{R}^{M\times\tau}$ is the generic $j-th$ window.

Our EB-PBF-AE should predict the final state of p^* by assigning a label $\hat{y}_{p^*} \in \{0,1\}$, where 0 means success and 1 means failure.

Nonetheless, as above-mentioned, the ground-truth binary label $y_{p^*} \in \{0,1\}$, specifying if the given EB-PBF process p^* ends successfully or not, omits any information related to time instants or time intervals. Consequently, we do not have any a priori information expressing if the given sub-sequence $\boldsymbol{w}_j$ is anomalous or not. Thus, we need an anomaly detector that does not explicitly depend on annotations.

As it can be gathered from Figure 4.4(d), our EB-PBF-AE is built on top of an auto-encoder (AE) architecture, where the encoder is devoted to compress the original input ($\boldsymbol{w}_j$), obtaining a corresponding latent representation. At the same time, the decoder produces an output ($\hat{\boldsymbol{w}}_j$) that mimics the original input.

Since the training of the AE neural network makes use of inputs as targets for reference by reducing the reconstruction error between the output and input, no explicit annotations are required in the learning process. This consideration makes the AE paradigm the ideal candidate to learn a proficient representation of our multivariate time series data, missing any annotations concerning time points or intervals.

To handle the multivariate time series nature of our data, we put into effect both the encoder and the decoder of our EB-PBF-AE via Long-Short-Term Memory (LSTM) neural networks, known to be capable of capturing the complex temporal dynamics of current data by exploiting historical values [188]. Furthermore, as previously mentioned, rather than considering each data stream individually, our EB-PBF-AE leverages the complete variable set made up of 16 different time series to apprehend the latent interaction among the physical properties described by the sensors.

As shown in Figure 4.4(e), we leverage the magnitude of the reconstruction error between the original tested signal $\boldsymbol{w}_j$, and the AE reconstructed one $\hat{\boldsymbol{w}}_j$ to trigger anomalous events. Namely:

$$\hat{y}_t = \begin{cases} 1 & \text{if } \Delta w_t \geq \alpha \\ 0 & \text{otherwise} \end{cases} \tag{4.1}$$

Where $\Delta w_t = \left\| \boldsymbol{w}_j - \hat{\boldsymbol{w}}_j \right\|$ is the absolute error between the original and reconstructed signal, $\hat{y}_t$ the predicted anomaly label for time-stamp t (where 0 means normal and 1 means anomaly), and α a predefined threshold, later detailed.

As above-mentioned, the LSTM-based AE is trained over a windowed multivariate time series. This is conventionally done in time series analysis to reduce the back-propagating effort, as well as to address the weights update to the most recent time-steps, assumed to be more relevant for the prediction [107, 121]. Nonetheless, the absolute error Δw_t is continuous over time by definition, as also shown in Figure 4.4(e).

Since the backbone AE was trained only on the data collected during successful processes, large reconstruction errors were expected corresponding to anomalous events, which are not included in the training set.

To proficiently set the threshold α (see Equation 4.1), a well-known outlier detector was exploited leveraging the mean and standard deviation of the distribution of Δw_t: when the reconstruction error deviates away from its mean by 3 standard deviations, the given event is tagged as anomalous [121, 91].

Once an effective anomaly score is defined, it is possible to proceed in the following way to predict the final state of p^*:

1. For all the sub-sequences $\boldsymbol{w}_j \in p^*$ compute the anomaly score Δw_t.
2. Count the number of time-stamps predicted as anomalous and compare such value with a threshold δ empirically set on an independent validation dataset (see Figure 4.4(f)). Formally:

$$\hat{y}_{p^*} = \begin{cases} 1 & \text{if } (\sum_{t=0}^{T} \hat{y}_t) \geq \delta, \text{with } \hat{y}_t \in \{0,1\} \\ 0 & \text{otherwise} \end{cases}$$

4.5.2 Experimental setup

As already anticipated, the backbone of the proposed solution is an LSTM-based AE, whose structure and architectural parameters are reported in Table 4.3. The AE is randomly initialized and then trained for 100 epochs using Adam optimizer with a learning rate equal to 0.001, β_1 and β_2 parameters respectively set to 0.9 and 0.999,

batch-size to 32, and leveraging the mean squared error as loss. The early stopping criterion was exploited based on the training loss to prevent model over-fitting (i.e., loss no longer decreasing for more than 10 epochs stops the training).

Table 4.3 LSTM-AE parameters

Layer	**Output**	**# Parameters**
Encoder 1	(None, 64, 256)	279552
Encoder 2	(None, 128)	197120
Repeated Vector	(None, 64, 128)	0
Decoder 1	(None, 64, 128)	131584
Decoder 2	(None, 64, 256)	394240
Time Distributed	(None, 64, 16)	4112
Total Parameters		1006608

Based on the data available, we randomly split the jobs in order to have 44 successful jobs as the training set for hyper-parameters optimization, 12 successful jobs and 12 failed jobs for validation and optimization of the anomaly classifier, and 6 successful jobs and 6 failed jobs for test and performance evaluation.

The AE was optimized on the training set with a grid search on the following hyper-parameters: layers size and batch size. The hyper-parameters tested range from a maximum of 256 to a minimum of 8. From these tests, the optimal parameters were found to be the following: 256 for the first layer, 128 for the second layer, and 32 for the batch size.

Thus, the threshold δ was optimized based on the optimal model. We tested different values on the validation test and found the optimal one to be $\delta = 12$. Finally, the optimal model and threshold were picked up and exploited on the test set for the evaluation of the performances.

In this study, the following indicators were adopted to evaluate the performances of our methodology: i) Accuracy (ACC), ii) Precision (PPV), iii) Recall (TPR), and iv) F1-score (F1).

The metrics are calculated as follows:

$$ACC = \frac{TP+TN}{TP+FN+FP+TN} \quad PPV = \frac{TP}{TP+FP}$$

$$TPR = \frac{TP}{TP+FN} \quad F1 = 2 \times \frac{PPV \times TPR}{PPV+TPR}$$

Where True Positives (TP) are the Successful jobs, True Negatives (TN) are the Failed jobs, False Positives (FP) are the Failed jobs predicted as Successful ones, and False Negatives (FN) are the Successful jobs predicted as Failed ones.

4.6 Results

This section presents the experimental results of the EBM-AE methodology (see Section 4.5) in predicting the future status of an EB-PBF process and categorizes them into two categories: *Successful* and *Failed*.

First, we trained our model on the training set composed of successful jobs in order to understand the behavior of all the signals in steady-state conditions. The model learns the correlation between each signal and reconstructs them with a loss of 5.1×10^{-4} and a validation loss of 4.7×10^{-4}.

After that, the model is optimized on the validation set, as explained in Section 4.5. It was evaluated with the metrics reported in the previous section obtaining the results reported in Table 4.4. In general, we can notice that the performances are very interesting considering the high variance and complexity of the signals analyzed and the process, with all metrics above 83%.

Table 4.4 LSTM-AE results

Accuracy	**Precision**	**Recall**	**F1-score**
83.3%	87.5%	83.5%	83.0%

The model correctly identifies all the successful jobs that have a standard behavior, while it has some issues in classifying the jobs with a quick variation of the signals and a sudden critical failure that is not predictable from the signal's state history.

In addition to the jobs classification, the performances of our methodology were also evaluated based on how much time before it predicts the failure of a job with respect to the real failure and the expected job ending at that time instant. The results are reported in Table 4.5. In the worst case scenario, our methodology predicts with a margin of 27 minutes before the real failure occurs, while in the best case, with a margin of 19 hours and 23 minutes.

As previously said, the prediction time is highly influenced by the overall behavior of the signals during the job, with a high prediction time in case of jobs with an

Table 4.5 Example of failed job prediction

Job Failed	Prediction time before failure		Expected time before ending		
	hours	minutes	days	hours	minutes
Job 1	0	27	1	19	23
Job 2	1	28	0	14	51
Job 3	3	21	0	23	50
Job 4	19	23	0	19	53

unstable trend from the start and a low prediction time for jobs with stable progress and sudden failure.

This rapid process drift could depend on unexpected anomalies that affect the process stability and lead to an out-of-control process. As examples, these anomalies could be in the raw material (powder contamination, humid or oxidized powder), in an incorrect design of the job (the components to be produced are not uniformly distributed in the build volume, or the distribution is not symmetrical that involve areas with different temperatures and heat accumulation).

The methodology shows promising results in predicting the future status of complex jobs like the ones in the EB-PBF process. In the EB-PBF context, a quick prediction of a failure allows to save a lot of time, materials, and money, otherwise wasted and avoids possible damage to the machine.

4.7 Discussion and Remarks

In this Chapter, we report a new reconstruction-based methodology for the early detection of anomalies in smart manufacturing EB-PBF systems. Following our methodology, we proposed an Autoencoder to reconstruct the signals provided by the EB-PBF machine. Based on the reconstruction error, if the error exceeds a set threshold, an anomaly is detected. Then, based on the number of anomalies, the future status of the job is predicted as Successful or Failed. The proposed EB-PBF-AE solution results to be robust and reliable to detect anomalies during a printing job. Overall, the results are very promising in the AM context, where a quick prediction of a failure allows to save a lot of time, materials, and money, otherwise wasted, and avoids possible damage to the machine. At the research level, such a tool, therefore, becomes of utmost meaning when it is coupled with material development, in which

times are strongly dependent on the trial-and-error approach. The ability to predict job failure could speed up the development of new alloys by limiting the number and time spent running unsuccessful jobs. An advanced prediction of the job's failure may be used as the possibility to regulate online the process parameter and prevent the failure event. At the industrial level, this tool may be coupled with the internal control of the machine, designing closed-loop controls of the process and rules for implementing an adaptive control of the system.

Chapter 5

Automation of post-production inspection in Additive Manufacturing

5.1 Introduction

Some of the work described in this Chapter was also previously published in [30].

The quality of products has become a key factor for success in the Additive Manufacturing (AM) field in Industry 4.0.

This is particularly true in the aerospace and aviation field, where complex and critical parts are produced. However, the inspection of the parts by a human operator could lead to misjudgment due to subjective analysis, human tendencies, environment, and characteristics of the part [112, 117].

The presence of defective parts could be highly expensive for companies aside from being life-threatening for people if a faulty part is assembled on an engine.

The visual inspection may vary depending on the application and part produced [88]. For example, in the aerospace field, the production of honeycomb parts for engines is one of the most demanding in terms of accuracy, complexity, and time required [106].

A honeycomb part is typically composed of several hexagonal cells connected to each other with different heights and slopes (see Figure 5.1).

During production, the honeycomb structures printed with AM technologies are connected to metal support through a hot metal fluid, where solidifying ensures the bounding between the two parts [215]. Since this process involves high temperatures, the two structures are exposed to strong thermal phenomena, leading to different types of structural defects that badly impact the performance of the honeycomb [213]. Such defects can either affect the external shape and morphology of the part or the internal arrangement of the cells. Among external defects, one could count holes in structure walls or missing/damaged cells (see Figure 5.1a-b, respectively) caused by the erosion process. On the other hand, typical internal defects may include not bonded cells or areas (Figure 5.1e) due to a non-homogeneous distribution of the brazing liquid between the honeycomb and its support.

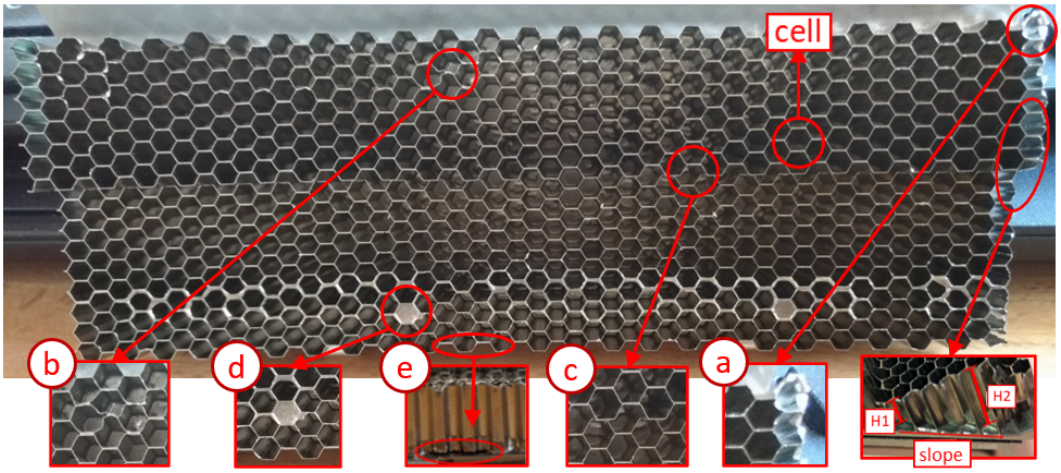


Figure 5.1 Honeycomb structure and defects: a) missing cells, b) damaged cells, c) opened cells, d) excessive bond material, and e) not bonded cells

In order to ensure the quality of the finished part, the structure is thoroughly examined by an operator [55], putting in place specific visual inspection procedures tailored for the different categories of defects:

- *External defects* are inspected by visually observing the part and measuring the size and type of faults as well as the number of missing or damaged cells.
- *Internal defects* are inspected by manually passing an inspection light through each and every contiguous cell of the honeycomb, looking for any light dispersion due to incorrect bonding of the structure (*light-bleeding* approach).

The manual inspection can address the damages that are either in the honeycomb wall basements (internal defects) or in the honeycomb wall medium part (external defects), which are the ones that are most critical and frequent.

However, in both cases, the successful detection of the defects is highly influenced by the ability and experience of the human operator [64]. Hence, it is scarcely reproducible and prone to failure.

To overcome the problems of manual inspection, several studies proposed solutions to automatically analyze the parts produced. In [97], the authors propose a standard program called General Aviation Inspection Training System to train the human operator with advanced technology and reduce human errors. However, this program could only help to slightly reduce human errors as the operator's role is still central during the inspection. In this context, the integration of Machine Learning (ML) and machine vision is expected to provide many advantages to drastically improve the detection of faulty parts and reduce inspection time. For example, in [2], the authors focus on the integration of artificial intelligence and augmented reality to analyze and inspect the part produced; in [95], the authors propose a general scheme to develop a robotic system for vision-based inspection.

While these early works show very promising results in combining robotics, artificial intelligence, and complex systems to achieve full automation of the simplest inspections tasks, the provided solutions are still in the early stages, and they are not tested in a real industrial environment [58]. Indeed, complex inspection tasks like the ones targeting both internal and external defects of a honeycomb still remain unaddressed.

Starting from these considerations, the objective of this work is to propose a completely new approach that can fully automatize the inspection of honeycomb parts (both external and internal defects) in the Industry 4.0 context, possibly making it more objective and less prone to human errors.

5.2 Case study analysis and design concept

In the first stages of our project, we analyzed a limited set of significant samples provided by an aviation company to characterize the most typical defects affecting the parts (see Figure 5.1). Based on the characteristics of such defects and

their acceptability requirements, we conceived and designed the automation of the inspection procedure.

The observed defects can be divided into two main broad categories:

1. *External defects*: this category includes defects that can be visually assessed by frontally observing the part. This includes missing cells (Figure 5.1a), damaged cells (Figure 5.1b), opened cells (Figure 5.1c) and portions with excessive bond material (Figure 5.1d). While the specific characteristics and location of the defects in this category vary, they have a common trait in that they are all represented by discontinuities in the cell profile patterns, which can be either in the form of cell profile breaks (missing/opened cells) or cell profile shape and dimension (damaged cells, excessive bond materials). This makes it possible to conceive defect detection as an Artificial Intelligence framework taking a high-resolution frontal image of the honeycomb as the input. The framework automatically identifies the profiles of the single cells to detect deformed profile patterns specific to different types of external defects.

2. *Internal defects*: this type of defects (not bonded cells, Figure 5.1e, and opened cells, Figure 5.1c) involves the internal structures of the cells. Hence, it cannot possibly be detected from the same frontal view that is used for internal defects. To solve this problem, we exploit an automatic light-bleeding approach similar to the manual one. For this purpose, a special sensor has been developed that uses a set of glass fibers to illuminate the internal walls of the cells. A defective cell is detected if the light is observed in the not illuminated cells.

To fully automatize the inspection, we envisioned an innovative process that integrates the use of Cyber-Physical Systems (robotics), machine vision, artificial intelligence, and smart sensors. The process is conceived as a sequence of two consecutive phases (see Figure 5.2):

1. *Phase 1*: In the first phase, a robotic arm places the camera frontally to the honeycomb and triggers the image acquisition. The proprietary camera software allows to: i) output a high-resolution image showing the frontal view of the honeycomb, ii) obtain a *disparity map*, that provides high-resolution depth information about each pixel of the first image. The two images are processed by a purposely developed Artificial Intelligence framework to automatically

identify the presence of external defects. In case several defects are detected, the part is rejected, and the inspection ends with Phase 1. Otherwise, the coordinates of the centers of the cells, which are identified by the Artificial Intelligence framework, are used as the input to Phase 2.

2. *Phase 2*: The second phase detects internal defects. More specifically, the same robotic arm inserts a glass fiber sensor, specifically designed and developed for this purpose, inside the cells of the honeycomb, scanning the whole part. A light-bleeding method is used to highlight the presence of open cells or not bonded areas in each cell, which is automatically detected and registered by the system.

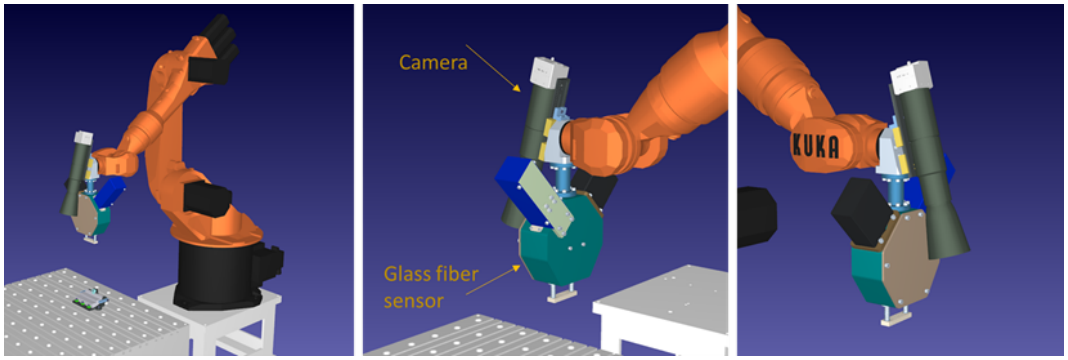


Figure 5.2 Scheme of the inspection process: robotic tools

Starting from this concept, we developed a fully functioning demonstrator in a real industrial environment.

This system can also be used to monitor an AM printing process over time, allowing manufacturers to identify trends and make adjustments as needed rather than waiting until the entire object is printed. For example, the inspection system could be programmed to move the camera around the 3D printed object, capturing images of each layer and using AI and ML algorithms to analyze these images in order to detect any defects, such as warping or distortion, that may be present. This can save time, reduce material waste, and ultimately improve the quality of the final product.

In the following Sections, we describe in detail the components of the prototype inspection system (Hardware and Software) as well as the overall experimental setup of the demonstrator.

5.3 Automated inspection system

In order to develop an automatic procedure for defect detection, a robotic setup was developed in a real industrial plant. The setup is composed of the following hardware and software components (Figure 5.3):

1. *Kuka KR16-2 robot with KRC4 robot controller*: the camera and the glass fiber sensor are attached to the robot flange. The robot will move the attached tools and properly place them for inspection.

2. *Ingesys IC3 controller*: the camera and the glass fiber sensor require the activation of digital signals in order to start the acquisition for synchronization purposes between the robot moves and the inspection procedures. This PLC is used by the robot as an interface due to the lack of available digital input/output ports.

3. *SVS-Vistek hr455CXGE camera and 3D lens*: by means of the specific 3D lens, the camera acquires images that are used in Phase 1 external defect detection procedure as well as for the cell location.

4. *Optical fiber sensor*: the sensor developed for Phase 2 internal defect detection is composed of several glass fibers and a microcontroller for data processing. The fibers are inserted inside the cells, and light reception is analyzed for defect detection.

5. *Disparity Map software*: the disparity map software provides depth information about each cell in the honeycomb that is used in Phase 1 for defect detection.

6. *Artificial Intelligence framework*: the framework is in charge of detecting external defects in Phase 1 by means of automated segmentation techniques. It rejects a part in case the requirements about the number and severity of the defects are not met.

A special tool holder was developed to attach both the glass fiber sensor and the camera to the same robot. The whole integrated system will be hereafter referred to as *the tool*.



Figure 5.3 Inspection setup

5.3.1 Robotic setup

The robot is in charge of moving the tool during the inspection process. The robot trajectory is programmed offline with a predefined trajectory and executed on the KRC4 controller. The robot motion and the process are synchronized through digital input/output signals and TCP/IP communication with the Ingesys IC3 external controller. The visual robotic inspection procedure consists of the following steps:

1. Phase 1 starts, and the robot moves the tool with the camera to acquire an image of the part.
2. The camera acquires the image, and a disparity map is calculated to obtain the depth of each cell.
3. Both the image and the disparity map are analyzed by the Artificial Intelligence framework to detect external defects:
   (a) If the part is not OK, it is rejected, and the inspection process is finished.
   (b) If the part is OK, the inspection process continues with Phase 2.

4. The Artificial Intelligence framework computes the location of the part precisely and provides this information (i.e., coordinates of the cells) to the robot to adjust the trajectory for Phase 2.

5. The robot moves the tool in order to insert the glass fibers inside each honeycomb cell to detect internal defects with a light-bleeding approach.

6. When the Phase 2 inspection is finished:

   (a) If the part is not OK, it is rejected, and the inspection process is finished.

   (b) If the part is OK, the part has successfully passed the inspection process.

5.3.2 Camera setup

As mentioned in the previous section, each honeycomb part has a complex structure composed of several hexagon-shaped cells. In order to obtain a reliable part location and detect if a cell is defective or not, it is required to analyze in detail the part produced and measure the shape and height of each cell. Due to the small size of each cell (6 mm width and up to 12 mm depth), standard stereo systems are physically unable to inspect those structures and calculate the depth of each cell. This is mostly due to the large baseline typical of those systems.

To overcome these problems, a novel multi-view technology, having a very short baseline, is used. This allows to analyze the structure and depth of each cell, avoiding any occlusion problems, and it also brings the following advantages:

1. As a multi-view system, it records 9 images having different perspectives in one shoot. This rich data improves the output of machine learning algorithms.

2. Standard camera components and lenses can be used together, reducing general costs and increasing the integration flexibility in automated systems.

3. It is robust against illumination changes and reflections.

In order to ensure high accuracy in the depth estimation, a full frame industrial camera (SVS Vistek) is used that is able to acquire images with a resolution of 61Mpix (see Figure 5.4).

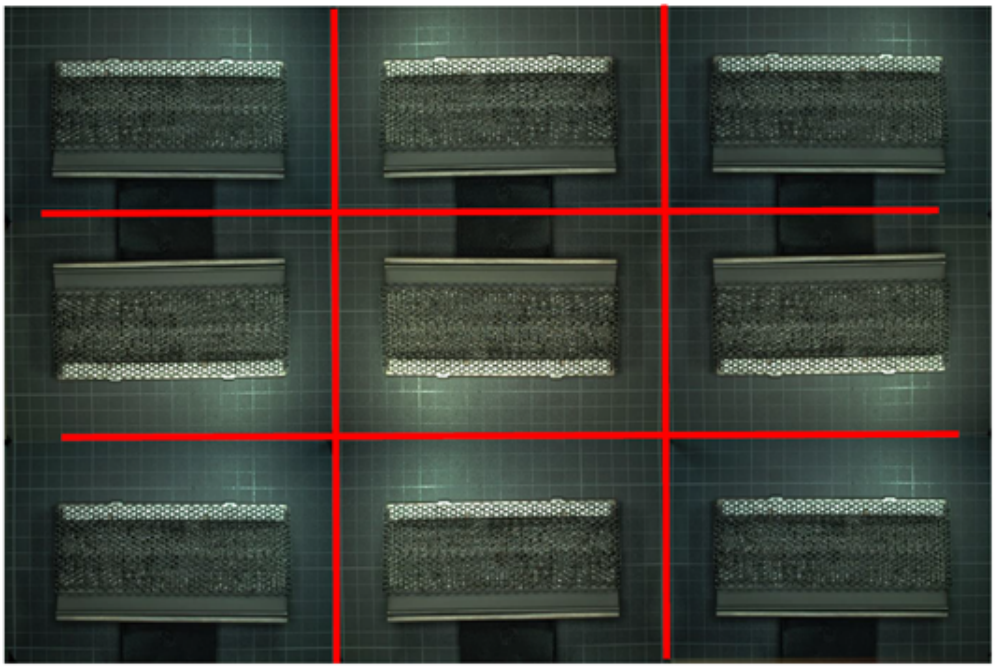


Figure 5.4 Multi-view image of a honeycomb

5.3.3 Disparity Map software

The acquisition of images using a multi-view stereo system allows to extract 3D information from digital images in a manner similar to human binocular vision. By comparing information about a scene from two different points, the relative depth information can be obtained in the form of a disparity map, which encodes the difference in horizontal and vertical coordinates of corresponding image points [82]. The values in this disparity map are inversely proportional to the scene depth at the corresponding pixel location. By calibrating the system, it is possible to convert a disparity map into true depth.

As described before, it is possible to capture with one physical lens 9 different perspectives, as shown in Figure 5.4. In this Figure, each sub-image looks slightly different from the others, allowing to obtain a disparity map that is more robust and contains richer data than the one derived from two perspectives. By using a single camera, a mirror system called Image Multiplier, splits the light rays into separate images, which are captured simultaneously on the camera sensor. The high number of viewpoints allows to deal with occlusions as well as complex surface materials such as metal or glass (transparent, glossy etc.).

In Figure 5.5, an example of the obtained disparity map using proprietary software is shown. The depth of each cell is shown using a color scale, where lighter color indicates a lower distance to the viewer. In this image, it is also possible to see that the base of each cell has a darker color than its walls, indicating that the cells of this particular honeycomb do not have an excessive deposit of brazing material.

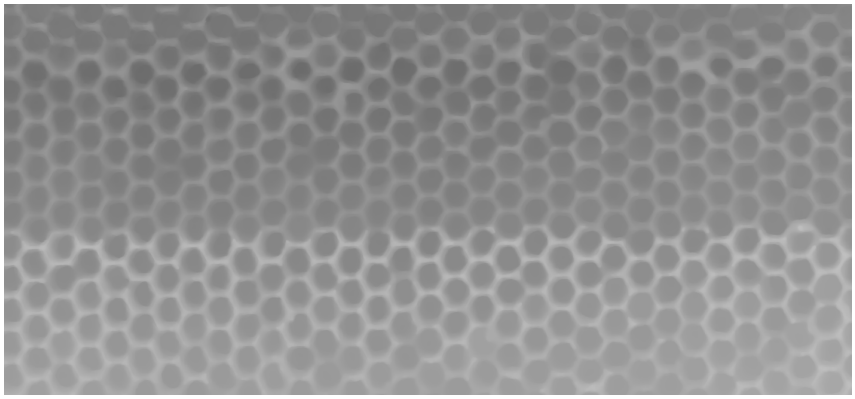


Figure 5.5 Disparity map

5.3.4 Optical fiber sensor

In order to automatize the inspection of internal defects for Phase 2, an optical fiber sensor was designed and specifically manufactured for this purpose. If a cell is internally damaged, by illuminating it, rays of light are reflected in its neighboring cells. Therefore, monitoring the neighboring cells makes it possible to understand whether one cell is healthy or damaged. The optical fiber sensor (see Figure 5.6) is composed of the following parts:

1. A powerful LED light generator.
2. Two types of glass fibers, the emitting ones (actuators) and the passive ones (sensors).
3. A closed box with a mini camera and a microcontroller.
4. A sensor holder, which enables the connection between the robot end-effector, easily adaptable to any robot model and brand.
5. The fiber driving adapter, in which the fibers are collected and driven in order to simplify their insertions in the honeycomb cells.

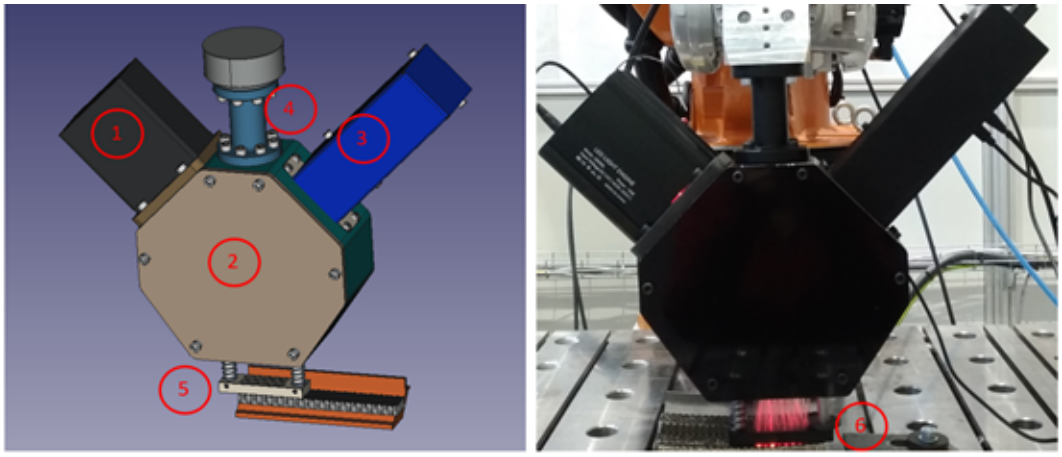


Figure 5.6 Comparison between the sensor 3D CAD design and the first manufactured prototype

In Figure 5.7, the holder and its fibers at work are shown. The holder has been rigidly connected through screws to metal cylinders. Instead, those ones are connected to the sensor body through ball bearings, which enable the cylinders to slide when the spring is pressed. The bearings also avoid the transmission of undesired forces to the holder. When the robot end effector moves upward, the spring is released, and therefore, the holder regains its original position.

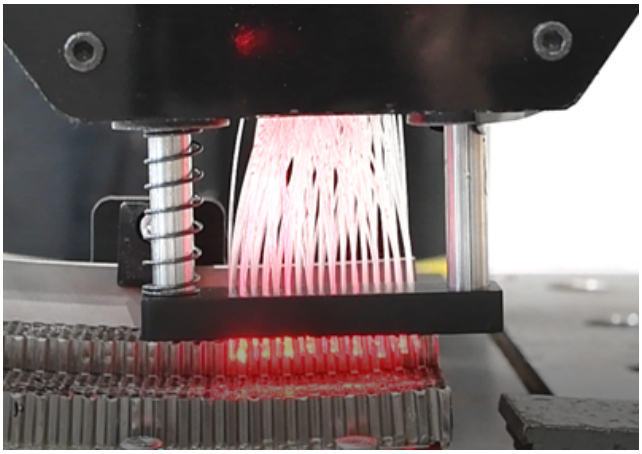


Figure 5.7 Fiber holder with emitting fibers (lighted) and sensor fibers (not lighted)

5.3.5 Artificial Intelligence framework

The Artificial Intelligence framework is in charge of detecting external defects in Phase 1. It is implemented using a modular framework, with a cell segmentation module followed by a defect detection module. More specifically:

1. The *segmentation module* takes the acquired honeycomb image and disparity map as the input and provides a segmentation of the single cell profiles/areas, as well as the full list of coordinates of their centers, as the outcome.

2. The *defect detection module* takes as input a set of quantitative features extracted from the segmentation results (e.g., number of connected regions, dimension of the regions, etc.) and identifies the faulty regions accordingly.

To improve the results obtained from images with different characteristics, the segmentation module is based on a combination of classic approaches (standard edge detectors, adaptive thresholding techniques) to detect the main characteristics of the honeycomb, as well as on a deep learning architecture specialized for semantic segmentation (U-Net, see Figure 3.4).

While the classic segmentation strategies have the advantage of being unsupervised in principle and hence do not need to be trained on pre-annotated images, U-Net is a supervised framework that learns the segmentation strategy based on a training set of input images and corresponding segmented maps. Nonetheless, unlike other types of deep architectures, it is known to be quite accurate even when trained on a relatively small number of images.

To address the issue of low availability of training samples, the defect detection module is developed as an anomaly detection system leveraging a set of ad-hoc rules for each category of defects (e.g., if the area of the cell is out of the expected range, then it is a defect).

To speed up the development of the framework, the images used for training the model were acquired in a laboratory environment, using a fixed frontal camera with the same characteristics as the one used in the real demonstrator (*off-process setting*). Then, for testing purposes, we used images acquired in a real industrial environment, using the camera mounted on the robotic arm (*in-process setting*).

The model is trained on small annotated Regions of Interest (ROI) of the honeycomb image, together with a binary mask of each cell wall. The unsupervised module used for segmentation allows a first analysis of the structure of the honeycomb, while the supervised network refines the results and detects the profile of each cell. Finally, the defects detection module receives in input the results from the segmentation and detects the anomalies. Both approaches are built on top of standard Python libraries and frameworks for Computer Vision and Deep Learning (OpenCV, Keras, etc.), allowing easy integration and automation of the Machine Vision system.

5.4 Experimental inspection process

To demonstrate the functionality of the automated process, we performed a full demonstration (Phase 1 + Phase 2) on a few test samples in a real industrial plant. In the following, we describe the obtained results.

5.4.1 1st phase inspection

In the first phase, the robot starts from the initial position and moves the fixture with the camera above the honeycomb, as shown in Figure 5.8.

When the robot is in position, the camera is triggered, and a new multi-view image (kaleidoscopic image) of the part is acquired (Figure 5.9).

The nine views captured with the camera are pre-processed, cut into individual sub-images, rectified, and used to compute the disparity map of the honeycomb as described in the previous section. As visible in Figure 5.10, in the calculated disparity map, most of the cells look regular, with a light border and a dark floor (the border is closer to the camera, therefore, is lighter; instead, the floor is farther and, therefore, darker). Some cells instead show a floor depicted with lighter color. This is a first indication that those cells are damaged. At this stage, using the disparity map, the height of every cell can be computed, and possible cells filled with brazing material can be identified.



Figure 5.8 Image acquisition

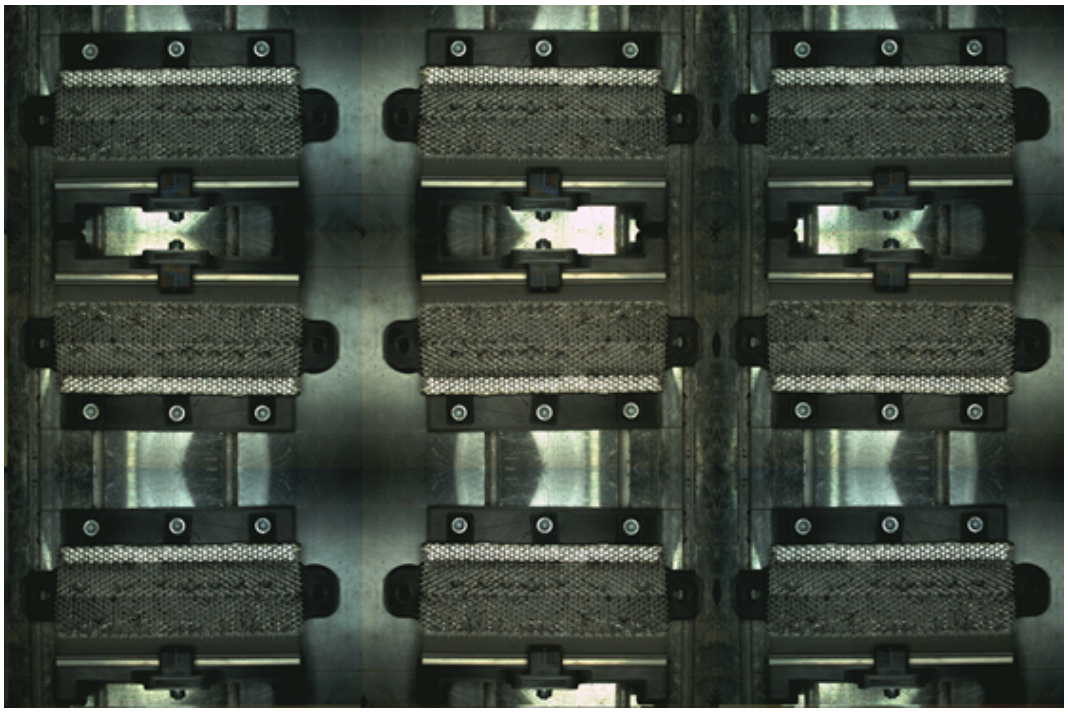


Figure 5.9 Honeycomb image with different perspectives

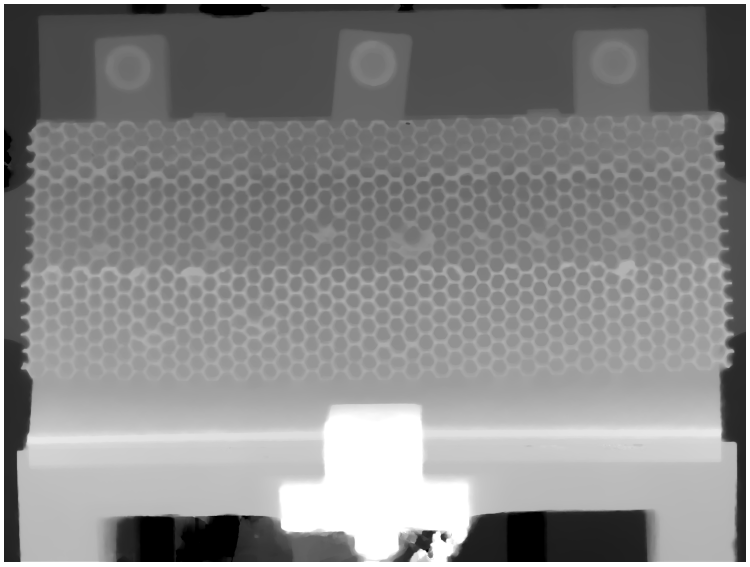


Figure 5.10 Disparity map of honeycomb in a real plant

5.4.2 Inspection results

After the calculation of the disparity map, the two images are collected and provided to the Artificial Intelligence framework.

The algorithm analyzes the two images through a segmentation module that provides a segmentation of the single cell profiles/areas as the outcome (Figure 5.11(a)) and a defect detection module that takes, as input, quantitative features extracted from the segmentation results (e.g., number of connected regions, dimension of the regions, etc.).

The defects detection module identifies the faulty regions accordingly, detects the shape of each healthy cell in green, and shows the defected cells based on the gravity of cell distortion (red $\geq 30\%$, yellow $< 30\%$ and purple $< 5\%$) (Figure 5.11(b)).

The analysis allows to detect superficial defects like damaged cells, missing cells, and opened cells. The algorithm also calculates the coordinate of the center of each cell and stores them in a text file. These coordinates are used by the robot in the next phase to correctly locate the honeycomb.

The first phase inspection requires $\sim 15s$ on a basic CPU or $\sim 8s$ on a basic GPU, which is well below the time required for a manual inspection.

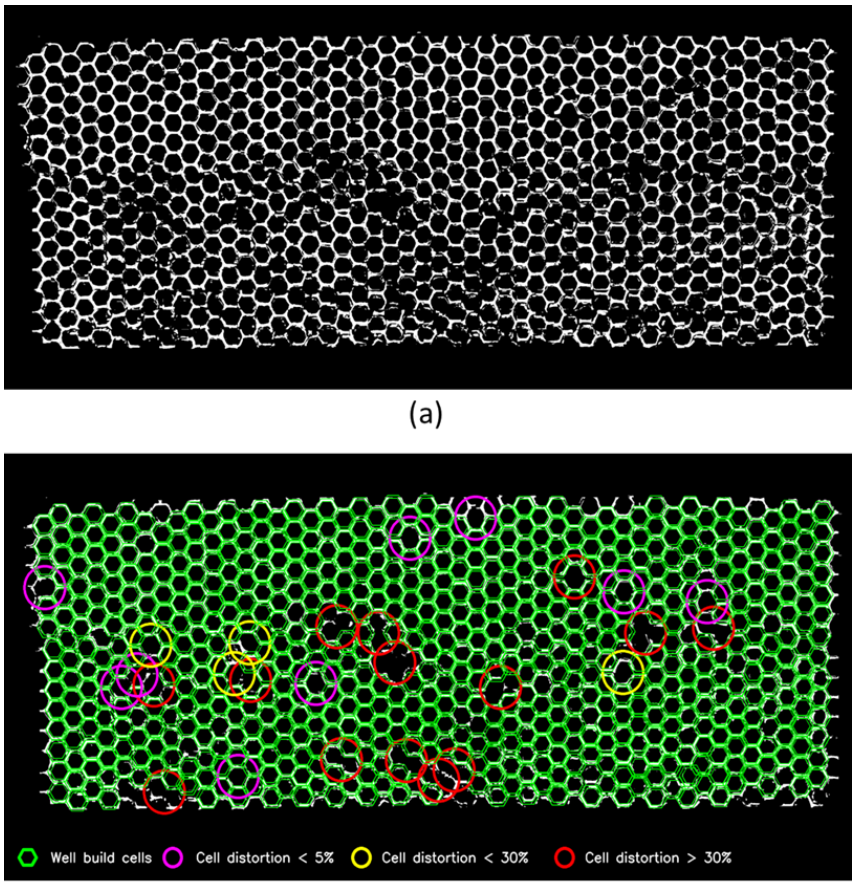


Figure 5.11 Honeycomb cells segmentation (a) and honeycomb defects analysis (b)

5.4.3 1st phase validation

To train and test the Artificial Intelligence framework, images of defected honeycombs were taken, and a mask of the part was created manually (Figure 5.12) to serve as the ground truth for the honeycomb segmentation.

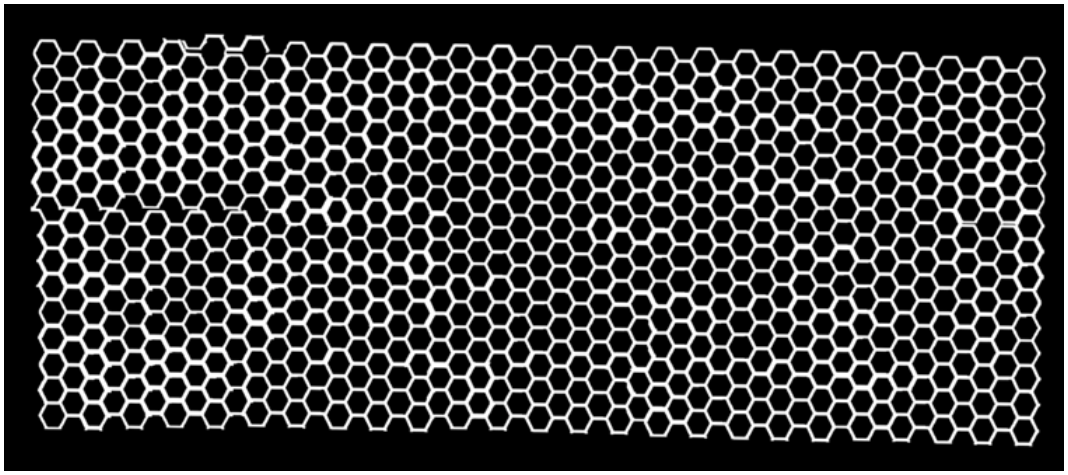


Figure 5.12 Honeycomb mask

In the aerospace industry, the number of defected parts available is very small. Thus, the dataset was built by taking images of 8 different honeycombs, split into two completely independent groups:

1. *Training honeycombs*: the group is composed of 6 honeycombs that are used for training the algorithm and optimizing the parameters.
2. *Testing honeycombs*: the group is composed of the remaining 2 honeycombs that are used solely for testing purposes.

To boost the significance of the test, the two honeycombs chosen for testing are very different in terms of the level of defectiveness. The training/testing images were acquired with the two different experimental settings described previously:

1. *Off-process*: this setting was implemented in a laboratory with the camera mounted in a fixed position above the part. Of the 95 off-process images that were acquired, 75 were included in the training group and 20 in the testing one.

2. *In-process*: this is the setting of the integrated demonstrator built in a real industrial environment, but due to logistics difficulties, only 20 images were taken and added only to the testing group.

Finally, we performed two different testing experiments:

1. The first experiment (*Off-process testing*) trains the model with the training group (that is composed of 100% off-process images) and tests it with the off-process testing ones. In this case, the training and testing images are from the same acquisition environment, which provides the best conditions possible for the algorithm to be successful.

2. The second experiment (*In-process testing*) trains the model with the off-process training images and tests it with the in-process testing ones. While the choice of not re-training the model with more in-process images represents an additional challenge for the algorithm and hence may be considered not ideal, this experimental configuration allows to verify further the robustness of the model to environmental conditions that are sensibly different from the ones used for training.

The defective cell detection results of such two experiments are reported in the following table in terms of accuracy, sensitivity, and specificity.

Table 5.1 Experimental results

	Off-process setting	**In-process setting**
Accuracy [%]	87.99	76.89
Sensitivity [%]	70.31	60.68
Specificity [%]	88.37	77.36

As can be seen in Table 5.1, the off-process testing metrics showed better results, as expected. On the other hand, the in-process testing experiment shows a reduction of accuracy that is reasonably limited (around 11%). On top of that, the obtained values are above 60% for all the figures of metrics, which suggests a good level of robustness of the algorithm. One can conclude that the model performance is reasonably good in the given challenging experimental conditions, with a large room for future improvement when the framework is trained on a larger number of parts, with the same acquisition settings as in the ones used for testing.

5.4.4 2nd phase inspection

After the inspection of the honeycomb in Phase 1, if no critical external defects are detected, Phase 2 is carried on. In this case, the robot rotates its end effector in order to place the optical fiber sensor parallel to the honeycomb surface and to start the automatic inspection inside the cells. When the glass fibers are inside the cells, 24 fibers are lit, and the neighbors' cells are inspected. At the same time, the second end of the sensing fibers is monitored by the mini camera located inside the body of the tool to detect unexpected light. The images acquired by the mini camera are processed by the microcontroller. If no light is detected from the sensing fibers, the microcontroller returns that the analyzed cells are healthy (Figure 5.13, left side). Instead, if one or more sensing fibers light up, the microcontroller returns that the cells are damaged (Figure 5.13, right side). Since each fiber is in charge of inspecting one individual cell, it is possible to precisely locate the position of the damaged ones. In Figure 5.13, the blue dots represent healthy cells, while the red dots represent damaged cells.

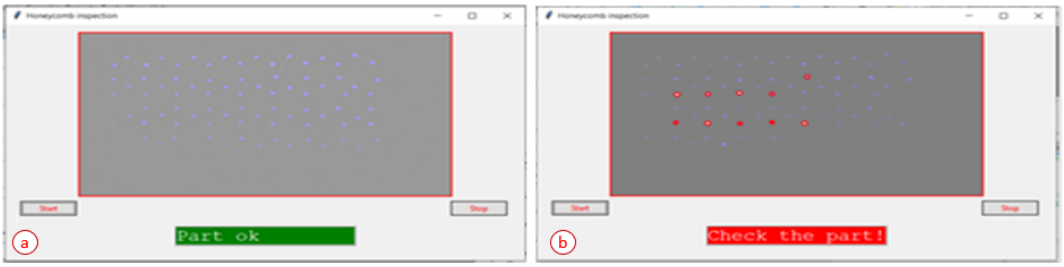


Figure 5.13 Interface with the microcontroller output: (Left side) output for healthy cells, (Right side) output for heavily damaged cells

With this procedure, the whole inspection for Phase 2 can be carried on. First, the optical fibers sensor is moved in order to position the fiber inside the honeycomb cells.

As soon as the floor of the cell is reached, the robot waits for $3s$. At this moment, the camera acquires an image of fibers and processes it in order to detect potentially damaged cells. The result (OK/NOT OK) is then displayed in the user interface together with the acquired image. After $3s$ from the image acquisition, the robot releases the fibers and moves them to a new group of cells to repeat the procedure until all the cells of the honeycomb are inspected. The described process can be

easily sped up by increasing the number of used fibers and/or improving the robot motion algorithm.

In Figure 5.14, an example of the glass fiber sliding process inside the honeycomb cells is depicted. Since the glass fiber ends are contained and driven inside the black holder, it is possible to make them slide inside the honeycomb cells avoiding any possible fiber deformation or damage. The end part of the fibers can be cut in order to have a pointed shape, and this ensures easier sliding inside the cells. All the experiments for Phase 2 were directly carried out on the real plant, showing promising results in detecting internal defects with a 100% accuracy.

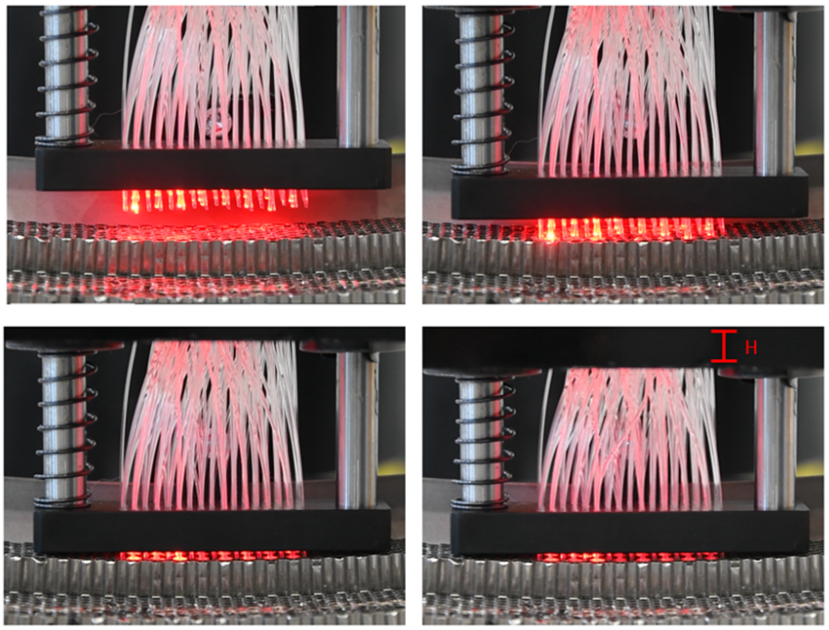


Figure 5.14 Glass fiber sliding inside the honeycomb cells

5.5 Discussion and Remarks

In this Chapter, I propose a new approach to automatize the inspection procedure of complex honeycomb parts in the aviation industry. The proposed 2-phases inspection process results to be fast and robust, less prone to typical human errors. The automated inspection is able to identify with a good level of accuracy and robustness all different types of defects (either internal or external) that may be the cause of

rejection of the part, using a two phases procedure. While Phase 1 addresses external defects, Phase 2 aims to identify potential internal defects.

The performance of Phase 1 can be largely improved in the future by enlarging the training set with many more honeycomb parts and defect examples. The system can also be easily extended to new categories of defects.

For Phase 2, the first prototype of the honeycomb sensor is designed and manufactured, which is able to successfully detect internal damage of the cells. It has been experimentally demonstrated that sliding the fibers inside the cells is an easy and quite robust task. In the future, sensor shape and weight will be largely improved.

At the present stage, the total estimated inspection time, which takes into account all the steps of the procedure, is in the range of 40-60s, which is a very promising result. Future works addressing better automation of the acquisition and image pre-processing stage, as well as the optimization of the computational aspects of the processing stages, are expected to improve the performance and reduce inspection time by at least half.

Unlike manual inspection, the procedure is completely repeatable, and the obtained results are traceable and documentable in every part, as expected in an Industry 4.0 context.

Chapter 6

Industrial indoor localization

6.1 Introduction

In recent years, similarly to what is happening in cities and public spaces, smart and connected objects are increasingly being also deployed in industrial environments. One of the effects of the development of this so-called Industry 4.0 is an increase in the number and type of automated vehicles that move in industrial environments. In this context, indoor localization systems become important to improve logistics effectiveness and productiveness. In many industrial sectors, the quick and automatic localization of goods (components, instruments, partial assemblies, etc.) in large environments would provide a critical production speed up.

For instance, a critical problem in the automotive industry is the quick localization of partially assembled vehicles which are temporarily removed from the pipeline and placed in large warehouses, e.g., while waiting for the availability of some of the parts required for their completion.

In literature, several existing technologies like GPS [155], Wi-Fi [200], and Bluetooth [223] are adapted to provide indoor localization services in domestic environments. While GPS performance indoors is poor [155], Wi-Fi and Bluetooth show promising results, with the latter being particularly interesting due to its low cost and energy consumption features [199].

The use of Wi-Fi for indoor positioning is well established and is known to provide errors in the order of a few meters [39]. However, the requirements in

terms of topological distribution, the number of Wi-Fi access points associated with the costs, and power consumption make this solution unfeasible without consistent retrofitting, which is often undesirable in existing industrial plants (e.g., making power outlets available at deployment points).

Bluetooth Low-Energy (BLE) uses the same frequency of Wi-Fi ($2.4GHz$) but is designed as a short-range energy-efficient communication protocol, allowing compatible devices to communicate through short messages with minimal overhead [74]. Thanks to these features, BLE-compatible devices can transmit periodic messages for years within a single cycle of a small form-factor battery [131], thus being perfectly suited for easy installation in any environment.

BLE-based localization is typically performed by installing a set of proximity beacons (i.e., BLE transmitters) at known locations. Receivers extract the RSSI (which is a proxy of the distance from the sender) from the nearest beacons and use these values to predict their own position [211].

In general, BLE-based localization algorithms can be split into two categories: distance-based and fingerprinting-based [27].

Distance-based algorithms directly translate RSSI values into position coordinates for the object being localized. These methods require that at least three RSSI measurements from nearby beacons are available and apply either trilateration or min-max to transform these measurements into position estimates [171].

In contrast, fingerprinting-based algorithms exploit a vector of RSSI measurements in known fingerprint positions to create a so-called Reference Fingerprint Map (RFM). Then, a machine-learning regressor is fed with the RFM data to build an association rule between new RSSI measurements and their corresponding position estimates [110].

Although these techniques have been proven effective in domestic settings, smart industry environments pose several new challenges that need to be addressed: the areas where objects have to be localized are much larger than in the domestic case, and they constitute a harsh environment for BLE-based localization, due to the presence of large metal objects (robots, carts, etc.) causing signal interference, multi-path propagation, and shadowing.

Despite the vast literature on BLE-based localization, very few works perform detailed comparisons between different algorithms under the same experimental

conditions (i.e., target environment, type and number of beacons and receivers, etc.). In particular, to the best of our knowledge, no one focuses on applying these methods in real industrial environments.

In this work, we perform such a comparison considering two real industrial settings (i.e., a shed and a workshop). We evaluate the effectiveness of several state-of-the-art algorithms (both distance-based and fingerprinting-based) under varying experimental parameters, changing the number of installed beacons and the density of the RFM. Moreover, we assess the robustness of fingerprint-based algorithms to variations in the environment and random noise.

Our goal is to assess the feasibility of indoor localization in industrial environments using commercial off-the-shelf BLE transmitters and receivers due to their low cost, ready availability, and ease of installation and usage. The experiments performed are consequently tailored for this target and are based on commercial devices and protocols currently available in the market.

Similarly, the goal of this work is to test the effectiveness of state-of-the-art localization algorithms for BLE localization in an industrial scenario. Therefore, we do not aim to design a new specialized algorithm for this task, and we just optimize the hyper-parameters of existing solutions to the best values for our target.

Our results confirm that fingerprinting-based methods reduce the positioning error compared to distance-based methods, even in an industrial setting. However, we could not identify a clear winner among the three fingerprinting-based algorithms considered in our experiments, as they all perform similarly both in terms of positioning error and computational complexity. Therefore, we base our final selection on simplicity and flexibility, which leads us to choose k-NN, as explained below.

The rest of this Chapter is organized as follows. Section 6.2 reviews relevant literature solutions. Section 6.3 presents the localization algorithms analyzed in this study, and Section 6.5 and Section 6.6 discuss the experimental results. Finally, Section 6.7 offers our concluding remarks. Some of the work described in this Chapter was also previously published in [33].

6.2 Background and Related Work

Indoor localization methods based on the received signal strength indication (RSSI) from BLE devices have become mainstream thanks to their low cost, wide coverage, and simple hardware required. In comparison, more accurate alternatives, such as those based on the Angle Of Arrival (AOA) or on the Time Of Arrival (TOA), require significantly more expensive hardware since they rely either on an array of BLE antennas or on a very accurate synchronization between the transmitter and receiver [40].

The RSSI is a measurement of the power present in a received radio signal. It is typically computed as the ratio between the received signal power and the transmission power, which is a known value set by the manufacturer and attached to each packet sent by the beacons [99].

As mentioned above, RSSI is generally used for two types of localization methods: distance-based methods [171] and fingerprint-based methods [189]. While the former tries to directly convert RSSI values to distances, the latter uses an RFM to build a model that associates new RSSI measurements to a position.

The two main distance-based methods are trilateration [110] and min-max [157]. Both convert RSSI values to distances between the beacon and localized device using the quadratic decay law of power with respect to distance.

Trilateration determines the location of an object by combining measurements from three spatially-separated known locations (beacons). The position of the device is obtained as the intersection of the three circumferences having the location of one beacon as the center and the computed distance as radius [171].

The min-max algorithm constructs a square bounding box around each beacon using its known coordinates as the center and two times the distance from the target as the side length of the box. Next, the intersection of three boxes is computed by selecting the min-max coordinates. This approach is computationally simpler than trilateration, but it is also more prone to errors.

In both cases, the error obtained by distance-based methods is influenced by the number of installed beacons and, in particular, by their *density* (number of beacons per square meter). Moreover, trilateration and min-max are influenced by multi-paths and fading. Multi-paths are propagation phenomena that occur when radio signals

reach the receiving antenna by two or more paths. Multi-path propagation causes constructive or destructive interference and phase shifting of the signal, which in turn can vary the measured RSSI. Fading is a more generic variation of the attenuation rate of a signal due to obstacles or frequency disturbances, which can result in a loss of signal power.

Several works use distance-based positioning with BLE beacons in various domestic or public environments. In [79], the authors use trilateration in a hall of $7m \times 5m$ where the maximum distance from the beacons to the target is less than $9m$. Despite the small size of the environment, their experiments using three beacons produced an average error of $2.766m$. In [110], the authors optimize the trilateration algorithm by applying a Kalman filter to reduce noise in the RSSI. Park et al. [159] calculate the distance using the RSSI, and then they refine the result by applying two Kalman filters to reduce the error. Using six beacons in a polygonal living room ($8.5m \times 13m$), they obtain an average error of $1.77m$. In [134], the authors use the min-max algorithm to obtain the object position in three environments: a classroom and two construction sites. The classroom has a size of $7m \times 6.4m$, and the distance between beacons varies from $3m$ to $4.8m$; the average error obtained is $1.55m$. The two construction sites have the same size ($6.3m \times 5.1m$) and are tested with six and three beacons, respectively, positioned with a spacing ranging from $1m$ to $3.4m$; the average errors obtained are $1.22m$ and $2.58m$.

Fingerprinting refers to the type of algorithms that estimate the location of an object by exploiting previously recorded data at known positions. Therefore, RSSI-based location fingerprinting requires a preliminary setup stage. An inspection of the environment is performed to select reasonable data collection points to create the RFM. The coordinates of these points and the corresponding RSSI values from nearby beacons are collected. Then, these data are used to build a model that associates a new RSSI measurement to its corresponding location. Several types of models, typically based on machine learning, are considered in the literature. Therefore, fingerprinting allows for partially accounting for multi-paths and fading due to fixed (not moving) elements of the environment, as these effects are also measured in the "training" RFM data.

In [43], the authors use the k-Nearest Neighbors (k-NN) algorithm with $k = 8$ and different numbers of fingerprints points in a living room of $6.66m \times 5.36m$. With an RFM of 50 points, they obtain an average localization error of $1.03m$, while with

an RFM of just eight points, they obtain an average error of $1.91m$. Dierna and Machì in [133] set up an experiment with 31 beacons in a museum of $670m^2$ using two different sampling densities to create the RFM and a k-NN model with $k = 3$. In the first experiment, the density is varied from 0.7 reference points per square meter in museum halls to one point every four square meters in the courtyard (224 total RFM points). The authors also experiment with averaging RSSI measurements over multiple samples, obtaining an error that varies from $1.72m$ (1 sample) to $1.28m$ (10 samples). Then, the authors try to increase the RFM sampling density in the proximity of structural points, obtaining 239 total points and reducing the error to $1.70m$ (without averaging).

In [161], authors develop an Iterative Weighted k-NN (IW-kNN) for fingerprinting-based localization. They select $k = 4$ and use 10 beacons in a test room of $18.8m \times 12.6m$, obtaining an average error of $2.52m$. In [116], the authors propose a k-NN with $k = 1$ in a small office room. Using 20 beacons and an RFM of 120 points, the average error obtained is $1.9m$.

Other studies map the localization problem to regression or classification using Support Vector Machines (SVM) or Neural Networks (Multi-Layer Perceptrons - MLPs). In [141], the authors test different configurations of an MLP in two different locations, a room of $4m \times 4m$ and a $36m$-long corridor. In the test room, they use four beacons and 16 RFM points, obtaining an average error of $2.48m$. In the test corridor, nine beacons and 36 RFM points yield an average error of $1.25m$. In [175], the authors use an SVM in a living room of $3.53m \times 8.09m$ divided into six zones, with two beacons placed in the middle of the room, equidistant from the walls. A set of RSSI values are taken in different zones of the room and out of it to test the algorithm. The SVM shows an accuracy of 88.54% in classifying the zone.

Both distance-based and fingerprint-based localization algorithms can be applied either directly to the raw RSSI values measured at the receiver or after extracting some higher-level features by aggregating or processing those raw values. Table 6.1 summarizes the feature extraction methods adopted in the relevant literature works on this problem.

As shown in the table, many works [199, 39, 27, 40, 168, 110, 134, 175, 60, 59, 17] do not perform any feature extraction and work directly on raw RSSI data. Another common solution [171, 110, 189, 159, 133, 116] is to use as a feature the average of RSSI measures over a short interval of time (from a few seconds to some

minutes) for noise rejection. Other authors [157, 161] also use the average RSSI over a time window but remove outliers before computing it. Mazan et al. [141] use the RSSI sample maximum rather than the average, while Grzechca et al. [79] consider several different summary statistics as possible features for localization (sample average, median, etc.). Eventually, however, they determine that the average/moving average is the feature that yields the best results. Lastly, Daniş et al. [43] store the entire histogram of RSSI measurements over a time interval, where each bin of the histogram could be considered as a different feature. However, they do not use these features to perform localization. Rather, they use them to generate a dense RFM from a sparse one.

According to literature [60], there are many factors affecting the accuracy of BLE-based localization. When a receiver moves through the environment, the RSSI signal can vary abruptly due to the presence of walls and objects. The signal can penetrate differently through different materials and interact with objects as it moves along different paths through the environment. Furthermore, fingerprinting schemes require regular review to ensure the accuracy of the map. If environmental changes (e.g., movements of large objects) occur, fingerprinting methods' performances can degrade drastically. More specifically, the positions of furniture elements, the density of people, and even the positions of walls and partitions can influence the measurements. In [59], the authors analyze the problem of multi-paths and possible mitigation schemes when the beacon signals are transmitted using different frequencies (2402 MHz, 2426 MHz, and 2480 MHz). They also verify that positioning accuracy increases with the number of beacons measured in each fingerprint point, up to a threshold of around 6-8, beyond which there is no quantifiable improvement.

More advanced solutions are also proposed in the literature; in [45], the authors introduce a radio-based positioning algorithm for localization systems suitable for low-cost and resource-bounded applications. The proposed approach uses a graph-based representation of the environment to localize the user by analyzing the received signal level of a mobile device with respect to real and virtual landmarks. The algorithm is compared with commercial localization systems, and it is shown to be effective and accurate. However, it requires to compute and setup virtual landmarks and the movement of the users is not taken into consideration.

Table 6.1 Summary table including relevant works on indoor localization based on Bluetooth low-energy (BLE) received signal strength indication (RSSI) and the corresponding feature extraction methods

Authors	**Feature Extraction**
Varshney et al. [199]	None / Not Mentioned
Chintalapudi et al. [39]	None / Not Mentioned
Čabarkapa et al. [27]	None / Not Mentioned
Er Rida et al. [171]	Sample Average
Kriz et al. [110]	Sample Average
Cho et al. [40]	None / Not Mentioned
Subhan et al. [189]	Sample Average
Palumbo et al. [157]	Sample Average (removing outliers)
Rattanalert et al. [168]	None / Not Mentioned
Grzechca et al. [79]	Sample Average, Moving Average, Sample Median, etc.
Ibrahim et al. [110]	None / Not Mentioned
Park et al. [159]	Sample Average
Luo et al. [134]	None / Not Mentioned
Daniş et al. [43]	Entire Histogram
Dierna et al. [133]	Sample Average
Peng et al. [161]	Sample Average (removing outliers)
Larsson [116]	Sample Average
Mazan et al. [141]	Sample Maximum
Rodríguez-Damián et al. [175]	None / Not Mentioned
Faragher et al. [60]	None / Not Mentioned
Faragher et al. [59]	None / Not Mentioned
Bailey et al. [17]	None / Not Mentioned

As mentioned in Section 6.1, while all these factors are relevant even in the case of a domestic or public space, their impact is expected to increase in an industrial scenario due to the larger size of environments and the abundance of reflecting materials such as metal. However, none of the aforementioned papers compared different BLE-based localization algorithms in the context of a Smart Industry application, which is exactly the main contribution of this work.

6.3 Localization Algorithms

In this work, we considered one distance-based algorithm (Trilateration) and three fingerprint-based algorithms (k-NN, SVM, MLP). These algorithms have been selected as those that provide the most consistent results across different types of environments in literature (see Section 6.2). All considered fingerprint-based algorithms could be used for either classification or regression. The former solution could be used when the goal is simply to identify the area (e.g., room) where the target object is located, whereas the latter approach is needed to obtain precise coordinates. Consequently, we use all three fingerprint-based algorithms in regression mode. In the following, we describe each algorithm in detail, focusing in particular on the hyper-parameters settings used in our experiments.

The trilateration algorithm (see Figure 6.1) is a sophisticated version of triangulation that calculates the distance from the target object directly rather than indirectly through the angles. Data from a single beacon provide a general location of a point within a circle area. Adding data from a second beacon allows for reducing the plausible area to the overlap region of two circles, while adding data from a third beacon reduces it to a point. In the figure, U is the known distance between beacon 1 (B_1) and beacon 2 (B_2), while V_x and V_y are the coordinates of beacon 3 (B_3) with respect to B_1. The radii of the three circles obtained from RSSI measurements are r_1, r_2, and r_3. With this notation, the coordinates of the object P in a 2D space are calculated using the following equations:

$$
\begin{aligned}
V^2 &= V_x^2 + V_y^2, \\
x &= \frac{r_1^2 - r_2^2 + U^2}{2U}, \\
y &= \frac{r_1^2 - r_3^2 + V^2 - 2V_x x}{2V_y}.
\end{aligned}
$$

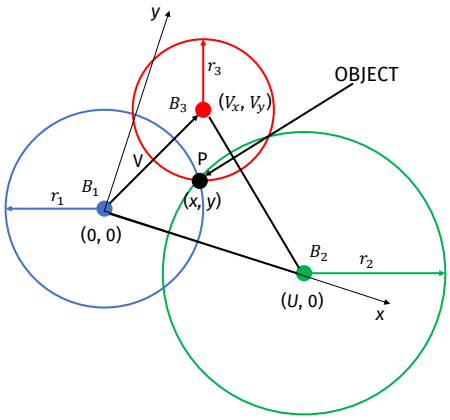


Figure 6.1 Trilateration algorithm for object localization using three beacons B_1, B_2, and B_3

The k-Nearest Neighbors algorithm (k-NN) is a non-parametric supervised learning method that can be used for both regression and classification [17]. When k-NN is used for regression, the output of the algorithm is the property value of the object (i.e., its position) calculated as a weighted average of the values of its k nearest neighbors, where k is the main hyper-parameter of the algorithm.

K-NN does not require proper training, as it just requires storing the RSSI values and corresponding coordinates for the training data. At inference time, the k nearest neighbors of a new sample are simply identified by evaluating a distance metric between its RSSI measurements and those of all training data.

In our work, we tested different values of k and selected the one that yielded the lowest median error on the test data set. We obtained that the optimal value of k did not depend on the density of beacons present in the environment. However, our tests only included either three or four beacons in two different environments. Hence the optimal value may change in a different scenario. Nonetheless, finding the best value of k is a straightforward process if the number of test points is not extremely large since this is the only hyper-parameter of the algorithm. The value of k used in our experiments is reported in Section 6.4.

The Multi-Layer Perceptron (MLP) is a type of feed-forward artificial neural network used in supervised regression and classification tasks [83]. As shown in Figure 6.2, an MLP consists typically of three or more layers of neurons: i) an input layer, ii) an output layer, and one or iii) more hidden layers. MLPs are trained using backpropagation and exploit nonlinear activation functions (Rectified Linear

Unit - ReLU or sigmoid functions) in the hidden and output layers to learn complex input/output relationships.

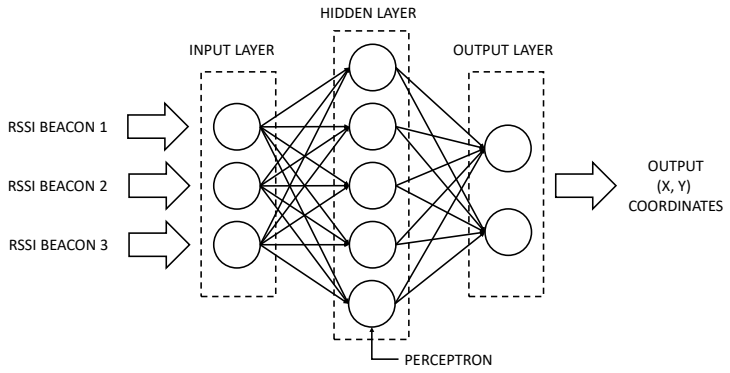


Figure 6.2 Multi-Layer Perceptron (MLP) schematic model

In our work, the number of neurons in the input layer is equal to the number of beacons whose RSSI measurements are recorded (which varies among different environments and experiments), while the number of output neurons is fixed to 2, i.e., the predicted (x,y) coordinates of the target object. Concerning the hidden layers, we tried different combinations of the number of layers and neurons. The exact type and number of layers used in our experiments are reported in Section 6.4.

Lastly, Support-Vector Machines (SVMs) can also be used for supervised classification and regression [83].

When used for regression, SVMs are more properly called Support-Vector Regressors (SVRs) and are trained to predict the output variables with a given margin of tolerance, called ε. In its basic form, the SVR training algorithm finds a hyperplane that maximizes the number of samples that fall within the margin (see Figure 6.3). In practice, this is a transformation of the type:

$$\mathbf{p} = w\mathbf{r} + b,$$

Where $\mathbf{r}$ is the vector of RSSI values measured from the beacons, $\mathbf{p}$ is the corresponding vector of coordinates (x,y) of the object, and w and b are a matrix and a vector of weights, which are optimized to maximize the number of training samples that fall within the margin.

Nonlinear relationships are dealt with by projecting the input variables using a kernel function. For instance, the common Radial Basis Function (RBF) kernel transforms the input $\mathbf{r}$ onto:

$$K(\mathbf{r}-\mathbf{r}') = \exp(-\gamma\|\mathbf{r}-\mathbf{r}'\|^2),$$

where γ is a hyper-parameter used to optimize the hyperplane and $\|\mathbf{r}-\mathbf{r}'\|^2$ is the squared Euclidean distance between two input samples ($\mathbf{r}$ and $\mathbf{r}'$).

For a sufficiently complex problem, some data will inevitably fall outside of the tolerance margin (see Figure 6.3). SVR training tries to minimize the total distance of these points from the margin (ξ), including a penalization term $C\sum\xi$ to the objective function. The hyper-parameter C is used to weigh the importance of these outliers.

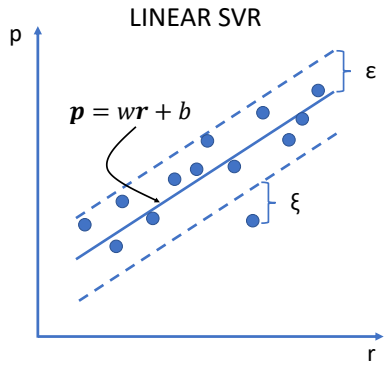


Figure 6.3 Support-Vector Regressor (SVR) example for a two-dimensional problem, using a linear kernel

In our work, we used a grid search to find the best combination of hyper-parameters for an SVR. Specifically, we varied the kernel function, the kernel coefficient γ (only when using an RBF kernel), the boundary limit ε, and the penalty parameter C. We considered linear and RBF kernels, whereas γ was varied in the interval $[0.01-0.08]$, ε in $[0.02-0.08]$, and C in $[1-100]$. The exact kernel functions and the optimal values of the parameters found in our experiments are reported in Section 6.4.

6.4 Experimental Setup

In our experimental setup, we used up to four beacons as BLE transmitters. As mentioned in Section 6.1, we wanted to focus on off-the-shelf BLE-based devices that could be bought on the market today at a low cost to implement indoor localization in industrial environments. Specifically, we selected Estimote beacons [56] because of their low cost and long battery lifetime. These devices adopt the Apple proprietary "iBeacon" protocol [154] to broadcast information packets over BLE. Each packet includes a universally unique identifier (UUID) of the beacon and an indication of the nominal transmitted power at $1m$ from the device, which is needed to convert RSSI measurements into distances.

As a target device to be localized, we used a Raspberry Pi 3 B (quad-core ARM A-53@1.2 GHz) with its onboard Bluetooth receiver. All localization algorithms have been implemented in Python using the Scikit-learn package. In order to improve the correlation between the measurements collected when building the RFM and those exploited to localize the device, the Raspberry Pi has been used for both tasks.

The localization algorithms described in Section 6.3 have been compared in two industrial environments:

- *Environment A* is an industrial shed of $100m^2$, with a $9m$ high ceiling. The walls are partly masonry and partly metal, with a doorway of $4m^2$ on a side. The area is partially occupied by pallets and building materials of different types.
- *Environment B* is an industrial workshop of $288m^2$ with a $10m$ high ceiling. The area contained electronic and mechanical equipment, computers, and metal structures and was continuously in use by workers during our experiments. The walls are partly masonry and partly glass.

Beacons have been positioned as equidistant as possible from each other, given the restrictions caused by obstacles in the two environments. When using four beacons, the minimum distance between them was $\approx 3.9m$ and $\approx 7.7m$ for Environment A and Environment B, respectively.

Notice that a simple localization method based on proximity (i.e., one that simply detects the area of the environment based on the beacon that generates the strongest RSSI) would incur an error that is always greater than half of these distances (i.e., $> 1.85m$ and $> 3.85m$ respectively for the two Environments). Therefore, these half-distances represent a sort of lower bound on the acceptable performance of any localization algorithm that exploits more than one beacon, such as trilateration and fingerprint-based ones.

Fingerprint methods have been tested with different densities of points in the RFM. Specifically, for both environments, we considered three grids of equidistant points with a distance between adjacent points of $0.5m$, $1m$, and $2m$, respectively. RFM points were collected whenever the corresponding location was not obstructed by furniture or other obstacles. This gave us a total of 192 RSSI values for training and 31 for testing in Environment A, while, in Environment B, we measured 293 RSSI points for training and 50 for testing.

In accordance with many literature papers on indoor localization using BLE RSSI [171, 110, 189, 157, 79, 159, 133, 161, 116], we used a simple yet effective "feature extraction" method. Specifically, rather than using raw RSSI measurements to perform localization, we averaged the measurements over an interval of time to filter noise partially. For training measurements, we averaged the RSSI over intervals of $1min$. Testing measurements have been performed for intervals of $20s$ to avoid increasing too much the time required to obtain an estimate of the position.

In both training and testing measurements, the Raspberry Pi was randomly oriented to simulate a realistic localization scenario. The ground truth positions of training and testing points have been measured using a professional meter.

For the k-NN algorithm, we tested different values of k, and we obtained the best performance using $k = 3$. For the MLP, the lowest error was obtained with a model including three hidden layers composed of 8, 8, and 6 neurons, respectively, and using ReLU activation functions. For the SVR, the lowest median localization error was obtained with the RBF kernel, $\gamma = 0.01$, $\varepsilon = 0.02$, and $C = 1$.

6.5 Trilateration results

The error box plots for the trilateration algorithm, computed on all test points, are reported in Figure 6.4 for both environments.

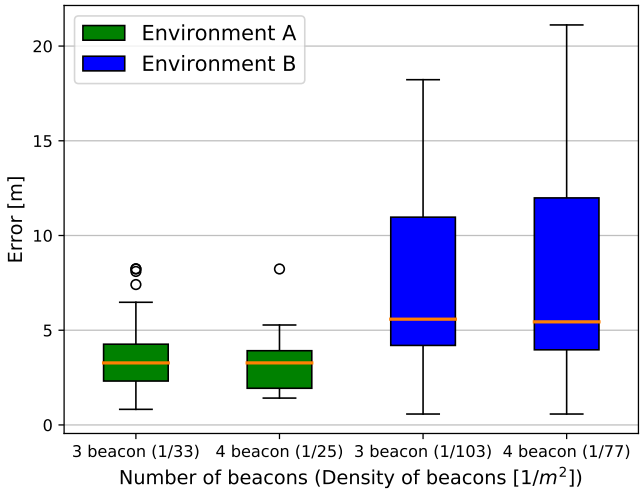


Figure 6.4 Error box plots for the trilateration algorithm. The x-axis reports the number and the corresponding density of the beacons in the environment

As shown in the graph, we performed measurements with either three or four beacons installed in each of the two environments, resulting in different beacons' densities. In the case with four beacons, trilateration was performed using the three strongest RSSI measurements at a given point.

As expected, the localization error is influenced by the density of beacons in the area, but the impact is minimal, and the trends are not straightforward. In Environment A, with four beacons covering an area of $25m^2$ each, we obtain a median error of $3.3m$ and an interquartile range (IQR) of $2.0m$. With only three beacons covering an area of $33m^2$ each, the error median and IQR are practically unchanged, but the two extremes of the IQR are slightly shifted towards larger errors. For Environment B, the error clearly increases due to its wider area. For this environment, 4 beacons cover an area of $77m^2$ each, obtaining a median error of $5.4m$ and an IQR of $8.0m$. As for Environment A, when using only three beacons (each covering an area of $103m^2$), the error median is practically identical, and surprisingly the IQR decreases to $6.7m$. However, the difference is not so significant, so also in this case, the two results can be considered substantially equivalent.

Such large errors are clearly unacceptable for most indoor localization tasks and clearly show the limits of simple RSSI-based distance algorithms. In fact, the median errors obtained with trilateration are significantly larger than the minimum half-distance between beacons (reported in Section 6.4), showing that this method does not reduce the error compared to a simple proximity-based localization. This is probably due to the fact that RSSI measurement precision is strongly affected by external noise sources, shadowing effects, and multi-paths, which are particularly relevant in industrial environments. The fact that the RSSI is measured with a random device orientation also worsens the results.

Moreover, the results in Figure 6.4 show that increasing the density of beacons does not yield significant improvements to the localization accuracy. Of course, if these experiments were to be extended to a much higher number of beacons (e.g., > 10), trilateration could indeed perform better. However, having such a high density of BLE devices would be unlikely in an industrial context, which is characterized by large environments and where the positioning of beacons is not always easy due to the obstructions caused by industrial machinery, etc. Moreover, the experiments of Section 6.6 show that fingerprint methods can significantly reduce the median error without resorting to a larger number of beacons, hence providing a lower-cost solution.

For all these reasons, we conclude that the trilateration algorithm is not suitable for this type of environment and that a fingerprint method is indeed necessary.

6.6 Fingerprint Methods results

Table 6.2 reports the median error and the corresponding IQR for the three fingerprinting algorithms considered in this work. Errors are reported for both environments and refer to the case where 4 beacons are installed and the RFM measurements are collected every $0.5m$.

Table 6.2 Median error and interquartile range (IQR) of fingerprint-based methods in Environment A and Environment B using four beacons and a fingerprint grid with one measurement every $0.5m$

Method	Environment A	Environment B
SVM	median = 1.40 m IQR = 0.82 m	median = 3.52 m IQR = 3.30 m
MLP	median = 1.82 IQR = 1.02 m	median = 3.66 m IQR = 2.85 m
k-NN	median = 1.33 IQR = 1.21 m	median = 3.18 IQR = 3.37 m

These results clearly show that fingerprinting helps in improving localization accuracy significantly, both in terms of median error and in terms of localization stability (i.e., IQR). SVM and k-NN show similar performance, with k-NN achieving the lowest median error in both environments. The MLP obtains the worst performance among the three tested algorithms. This is due to the low amount of training data available, which is typical for this application, where the amount of data is limited by the number of sites RSSI characterization recordings.

In general, the error increases for Environment B due to the lower density of beacons and the presence of metal objects causing disturbances in the signals.

6.6.1 Error Maps

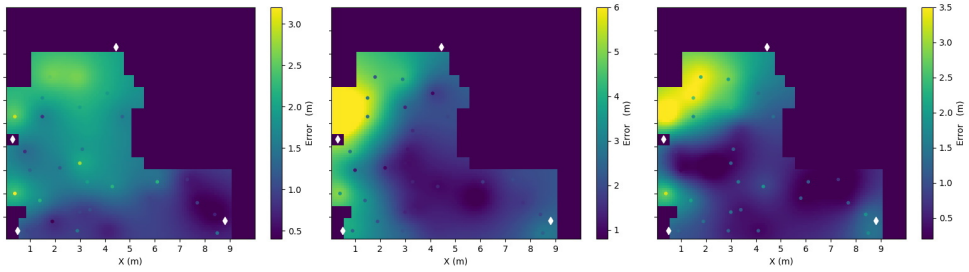
Figure 6.5 and Figure 6.6 show the map of the localization error of fingerprint algorithms for both environments. The positions of the four beacons are represented by white diamonds in the figures. The plots refer to the same experiment described in the previous Section (four beacons, one RFM measurement every $0.5m$).

Interestingly, Figure 6.5 shows that the area with the largest error ($> 3m$) for Environment A is towards the left side of the room, where the wall is made of metal. Similarly, Figure 6.6 shows that SVM and MLP obtain a larger error ($> 10m$) in the bottom-right area, where there is a doorway made of metal and glass. k-NN seems to be less affected by the disturbances caused by these materials. In general, the map highlights that k-NN obtains the most stable estimate across the area of the two environments. Table 6.3 reports the percentage of testing points whose error is

lower than the minimum half-distance between beacons (reported in Section 6.4). As shown, for both environments, more than half of the testing points achieve a localization error that is smaller than the *best result* achievable with proximity-based localization, thus proving the effectiveness of fingerprint-based methods.

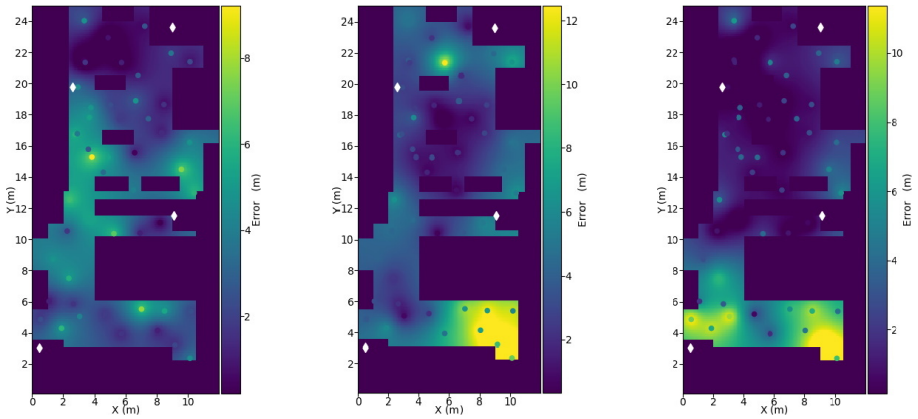
Table 6.3 Percentage of testing points lower than minimum half-distance between beacons

Method	**Environment A [%]**	**Environment B [%]**
k-NN	≈ 67	≈ 56
MLP	≈ 57	≈ 56
SVM	≈ 77	≈ 56



(a) k-NN (b) MLP (c) SVM

Figure 6.5 Error map of the three fingerprint-based methods in Environment A



(a) k-NN (b) MLP (c) SVM

Figure 6.6 Error map of the three fingerprint-based methods in Environment B

6.6.2 Impact of Beacons' Density

Similarly to trilateration, we evaluated the impact of beacon density on the localization error in fingerprinting algorithms. As before, we tried decreasing the number of beacons in each environment from 4 to 3, leaving the density of the RFM grid unchanged (one measurement every $0.5m$). The results of this experiment are reported in Figure 6.7.

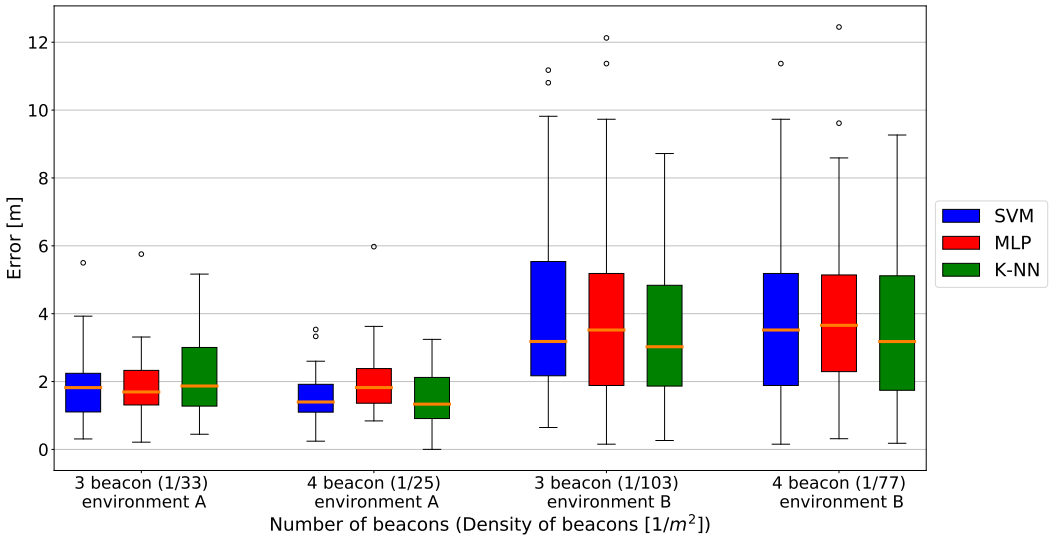


Figure 6.7 Error box plots for the three fingerprinting algorithms with different beacon densities. Results for a fingerprint grid with one measurement every $0.5m$

As shown, the error trends for SVM and k-NN are similar, and it is difficult to identify a clear winner. With either three or four beacons, SVM is slightly better than k-NN for Environment A and vice versa for Environment B. For all three algorithms, the impact of having a larger number of beacons on the results is minimal. In general, the boxes identified by the IQR, as well as the "whiskers" of the box plots, tend to shift towards lower errors when the number of beacons increases from three to four. However, the medians are often very similar and sometimes even increase. For instance, the median error of the SVM reduces by $0.4m$ when going from three to four beacons in Environment A but increases by $0.3m$ in Environment B.

6.6.3 Impact of Fingerprint Grid Density

In fingerprint-based methods, the density of the grid map influences the performance of localization. To this purpose, we keep the number of beacons to four, and we increase the number of RSSI measurements. In this section, we analyze the impact of this design parameter considering three fingerprint grids' densities (one measurement every $0.5m$, $1m$, and $2m$). Figure 6.8 and Figure 6.9 show the results of this analysis for the two environments using four beacons.

Reducing the number of fingerprint points tends to cause a slight increase in the median error in the values of the first and third quartiles and the length of the box plot whiskers. However, this is not always true, and in general, the impact is not dramatic, showing that even a quite "sparse" RFM is sufficient to obtain a decent localization accuracy. Once again, it is not easy to identify a clear winner among the three fingerprint algorithms. For example, in the smallest environment, SVM shows the most stable results with a dense grid ($IQR = 0.8m$), but when the distance between the points increases, it is outperformed by k-NN (e.g., $IQR = 1.1m$ versus the $1.9m$ of the SVM with one fingerprint point every $2m$).

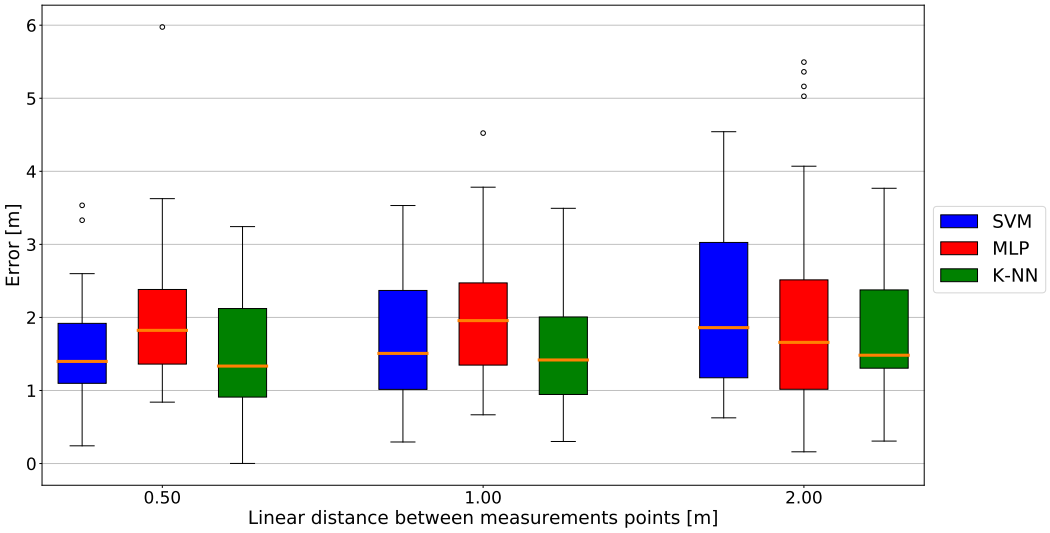


Figure 6.8 Error box plots for the three fingerprinting algorithms with different fingerprint grid densities in Environment A

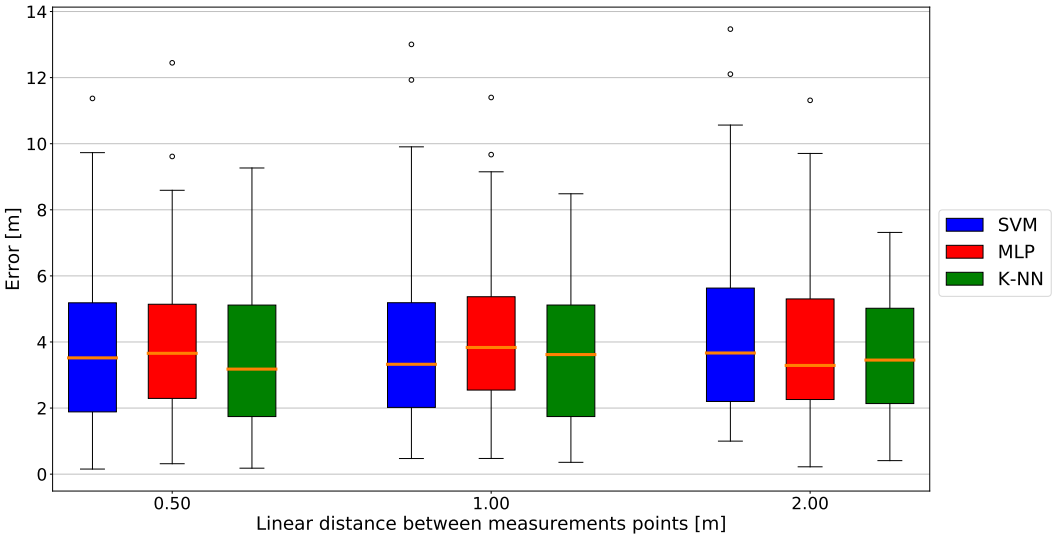


Figure 6.9 Error box plots for the three fingerprinting algorithms with different fingerprint grid densities in Environment B

6.6.4 Impact of Environmental Perturbations

In an industrial environment, movements of large objects and machines may occur. Thus, a good localization algorithm should be resilient to environmental changes. This is clearly relevant, especially for fingerprint-based algorithms, since when something changes within the environment, the fingerprint data collected previously become obsolete and could lead to incorrect positioning. Therefore, we test the effect of changes in the environment on the three fingerprint algorithms, first by simulation and then in the field.

The simulation is performed by adding a zero mean Gaussian noise to the fingerprint RSSI values to mimic the scenario in which a variation in the environment makes them obsolete. Figure 6.10 and Figure 6.11 report the trends of the error box plots of the three algorithms as a function of the Gaussian standard deviation σ.

The three algorithms show a similar trend when Gaussian noise is introduced, and once again, it is hard to state which algorithm performs better since the differences in the box plots for a given value of σ do not justify the choice of one algorithm over the other.

As a field experiment to test the influence of environmental perturbations, we introduce a large item (a $4m \times 4m$ vehicle emulating industrial operations) in Envi-

ronment A, then perform the localization without updating the RFM. The results obtained with the three algorithms in this scenario are reported in Table 6.4.

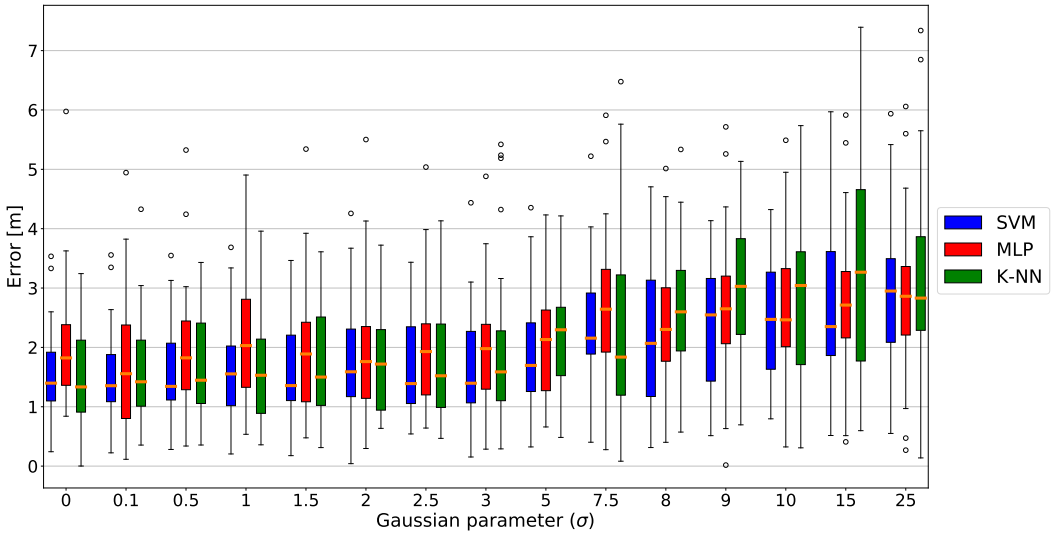


Figure 6.10 Error box plots for the three fingerprinting algorithms as a function of the standard deviation of the Gaussian noise in Environment A

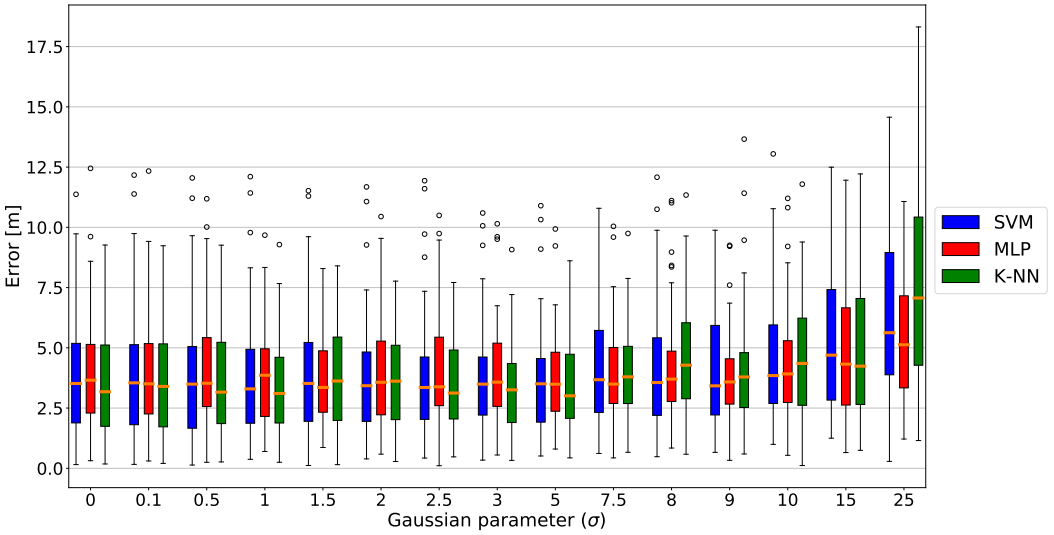


Figure 6.11 Error box plots for the three fingerprinting algorithms as a function of the standard deviation of the Gaussian noise in Environment B

Table 6.4 Error median and IQR for the three fingerprinting-based algorithms in Environment A after adding a large object not present during the collection of the RFM

Method	Environment A
SVM	median = 2.16 m IRQ = 1.02 m
MLP	median = 2.64 m IRQ = 1.39 m
k-NN	median = 1.83 m IRQ = 2.02 m

As expected, the median error in the test points increases for all three algorithms compared to the results of Table 6.2. Still, k-NN is slightly less affected by the perturbation than SVM and MLP since its median error only increases by $0.5m$.

6.7 Discussion and Remarks

In this Chapter, we have presented a study that compares state-of-the-art indoor localization methods based on BLE in an Industry 4.0 setting. Our results show that fingerprinting-based methods are sharply better than distance-based ones. This is motivated by the fact that the latter are affected by shadowing effects and reflections, which are particularly relevant in industrial environments due to the presence of large moving metal objects.

Vice versa, the three considered fingerprint algorithms achieve very similar localization errors and show similar resilience to environmental variations and noise. Therefore, we argue that the selection of the best algorithm should be based on aspects other than sheer localization accuracy.

Even computational complexity considerations do not help the selection, as all three algorithms have a negligible time and memory complexity for the selected target device (the Raspberry Pi). Specifically, given the relatively small size of the training data (and consequently of the prediction models), we obtain that the inference execution time is $< 2ms$ for all three algorithms, and the memory occupation is $< 40MB$.

Therefore, the final selection of an algorithm should be made based on training complexity and ease of adaptability, which directs the choice towards k-NN. In fact, this algorithm has a much simpler set of hyper-parameters (consisting solely of the number of neighbors k) compared to both SVM and MLP. This eases the deployment of a k-NN-based localization system in a new environment, as the only tests that need to be performed are those to identify the optimal value of k. Both MLP and SVM require a much more time-consuming parameter tuning for setting the number of layers, the number of neurons in each layer, and the activation functions in the former case, or the basic function and values of γ, C, and ε in the latter.

Moreover, k-NN does not require an actual training phase, which makes it more flexible for online adaptation. Specifically, whenever new or updated fingerprinting measurements become available, it is sufficient to update the database of reference points used by k-NN in order to obtain an updated location estimate. In contrast, such kind of update would require re-training in the case of SVM and MLP.

Therefore, we conclude that k-NN is the most suitable algorithm for the localization of machines and objects in the Industry 4.0 scenario.

Chapter 7

Solar radiation forecasting for energy efficiency in Industry 4.0

7.1 Introduction

To contrast the negative effects of climate change, renewable energy is becoming a very attractive solution to produce clean and green energy. Renewable Energy Sources (RES) are increasingly cost-competitive compared to conventional fossil fuel-fired power plants. By the end of 2018, electricity generated from new wind power plants and Photovoltaic (PV) systems had become more economical than power from fossil fuel-fired plants in many places [169]. However, the widespread propagation of RES, such as PV systems, is changing electrical energy management in cities and industries, in particular with the widespread of smart cities and Industry 4.0. This contributes to the development of the so-called smart grids, in which strict rules deal with electrical energy production, consumption, and distribution. In this context, the availability of RES needs to be predicted at various time horizons to keep the power grid in balance, and an important role is played by electrical energy operators [204].

For example, for efficient conversion and utilization of solar energy, the Global Horizontal Solar Irradiance (GHI) should be estimated and forecasted through accurate methods and tools. In this context, new control strategies can be developed to maximize the consumption of energy when this is produced by RES. Demand-Response (DR) and Demand Side Management (DSM) are examples of applications

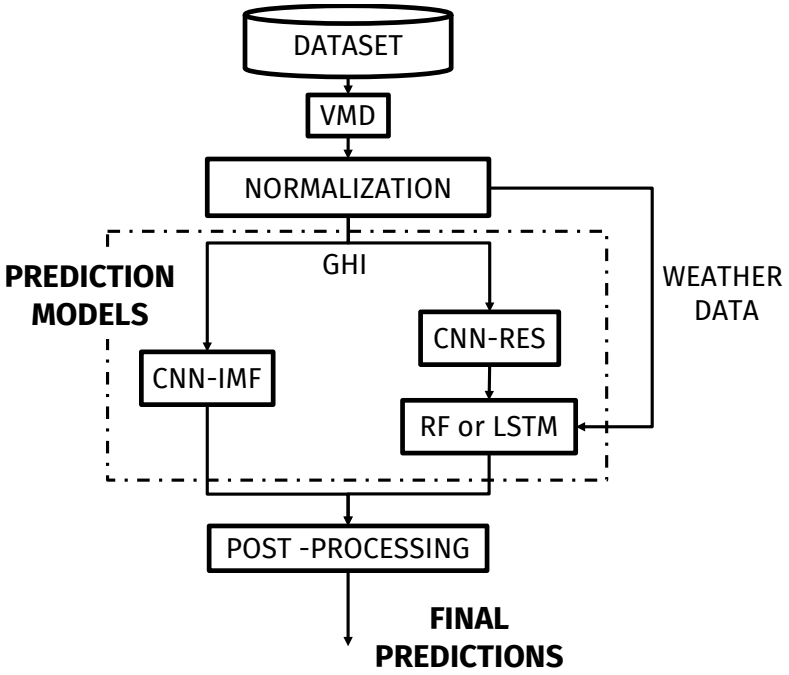


Figure 7.1 Schema of the proposed methodology

to achieve this purpose [46, 128, 183] that can take advantage of such prediction models.

In recent years, many models have been developed to increase the performance to forecast GHI. In [147], authors developed a model that uses signal decomposition and combines the Autoregressive process and Neural Network to improve short-term GHI forecasts. In [3], a two-stage day-ahead solar forecasting method is developed. It implements Nonlinear Autoregressive Neural Network and Autoregressive Moving Average with Exogenous Input to break down the prediction into linear and nonlinear components and improve the accuracy of the final results. Currently, the research is focused on building models to predict only short-term or long-term forecasts separately.

Given the promising results using signal decomposition and hybrid models, we proposed an innovative methodology for both short- and long-term GHI predictions (from $15min$ to $24h$) using a dataset with $15min$ or $1h$ time-resolution. Figure 7.1 presents the schema of our methodology for solar radiation forecasts using Variational Mode Decomposition (VMD) and a combination of Convolutional Neural Networks (CNN) and two regression models like Random Forest (RF) or Long Short-Term Memory (LSTM). Typically CNNs have applications in image and video recognition,

but in this work, we use them for time-series forecasting in combination with the other models. For the decomposition, we applied VMD, a recently developed algorithm that turns out to be more promising than Ensemble Empirical Mode Decomposition (EEMD) and Wavelet Decomposition (WD) [115]. Indeed, VMD does not require heavy computation and exhibits good robustness compared to WD and EEMD, as analyzed by Dragomiretskiy and Zosso in [54].

The inputs given to our solution are processed through VMD, and the decomposition returns a set of sub-signals (or modes) that have different bandwidths and a residue that is the difference between input data and modes. As shown in Figure 7.1, the set of modes is given as input to a first CNN (i.e., CNN-IMF in the figure), while the residue to a second CNN (i.e., CNN-RES). The output of the CNN-RES is given as input, together with weather data, to the chosen regression model (i.e., RF or LSTM). The results from the regression model and CNN-IMF are post-processed to obtain the final prediction. We built a single set of models to be used to predict all the resolution periods requested, increasing the performance. We choose RF because it runs efficiently on large datasets. It is a classifier that evolves from decision trees and correlates them to form a better model. The output model of RF is the class that is the mean prediction of the individual trees. Moreover, as an alternative to RF, we also tested the LSTM due to its ability to predict time-series data [90].

LSTM belongs to a group of information processing techniques that can be used to find knowledge, patterns, or models from a large amount of data in combination with other computational models or methodologies [102]. In literature, RF and LSTM showed promising results in forecasting solar radiation, in particular when they are involved in hybrid models that combine multiple learning models in a best-of-breed approach [201, 187].

We trained and tested the neural networks with a dataset consisting of six years of GHI and weather samples collected by a real weather station on our University campus. The CNN is a regularized version of multi-layer perceptrons exploiting a high number of regressors fed with GHI data pre-processed with VMD. LSTM and RF are trained using meteorological data provided by the weather station. Then, the results obtained are post-processed to provide the final GHI with the prediction range desired. The GHI predictions obtained by the proposed model can be given as input to tools for forecasting PV production, for example, PVsim [25], and used to manage energy consumption in industry and reduce costs and waste.

Two models are proposed to test different time-resolutions (15*min* and 60*min*, a.k.a. *M15* and *M60*, respectively) and prediction horizons (from 15*min* to 24*h*). The two models differ only on the granularity of both i) sampling time of measurements in the input dataset and ii) time-step of the output forecast. Thus, M15 has a 15*min* time resolution which makes it suitable for sub-hourly predictions in the short term, while M60 has a 60*min* time resolution suitable for hourly forecasts in the long term.

The remaining part of the Chapter is organized as follows: Section 7.2 reviews literature solution on solar radiation forecast; Section 7.3 introduces the theoretical background behind our solution; Section 7.4 provides the performance indicators to evaluate the accuracy of GHI predictions. Section 7.5 describes the adopted structure and details all the steps performed to initialize, train and test the final model. Section 7.6 discusses the results obtained on solar radiation forecast and accuracy on GHI exploiting different structures. Finally, Section 7.7 discusses the concluding remarks. Some of the work described in this Chapter was also previously published in [29].

7.2 Related works

For decades, solar radiation forecasting methods have been studied in the literature using a variety of techniques. Hence, several solutions in literature analyze mathematical and physical models to forecast solar radiation, such as stochastic models based on time-series [103, 201, 15]. However, these methodologies often are not sufficient in predicting GHI values due to time-series data that are non-stationary and non-linear [136, 153].

Several supervised machine learning techniques [122], Support Vector Regression methods [100], and neural networks [165] have been introduced to forecast GHI using historical weather data as input information. In [123], authors exploit a deterministic model and a bootstrap multi-layer perceptron to forecast in a very short-term (i.e., 5, 10, and 15*min*) the Diffuse Horizontal Irradiance component (DHI). Generally, DHI is less investigated in modeling solar energy systems because DHI is part of GHI. Although [123] presents very interesting results as the authors investigate how both cloud transmittance and cloud velocity impact forecasting. However, this methodology is very limited because it allows only short-term prediction (from 5 to 15*min*). In addition, in order to have a high forecast fidelity, the

authors themselves recommended incorporating real-time sky and cloud data into the forecasting models and calibrating the cloud velocity derivations.

In [202], Voyant et al. reviewed several machine learning methods based on i) Artificial Neural Networks (ANN); ii) Autoregressive Integrated Moving Average networks (ARIMA); iii) different predictor ensemble methodologies; iv) Support Vector Machine; v) regression trees and vi) Random Forests (RF). In general, the authors state that in terms of the quality of prediction in variable conditions, ANN and ARIMA are equivalent. However, ANNs are very efficient in solving nonlinear problems [225], while ARIMA assumes that the series are generated from linear processes, thus not suitable for most real-world problems that are non-linear [105]. Moreover, the predictor ensemble methods are more promising than simple predictors. Finally, Support Vector Machines, regression trees, and RF present very promising potentials. Currently, numerous studies have been developed using RF to forecast GHI [20].

Other studies try to reduce the complexity of forecasting problems and analyze the correlation between the input data. In [70], García-Hinde et al. try to improve the prediction accuracy of solar irradiance, obtaining a robust set of features by performing dimensionality reduction on the input variables. Real data from the weather station and forecasting model, which produce a very large number of variables, are used as input.

In [173], authors suggested dividing the complex problem of seasonal forecasting (like solar irradiance) into sub-problems that are easier to deal with by forecasting each undetectable component separately. They propose a methodology based on a Neuro-Fuzzy Takagi-Sugeno (NF-TS). The NF-TS considers the undetectable components extracted from the time-series data to evolve because they present dynamic behaviors that are simpler to forecast.

In [69], instead of providing a single measure for the prediction, the lower and upper bound of solar radiation are used (prediction interval). The algorithm uses neural networks to create complex nonlinear models whose outputs are the upper and lower limits of the prediction intervals.

Numerous machine learning algorithms are tested to increase the performance in forecasting solar irradiation. In [34], the authors established a Diagonal Recurrent Wavelet Neural Network so as to carry out forecasting of the hourly global solar

irradiance. The authors themselves recommended studying further the application of more weather information to forecast solar irradiation.

Other algorithms are tested in literature, like Support Vector Regression and regression trees, that show advantages in GHI forecasting. Still, they are not suitable for dealing with a large volume of incoming data. Unlike them, CNN allows to have fewer weights and shared parameters. CNN is particularly suitable to deal with two-dimensional arrays and uncertainties due to weather in time-series data like wind [208] and PV power [207]. Moreover, using time-series decomposition, such as VMD, to pre-process data, it is possible to improve the accuracy of predictions made by CNNs [191, 138]. In [224], the authors developed a set of hybrid models that are a combination of different neural networks, and they use them to forecast PV power production with different time resolutions. In [6], authors developed a Nonlinear Autoregressive (NAR) neural network for short-term forecast of GHI and Photovoltaic energy prediction.

In literature, the models proposed focus only on predicting a short range of prediction horizons. Moreover, such models develop only short- or long-term predictions. Our methodology overcomes these problems. With respect to literature solutions, we propose a methodology to forecast GHI in both short- and long-term time horizons with a single set of models.

It takes advantage of a recently developed decomposition algorithm, VMD, that performs better than previous techniques like Ensemble Empirical Mode Decomposition and Wavelet Decomposition [115] in both computational performances and robustness in decomposition [54]. The novelty of our approach is that there are no other examples of models in literature developed to predict GHI data for both short- and long-term using a combination of multiple models and weather information.

At its core, our methodology consists of a combination of two Convolutional Neural Networks and two regression models (i.e., Random Forest and Long Short-Term Memory). To improve the performance of the forecasts, our methodology distinguishes the input data between the four meteorological seasons. Both training and test have been performed by exploiting a dataset consisting of six years of real weather information sampled every $15min$ by a real weather station in our University campus.

The meteorological measures we consider are: i) GHI, ii) air temperature, iii) relative humidity, iv) air pressure, v) wind speed, and vi) cloud coverage. While solutions like [225, 105, 20, 6] only exploit GHI time-series.

Finally, with this methodology, we proposed two models to test two different time resolutions, 15min and 60min, respectively, with a prediction horizon ranging from next $15min$ up to $24h$. Thus, our methodology allows both sub-hourly and hourly predictions in the short- and long-term. It is worth noting that these two models differ only on the time resolution of measurements in the input dataset that affects the time-step in the output forecast (i.e., predictions every $15min$ or $60min$, respectively).

7.3 Theoretical Background

In this section, we analyze in detail the main algorithms used to create the model proposed in Section 7.5. We used the VMD as decomposition algorithm and a combination of i) CNNs and RF, ii) CNNs and LSTM to finally forecast the GHI.

7.3.1 Variational Mode Decomposition

As proposed by [54], the role of the Variational Mode Decomposition (VMD) is to decompose an input signal into a discrete number of sub-signals that have specific sparsity properties. The use of VMD allows one to cope with noise properly, determine the relevant bands adaptively, and estimate the corresponding modes concurrently while balancing errors. VMD leads to a decomposition of a signal f into its K principal modes, u_k, called Intrinsic Mode Function (IMF) based on modulation criteria. Each signal is expressed as:

$$u_k(t) = A_k(t)cos[\phi_k(t)]$$

where A_k is a non-negative envelope and ϕ_k is a non-decreasing function with $\omega_k(t) := \phi(t)'_k$ the center pulsation of the mode. In a nutshell, the mode can be considered to be a pure harmonic signal with amplitude A_k and instantaneous frequency ϕ_k. To obtain the final modes, it is required to solve the following minimization

problem:

$$\min_{\{u_k\},\{\omega_k\}} \left\{ \sum_{k=1}^{K} \left\| \partial_t \left[\left(\delta(t) + \frac{j}{\pi t} \right) * u_k(t) \right] e^{-j\omega_k t} \right\|_2^2 \right\} \quad s.t. \sum_{k=1}^{K} u_k = f \tag{7.1}$$

where $\{u_k\} := \{u_1, \cdots, u_K\}$ and $\{\omega_k\} := \{\omega_1, \cdots, \omega_K\}$ are shorthand notations for the set of all modes and their center frequencies, δ is the Dirac distribution, π is the mathematical constant and j the imaginary unit. To address the reconstruction constraint, a quadratic penalty term (α) and Lagrangian multipliers (λ) are introduced in order to render the problem unconstrained. Therefore, the augmented Lagrangian $\mathscr{L}$ is set up as follows [156]:

$$\begin{aligned} \mathscr{L}(\{u_k\},\{\omega_k\},\lambda) := \alpha \sum_{k=1}^{K} \left\| \partial_t \left[\left(\delta(t) + \frac{j}{\pi t} \right) * u_k(t) \right] e^{-j\omega_k t} \right\|_2^2 \\ + \left\| f(t) - \sum_{k=1}^{K} u_k(t) \right\|_2^2 + \left\langle \lambda(t), f(t) - \sum_{k=1}^{K} u_k(t) \right\rangle \end{aligned}$$

The solution to the original minimization problem (Equation 7.1) is now found as the saddle point of the augmented Lagrangian $\mathscr{L}$ in a sequence of iterative sub-optimizations called Alternate Direction Method of Multipliers [21]. In order to achieve the final solution, the iterative process is applied to u_k and ω_k, taking u_k as an example. This problem can be solved in the spectral domain:

$$\begin{aligned} \hat{u}_k^{n+1} = arg \min_{\hat{u}_k, u_k \in X} \left\{ \alpha \| j(\omega - \omega_k)[(1 + sgn(\omega))\hat{u}_k(\omega)] \|_2^2 \right. \\ \left. + \left\| \hat{f}(\omega) - \sum_i \hat{u}_i(\omega) + \frac{\hat{\lambda}(\omega)}{2} \right\|_2^2 \right\} \end{aligned}$$

where n is the number of iterations.

Firstly, we select the number of K modes. Typically this step is done by observing the central frequency of the IMFs. We observe the decomposition of the modes with different values of K. When the value selected is too high, duplicated modes with a similar central frequency appear (Over-segmentation); instead, when the value chosen is too low, some components are not correctly decomposed, and they are contained in other modes (Under-segmentation). Then, the VMD algorithm is applied: the signal is extended by mirroring and analyzed through a Fast Fourier transform to

allow the decomposition in the frequency domain. The signal is reconstructed, and the spectrum is analyzed to check the consistency of the decomposition. The decomposition of the signal in K modes plus a residue is obtained as:

$$f(t) = \sum_{k=1}^{K} u_k(t) + res(t)$$

where $u_k(t)$ is the k^{th} IMF and $res(t)$ is the residue.

7.3.2 Convolutional Neural Network

In deep learning, a Convolutional Neural Network (CNN) is a feed-forward deep neural network. CNNs are regularized versions of multi-layer perceptrons inspired by biological processes. Firstly researched by Hubel and Wiesel [93], CNN has applications in image and video recognition, image classification, and natural language processing [111].

A CNN consists of an input layer and an output layer with multiple hidden layers in the between. The hidden layers are typically a set of convolutional layers with a Rectified Linear Unit (ReLU) as the activation function, followed by a flatten layer and fully connected layers (dense layers).

All these layers are stacked sequentially to create the CNN. There are more layers than the previously described, such as dropout, but the usage may not be suitable for all the situations and worse the model results.

During the development of the network, some parameters have to be selected manually, such as the number of filters, filter shape, batch size, iteration epoch, and initial learning rate. These parameters, named hyper-parameters, can be determined through optimization algorithms like grid search, random search, or gradient-based optimization. In our methodology, grid search is applied to obtain the optimal parameters for the task.

7.3.3 Random Forest regressor

Random Forest (RF) regression is a classifier that evolves from decision trees. It combines the predictions from multiple decision tree algorithms together to obtain more accurate results than any individual model.

This technique uses multiple models to obtain better performance and is called model ensembling. In RF, all the base trees are constructed independently using a deterministic algorithm by selecting a random set of input data from the training set. Firstly researched by Ho [87], RF runs efficiently on large datasets and can handle thousands of input variables without variable deletion.

In RF, typically, three parameters need to be optimized:

- *Number of estimators*: They represent the number of trees in the forest.
- *Max depth*: It restricts the depth of each individual tree to prevent over-fitting (if set to none, there is no restriction).
- *Min samples leaf*: It sets the minimum size of the terminal nodes of the trees.

These parameters can be optimized through the grid search cross-validation method to improve the predictions of the model.

7.3.4 Long Short-Term Memory Neural Network

Long Short-Term Memory (LSTM) is an artificial Recurrent Neural Network architecture capable of learning long-term dependencies. LSTM is introduced by Hochreiter and Schmidhuber [90]. It is explicitly designed to avoid the long-term dependency problems that led to the vanishing gradient problem that emerged as a major obstacle to Recurrent Neural Network performance [89].

All Recurrent Neural Networks have the form of a chain of repeating modules of neural networks called cells. Typically, an LSTM is composed of a set of cells (hidden layer) and a fully connected layer (dense layer).

In an LSTM cell, three gates are present:

- *Input gate*: It chooses the input value that should be used to modify the memory.

- *Forget gate*: It chooses the details to be discarded from the cell.
- *Output gate*: The input and the memory of the cell are used to decide the output.

7.4 Performance Indicators

In this study, we adopted the following performance indicators to evaluate the accuracy of our methodology: i) Root Mean Square Error (RMSE), ii) Normalized Root Mean Square Error percentage (NRMSE %), iii) Mean Absolute Error (MAE) and iv) Coefficient of determination (R^2).

RMSE is calculated as:

$$RMSE = \sqrt{\frac{\sum_{n=1}^{N}(\hat{y}_n - y_n)^2}{N}} \quad [W/m^2]$$

where N is the sample size, $\hat{y}_n$ is the n^{th} predicted value and y_n is the n^{th} real value of GHI.

NRMSE is given as follows:

$$NRMSE = \frac{100\%}{\hat{y}_{max} - \hat{y}_{min}} RMSE \quad [W/m^2]$$

where $\hat{y}_{max}$ and $\hat{y}_{min}$ are, respectively, the maximum and minimum values of the predicted samples.

MAE is defined as:

$$MAE = \frac{\sum_{n=1}^{N} |\hat{y}_n - y_n|}{N} \quad [W/m^2]$$

Finally, R^2 is given as:

$$R^2 = \frac{\sum_{n=1}^{N}(\hat{y}_n - \bar{y})^2}{\sum_{n=1}^{N}(y_n - \bar{y})^2}$$

Where $\bar{y}$ is the mean value of real GHI, with low RMSE, NRMSE, and MAE, the model is more accurate. R^2 values range between 0 and 1; where 0 indicates no correlations and 1 a perfect correlation.

7.5 Proposed model

Figure 7.1 reports the schema with the main steps of our methodology to forecast GHI. In its core, it exploits VMD to decompose the GHI into a set of sub-signals (or modes) and a residue (i.e., the difference between input data and modes). After data normalization, the sets of modes and residue are given as input to CNN-IMF and CNN-RES, respectively. The output of the CNN-RES, together with the other meteorological measures in the dataset, feeds one of the chosen regression models (i.e., RF or LSTM). Finally, the outputs of both CNN-IMF and regression model are combined together and post-processed, providing the final GHI forecast.

Two proposed models to forecast GHI, called M15 and M60, are built with the same structure, and they use the same dataset differing only on the time resolution. M15 is suitable for sub-hourly forecasts with a granularity of $15min$, while M60 has a granularity of $1h$. These models are used for short- and long-term forecasts (up to the next $24h$). The rest of this section presents in detail the dataset and the proposed methodology.

7.5.1 Dataset and pre-processing

The dataset used in this study consists of six years of meteorological data collected by a real weather station in our University Campus in Turin, Italy, from December 2009 to November 2015, with a time resolution of $15min$. It is composed of 6 time series, one for each of the following meteorological measures: i) GHI, ii) air temperature, iii) relative humidity, iv) air pressure, v) wind speed and vi) cloud coverage.

This dataset has been split into training- and test-set with 175301 and 35041 samples for each meteorological measure, respectively. In particular, the training set ranges from December 2009 to November 2014, test-set from December 2014 to November 2015. Both training- and test-set report data for the full four seasons.

We fill missing data with the mean value of previous and next samples in the time series, thus allowing to keep the data aligned with the overall behavior. In [119], the authors study the accuracy of filling the missing GHI samples with mean values. The results show that this method provides decreased errors and leads to better solar forecasting performance compared to other imputation techniques. In [44], the authors compare the accuracy of different imputation methods for missing solar

irradiance data with different time resolutions. They identify linear interpolation as the most reliable and robust approach for the imputation of minute solar irradiance values.

To feed both CNNs, after applying VMD, the dataset is normalized between 0 and 1 using the standard normalization obtained with min-max scaler [120], and then it is split according to meteorological seasons. Meteorological seasons are different from astronomical seasons. Indeed for temperate areas in the northern hemisphere, spring begins on March 1^{st}, summer on June 1^{st}, autumn on September 1^{st}, and winter on December 1^{st}. This normalization is needed to have the same range of values for each of the inputs given to the CNNs. Without normalization, the ranges of our GHI signals decomposed would be different from one another. Thus the learning rate would cause corrections in each dimension that would differ from one another.

With the meteorological seasons, the statistically colder, warmer, and intermediate months are those identified by these periods, with the months at extreme thermal averages (usually January and July) that fall in the middle of the respective meteorological season. With this division, extracting seasonal statistics from the monthly ones becomes much easier, and both of them are very useful for forecasting.

As discussed previously, the dataset is composed of weather elements (i.e., air temperature, relative humidity, air pressure, wind speed, and cloud coverage) and GHI. As the input of both CNNs, we use only the GHI values, while, based on the correlation analysis, the weather elements are used to feed the regression model (RF or LSTM) in combination with the output of CNN-RES.

To feed the GHI time series to the CNNs, it is necessary to create a 2D input vector and then decompose it through VMD. VMD is required to decompose in frequency the behavior of the GHI. Then the CNNs can fully analyze the signals and extract the relevant features.

The input data are built to create a 2D array as shown in Figure 7.2, one input array is composed by $7days$ with 96 GHI points corresponding to a time-resolution of $15min$ in $24h$. Choosing $7days$ is a good compromise that allows to maintain the size of the array small and to find repetitive patterns between data and daily correlation.

To generalize, the input vector is composed by $n_{day} \times p_{GHI}$, where n_{day} is the number of the day chosen and p_{GHI} is the number of GHI points that depends by the

chosen time-resolution. To forecast GHI, we use an iterative approach. The proposed model has a single output, so it can only predict one step ahead. For subsequent steps, the predicted value for time $t+1$ is used as one of the inputs for the prediction at time $t+2$, and so on. The input data are shifted of one resolution point, and the process is repeated until the prediction period is reached (i.e., following a sliding window approach).

It is worth saying that to train the CNN with $1h$ time-resolution, the data are reduced by taking the arithmetic mean of the values in $1h$ (i.e., from 0:00 to 0:45 to obtain the GHI value at 0:00) and obtaining 24 samples. Then, the same process used for $15min$ time resolution is applied to build the input vector. To apply VMD, we need to set the number of K modes of the decomposition reasonably. To avoid aliasing problems, a K value not too high is needed. Therefore $K = 9$ is chosen after observing the central frequency of the signal decomposition, as reported in [224].

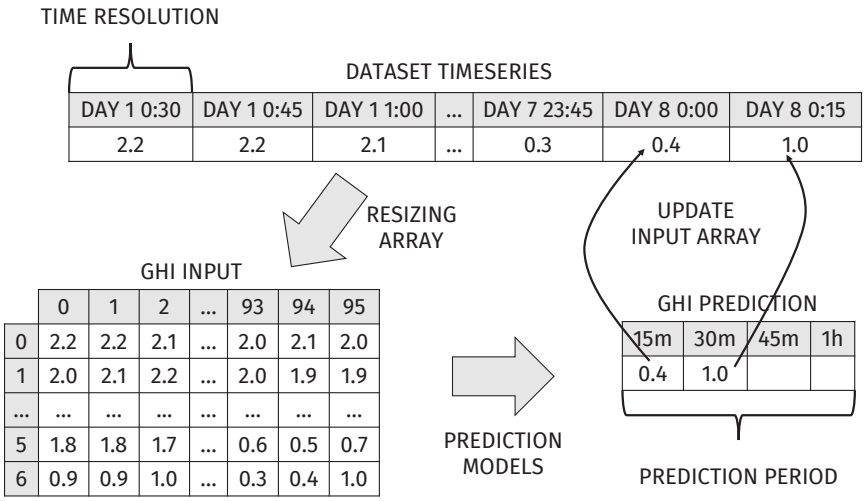


Figure 7.2 Schema to create the input data ($15min$ time-resolution)

Because all the days have similar trends independently from the season, a period of GHI time series is taken as an example, which represents the first day of December 2009 with a time-resolution of $15min$, and its VMD results are shown in Figure 7.3. Each IMF has its own central frequency, the sum of all the modes, and the residue gives the original GHI signal. The frequencies of the modes increase from the first IMF (low frequency) to the last IMF (high frequency). As previously explained, the original GHI time series is decomposed into 9 IMFs and 1 residue.

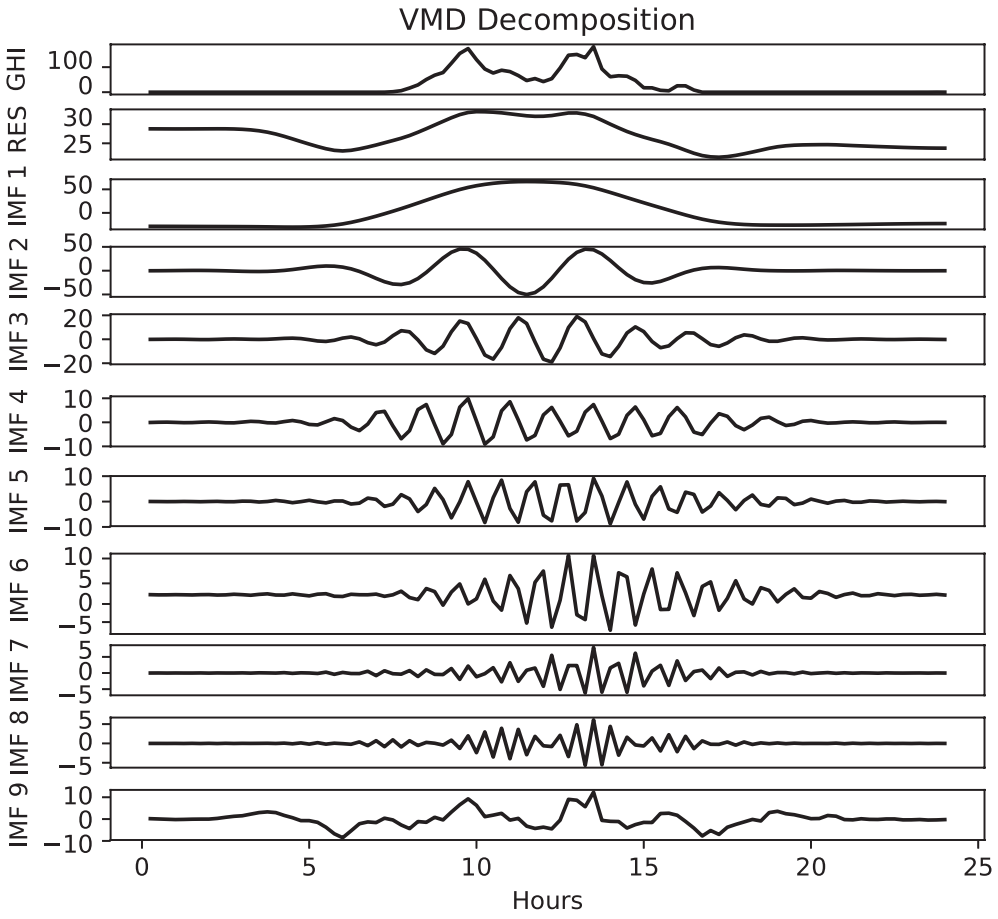


Figure 7.3 VMD results for the first day in December 2009 (15*min* time-resolution)

In order to make the best use of the meteorological data, we analyzed the correlation between the IMFs, the residue, and the weather elements. We used the standard Pearson correlation coefficient calculated using the following equation:

$$r_{xy} = \frac{\sum x_i y_i - n\bar{x}\bar{y}}{\sqrt{(\sum x_i^2 - n\bar{x}^2)}\sqrt{(\sum y_i^2 - n\bar{y}^2)}}$$

where n is the sample size, x_i and y_i are the individual sample points indexed with i and $\bar{x}$ and $\bar{y}$ are the sample mean. The correlation is strong when r_{xy} is close to ± 1; if it is near 0, there is no correlation.

Table 7.1 reports the results of the correlation analysis for one week in December 2009 (reported as an example). We can notice a strong correlation between the residue and the air temperature (≥ 0.9). While there is a moderate correlation

Table 7.1 Correlations between decomposed components and weather data

Component	Temp.	Hum.	Pres.	Wind sp.	Cloud cov.
Residue	**0.949**	**0.314**	**−0.266**	**−0.200**	**0.494**
IMF 1	0.853	0.043	0.082	0.022	0.215
IMF 2	0.008	0.038	0.003	−0.154	0.030
IMF 3	0.026	−0.027	0.005	−0.069	0.006
IMF 4	0.006	−0.010	0.004	−0.031	−0.003
IMF 5	0.007	0.005	0.003	−0.026	0.007
IMF 6	0.007	−0.004	−0.001	−0.008	0.007
IMF 7	0.006	0.011	−0.001	−0.009	0.003
IMF 8	0.001	0.009	0.001	−0.011	0.001
IMF 9	0.343	−0.057	0.030	−0.133	0.241

(≥ 0.2) with the other meteorological measures. Moreover, the residue shows a particular correlation with air temperature (0.949) and cloud coverage (0.494). The IMF signals instead show almost no correlation with the meteorological data (< 0.2). Therefore, we choose to use together the feature extracted from the last layer of the CNN-RES and the meteorological data to feed the regression model (one among RF or LSTM) and compute the second part of the GHI.

7.5.2 CNN-IMF and CNN-RES

CNN possesses excellent abilities in analyzing linear and nonlinear features using a convolutional process. CNNs are a particular type of neural network for processing data with grid-like topology. CNNs are suitable for time series data that can be ordered as a 1D grid or 2D grid-like image data.

As discussed in Section 7.3, the hyper-parameters of both CNNs have to be chosen. One of them is the pooling layer, which reduces the dimensions of the data by combining the outputs of neurons at one layer into a single neuron in the next layer. The pooling layer is widely utilized in image processing, but it is not suitable for regression problems like GHI forecasts.

This is demonstrated by the following analysis. We tested two different types of pooling layers: max-pooling and average-pooling, with pool sizes of 2×2 or 1×2, then the pooling layer is removed for comparison. Those pooling layers are added after the first convolutional layer. In Table 7.2, we reported the comparison between various pooling layers. The results reported are obtained using only the

Table 7.2 Comparisons of CNNs with different pooling layers

Model	Pooling layer	RMSE $[W/m^2]$	MAE $[W/m^2]$
CNN-IMF	max-pooling (2×2)	0.004	0.048
	average-pooling (2×2)	0.005	0.049
	max-pooling (1×2)	0.001	0.021
	average-pooling (1×2)	0.001	0.021
	no-pooling	**0.001**	**0.020**
CNN-RES	max-pooling (2×2)	0.006	0.054
	average-pooling (2×2)	0.006	0.056
	max-pooling (1×2)	0.002	0.019
	average-pooling (1×2)	0.002	0.019
	no-pooling	**0.001**	**0.019**

GHI from the dataset to verify which pooling layer is most suitable to our case. To obtain the final result, we also need the contribution of the RF or LSTM models with the weather elements.

For both CNNs, according to the comparison, the best results are obtained using no pooling layer (RMSE: 0.001 for both CNNs, MAE: 0.020 and 0.019 for CNN-IMF and CNN-RES, respectively). This is probably due to the characteristic of the GHI signal. With pool size (2×2), both the pooling methods resulted in information loss and reduced prediction accuracy (RMSE ≥ 0.005 and MAE ≥ 0.048), whereas average-pooling performed worse than max-pooling. Otherwise, with pool size (1×2), the results are comparable with the no pooling CNN (RMSE: 0.001 and 0.002, MAE: 0.021 and 0.019 for CNN-IMF and CNN-RES, respectively). Thus to achieve high accuracy, no pooling layers are utilized in both CNNs. Table 7.3 and Table 7.4 report the final structure, including all the other possible hyper-parameters for both CNN-IMF and CNN-RES for M15 and M60, respectively.

7.5.3 RF and LSTM

RF and LSTM are two alternative regression models combined with CNN-RES in our methodology. Thus the output of CNN-RES, together with the other meteorological data, is given as input to RF or LSTM in both M15 and M60 models.

The RF model gives more robust performances in different seasons than other models and produces accurate predictions that do not overfit the data. As shown

Table 7.3 Structure of CNN-IMF

Layer	CNN 15min (M15)		CNN 1h (M60)	
	Param	Train	Param	Train
Conv2D 1	Input: $7 \times 96 \times 9$ Filters: 16 Kernel size: 3×3 Padding: valid Activation: ReLU	1312	Input: $7 \times 24 \times 9$ Filters: 16 Kernel size: 3×3 Padding: valid Activation: ReLU	1312
Conv2D 2	Filters: 32 Kernel size: 2×2 Padding: valid Activation: ReLU	2080	Filters: 32 Kernel size: 2×2 Padding: valid Activation: ReLU	2080
Flatten	None	0	None	0
Dense 1	Units: 6 Activation: None	71430	units: 6 Activation: None	16134
Dense 2	Units: 1 Activation: None	7	units: 1 Activation: None	7
Total	None Epochs: 20 Batch size: 32	74829	None Epochs: 20 Batch size: 32	19533

in [190], RF models are indicated to forecast solar radiation estimation, but the performances reduce when a huge number of weather parameters are involved.

Instead, LSTM models exploit the temporal and spatial dependence of data, which represents a major asset in utilizing contextual information. LSTM models have shown the potential to deal with a large number of parameters, such as in the case of solar radiation forecast [187].

A small set of parameters with the most significant weather data is selected to exploit both RF and LSTM models. As discussed in the next Section 7.6, the RF model is more suitable for a dataset with a granularity of $15min$ while the LSTM model works better using a dataset with a granularity of $1h$.

The parameters to choose needed by RF are the number of estimators, max depth, and min samples leaf. While LSTM only needs the number of units. All the chosen parameters and the final structures of RF and LSTM are reported in Table 7.5.

Table 7.4 Structure of CNN-RES

Layer	CNN 15min (M15)		CNN 1h (M60)	
	Param	Train	Param	Train
Conv2D 1	Input: $7 \times 96 \times 1$ Filters: 16 Kernel size: 3×3 Padding: valid Activation: ReLU	160	Input: $7 \times 24 \times 1$ Filters: 16 Kernel size: 3×3 Padding: valid Activation: ReLU	160
Conv2D 2	Filters: 32 Kernel size: 2×2 Padding: valid Activation: ReLU	2080	Filters: 32 Kernel size: 2×2 Padding: valid Activation: ReLU	2080
Flatten	None	0	None	0
Dense 1	units: 6 Activation: None	71430	units: 6 Activation: None	16134
Dense 2	units: 1 Activation: None	7	units: 1 Activation: None	7
Total	None Epochs: 20 Batch size: 32	73677	None Epochs: 20 Batch size: 32	18381

Table 7.5 Structures for both RF and LSTM

Model	Parameters	Trainable parameters
Random Forest	Number of estimators: 100 Max depth: None Min samples leaf: 1 Criterion: MSE	None
LSTM Dense 1 Total	Units: 100 Units: 1 Epochs: 100 Batch size: 10	45200 101 45301

7.5.4 Post-processing

To obtain the final prediction, we need to post-process the output from CNN-IMF and the regression model (RF or LSTM). We sum the partial results coming from them to obtain the complete prediction, and then we denormalize it. The data are

denormalized using the reverse of the min-max algorithm used for normalization, as discussed in Section 7.5.1. The final result is the GHI forecast.

7.5.5 Creation and utilization of the proposed model

As discussed in Section 7.3, the core of our solution consists of VMD and both CNNs. As shown in Figure 7.4, the dataset is composed of two parts: i) GHI and ii) Weather elements.

The GHI data array ($n_{day} \times p_{GHI}$) is fed to the VMD algorithm, and the signal is decomposed in the number of K modes chosen and a residue. Because CNNs are typically used for image processing, the IMFs signals are built together to create a 3D image using K as a further dimension ($n_{day} \times p_{GHI} \times K$) and the resulting array is used as input to the channels of CNN-IMF. The output of the proposed model has only one dimension that corresponds to the sum of all the K IMFs. As an advantage, using CNN allows for reducing the complexity of feeding the model with a 3D array of all the IMFs, instead of creating a model for each IMF signal. As shown in Figure 7.4, channels work in parallel to train the CNN-IMF using all the IMFs. Only a CNN is used to obtain the first part of the final GHI prediction.

The residue that is not involved in the previous part is used to train a second CNN (i.e., CNN-RES). The remaining part of the signal varies over days based on weather conditions and climate changes. Therefore, CNN-RES is trained to extract model features from the signal. The final layer of the CNN-RES contains the major characteristic of the signal. As shown in Figure 7.4, the regression model (RF or LSTM) is fed with the final layer output of CNN-RES and the second part of the dataset, the meteorological data (i.e., i) air temperature, ii) relative humidity, iii) air pressure, iv) wind speed and v) cloud coverage), to predict the second part of the GHI. Finally, the prediction outputs of the two parts are post-processed, as described in Section 7.5.4.

As shown in Figure 7.2, the proposed model forecasts the GHI until the desired prediction period is reached (e.g., $15min$, $1h$ or $24h$), the time-resolution depends on the chosen model (M15 or M60). If the prediction period is longer than the chosen resolution, the algorithm uses the previous GHI predictions to build the next input arrays. Figure 7.5 shows the flowchart of the applied algorithm. The algorithm runs

until the prediction time is reached, and the output of all the steps composes the final prediction.

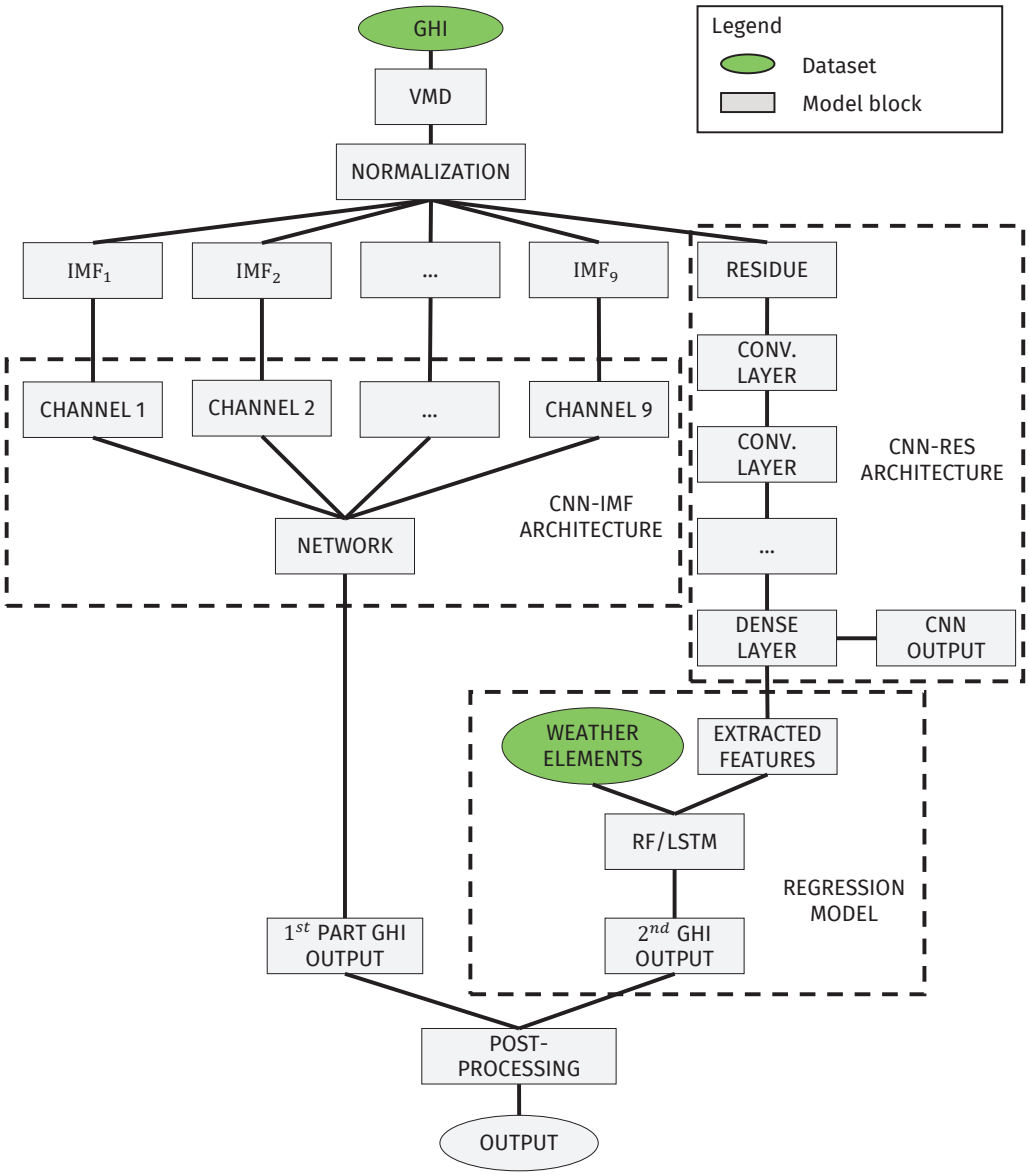


Figure 7.4 Detailed schema of the proposed methodology

7.6 Experimental results and analysis

This section presents the experimental results of both *M15* and *M60* models (see Section 7.5) in forecasting GHI in short- and long-term (i.e., from the next 15*min* up

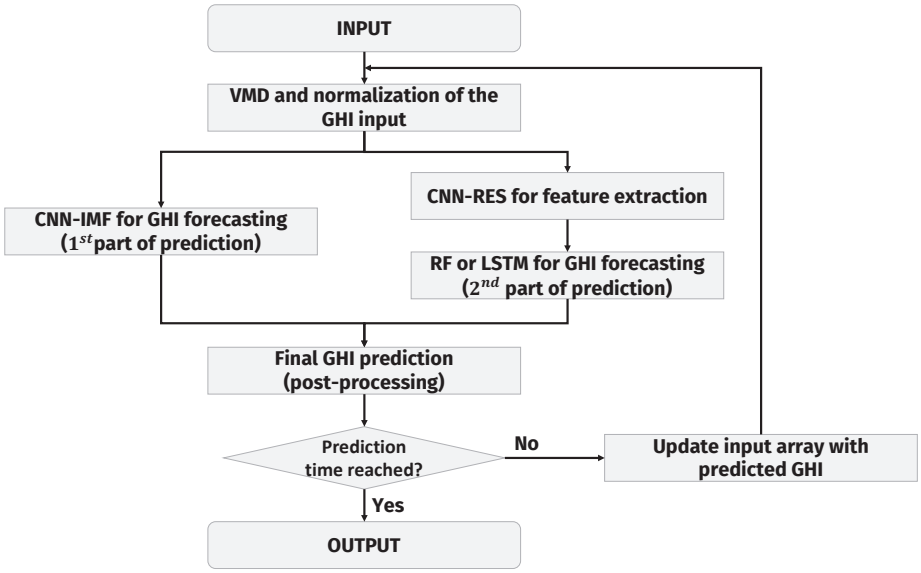


Figure 7.5 Flow chart of our model to forecast GHI

to the next 24*h*). We also evaluate and compare the performance in exploiting either RF or LSTM, as they can be exploited alternatively.

To assess the accuracy of our models, we adopted the performance indicators presented in Section 7.4. First, we analyzed each individual season and then the whole year. Finally, we compare our M15 and M60 models with two models in literature [6] based on Nonlinear Autoregressive neural network (NAR).

7.6.1 Performance assessments of the M15 model

Table 7.6 reports the performance indicators obtained by forecasting GHI with the M15 model for different time-horizons. As described in Section 7.5, M15 performs its analysis with 15*min* time resolution. In general, we can notice that M15 performs better during the winter because there are small variations in GHI trends, making easier the learning phase of our model and consequently increasing the accuracy during tests. Indeed, varying the time horizon from the next 15*min* up to the next 24*h*, R^2 decreases from 0.96 to about 0.63.

On the other hand, summer provides worse results because of high variations in GHI trends among sunny, cloudy, and rainy days. Thus, R^2 varies from 0.90 for the next 15*min* to about 0.66 for next 24*h*.

Autumn and spring have similar GHI trends; thus, their performances are quite comparable.

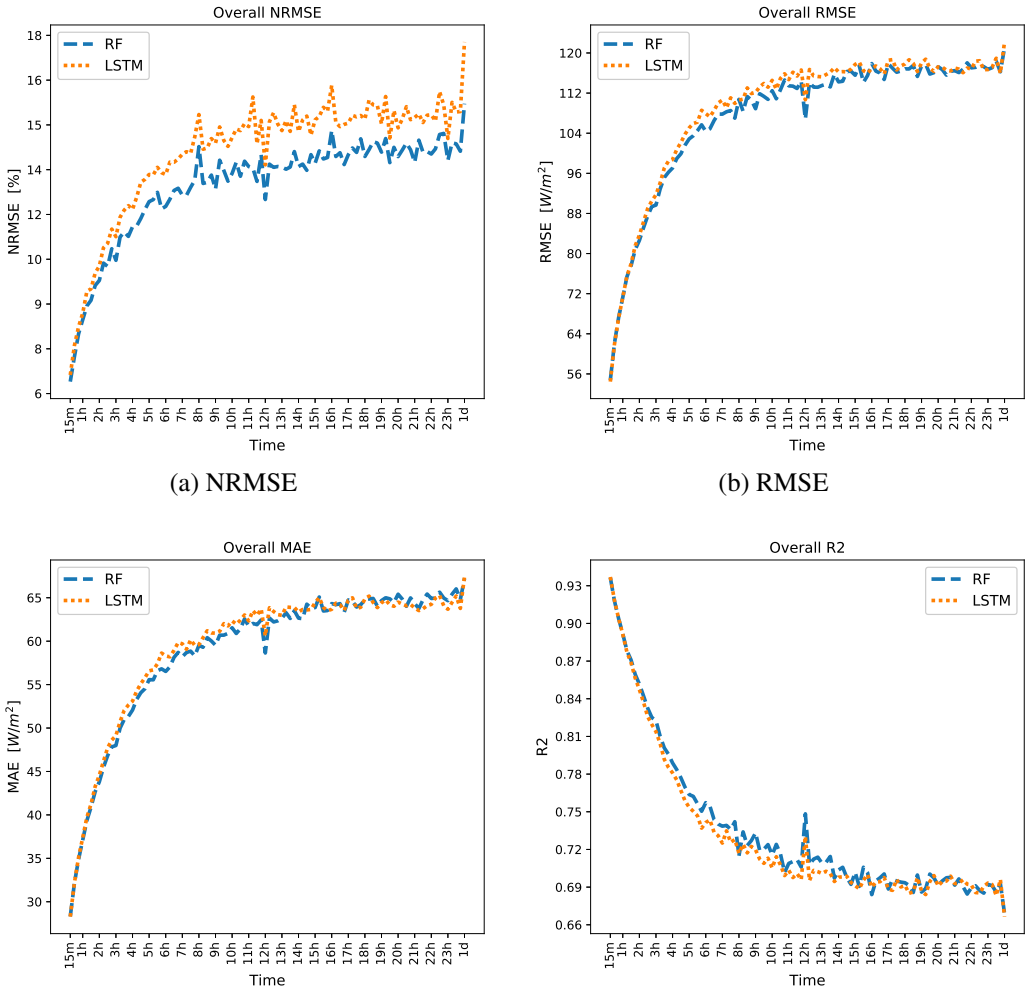
Finally, considering the whole year, R^2 decreases from about 0.94 to 0.67; RMSE varies from 54.464 W/m^2 to 121.066 W/m^2; MAE increases from 28.286 W/m^2 to 67.208 W/m^2; NRMSE rises from 6.401% to 15.711%.

As reported in Table 7.6 and shown in Figure 7.6, there are no remarkable differences in exploiting RF or LSTM in terms of RMSE, MAE, and R^2. Indeed, M15 provides similar results. However, if we consider NRMSE, we can notice that RF provides slightly better performance than LSTM, especially for long-term forecast, e.g., 15.711% and 17.779%, respectively, at the next 24h. In general, Figure 7.6 shows that the gap between RF and LSTM becomes almost constant for predictions with time-horizons between 6h and 24h (e.g., ~ 0.1 for R^2 and $\sim 2\%$ for NRMSE).

Finally, Figure 7.7 and Figure 7.8 show the trends for one week, taken as an example, for each meteorological season at next 15min and next 24h, respectively. Real GHI samples are represented by the blue dashed line, while forecasts with RF and LSTM are depicted by orange and green dashed lines, respectively.

Table 7.6 Performance indicator of M15 (i.e. 15*min* time-resolution) at different time-horizons

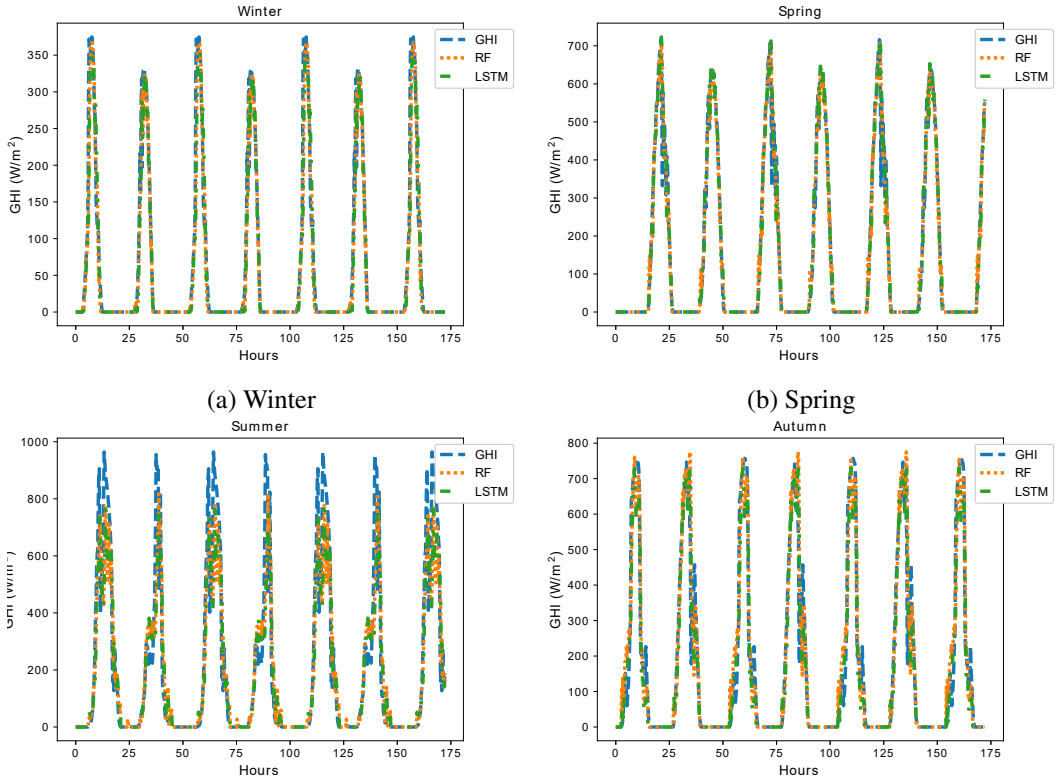
Time resolution	Time-horizon	Season	RMSE $[W/m^2]$		MAE $[W/m^2]$		NRMSE [%]		R^2	
			RF	LSTM	RF	LSTM	RF	LSTM	RF	LSTM
15m	15m	winter	24.314	24.040	10.527	10.477	3.978	4.076	0.960	0.960
		spring	64.190	62.142	31.402	30.240	6.680	6.709	0.942	0.944
		summer	89.493	90.591	52.878	53.922	9.647	10.319	0.900	0.898
		autumn	41.076	41.084	18.433	18.505	5.297	5.385	0.944	0.944
		year	54.768	54.464	28.310	28.286	6.401	6.622	0.936	0.936
	30m	winter	28.313	27.976	12.213	12.142	4.632	4.771	0.945	0.946
		spring	74.273	72.019	36.545	35.341	7.729	7.878	0.922	0.927
		summer	98.627	99.986	58.405	59.649	10.632	11.540	0.879	0.876
		autumn	47.341	47.717	21.432	21.643	5.953	6.255	0.926	0.925
		year	62.138	61.924	32.149	32.194	7.237	7.611	0.918	0.918
	45m	winter	31.400	30.973	13.637	13.589	5.137	5.282	0.932	0.934
		spring	81.859	78.784	40.250	39.085	8.476	8.457	0.905	0.912
		summer	104.691	107.164	62.078	63.915	11.823	12.233	0.864	0.857
		autumn	50.975	51.592	23.583	23.868	6.574	6.987	0.914	0.912
		year	67.231	67.128	34.887	35.114	8.002	8.240	0.904	0.904
	1h	winter	33.140	32.897	14.369	14.353	5.422	5.610	0.925	0.926
		spring	87.294	84.075	43.275	42.125	9.049	8.888	0.892	0.900
		summer	109.150	111.714	65.105	66.984	12.449	12.894	0.852	0.845
		autumn	55.479	56.354	25.702	26.054	6.977	7.387	0.898	0.895
		year	71.266	71.260	37.113	37.379	8.474	8.695	0.892	0.891
	3h	winter	48.721	49.419	21.013	21.262	7.971	8.427	0.837	0.832
		spring	106.852	103.522	55.520	54.383	10.782	10.436	0.839	0.848
		summer	131.294	139.784	80.215	84.867	14.978	16.134	0.785	0.756
		autumn	71.790	74.183	35.214	36.013	8.122	10.024	0.829	0.818
		year	89.664	91.727	47.991	49.131	10.463	11.255	0.823	0.814
	12h	winter	60.163	60.834	25.939	26.496	9.843	10.374	0.752	0.746
		spring	132.763	129.236	70.254	68.955	12.909	12.789	0.751	0.764
		summer	153.428	167.490	97.040	103.618	17.503	19.332	0.707	0.651
		autumn	80.855	85.127	41.317	43.393	9.745	12.149	0.784	0.760
		year	106.802	110.672	58.638	60.616	12.500	13.661	0.748	0.730
	24h	winter	73.906	72.896	32.699	32.908	13.226	14.633	0.625	0.635
		spring	155.058	144.480	83.022	78.514	14.719	14.297	0.660	0.705
		summer	164.377	183.571	104.970	113.210	23.940	29.265	0.664	0.581
		autumn	90.922	87.841	48.140	44.937	10.959	12.920	0.726	0.745
		year	121.066	122.197	67.208	67.392	15.711	17.779	0.669	0.666



(a) NRMSE (b) RMSE

(c) MAE (d) R^2

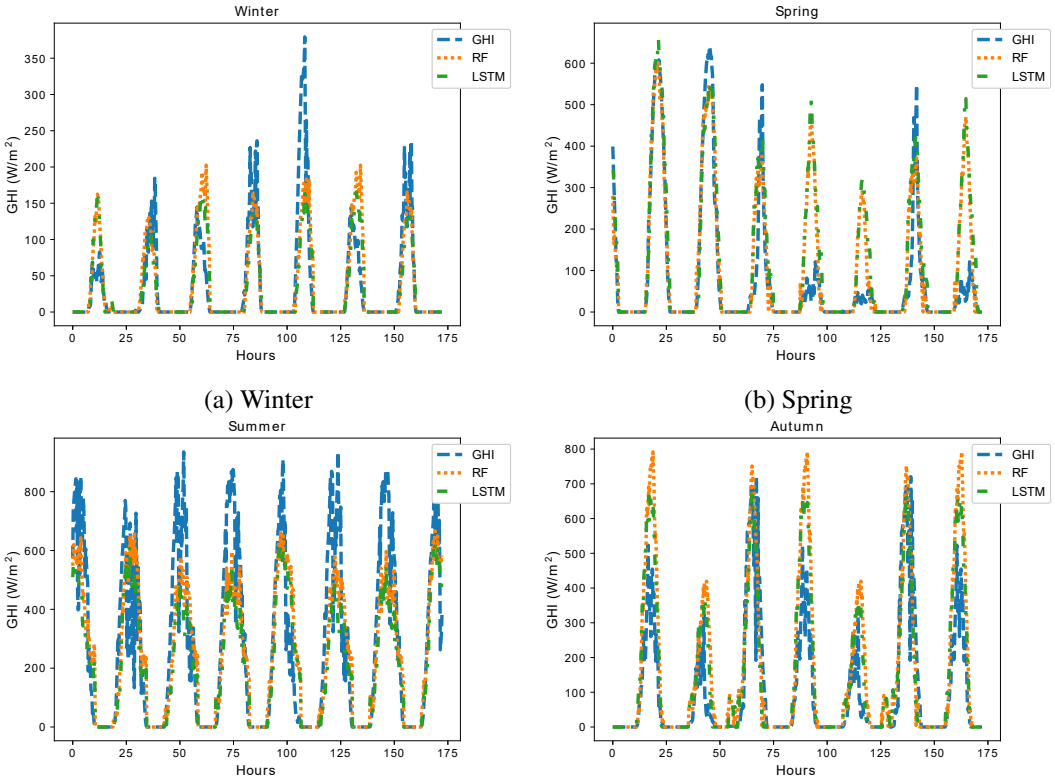
Figure 7.6 Overall performance trends for M15



(a) Winter (b) Spring

(c) Summer (d) Autumn

Figure 7.7 GHI forecast trends for M15 at next 15*min*



(a) Winter (b) Spring

(c) Summer (d) Autumn

Figure 7.8 GHI forecast trends for M15 at next 24*h*

7.6.2 Performance assessments of the M60 model

Table 7.7 reports the performance indicators given by the M60 for GHI predictions at different time-horizons, from next $1h$ up to next $24h$. Differently than M15, M60 performs its forecasts exploiting a 60min time-resolution (see Section 7.5). As for M15, M60 provides the best results in forecasting GHI in winter, while worse performances are given during summer for the very same reasons discussed in Section 7.6.1. Indeed, R^2 decreases from 0.93 for the next 1h to about 0.73 for the next $24h$ in winter, while it drops from approx. 0.90 to about 0.79 in summer.

Also, in this case, forecasts obtained in autumn and spring are quite comparable because of their similarities in GHI trends.

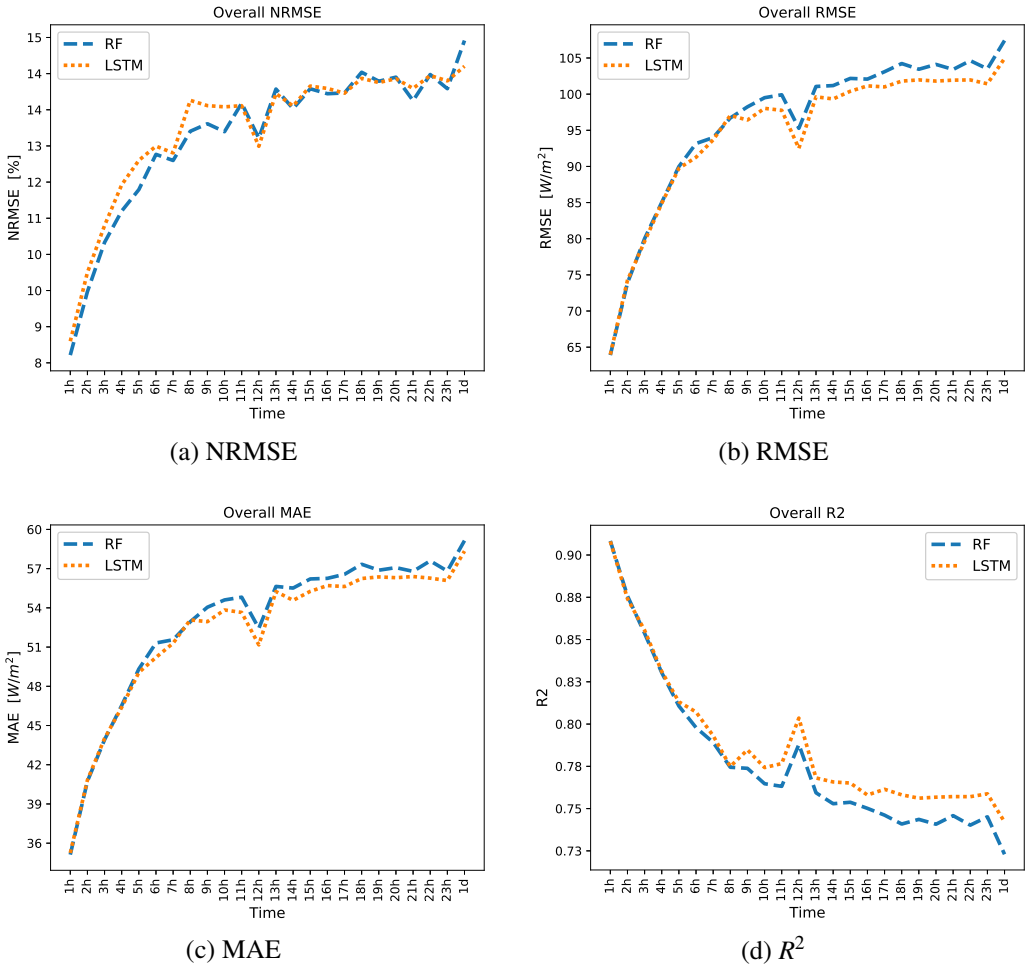
Finally, considering the whole year, R^2 falls from about 0.91 to 0.74; RMSE rises from 63.899 W/m^2 to 104.962 W/m^2; MAE varies from 35.126 W/m^2 to 58.355 W/m^2; NRMSE increases from 8.171% to 14.565%.

Differently from M15, M60 performs better with LSTM than RF for long-term forecasts. This is highlighted by the R^2 in Figure 7.9 for time-horizons over the $10h$. While, from $1h$ to $10h$, RF slightly outperforms LSTM in the NRMSE; indeed, the difference is $\sim 0.3\%$, which is negligible. More in general, results point out that, over the $10h$, the error rate is almost constant (e.g., ~ 0.5 for R^2 and $\sim 2\%$ for NRMSE).

Finally, Figure 7.10 and Figure 7.11 report the trends for one week for each meteorological season at $1h$ and $24h$ time-horizon, respectively. The blue dashed line represents the real GHI samples, orange and green dashed lines depict forecasts with RF and LSTM, respectively.

Table 7.7 Performance indicator of M60 (i.e. 60*min* time-resolution) at different time-horizons

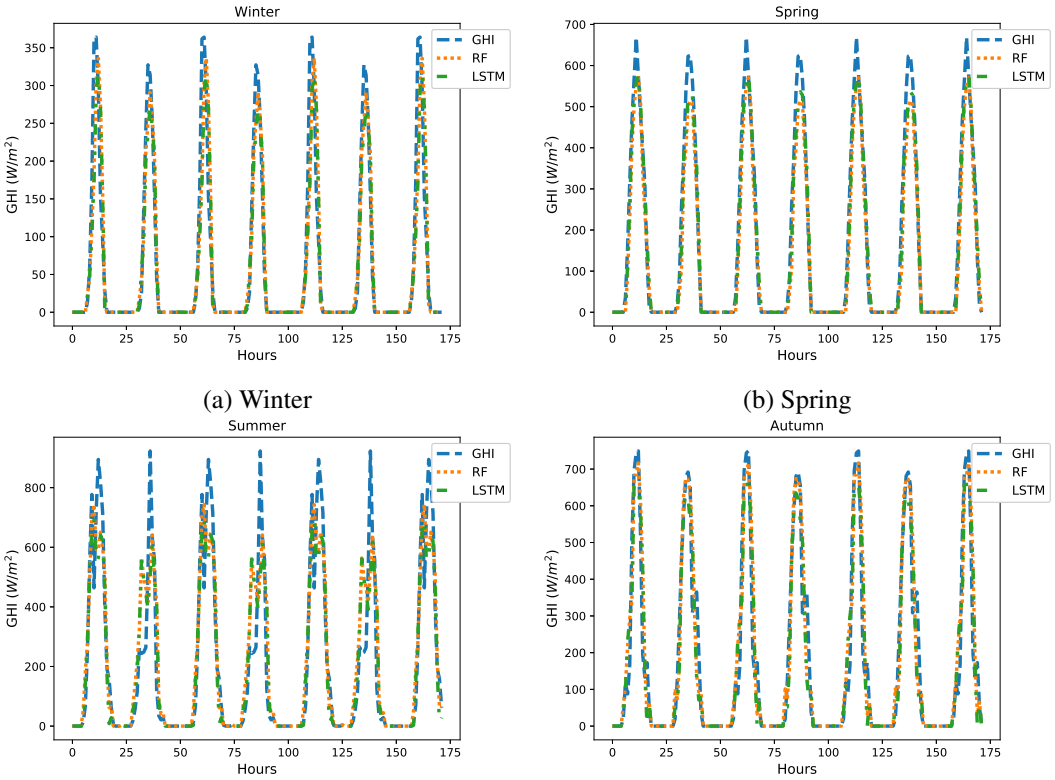
Time resolution	Time-horizon	Season	RMSE $[W/m^2]$		MAE $[W/m^2]$		NRMSE [%]		R^2	
			RF	LSTM	RF	LSTM	RF	LSTM	RF	LSTM
1h	1h	winter	31.739	31.420	14.862	14.642	5.502	5.726	0.929	0.931
		spring	82.591	80.132	47.378	46.199	9.282	9.009	0.901	0.906
		summer	88.222	90.350	52.735	54.066	10.501	11.140	0.899	0.894
		autumn	53.046	54.011	25.528	26.108	7.399	8.037	0.904	0.901
		year	63.899	63.978	35.126	35.254	8.171	8.478	0.908	0.908
	3h	winter	43.389	42.278	19.800	19.056	7.521	7.363	0.868	0.874
		spring	106.027	100.538	60.989	59.263	13.112	12.109	0.836	0.853
		summer	105.532	109.510	63.380	65.662	13.079	14.622	0.855	0.844
		autumn	65.108	66.204	31.476	32.065	8.947	10.059	0.855	0.850
		year	80.014	79.633	43.911	44.012	10.665	11.038	0.854	0.855
	12h	winter	57.074	51.697	25.573	23.054	9.893	9.449	0.771	0.812
		spring	132.854	121.156	77.804	71.629	17.170	14.562	0.743	0.787
		summer	118.344	122.719	71.020	73.697	14.667	16.300	0.818	0.805
		autumn	72.811	74.457	35.352	36.288	10.156	10.856	0.819	0.811
		year	95.271	92.507	52.437	51.167	12.972	12.792	0.788	0.804
	24h	winter	68.838	62.266	31.694	28.379	12.354	11.432	0.667	0.728
		spring	153.807	139.886	89.516	83.129	20.750	16.813	0.656	0.716
		summer	125.567	132.757	74.702	79.477	15.640	17.633	0.795	0.771
		autumn	81.556	84.939	40.741	42.436	11.800	12.384	0.773	0.754
		year	107.442	104.962	59.163	58.355	15.136	14.565	0.723	0.742



(a) NRMSE (b) RMSE

(c) MAE (d) R^2

Figure 7.9 Overall performance trends for M60



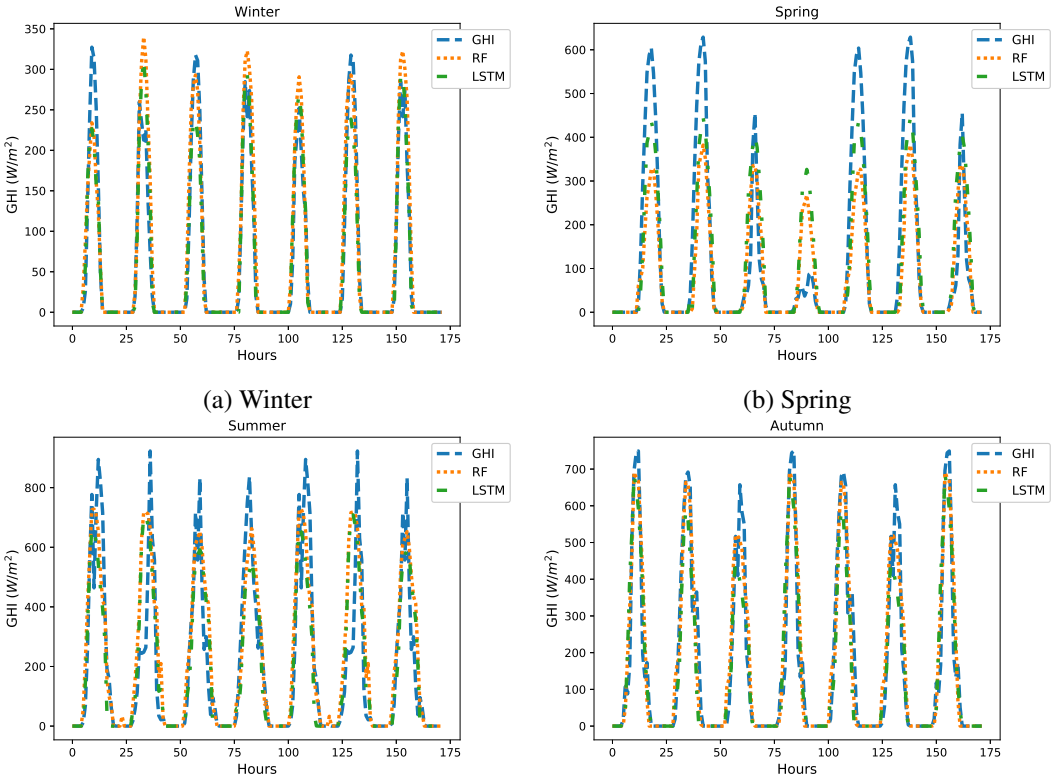
(a) Winter

(b) Spring

(c) Summer

(d) Autumn

Figure 7.10 GHI forecast trends for M60 at next 1h



(a) Winter (b) Spring

(c) Summer (d) Autumn

Figure 7.11 GHI forecast trends for M60 at next 24*h*

7.6.3 Comparison with NAR-based model in literature

In this section, we compare the performance of our models with the literature solution proposed in [6], where authors presented a Nonlinear Autoregressive neural network (NAR) for short-term GHI forecasts. To perform a fair comparison, we implemented and trained two NAR models, NAR-15 and NAR-60, following the methodology provided in [6] that have been trained and tested with our dataset with 15*min* and 60*min* time-resolution, respectively. Hence, we compare our M15 with NAR-15 and our M60 with NAR-60. As reported in [6], both NAR-15 and NAR-60 only need GHI trends in input, neglecting the other meteorological elements.

Table 7.8 summarizes this comparison, reporting the results of the performance indicators in forecasting GHI for different time horizons. Considering the 15*min* time resolution, we can notice that our M15, in both RF- and LSTM-based implementations, improves the forecast in terms of error rate and time horizon. Indeed, NAR-15 provides good results in the very short term, i.e., up to the next 45*min*. In forecasts after the next 1*h*, our M15 outperforms NAR-15 in all the indicators. As can be noted, the NRMSE trend for NAR-15 reports some fluctuations due to wrong forecasts that can be considered outliers. These errors are present, in particular when the prediction is close to the max value of the real GHI. In detail, R^2 for NAR-15 decreases from about 0.96 to 0.72 (M15-RF: from 0.94 to 0.85, M15-LSTM: from 0.94 to 0.85); RMSE varies from 47.70 W/m^2 to 123.64 W/m^2 (M15-RF: from 54.77 W/m^2 to 82.62 W/m^2, M15-LSTM: from 54.46 W/m^2 to 83.66 W/m^2); MAE increases from 17.19 W/m^2 to 77.45 W/m^2 (M15-RF: from 28.31 W/m^2 to 43.81 W/m^2, M15-LSTM: from 28.29 W/m^2 to 44.57 W/m^2); NRMSE rises from 5.24% to 6.43% (M15-RF: from 6.40% to 9.79%, M15-LSTM: from 6.62% to 10.27%).

Regarding 60*min* time-resolution (see Table 7.8), similar considerations can be made. We can notice that our M60 outperforms NAR-60 from the next 2h predictions (i.e., after two prediction steps). Indeed, NAR-60 performances decrease considerably in the long term, making it suitable only for short-term forecasts. Also, in this case, the NRMSE trend for NAR-60 is affected by fluctuations due to some outliers in predictions close to the max value of the real GHI. After 6*h*, NAR-60 performances are completely degraded. This is highlighted by R^2, which values are lower than 0. In detail, R^2 for NAR-60 decreases from about 0.93 to -2.59 (M60-RF: from 0.91 to 0.72, M60-LSTM: from 0.91 to 0.74); RMSE varies from 61.31 W/m^2 to 432.790 W/m^2 (M60-RF: from 63.90 W/m^2 to 107.42 W/m^2, M60-

Table 7.8 Comparison of our methodology with [6] at different time-horizons for both 15min and 60min time-resolutions

Time resolution	Time-horizon	RMSE [W/m^2]			MAE [W/m^2]			NRMSE [%]			R^2		
		RF-based	LSTM-based	NAR	RF-based	LSTM-based	NAR	RF-based	LSTM-based	NAR	RF-based	LSTM-based	NAR
15min	15m	54.768	54.464	47.700	28.310	28.286	17.185	6.401	6.622	5.239	0.936	0.937	0.958
	30m	62.138	61.924	64.830	32.149	32.194	26.493	7.237	7.611	7.442	0.918	0.918	0.922
	45m	67.231	67.128	75.284	34.887	35.114	33.661	8.002	8.240	5.145	0.904	0.904	0.895
	1h	71.266	71.260	84.777	37.113	37.379	40.800	8.474	8.695	4.386	0.892	0.891	0.866
	2h	82.620	83.661	123.636	43.808	44.572	77.450	9.789	10.272	6.433	0.852	0.847	0.716
60min	1h	63.899	63.978	61.305	35.126	35.254	28.094	8.171	8.478	6.261	0.908	0.908	0.928
	2h	73.880	74.249	85.131	40.739	40.823	49.093	9.576	10.000	10.283	0.876	0.874	0.861
	3h	80.014	79.633	114.732	43.911	44.012	81.799	10.665	11.038	14.427	0.854	0.855	0.748
	6*h*	93.170	91.267	230.197	51.314	50.209	171.747	12.613	12.795	19.419	0.798	0.807	−0.016
	12h	95.271	92.507	405.587	52.437	51.167	255.536	12.972	12.792	15.372	0.788	0.804	−2.153
	24h	107.415	104.942	432.790	59.141	58.340	298.978	15.132	14.563	12.909	0.723	0.742	−2.590

LSTM: from 63.98 W/m^2 to 104.94 W/m^2); MAE increases from 28.09 W/m^2 to 298.98 W/m^2 (M60-RF: from 35.13 W/m^2 to 59.14 W/m^2, M60-LSTM: from 35.25 W/m^2 to 58.34 W/m^2); NRMSE rises from 6.26% to 12.91% (M60-RF: from 8.17% to 15.13%, M60-LSTM: from 8.48% to 14.56%).

7.7 Discussion and Remarks

In this work, we presented a methodology to forecast the Global Horizontal Solar Irradiance in the short- and long-term, combining VMD, CNNs, and two regression models (used alternatively). Following our methodology, we proposed two models, M15 and M60, to test different time resolutions (15*min* and 60*min*, respectively). The analysis of the experimental results highlighted that the performances of both models are comparable. In particular, M15 is suitable for sub-hourly forecasts in the short term, while M60 is for hourly previsions in the long term.

GHI is the renewable source needed by PV systems to produce energy. To move forward more sustainable cities and industries, next-generation energy distribution systems will include different RES distributed across the network. Consequently, current energy distribution systems must be enhanced with novel and smart control systems by taking advantage of rising ICT technologies, thus becoming smart grids. In this view, DR and DSM are examples of possible applications to match the energy produced by RES with the total energy requested by the city [46, 128, 183]. Usually, DR strategies need sub-hourly information to work in the short-term (i.e., from the next 15*min*), while DSM needs hourly information up to the next 24*h* to perform load balancing. In this context, M15 and M60, alternatively combined with tools to estimate PV production, like PVsim [25], can unlock the design and development

of more accurate policies for DR and DSM, respectively, and also improve energy management in Industry 4.0 scenario.

Conclusions

In the last years, the research towards Industry 4.0 and process digitalization gained a strong momentum, also pushed by the European and national communities through several funding initiatives (e.g., Horizon programme, PNRR, Regional development funds). All of these projects have as their primary purpose the aim to improve the overall production plant by integrating new smart technologies, retrofitting already existing machines, and digitalizing and rethinking entire industrial processes.

In this context, several technologies have received strong impulse at different levels and were integrated into industrial scenarios starting from infrastructures like Cloud Computing and Additive Manufacturing (AM) production lines to local hardware and software like Internet of Things (IoT) devices, Cyber-Physical systems (CPS), Artificial Intelligence (AI) and Machine Learning (ML) algorithms.

Consequently, by collaborating on several Industry 4.0 projects, my research activity had the opportunity to touch on all these issues in several contexts: from industry and manufacturing digitalization to Smart Sensors integration and energy prediction. In order to improve the overall processes and collect the data produced by different sources, I primarily focused on the design and development of a management framework on cloud computing for the collection, management, and analysis of data from heterogeneous sources. Then, starting from these data, I proposed and integrated into the platform a set of AI and ML algorithms to address different industrial problems like AM process defects detection or energy production forecasting.

In more detail, one of the main issues faced by Industry 4.0 is the acquisition of data and retrofitting of machines. To address these challenges, I conducted an analysis of various aspects of smart factories in the automotive and aerospace industries, including production plants, available machines, data types, and exchange protocols. I also examined user experiences and requirements to propose a gen-

eral management platform suitable for all the different industrial scenarios. The implemented infrastructure comprises a set of Internet of Things (IoT) devices connected to available machines, which facilitates data collection and exchange via standard protocols such as HTTP and MQTT, and a management platform, which has been implemented in both research and industrial settings, that utilizes cutting-edge open-source technologies such as Docker and allows to perform analysis on the data stored.

To populate the database and show the potential of the platform, I first collected and analyzed data from smart manufacturing processes such as Laser Powder Bed Fusion (L-PBF) and Electron Beam Powder Bed Fusion (EB-PBF). For the first process, I incorporated an IoT device connected to a camera within the printer chamber to monitor the powder bed and collect images in the management platform of each layer printed, both before and after the printing process was completed. Then, I applied a set of algorithms (image processing, U-Net) to identify potential defects in the layers printed and monitor the profile of the part during the process. Furthermore, because producing defects is costly and not practicable, I generate synthetic data using a GAN model (ConSinGAN) based on the actual data to increase the dataset available. These algorithms were incorporated as services in the framework established to be easily executed by platform users. For the second process, the data obtained from all the sensors already installed in the machine, such as temperature, power, and log files, were analyzed by means of a methodology that utilizes an autoencoder neural network and ad-hoc thresholds to predict the future status (failure or success) of a printing job in real-time.

The results from both processes showed that new capabilities could be simply integrated into older equipment by incorporating IoT devices in the manufacturing facility. Additionally, the algorithms developed illustrate the potential of the methodologies and yield positive results in identifying various types of defects and forecasting the status of a printing job through the use of ML and AI algorithms.

Consequently, I extended the data collection and analysis to other industrial scenarios, such as aerospace production lines, industrial localization, and energy predictions.

I analyzed the quality inspection procedures used for aircraft engine components constructed with AM technology to decrease the overall inspection time, enhance quality, and ensure standardization and process control. The solution developed

integrates Cyber-Physical Systems (CPS), a camera, and an especially designed optical fiber sensor to collect process data and analyze them with computer vision and AI algorithms (image-processing, U-Net) for the detection of external and internal defects that could occur during the production.

In the context of industrial localization, I assessed the performances of multiple ML algorithms (k-NN, SVM, etc.) for locating commercial Bluetooth Low Energy (BLE) IoT devices in industrial settings under varying experimental conditions. The results demonstrated that fingerprinting-based methods decrease positioning error in comparison to distance-based methods, even in an industrial environment.

Lastly, I proposed a methodology for performing short-, mid-, and long-term Global Horizontal Irradiation (GHI) forecasting, which is beneficial for photovoltaic energy predictions and for energy efficiency in Industry 4.0. I collected weather data (GHI, temperature, pressure, etc.) from a meteorological station at Politecnico di Torino and analyzed them with a Variational Mode Decomposition (VMD) and two Convolutional Neural Networks (CNN) in combination with either Random Forest (RF) or Long Short Term Memory (LSTM). By utilizing this methodology, it is possible to gain an understanding of the quantity and quality of energy available, which can be used to devise appropriate energy policies. Additionally, this information can be leveraged to manage production loads and machines more effectively. Thus, the study highlights the benefits of this methodology in improving energy efficiency and reducing costs in industrial contexts.

In conclusion, the integration of several technologies will play a huge role in Industry 4.0 and will provide opportunities for increased automation, data analytics, and connectivity in manufacturing and other industries, leading to improved capability, flexibility, and customization. However, all of these benefits also come with huge challenges that need to be addressed in the future to realize an industry on a human scale. Thus, companies are moving towards the next industrial revolution, called Industry 5.0, which emphasizes the collaboration between humans and machines in the production process (e.g., Augmented reality, voice and gesture recognition). This will allow for a more flexible and customizable manufacturing process that can respond quickly to changing customer demands. Additionally, Industry 5.0 emphasizes sustainability and social responsibility, ensuring that manufacturing processes are environmentally friendly and socially responsible.

In this context, this thesis provides a first example of a general management platform together with several AI and ML algorithms which can serve as a basis for a more extended framework by integrating people and environmental needs but also new technologies (e.g., Autonomous Robot, Augmented Reality) and other industrial processes, such as logistics and business management. While the initial focus of this management framework is on manufacturing, in the future, the framework could be expanded to other industries that generate large amounts of data, such as the financial, telecommunication, or healthcare sectors. This could help organizations in these industries to collect and analyze data more efficiently, leading to improved decision-making.

With a continuous collaboration between researchers and companies, it could be possible to manage all the data through the platform developed and integrate new and existing technologies and algorithms in order to reach a new level of automation and, with a look towards Industry 5.0, develop an environment to improve people welfare by reducing human fatigue and stress during shifts (e.g., using AR) and where machines will work in harmony with humans (e.g., Autonomous robots), augmenting their abilities (e.g., exoskeleton), providing insights and instructions (e.g., work instructions digitalization) and making their work easier and more efficient, while also improving safety and reducing the risk of accidents.

References

[1] Aboulkhair, N. T., Simonelli, M., Parry, L., Ashcroft, I., Tuck, C., and Hague, R. (2019). 3d printing of aluminium alloys: Additive manufacturing of aluminium alloys using selective laser melting. *Progress in materials science*, 106:100578.

[2] Aggour, K. S., Gupta, V. K., Ruscitto, D., Ajdelsztajn, L., Bian, X., Brosnan, K. H., Kumar, N. C., Dheeradhada, V., Hanlon, T., Iyer, N., et al. (2019). Artificial intelligence/machine learning in manufacturing and inspection: A ge perspective. *MRS Bulletin*, 44(7):545–558.

[3] Alanazi, M., Mahoor, M., and Khodaei, A. (2017). Two-stage hybrid day-ahead solar forecasting. *2017 North American Power Symposium, NAPS 2017*, pages 1–6.

[4] Albuquerque Filho, J., Brandão, L. C., Fernandes, B. J., and Maciel, A. M. (2022). A review of neural networks for anomaly detection. *IEEE Access*.

[5] Alexopoulos, K., Nikolakis, N., and Chryssolouris, G. (2020). Digital twin-driven supervised machine learning for the development of artificial intelligence applications in manufacturing. *International Journal of Computer Integrated Manufacturing*, 33(5):429–439.

[6] Aliberti, A., Bottaccioli, L., Cirrincione, G., Macii, E., Acquaviva, A., and Patti, E. (2018). Forecasting Short-term Solar Radiation for Photovoltaic Energy Predictions. In *Proceedings of the 7th International Conference on Smart Cities and Green ICT Systems - Volume 1: SMARTGREENS,*, pages 44–53. INSTICC, SciTePress.

[7] Aljabre, A. (2012). Cloud computing for increased business value. *International Journal of Business and social science*, 3(1).

[8] Alldredge, J., Slotwinski, J., Storck, S., Kim, S., Goldberg, A., and Montalbano, T. (2018). In-situ monitoring and modeling of metal additive manufacturing powder bed fusion. In *AIP Conference Proceedings*, volume 1949, page 020007. AIP Publishing LLC.

[9] Aminzadeh, M. and Kurfess, T. R. (2019). Online quality inspection using bayesian classification in powder-bed additive manufacturing from high-resolution visual camera images. *Journal of Intelligent Manufacturing*, 30(6):2505–2523.

[10] Arcam (2011). *Arcam A2X User Manual*.

[11] Arjovsky, M. and Bottou, L. (2017). Towards principled methods for training generative adversarial networks. *arXiv preprint arXiv:1701.04862*.

[12] Arnold, C. and Körner, C. (2021). Electron-optical in-situ metrology for electron beam powder bed fusion: calibration and validation. *Measurement Science and Technology*, 33(1):14001.

[13] Ashima, R., Haleem, A., Bahl, S., Javaid, M., Mahla, S. K., and Singh, S. (2021). Automation and manufacturing of smart materials in additive manufacturing technologies using internet of things towards the adoption of industry 4.0. *Materials Today: Proceedings*, 45:5081–5088.

[14] Avram, M.-G. (2014). Advantages and challenges of adopting cloud computing from an enterprise perspective. *Procedia Technology*, 12:529–534.

[15] Badescu, V. (2008). *Modeling solar radiation at the earth's surface: Recent advances*. Springer.

[16] Baheti, R. and Gill, H. (2011). Cyber-physical systems. *The impact of control technology*, 12(1):161–166.

[17] Bailey, T. A. and Jain, A. K. (1978). Note on distance-weighted k-nearest neighbor rules.

[18] Baturynska, I., Semeniuta, O., and Martinsen, K. (2018). Optimization of process parameters for powder bed fusion additive manufacturing by combination of machine learning and finite element method: A conceptual framework. *Procedia CIRP*, 67:227–232.

[19] Baumann, F. W., Sekulla, A., Hassler, M., Himpel, B., and Pfeil, M. (2018). Trends of machine learning in additive manufacturing. *International Journal of Rapid Manufacturing*, 7(4):310–336.

[20] Benali, L., Notton, G., Fouilloy, A., Voyant, C., and Dizene, R. (2019). Solar radiation forecasting using artificial neural network and random forest methods: Application to normal beam, horizontal diffuse and global components. *Renewable Energy*, 132:871–884.

[21] Bertsekas, D. P. (2014). *Constrained optimization and Lagrange multiplier methods*. Academic press.

[22] Bhavar, V., Kattire, P., Patil, V., Khot, S., Gujar, K., and Singh, R. (2017). A review on powder bed fusion technology of metal additive manufacturing. *Additive manufacturing handbook*, pages 251–253.

[23] Bhimani, P. and Panchal, G. (2018). Message delivery guarantee and status update of clients based on iot-amqp. In *Intelligent Communication and Computational Technologies*, pages 15–22. Springer.

[24] Bosi, F., Corradi, A., Foschini, L., Monti, S., Patera, L., Poli, L., and Solimando, M. (2019). Cloud-enabled smart data collection in shop floor environments for industry 4.0. In *2019 15th IEEE International Workshop on Factory Communication Systems (WFCS)*, pages 1–8. IEEE.

[25] Bottaccioli, L., Estebsari, A., Patti, E., Pons, E., and Acquaviva, A. (2017). A Novel Integrated Real-time Simulation Platform for Assessing Photovoltaic Penetration Impacts in Smart Grids. *Energy Procedia*, 111:780–789.

[26] Butt, J. (2020). Exploring the interrelationship between additive manufacturing and industry 4.0. *Designs*, 4(2):13.

[27] Čabarkapa, D., Grujić, I., and Pavlović, P. (2015). Comparative analysis of the bluetooth low-energy indoor positioning systems. In *2015 12th International Conference on Telecommunication in Modern Satellite, Cable and Broadcasting Services (TELSIKS)*, pages 76–79. IEEE.

[28] Caggiano, A., Zhang, J., Alfieri, V., Caiazzo, F., Gao, R., and Teti, R. (2019). Machine learning-based image processing for on-line defect recognition in additive manufacturing. *CIRP annals*, 68(1):451–454.

[29] Cannizzaro, D., Aliberti, A., Bottaccioli, L., Macii, E., Acquaviva, A., and Patti, E. (2021a). Solar radiation forecasting based on convolutional neural network and ensemble learning. *Expert Systems with Applications*, 181:115167.

[30] Cannizzaro, D., Simone, F., Illgner-Fehns, K., Mata, S., Mondino, I., Ghiazza, A., Poncino, M., and Di Cataldo, S. (2022). Quality inspection of critical aircraft engine components: towards full automation. In *2022 IEEE 27th International Conference on Emerging Technologies and Factory Automation (ETFA)*, pages 1–4. IEEE.

[31] Cannizzaro, D., Varrella, A. G., Paradiso, S., Sampieri, R., Chen, Y., Macii, A., Patti, E., and Di Cataldo, S. (2021b). In-situ defect detection of metal additive manufacturing: an integrated framework. *IEEE Transactions on Emerging Topics in Computing*, 10(1):74–86.

[32] Cannizzaro, D., Varrella, A. G., Paradiso, S., Sampieri, R., Macii, E., Poncino, M., Patti, E., and Di Cataldo, S. (2021c). In-situ monitoring of additive manufacturing. *Predictive Maintenance in Smart Factories: Architectures, Methodologies, and Use-cases*, pages 207–228.

[33] Cannizzaro, D., Zafiri, M., Jahier Pagliari, D., Patti, E., Macii, E., Poncino, M., and Acquaviva, A. (2019). A comparison analysis of ble-based algorithms for localization in industrial environments. *Electronics*, 9(1):44.

[34] Cao, J. and Lin, X. (2008). Application of the diagonal recurrent wavelet neural network to solar irradiation forecast assisted with fuzzy technique. *Engineering Applications of Artificial Intelligence*, 21:1255–1263.

[35] Carl, V. (2015). Monitoring system for the quality assessment in additive manufacturing. In *AIP Conference Proceedings*, volume 1650, pages 171–176. American Institute of Physics.

[36] Carlin, S. and Curran, K. (2013). Cloud computing security. In *Pervasive and Ubiquitous Technology Innovations for Ambient Intelligence Environments*, pages 12–17. IGI Global.

[37] Chaczko, Z., Mahadevan, V., Aslanzadeh, S., and Mcdermid, C. (2011). Availability and load balancing in cloud computing. In *International Conference on Computer and Software Modeling, Singapore*, volume 14, pages 134–140. sn.

[38] Chandrasekar, S., Coble, J. B., Yoder, S., Nandwana, P., Dehoff, R. R., Paquit, V. C., and Babu, S. S. (2020). Investigating the effect of metal powder recycling in Electron beam Powder Bed Fusion using process log data. *Additive Manufacturing*, 32(July 2019).

[39] Chintalapudi, K., Padmanabha Iyer, A., and Padmanabhan, V. N. (2010). Indoor localization without the pain. In *Proceedings of the sixteenth annual international conference on Mobile computing and networking*, pages 173–184.

[40] Cho, H., Ji, J., Chen, Z., Park, H., and Lee, W. (2015). Measuring a distance between things with improved accuracy. *Procedia Computer Science*, 52:1083–1088.

[41] Cordero, Z. C., Meyer, H. M., Nandwana, P., and Dehoff, R. R. (2017). Powder bed charging during electron-beam additive manufacturing. *Acta Materialia*, 124:437–445.

[42] Daminabo, S. C., Goel, S., Grammatikos, S. A., Nezhad, H. Y., and Thakur, V. K. (2020). Fused deposition modeling-based additive manufacturing (3d printing): techniques for polymer material systems. *Materials today chemistry*, 16:100248.

[43] Daniş, F. S. and Cemgil, A. T. (2017). Model-based localization and tracking using bluetooth low-energy beacons. *Sensors*, 17(11):2484.

[44] Demirhan, H. and Renwick, Z. (2018). Missing value imputation for short to mid-term horizontal solar irradiance data. *Applied Energy*, 225:998 – 1012.

[45] Demrozi, F., Costa, K., Tramarin, F., and Pravadelli, G. (2018). A graph-based approach for mobile localization exploiting real and virtual landmarks. In *2018 IFIP/IEEE International Conference on Very Large Scale Integration (VLSI-SoC)*, pages 249–254. IEEE.

[46] Deng, R., Yang, Z., Chow, M.-Y., and Chen, J. (2015). A survey on demand response in smart grids: Mathematical models and approaches. *IEEE Transactions on Industrial Informatics*, 11(3):570–582.

[47] Dev Singh, D., Mahender, T., and Raji Reddy, A. (2021). Powder bed fusion process: A brief review. *Materials Today: Proceedings*, 46:350–355.

[48] DeviceHive (Accessed: 2023-01-17). Devicehive. https://devicehive.com/.

[49] Di Cataldo, S., Vinco, S., Urgese, G., Calignano, F., Ficarra, E., Macii, A., and Macii, E. (2021). Optimizing quality inspection and control in powder bed metal additive manufacturing: Challenges and research directions. *Proceedings of the IEEE*, 109(4):326–346.

[50] Dilberoglu, U. M., Gharehpapagh, B., Yaman, U., and Dolen, M. (2017). The role of additive manufacturing in the era of industry 4.0. *Procedia Manufacturing*, 11:545–554.

[51] Dinwiddie, R. B., Dehoff, R. R., Lloyd, P. D., Lowe, L. E., and Ulrich, J. B. (2013). Thermographic in-situ process monitoring of the electron-beam melting technology used in additive manufacturing. In Stockton, G. R. and Colbert, F. P., editors, *Thermosense: Thermal Infrared Applications XXXV*, volume 8705, page 87050K. International Society for Optics and Photonics, SPIE.

[52] Dopico, M., Gómez, A., De la Fuente, D., García, N., Rosillo, R., and Puche, J. (2016). A vision of industry 4.0 from an artificial intelligence point of view. In *Proceedings on the international conference on artificial intelligence (ICAI)*, page 407. The Steering Committee of The World Congress in Computer Science, Computer ....

[53] Douzas, G. and Bacao, F. (2018). Effective data generation for imbalanced learning using conditional generative adversarial networks. *Expert Systems with applications*, 91:464–471.

[54] Dragomiretskiy, K. and Zosso, D. (2014). Variational mode decomposition. *IEEE Transactions on Signal Processing*, 62(3):531–544.

[55] Drury, C. G. and Watson, J. (2002). Good practices in visual inspection. *Human factors in aviation maintenance-phase nine, progress report, FAA/Human Factors in Aviation Maintenance*.

[56] Estimote (Accessed: 2023-01-17). Estimote beacons. https://estimote.com/.

[57] Everton, S. K., Hirsch, M., Stravroulakis, P., Leach, R. K., and Clare, A. T. (2016). Review of in-situ process monitoring and in-situ metrology for metal additive manufacturing. *Materials & Design*, 95:431–445.

[58] Fang, X., Luo, Q., Zhou, B., Li, C., and Tian, L. (2020). Research progress of automated visual surface defect detection for industrial metal planar materials. *Sensors*, 20(18):5136.

[59] Faragher, R. and Harle, R. (2014). An analysis of the accuracy of bluetooth low energy for indoor positioning applications. In *Proceedings of the 27th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+ 2014)*, pages 201–210.

[60] Faragher, R. and Harle, R. (2015). Location fingerprinting with bluetooth low energy beacons. *IEEE journal on Selected Areas in Communications*, 33(11):2418–2428.

[61] Feng, W., Mao, Z., Yang, Y., Ma, H., Zhao, K., Qi, C., Hao, C., Liu, Z., Xie, H., and Liu, S. (2022). Online defect detection method and system based on similarity of the temperature field in the melt pool. *Additive Manufacturing*, 54:102760.

[62] Fielding, R. T. and Taylor, R. N. (2002). Principled design of the modern web architecture. *ACM Transactions on Internet Technology (TOIT)*, 2(2):115–150.

[63] Forum, W. E. (2016). The Fourth Industrial Revolution: what it means, how to respond. https://www.weforum.org/agenda/2016/01/the-fourth-industrial-revolution-what-it-means-and-how-to-respond/. [Online; accessed 12-December-2022].

[64] Franciosi, C., Di Pasquale, V., Iannone, R., and Miranda, S. (2019). A taxonomy of performance shaping factors for human reliability analysis in industrial maintenance. *Journal of Industrial Engineering and Management*, 12(1):115–132.

[65] Fu, Y., Downey, A. R. J., Yuan, L., Zhang, T., Pratt, A., and Balogun, Y. (2022). Machine learning algorithms for defect detection in metal laser-based additive manufacturing: A review. *Journal of Manufacturing Processes*, 75:693–710.

[66] Galati, M. (2021). Chapter 8 - Electron beam melting process: a general overview. In Pou, J., Riveiro, A., and Davim, J. P., editors, *Additive Manufacturing*, Handbooks in Advanced Manufacturing, pages 277–301. Elsevier.

[67] Galati, M., Snis, A., and Iuliano, L. (2019a). Experimental validation of a numerical thermal model of the EBM process for Ti6Al4V. *Computers & Mathematics with Applications*, 78(7):2417–2427.

[68] Galati, M., Snis, A., and Iuliano, L. (2019b). Powder bed properties modelling and 3D thermo-mechanical simulation of the additive manufacturing Electron Beam Melting process. *Additive Manufacturing*, 30.

[69] Galván, I. M., Valls, J. M., Cervantes, A., and Aler, R. (2017). Multi-objective evolutionary optimization of prediction intervals for solar energy forecasting with neural networks. *Information Sciences*, 418–419:363–382.

[70] García-Hinde, O., Terrén-Serrano, G., Hombrados-Herrera, M., Gómez-Verdejo, V., Jiménez-Fernández, S., Casanova-Mateo, C., Sanz-Justo, J., Martínez-Ramón, M., and Salcedo-Sanz, S. (2018). Evaluation of dimensionality reduction methods applied to numerical weather models for solar radiation forecasting. *Engineering Applications of Artificial Intelligence*, 69:157–167.

[71] Gaytan, S. M., Murr, L. E., Medina, F., Martinez, E., Lopez, M. I., and Wicker, R. B. (2009). Advanced metal powder based manufacturing of complex components by electron beam melting. *Materials Technology*, 24(3):180–190.

[72] Gibson, I., Rosen, D. W., Stucker, B., Khorasani, M., Rosen, D., Stucker, B., and Khorasani, M. (2021). *Additive manufacturing technologies*, volume 17. Springer.

[73] Gobert, C., Arrieta, E., Belmontes, A., Wicker, R. B., Medina, F., and McWilliams, B. (2019). Conditional generative adversarial networks for in-situ layerwise additive manufacturing data. In *2019 International Solid Freeform Fabrication Symposium*. University of Texas at Austin.

[74] Gomez, C., Oller, J., and Paradells, J. (2012). Overview and evaluation of bluetooth low energy: An emerging low-power wireless technology. *Sensors*, 12(9):11734–11753.

[75] Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial networks. *arXiv preprint arXiv:1406.2661*.

[76] Grasso, M. and Colosimo, B. (2019). A statistical learning method for image-based monitoring of the plume signature in laser powder bed fusion. *Robotics and Computer-Integrated Manufacturing*, 57:103–115.

[77] Grasso, M., Gallina, F., and Colosimo, B. M. (2018). Data fusion methods for statistical process monitoring and quality characterization in metal additive manufacturing. *Procedia CIRP*, 75:103–107.

[78] Gröger, C. (2018). Building an industry 4.0 analytics platform. *Datenbank-Spektrum*, 18(1):5–14.

[79] Grzechca, D. E., Pelczar, P., and Chruszczyk, L. (2016). Analysis of object location accuracy for ibeacon technology based on the rssi path loss model and fingerprint map. *International Journal of Electronics and Telecommunications*.

[80] Guerreiro, B. V., Lins, R. G., Sun, J., and Schmitt, R. (2018). Definition of smart retrofitting: First steps for a company to deploy aspects of industry 4.0. In *Advances in Manufacturing*, pages 161–170. Springer.

[81] Haghnegahdar, L., Joshi, S. S., and Dahotre, N. B. (2022). From iot-based cloud manufacturing approach to intelligent additive manufacturing: Industrial internet of things—an overview. *The International Journal of Advanced Manufacturing Technology*, pages 1–18.

[82] Hamzah, R. A. and Ibrahim, H. (2016). Literature survey on stereo vision disparity map algorithms. *Journal of Sensors*, 2016.

[83] Hastie, T., Tibshirani, R., Friedman, J. H., and Friedman, J. H. (2009). *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer.

[84] Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. (2017). Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30:6626–6637.

[85] Hillar, G. C. (2017). *MQTT Essentials-A lightweight IoT protocol*. Packt Publishing Ltd.

[86] Hinz, T., Fisher, M., Wang, O., and Wermter, S. (2020). Improved techniques for training single-image gans. *arXiv preprint arXiv:2003.11512*.

[87] Ho, T. K. (1995). Random decision forests. *Proceedings of the International Conference on Document Analysis and Recognition*, 1:278–282.

[88] Hobbs, A. (2008). An overview of human factors in aviation maintenance. *ATSB Safty Report, Aviation Research and Analysis Report AR*, 55:2008.

[89] Hochreiter, S. (1998). The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 6(02):107–116.

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

[91] Hong, S. H., Kyzer, T., Cornelius, J., Zahiri, F., and Wang, Y. (2022). Intelligent anomaly detection of robot manipulator based on energy consumption auditing. In *2022 IEEE Aerospace Conference (AERO)*, pages 1–11. IEEE.

[92] Huang, J., Qin, Q., and Wang, J. (2020). A review of stereolithography: Processes and systems. *Processes*, 8(9):1138.

[93] Hubel, D. H. and Wiesel, T. N. (1968). Receptive fields and functional architecture of monkey striate cortex. *The Journal of physiology*, 195(1):215–43.

[94] Hundman, K., Constantinou, V., Laporte, C., Colwell, I., and Soderstrom, T. (2018). Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 387–395.

[95] Huo, C. and Li, T. (2021). Research on intelligent manufacturing equipment visual inspection control based on machine vision. In *2021 IEEE 4th International Conference on Automation, Electronics and Electrical Engineering (AUTEEE)*, pages 374–379. IEEE.

[96] Ibarra, D., Ganzarain, J., and Igartua, J. I. (2018). Business model innovation through industry 4.0: A review. *Procedia Manufacturing*, 22:4–10. 11th International Conference Interdisciplinarity in Engineering, INTER-ENG 2017, 5-6 October 2017, Tirgu Mures, Romania.

[97] Jacob, R. J., Raina, S., Regunath, S., Subramanian, R. C., and Gramopadhye, A. K. (2004). Improving inspector's performance and reducing errors-general aviation inspection training systems (gaits). In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, volume 48, pages 203–207. SAGE Publications Sage CA: Los Angeles, CA.

[98] Javaid, M., Haleem, A., Singh, R. P., and Suman, R. (2022). Artificial intelligence applications for industry 4.0: A literature-based study. *Journal of Industrial Integration and Management*, 7(01):83–111.

[99] Jeon, K. E., She, J., Soonsawad, P., and Ng, P. C. (2018). Ble beacons for internet of things applications: Survey, challenges, and opportunities. *IEEE Internet of Things Journal*, 5(2):811–828.

[100] Jiang, H. and Dong, Y. (2016). A nonlinear support vector machine model with hard penalty function based on glowworm swarm optimization for forecasting daily global solar radiation. *Energy Conversion and Management*, 126:991–1002.

[101] Kadir, A. Z. A., Yusof, Y., and Wahab, M. S. (2020). Additive manufacturing cost estimation models—a classification review. *The International Journal of Advanced Manufacturing Technology*, 107(9):4033–4053.

[102] Kamadinata, J. O., Ken, T. L., and Suwa, T. (2019). Sky image-based solar irradiance prediction methodologies using artificial neural networks. *Renewable Energy*, 134:837–845.

[103] Kaplanis, S. and Kaplani, E. (2010). Stochastic prediction of hourly global solar radiation for Patra, Greece. *Applied Energy*, 87(12):3748–3758.

[104] Khaing, M., Fuh, J., and Lu, L. (2001). Direct metal laser sintering for rapid tooling: processing and characterisation of eos parts. *Journal of Materials Processing Technology*, 113(1-3):269–272.

[105] Khashei, M., Bijari, M., and Raissi Ardali, G. A. (2009). Improvement of auto-regressive integrated moving average models using fuzzy logic and artificial neural networks (anns). *Neurocomputing*, 72:956–967.

[106] Kidangan, R. T., Krishnamurthy, C. V., and Balasubramaniam, K. (2020). Detection of dis-bond between honeycomb and composite facesheet of an inner fixed structure bond panel of a jet engine nacelle using infrared thermographic techniques. *Quantitative InfraRed Thermography Journal*, pages 1–15.

[107] Kim, S., Choi, K., Choi, H.-S., Lee, B., and Yoon, S. (2022). Towards a rigorous evaluation of time-series anomaly detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 7194–7201.

[108] Körner, C. (2016). Additive manufacturing of metallic components by selective electron beam melting - A review. *International Materials Reviews*, 61(5):361–377.

[109] Krishnamurthi, R., Kumar, A., Gopinathan, D., Nayyar, A., and Qureshi, B. (2020). An overview of iot sensor data processing, fusion, and analysis techniques. *Sensors*, 20(21):6076.

[110] Kriz, P., Maly, F., and Kozel, T. (2016). Improving indoor localization using bluetooth low energy beacons. *Mobile information systems*, 2016.

[111] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2017). ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90.

[112] Kujawińska, A., Vogt, K., and Hamrol, A. (2016). The role of human motivation in quality inspection of production processes. In *Advances in Ergonomics of Manufacturing: Managing the Enterprise of the Future*, pages 569–579. Springer.

[113] Kusiak, A. (2020). Convolutional and generative adversarial neural networks in manufacturing. *International Journal of Production Research*, 58(5):1594–1604.

[114] Kutscher, V., Olbort, J., Anokhin, O., Bambach, L., and Anderl, R. (2020). Upgrading of legacy systems to cyber-physical systems. *Proceedings of TMCE 2020*.

[115] Lahmiri, S. (2017). Comparing variational and empirical mode decomposition in forecasting day-ahead energy prices. *IEEE Systems Journal*, 11:1907–1910.

[116] Larsson, J. (2015). Distance estimation and positioning based on bluetooth low energy technology.

[117] Latorella, K. A. and Prabhu, P. V. (2017). A review of human error in aviation maintenance and inspection. *Human Error in Aviation*, pages 521–549.

[118] Lawton, G. (2008). Developing software online with platform-as-a-service technology. *Computer*, 41(6):13–15.

[119] Layanun, V., Suksamosorn, S., and Songsiri, J. (2017). Missing-data imputation for solar irradiance forecasting in thailand. *2017 56th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE)*, pages 1234–1239.

[120] LeCun, Y. A., Bottou, L., Orr, G. B., and Müller, K.-R. (2012). Efficient backprop. In *Neural networks: Tricks of the trade*, pages 9–48. Springer.

[121] Li, G. and Jung, J. J. (2022). Deep learning for anomaly detection in multivariate time series: Approaches, applications, and challenges. *Information Fusion*.

[122] Li, J., Ward, J. K., Tong, J., Collins, L., and Platt, G. (2016a). Machine learning for solar irradiance forecasting of photovoltaic system. *Renewable energy*, 90:542–553.

[123] Li, M., Chu, Y., Pedro, H. T., and Coimbra, C. F. (2016b). Quantitative evaluation of the impact of cloud transmittance and cloud velocity on the accuracy of short-term dni forecasts. *Renewable energy*, 86:1362–1371.

[124] Li, N., Huang, S., Zhang, G., Qin, R., Liu, W., Xiong, H., Shi, G., and Blackburn, J. (2019a). Progress in additive manufacturing on new materials: A review. *Journal of Materials Science & Technology*, 35(2):242–269.

[125] Li, S., Da Xu, L., and Zhao, S. (2018). 5g internet of things: A survey. *Journal of Industrial Information Integration*, 10:1–9.

[126] Li, S., Xu, L. D., and Zhao, S. (2015). The internet of things: a survey. *Information systems frontiers*, 17(2):243–259.

[127] Li, Z., Zhang, Z., Shi, J., and Wu, D. (2019b). Prediction of surface roughness in extrusion-based additive manufacturing with machine learning. *Robotics and Computer-Integrated Manufacturing*, 57:488–495.

[128] Lin, C.-C., Deng, D.-J., Liu, W.-Y., and Chen, L. (2017). Peak load shifting in the internet of energy with energy trading among end-users. *IEEE Access*, 5:1967–1976.

[129] Lins, T. and Oliveira, R. A. R. (2020). Cyber-physical production systems retrofitting in context of industry 4.0. *Computers & industrial engineering*, 139:106193.

[130] Liu, F., Tong, J., Mao, J., Bohn, R., Messina, J., Badger, L., Leaf, D., et al. (2011). Nist cloud computing reference architecture. *NIST special publication*, 500(2011):1–28.

[131] Liu, J., Chen, C., Ma, Y., and Xu, Y. (2013). Energy analysis of device discovery for bluetooth low energy. In *2013 IEEE 78th Vehicular Technology Conference (VTC Fall)*, pages 1–5. IEEE.

[132] Lu, Y. (2017). Cyber physical system (cps)-based industry 4.0: A survey. *Journal of Industrial Integration and Management*, 2(03):1750014.

[133] Luca, D. G. and Alberto, M. (2016). Towards accurate indoor localization using ibeacons, fingerprinting and particle filtering. In *2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*.

[134] Luo, X., O'Brien, W. J., and Julien, C. L. (2011). Comparative evaluation of received signal-strength index (rssi) based indoor localization techniques for construction jobsites. *Advanced Engineering Informatics*, 25(2):355–363.

[135] Ma, H., Mao, Z., Feng, W., Yang, Y., Hao, C., Zhou, J., Liu, S., Xie, H., Guo, G., and Liu, Z. (2022). Online in-situ monitoring of melt pool characteristic based on a single high-speed camera in laser powder bed fusion process. *Applied Thermal Engineering*, 211:118515.

[136] Madanchi, A., Absalan, M., Lohmann, G., Anvari, M., and Reza Rahimi Tabar, M. (2017). Strong short-term non-linearity of solar irradiance fluctuations. *Solar Energy*, 144:1–9.

[137] Mahale, T. R. (2009). *Electron Beam Melting of Advanced Materials and Structures, mass customization, mass personalization*. PhD thesis, NC State.

[138] Majumder, I., Dash, P. K., and Bisoi, R. (2018). Variational mode decomposition based low rank robust kernel extreme learning machine for solar irradiation forecasting. *Energy Conversion and Management*, 171:787–806.

[139] Mani, M., Lane, B. M., Donmez, M. A., Feng, S. C., and Moylan, S. P. (2017). A review on measurement science needs for real-time control of additive manufacturing metal powder bed fusion processes. *International Journal of Production Research*, 55(5):1400–1418.

[140] Maresova, P., Sobeslav, V., and Krejcar, O. (2017). Cost–benefit analysis–evaluation model of cloud computing deployment for use in companies. *Applied Economics*, 49(6):521–533.

[141] Mazan, F. and Kovarova, A. (2015). A study of devising neural network based indoor localization using beacons: First results. *Comput. Inf. Syst. J*, 19:15–20.

[142] McNeil, J. L., Sisco, K., Frederick, C., Massey, M., Carver, K., List, F., Qiu, C., Mader, M., Sundarraj, S., and Babu, S. (2020). In-situ monitoring for defect identification in nickel alloy complex geometries fabricated by l-pbf additive manufacturing. *Metallurgical and Materials Transactions A*, 51(12):6528–6545.

[143] Mell, P., Grance, T., et al. (2011). The nist definition of cloud computing. *National Institute of Science and Technology, Special Publication*.

[144] Meng, L., McWilliams, B., Jarosinski, W., Park, H.-Y., Jung, Y.-G., Lee, J., and Zhang, J. (2020). Machine learning in additive manufacturing: a review. *Jom*, 72(6):2363–2377.

[145] Milberg, J. and Sigl, M. (2008). Electron beam sintering of metal powder. *Production Engineering*, 2(2):117–122.

[146] Mirza, M. and Osindero, S. (2014). Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*.

[147] Monjoly, S., André, M., Calif, R., and Soubdhan, T. (2017). Hourly forecasting of global solar radiation based on multiscale decomposition methods: A hybrid approach. *Energy*, 119:288–298.

[148] Montazeri, M., Nassar, A. R., Dunbar, A. J., and Rao, P. (2020). In-process monitoring of porosity in additive manufacturing using optical emission spectroscopy. *IISE Transactions*, 52(5):500–515.

[149] MQTT (Accessed: 2023-01-17). Message queue telemetry transport (mqtt).

[150] Mustafa, W. A. and Yazid, H. (2017). Image enhancement technique on contrast variation: a comprehensive review. *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, 9(3):199–204.

[151] Naik, N. (2017). Choice of effective messaging protocols for iot systems: Mqtt, coap, amqp and http. In *2017 IEEE international systems engineering symposium (ISSE)*, pages 1–7. IEEE.

[152] Narayanan, V. and Bobba, R. B. (2018). Learning based anomaly detection for industrial arm applications. In *Proceedings of the 2018 Workshop on Cyber-Physical Systems Security and PrivaCy*, pages 13–23.

[153] Nazaripouya, H., Wang, B., Wang, Y., Chu, P., Pota, H. R., and Gadh, R. (2016). Univariate time series prediction of solar power using a hybrid wavelet-ARMA-NARX prediction method. In *Proceedings of the IEEE Power Engineering Society Transmission and Distribution Conference*, volume 2016-July, pages 1–5. IEEE.

[154] Newman, N. (2014). Apple ibeacon technology briefing. *Journal of Direct, Data and Digital Marketing Practice*, 15(3):222–225.

[155] Nirjon, S., Liu, J., DeJean, G., Priyantha, B., Jin, Y., and Hart, T. (2014). Coin-gps: Indoor localization from direct gps receiving. In *Proceedings of the 12th annual international conference on Mobile systems, applications, and services*, pages 301–314.

[156] Nocedal, J; Wright, S. J. (2006). *Numerical Optimization*. Springer, Berlin, Germany, 2nd ed. edition.

[157] Palumbo, F., Barsocchi, P., Chessa, S., and Augusto, J. C. (2015). A stigmergic approach to indoor localization using bluetooth low energy beacons. In *2015 12th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–6. IEEE.

[158] Parandoush, P. and Lin, D. (2017). A review on additive manufacturing of polymer-fiber composites. *Composite Structures*, 182:36–53.

[159] Park, J., Kim, J., Kang, S., et al. (2015). Ble-based accurate indoor location tracking for home and office. *Computer Science & Information Technology (CS & IT)*, pages 173–181.

[160] Pease, S. G., Trueman, R., Davies, C., Grosberg, J., Yau, K. H., Kaur, N., Conway, P., and West, A. (2018). An intelligent real-time cyber-physical toolset for energy and process prediction and optimisation in the future industrial internet of things. *Future Generation Computer Systems*, 79:815–829.

[161] Peng, Y., Fan, W., Dong, X., and Zhang, X. (2016). An iterative weighted knn (iw-knn) based indoor localization method in bluetooth low energy (ble) environment. In *2016 Intl IEEE Conferences on Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress (UIC/ATC/ScalCom/CBDCom/IoP/SmartWorld)*, pages 794–800. IEEE.

[162] Peres, R. S., Jia, X., Lee, J., Sun, K., Colombo, A. W., and Barata, J. (2020). Industrial artificial intelligence in industry 4.0-systematic review, challenges and outlook. *IEEE Access*, 8:220121–220139.

[163] Qian, L., Luo, Z., Du, Y., and Guo, L. (2009). Cloud computing: An overview. In *IEEE international conference on cloud computing*, pages 626–631. Springer.

[164] Qin, Y. and Lou, Y. (2019). Hydrological time series anomaly pattern detection based on isolation forest. In *2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, pages 1706–1710. IEEE.

[165] Qing, X. and Niu, Y. (2018). Hourly day-ahead solar irradiance prediction using weather forecasts by lstm. *Energy*, 148:461–468.

[166] Rahman, R. A. and Shah, B. (2016). Security analysis of iot protocols: A focus in coap. In *2016 3rd MEC international conference on big data and smart city (ICBDSC)*, pages 1–7. IEEE.

[167] Rao, P. K., Liu, J. P., Roberson, D., Kong, Z. J., and Williams, C. (2015). Online Real-Time Quality Monitoring in Additive Manufacturing Processes Using Heterogeneous Sensors. *Journal of Manufacturing Science and Engineering*, 137(6).

[168] Rattanalert, B., Jindamaneepon, W., Sengchuai, K., Booranawong, A., and Jindapetch, N. (2015). Problem investigation of min-max method for rssi based indoor localization. In *2015 12th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, pages 1–5. IEEE.

[169] REN21 (2019). Renewables 2019 global status report. Technical report, Paris, REN21 Secretariat.

[170] Rezaeifar, H. and Elbestawi, M. (2021). On-line melt pool temperature control in l-pbf additive manufacturing. *The International Journal of Advanced Manufacturing Technology*, 112(9):2789–2804.

[171] Rida, M. E., Liu, F., Jadi, Y., Algawhari, A. A. A., and Askourih, A. (2015). Indoor location position based on bluetooth signal strength. In *2015 2nd International Conference on Information Science and Control Engineering*, pages 769–773. IEEE.

[172] Rizza, G., Galati, M., and Iuliano, L. (2022). Simulating the sintering of powder particles during the preheating step of Electron Beam Melting process: review, challenges and a proposal. *Procedia CIRP*, 112:388–393.

[173] Rodrigues Júnior, S. E. and Serra, G. L. d. O. (2017). A novel intelligent approach for state space evolving forecasting of seasonal time series. *Engineering Applications of Artificial Intelligence*, 64:272–285.

[174] Rodriguez, E., Medina, F., Espalin, D., Terrazas, C., Muse, D., Henry, C., MacDonald, E., and Wicker, R. B. (2012). Integration of a thermal imaging feedback control system in electron beam melting. In *2012 International Solid Freeform Fabrication Symposium*, pages 945–961. University of Texas at Austin.

[175] Rodríguez-Damián, M., Vila Sobrino, X. A., and Rodríguez-Liñares, L. (2013). Indoor tracking persons using bluetooth: A real experiment with different fingerprinting-based algorithms. In *Ambient Intelligence-Software and Applications*, pages 25–32. Springer.

[176] 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.

[177] Salmi, M., Paloheimo, K.-S., Tuomi, J., Wolff, J., and Mäkitie, A. (2013). Accuracy of medical models made by additive manufacturing (rapid manufacturing). *Journal of Cranio-Maxillofacial Surgery*, 41(7):603–609.

[178] Schmidl, S., Wenig, P., and Papenbrock, T. (2022). Anomaly detection in time series: a comprehensive evaluation. *Proceedings of the VLDB Endowment*, 15(9):1779–1797.

[179] Scime, L. and Beuth, J. (2018). A multi-scale convolutional neural network for autonomous anomaly detection and classification in a laser powder bed fusion additive manufacturing process. *Additive Manufacturing*, 24:273–286.

[180] Seiger, R., Zerbato, F., Burattin, A., García-Bañuelos, L., and Weber, B. (2020). Towards iot-driven process event log generation for conformance checking in smart factories. In *2020 IEEE 24th International Enterprise Distributed Object Computing Workshop (EDOCW)*, pages 20–26. IEEE.

[181] Sendino, S., Martinez, S., Ukar, E., Lamikiz, A., and Gonzalez, J. J. (2022). Monitoring of the L-PBF process to analyze the effect of powder bed temperature on parts roughness. *Procedia CIRP*, 111:355–358.

[182] Shevchik, S. A., Kenel, C., Leinenbach, C., and Wasmer, K. (2018). Acoustic emission for in situ quality monitoring in additive manufacturing using spectral convolutional neural networks. *Additive Manufacturing*, 21:598–604.

[183] Siano, P. (2014). Demand response and smart grids - a survey. *Renewable and Sustainable Energy Reviews*, 30:461–478.

[184] Sigl, M., Lutzmann, S., and Zaeh, M. F. (2006). Transient physical effects in electron beam sintering. *17th Solid Freeform Fabrication Symposium, SFF 2006*, pages 464–477.

[185] Simeone, A., Caggiano, A., and Zeng, Y. (2020). Smart cloud manufacturing platform for resource efficiency improvement of additive manufacturing services. *Procedia CIRP*, 88:387–392.

[186] Singh, A. P. and Pervaiz, S. (2021). Current status and prospects of multi-jet fusion (mjf) based 3d printing technology. In *ASME International Mechanical Engineering Congress and Exposition*, volume 85550, page V02AT02A023. American Society of Mechanical Engineers.

[187] Srivastava, S. and Lessmann, S. (2018). A comparative study of lstm neural networks in forecasting day-ahead global horizontal irradiance with satellite data. *Solar Energy*, 162:232–247.

[188] Su, Y., Zhao, Y., Niu, C., Liu, R., Sun, W., and Pei, D. (2019). Robust anomaly detection for multivariate time series through stochastic recurrent neural network. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2828–2837.

[189] Subhan, F., Hasbullah, H., Rozyyev, A., and Bakhsh, S. T. (2011). Indoor positioning in bluetooth networks using fingerprinting and lateration approach. In *2011 International Conference on Information Science and Applications*, pages 1–9. IEEE.

[190] Sun, H., Gui, D., Yan, B., Liu, Y., Liao, W., Zhu, Y., Lu, C., and Zhao, N. (2016). Assessing the potential of random forest method for estimating solar radiation using air pollution index. *Energy Conversion and Management*, 119:121–129.

[191] Sun, S., Wang, S., Zhang, G., and Zheng, J. (2018). A decomposition-clustering-ensemble learning approach for solar radiation forecasting. *Solar Energy*, 163:189–199.

[192] Tammas-Williams, S., Zhao, H., Léonard, F., Derguti, F., Todd, I., and Prangnell, P. B. (2015). XCT analysis of the influence of melt strategies on defect population in Ti–6Al–4V components manufactured by Selective Electron Beam Melting. *Materials Characterization*, 102:47–61.

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

[194] Tofail, S. A., Koumoulos, E. P., Bandyopadhyay, A., Bose, S., O'Donoghue, L., and Charitidis, C. (2018). Additive manufacturing: scientific and technological challenges, market uptake and opportunities. *Materials today*, 21(1):22–37.

[195] Tukade, T. M. and Banakar, R. (2018). Data transfer protocols in iot—an overview. *Int. J. Pure Appl. Math*, 118(16):121–138.

[196] Tziolas, T., Papageorgiou, K., Theodosiou, T., Papageorgiou, E., Mastos, T., and Papadopoulos, A. (2022). Autoencoders for anomaly detection in an industrial multivariate time series dataset. *Engineering Proceedings*, 18(1):23.

[197] Uhlmann, E., Pontes, R. P., Laghmouchi, A., and Bergmann, A. (2017). Intelligent pattern recognition of a SLM machine process and sensor data Selection". *Procedia CIRP*, 62:464–469.

[198] Vallabh, C. K. P. and Zhao, X. (2022). Melt pool temperature measurement and monitoring during laser powder bed fusion based additive manufacturing via single-camera two-wavelength imaging pyrometry (STWIP). *Journal of Manufacturing Processes*, 79:486–500.

[199] Varshney, V., Goel, R. K., and Qadeer, M. A. (2016). Indoor positioning system using wi-fi & bluetooth low energy technology. In *2016 Thirteenth International Conference on Wireless and Optical Communications Networks (WOCN)*, pages 1–6. IEEE.

[200] Vasisht, D., Kumar, S., and Katabi, D. (2016). {Decimeter-Level} localization with a single {WiFi} access point. In *13th USENIX Symposium on Networked Systems Design and Implementation (NSDI 16)*, pages 165–178.

[201] Voyant, C., Darras, C., Muselli, M., Paoli, C., Nivet, M. L., and Poggi, P. (2014). Bayesian rules and stochastic models for high accuracy prediction of solar radiation. *Applied Energy*, 114:218–226.

[202] Voyant, C., Notton, G., Kalogirou, S., Nivet, M.-L., Paoli, C., Motte, F., and Fouilloy, A. (2017). Machine learning methods for solar radiation forecasting: A review. *Renewable Energy*, 105:569–582.

[203] Wagner, T., Herrmann, C., and Thiede, S. (2017). Industry 4.0 impacts on lean production systems. *Procedia Cirp*, 63:125–131.

[204] Wan, C., Zhao, J., Song, Y., Xu, Z., Lin, J., and Hu, Z. (2015). Photovoltaic and solar power forecasting for smart grid energy management. *CSEE Journal of Power and Energy Systems*, 1(4):38–46.

[205] Wang, C., Tan, X., Tor, S., and Lim, C. (2020a). Machine learning in additive manufacturing: State-of-the-art and perspectives. *Additive Manufacturing*, 36:101538.

[206] Wang, H., Xu, Z., Fujita, H., and Liu, S. (2016). Towards felicitous decision making: An overview on challenges and trends of big data. *Information Sciences*, 367:747–765.

[207] Wang, H., Yi, H., Peng, J., Wang, G., Liu, Y., Jiang, H., and Liu, W. (2017a). Deterministic and probabilistic forecasting of photovoltaic power based on deep convolutional neural network. *Energy conversion and management*, 153:409–422.

[208] Wang, H. z., Li, G. q., Wang, G. b., Peng, J. c., Jiang, H., and Liu, Y. t. (2017b). Deep learning based ensemble approach for probabilistic wind power forecasting. *Applied Energy*, 188:56–70.

[209] Wang, X., Jiang, M., Zhou, Z., Gou, J., and Hui, D. (2017c). 3d printing of polymer matrix composites: A review and prospective. *Composites Part B: Engineering*, 110:442–458.

[210] Wang, Y., Lin, Y., Zhong, R. Y., and Xu, X. (2019). Iot-enabled cloud-based additive manufacturing platform to support rapid product development. *International Journal of Production Research*, 57(12):3975–3991.

[211] Wang, Y., Ye, Q., Cheng, J., and Wang, L. (2015). Rssi-based bluetooth indoor localization. In *2015 11th international conference on mobile ad-hoc and sensor networks (MSN)*, pages 165–171. IEEE.

[212] Wang, Y., Zheng, P., Peng, T., Yang, H., and Zou, J. (2020b). Smart additive manufacturing: current artificial intelligence-enabled methods and future perspectives. *Science China Technological Sciences*, 63(9):1600–1611.

[213] Wang, Z., Li, Z., Zhou, W., and Hui, D. (2018). On the influence of structural defects for honeycomb structure. *Composites Part B: Engineering*, 142:183–192.

[214] Weiwei, H., Wenpeng, J., Haiyan, L., Huiping, T., Xinting, K., and Yu, H. (2011). Research on Preheating of Titanium Alloy Powder in Electron Beam Melting Technology. *Rare Metal Materials and Engineering*, 40(12):2072–2075.

[215] Wittenauer, J. and Norris, B. (1990). Structural honeycomb materials for advanced aerospace designs. *JOM*, 42(3):36–41.

[216] Wohlers, T., (Firm), W. A., Campbell, R., Diegel, O., Kowen, J., Huff, R., Mostow, N., Fidan, I., Bourell, D., Van Rensburg, J., et al. (2022). *Wohlers Report 2022: 3D Printing and Additive Manufacturing Global State of the Industry*. Wohlers Associates.

[217] Wohlers, T. and Gornet, T. (2014). History of additive manufacturing. *Wohlers report*, 24(2014):118.

[218] Wu, H., Yan, W., Xu, Z., Li, S., Cheng, T., and Zhou, X. (2022). Multimodal prediction-based robot abnormal movement identification under variable time-length experiences. *Journal of Intelligent & Robotic Systems*, 104(1):1–15.

[219] Xiyue, Z., Aidin, I., Mojtaba, K., Farhad, I., and Linkan, B. (2021). Automated Anomaly Detection of Laser-Based Additive Manufacturing Using Melt Pool Sparse Representation and Unsupervised Learning. *Solid Freeform Fabrication 2018: Proceedings of the 29th Annual International Solid Freeform Fabrication Symposium – An Additive Manufacturing Conference Reviewed Paper*, pages 376–387.

[220] Yang, L., Lo, L., Ding, S., and Özel, T. (2020). Monitoring and detection of meltpool and spatter regions in laser powder bed fusion of super alloy Inconel 625. *Progress in Additive Manufacturing*, 5(4):367–378.

[221] Ye, D., Hong, G. S., Zhang, Y., Zhu, K., and Fuh, J. Y. H. (2018). Defect detection in selective laser melting technology by acoustic signals with deep belief networks. *The International Journal of Advanced Manufacturing Technology*, 96(5-8):2791–2801.

[222] Yu, J. and Zhou, X. (2020). One-dimensional residual convolutional autoencoder based feature learning for gearbox fault diagnosis. *IEEE Transactions on Industrial Informatics*, 16(10):6347–6358.

[223] Zafari, F., Papapanagiotou, I., and Christidis, K. (2015). Microlocation for internet-of-things-equipped smart buildings. *IEEE Internet of Things Journal*, 3(1):96–112.

[224] Zang, H., Cheng, L., Ding, T., Cheung, K. W., Liang, Z., Wei, Z., and Sun, G. (2018). Hybrid method for short-term photovoltaic power forecasting based on deep convolutional neural network. *IET Generation, Transmission & Distribution*, 12(20):4557–4567.

[225] Zhang, G., Patuwo, B. E., and Hu, M. Y. (1998). Forecasting with artificial neural networks:: The state of the art. *International journal of forecasting*, 14(1):35–62.

[226] Zhang, Y., Fuh, J. Y., Ye, D., and Hong, G. S. (2019). In-situ monitoring of laser-based pbf via off-axis vision and image processing approaches. *Additive Manufacturing*, 25:263–274.

[227] Zhang, Y., Hong, G. S., Ye, D., Zhu, K., and Fuh, J. Y. H. (2018). Extraction and evaluation of melt pool, plume and spatter information for powder-bed fusion AM process monitoring. *Materials & Design*, 156:458–469.

[228] Zhao, H., Wang, Y., Duan, J., Huang, C., Cao, D., Tong, Y., Xu, B., Bai, J., Tong, J., and Zhang, Q. (2020). Multivariate time-series anomaly detection via graph attention network. In *2020 IEEE International Conference on Data Mining (ICDM)*, pages 841–850. IEEE.

[229] Zhu, Z., Anwer, N., Huang, Q., and Mathieu, L. (2018). Machine learning in tolerancing for additive manufacturing. *CIRP Annals*, 67(1):157–160.

Author's Publication List

Below is the complete list of research papers published during the Ph.D. experience. The list is divided by categories and in chronological descending order. Each document provides a brief description of the work carried out, the novelty introduced and the author's contributions.

Journals

[1] **Cannizzaro, D.**, Varrella, A. G., Paradiso, S., Sampieri, R., Chen, Y., Macii, A., Patti, E., & Di Cataldo, S. (2021). In-situ defect detection of metal Additive Manufacturing: an integrated framework. IEEE transactions on emerging topics in computing.

Metal Additive Manufacturing (AM) is a pillar of the Industry 4.0, with many attractive advantages compared to traditional subtractive fabrication technologies. However, there are many quality issues that can be an obstacle for mass production. The in-situ camera-based monitoring and detection of defects, taking advantage of the layer-by-layer nature of the build, can be an effective solution to this problem. In this context, the use of Computer Vision and Machine Learning algorithms have a very important role. Nonetheless, they are up to this date limited by the scarcity of data for the training, as well as by the difficulty of accessing and integrating the AM process data throughout the fabrication. To tackle this problem, this article proposes a system for in-situ monitoring that analyses images from an off-axis camera mounted on top of the machine to detect the arising defects in real-time, with automated generation of synthetic images based on Generative Adversarial Network (GAN) for dataset augmentation purposes. The computing functionalities are embedded into a

holistic distributed AM platform allowing the collection, integration and storage of data at all stages of the AM pipeline.

[2] **Cannizzaro, D.**, Aliberti, A., Bottaccioli, L., Macii, E., Acquaviva, A., & Patti, E. (2021). Solar radiation forecasting based on convolutional neural network and ensemble learning. Expert Systems with Applications, 181, 115167.

Nowadays, we are moving forward to more sustainable energy production systems based on renewable sources. Among all Photovoltaic (PV) systems are spreading in our cities. In this view, new models are needed to forecast Global Horizontal Solar Irradiance (GHI), which strongly influences PV production. For example, this forecast is crucial to develop novel control strategies for smart grid management. In this paper, we present a novel methodology to forecast GHI in short- and long-term time-horizons, i.e. from next 15 min up to next 24 h. It implements machine learning techniques to achieve this purpose. We start from the analysis of a real-world dataset with different meteorological information including GHI, in the form of time-series. Then, we combined Variational Mode Decomposition (VMD) and two Convolutional Neural Networks (CNN) together with Random Forest (RF) or Long Short-Term Memory (LSTM). Finally, we present the experimental results and discuss their accuracy.

[3] **Cannizzaro, D.**, Zafiri, M., Jahier Pagliari, D., Patti, E., Macii, E., Poncino, M., & Acquaviva, A. (2020). A Comparison Analysis of BLE-based Algorithms for Localization in Industrial Environments. Electronics, 9(1), 44.

Proximity beacons are small, low-power devices capable of transmitting information at a limited distance via Bluetooth low energy protocol. These beacons are typically used to broadcast small amounts of location-dependent data (e.g., advertisements) or to detect nearby objects. However, researchers have shown that beacons can also be used for indoor localization converting the received signal strength indication (RSSI) to distance information. In this work, we study the effectiveness of proximity beacons for accurately locating objects within a manufacturing plant by performing extensive experiments in a real industrial environment. To this purpose, we compare localization algorithms based either on trilateration or environment fingerprinting combined with a machine-learning based regressor (k-Nearest Neighbors, Support Vector Machines, or Multi-Layer Perceptron). Each algorithm is analyzed in two different types of industrial environments. For each environment, various configurations are explored, where a configuration is characterized by the number

of beacons per square meter and the density of fingerprint points. In addition, the fingerprinting approach is based on a preliminary site characterization; it may lead to location errors in the presence of environment variations (e.g., movements of large objects). For this reason, the robustness of fingerprinting algorithms against such variations is also assessed. Our results show that fingerprint solutions outperform trilateration, showing also a good resilience to environmental variations. Given the similar error obtained by all three fingerprint approaches, we conclude that k-NN is the preferable algorithm due to its simple deployment and low number of hyper-parameters.

Proceedings in International Conferences

[1] **Cannizzaro, D.**, Simone, F., Illgner-Fehns, K., Mata, S., Mondino, I., Ghiazza, A., Poncino, M., & Di Cataldo, S. (2022, September). Quality inspection of critical aircraft engine components: towards full automation. In 2022 27th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)

The quality of products has become a key factor for success in the current manufacturing industry. This is especially true in the aviation field where components for aircraft engines called honeycombs are produced. Due to their small dimension and peculiar shape, such components typically undergo a severe and cumbersome visual inspection by specialized operators. However, this process is highly prone to human error, requires a lot of time and a high number of undetected defects have been reported. In order to reduce the whole inspection time, ensure higher quality and guarantee standardization and process control, this paper presents an innovative strategy for the fully-automated inspection of honeycomb engine parts. The proposed solution is a two-phase process fully controlled by a robot, leveraging a camera as well as a purposely designed optic fibers sensor, coupled with Artificial Intelligence (AI) algorithms for the detection of different types of defects. To assess the functionality and validity of the proposed solution, a fully functioning prototype is described and characterized.

[2] **Cannizzaro, D.**, Varrella, A. G., Paradiso, S., Sampieri, R., Macii, E., Patti, E., & Di Cataldo, S. (2021, February). Image analytics and machine learning for in-situ defects detection in Additive Manufacturing. In 2021 Design, Automation & Test in Europe Conference & Exhibition (DATE) (pp. 603-608). IEEE.

In the context of Industry 4.0, metal Additive Manufacturing (AM) is considered a promising technology for medical, aerospace and automotive fields. However, the lack of assurance of the quality of the printed parts can be an obstacle for a larger diffusion in industry. To this date, AM is most of the times a trial-and-error process, where the faulty artefacts are detected only after the end of part production. This impacts on the processing time and overall costs of the process. A possible solution to this problem is the in-situ monitoring and detection of defects, taking advantage of the layer-by-layer nature of the build. In this paper, we describe a system for in-situ defects monitoring and detection for metal Powder Bed Fusion (PBF), that leverages an off-axis camera mounted on top of the machine. A set of fully automated algorithms based on Computer Vision and Machine Learning allow the timely detection of a number of powder bed defects and the monitoring of the object's profile for the entire duration of the build.

Book Chapters

[1] **Cannizzaro, D.**, Varrella, A. G., Paradiso, S., Sampieri, R., Macii, E., Poncino, M., Patti, E., & Di Cataldo, S. (2021). In-Situ Monitoring of Additive Manufacturing. In Predictive Maintenance in Smart Factories (pp. 207-228). Springer, Singapore.

Additive Manufacturing, in great part due to its huge advantages in terms of design flexibility and parts customization, can be of major importance in maintenance engineering and it is considered one of the key enablers of Industry 4.0. Nonetheless, major improvements are needed towards having additive manufacturing solutions achieve the quality and repeatability standards required by mass production. In-situ monitoring systems can be extremely beneficial in this regard, as they allow to detect faulty parts at a very early stage and reduce the need for post-process analysis. After providing an overview of Additive Manufacturing and of the state-of-the-art and challenges of in-situ defects monitoring, this chapter describes an in-house developed system for detecting powder bed defects. For that purpose, a low-cost camera has been mounted off-axis on top of the machine under consideration. Moreover, a set of fully automated algorithms for computer vision and machine learning enables allow the timely detection of a number of powder bed defects along with the layer-by-layer monitoring of the part's profile.