Boosting the Artificial Intelligence solutions training phase by means of process simulation methods

Albert Abió Rojo

Boosting the Artificial Intelligence solutions training phase by means of process simulation methods.

Albert Abio

Thesis submitted to Universitat de Barcelona in partial fulfilment of the requirements for the degree of Doctor of Philosophy

UNIVERSITAT DE BARCELONA

eurecat

Boosting the Artificial Intelligence solutions training phase by means of process simulation methods

UNIVERSITAT DE BARCELONA

By

Albert Abio

Departament de Matemàtiques i Informàtica
Universitat de Barcelona

Supervisor:
Dr. Oriol Pujol

Supervisor from Eurecat:
Francesc Bonada

A dissertation submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

Barcelona, January 2025

Abstract

Despite the emergence of Industry 4.0 and the rise of a data-driven manufacturing paradigm, the acquisition of valuable data in a cost-efficient and sustainable manner for manufacturing processes remains a challenge for many companies. Conducting non-productive tests on the production line in an industrial plant results in a waste of raw materials, energy, human resources, and time. Furthermore, executing high-fidelity manufacturing simulations entails a significant temporal and computational burden. Consequently, these drawbacks hinder the creation of knowledge in manufacturing processes and the development of technologies that aim to enhance and influence in the process performance, such as optimization or AI-based tools. This is especially critical for tools that benefit from the availability of large volumes of data and real-time responses, like Digital Twins and Reinforcement Learning agents. Therefore, it is necessary to provide methods that facilitate data generation in industrial environments.

This dissertation is devoted to present a set of general methods to companies and manufacturers to boost the data generation phase in the industrial context. Concretely, we focus on a fast and efficient way to model manufacturing processes through the development of Machine Learning-based Surrogate Models. We propose different general theoretical frameworks implementing or combining machine learning techniques for surrogate modeling applicable in distinct manufacturing process. The thesis demonstrates that the proposed methods enable significant cost and time reductions in different practical manufacturing applications while maintaining high accuracy in modeling and predicting process variables. We investigate the importance of the data chosen to construct the Surrogate Models and the transfer of the knowledge in the Surrogate Models from simulation to real plants by means of Transfer Learning. Overall, this supposes an improvement of the presented surrogate modeling methods and it facilitates the deployment of Surrogate Models in real-world industrial plants. The developed models during the thesis are a valuable asset in other studies, acting as a virtual environment to train Reinforcement Learning agents in hot stamping or supporting a Digital Twin of the high pressure die casting process. The thesis helps to advance towards the innovation of data-driven manufacturing by providing practical and efficient solutions in the direction of a better understanding of the manufacturing processes, leading to an enhancement in their performance and sustainability.

Keywords: *Surrogate Models, Machine Learning, Artificial Intelligence, Industry 4.0, Hot Stamping, High Pressure Die Casting, Plastic Injection Moulding*

Resum

Malgrat l'aparició recent de la Indústria 4.0 i l'auge d'un paradigma industrial basat en la fabricació en dades, l'adquisició de dades de valor de manera efficient en costos i sostenible en processos de fabricació continua sent un repte per a moltes empreses. La realització de proves no productives a la línia de producció d'una planta industrial comporta un malbaratament de matèries primeres, energia, recursos humans i temps. A més, executar simulacions de fabricació d'alta fidelitat suposa un cost temporal i computacional important. En conseqüència, aquestes limitactions dificulten la creació de coneixement en els processos de fabricació i el desenvolupament de tecnologies que tenen com a objectiu millorar i influir en el rendiment del procés, com l'optimització o eines basades en IA. Això és especialment crític per a eines que es beneficien de la disponibilitat de grans volums de dades i respostes en temps real, com ara els bessons digitals i els agents d'aprenentatge per reforç. Per tant, és necessari proporcionar mètodes que facilitin la generació de dades en entorns industrials.

Aquesta tesi està dedicada a presentar un conjunt de mètodes generals a empreses i fabricants per impulsar la fase de generació de dades en el context industrial. Concretament, ens centrem modelar els processos de fabricació d'una manera una manera ràpida i eficient mitjançant el desenvolupament de models substituts basats en l'aprenentatge automàtic. Proposem diferents marcs teòrics generals que implementen o combinen tècniques d'aprenentatge automàtic per al modelatge substitut aplicables en diferents processos de fabricació. La tesi demostra que els mètodes proposats permeten reduccions significatives de costos i temps en diferents aplicacions pràctiques de fabricació, mantenint una alta precisió en la modelització i predicció de variables de procés. Investiguem la importància de les dades escollides per construir els models substituts i la transferència del coneixement en els models substituts des de la simulació a les plantes reals a través del Transfer Learning. En conjunt, això suposa una millora dels mètodes de modelització substituts presentats i facilita el desplegament de models substitutius en plantes industrials reals. Els models desenvolupats durant la tesi són un actiu valuós en altres estudis, actuant com un entorn virtual per formar agents d'aprenentatge de reforç en l'estampació en calent o donant suport a un bessó digital del process injecció d'alumini fos a alta pressió. La tesi ajuda a avançar cap a la innovació de la fabricació basada en dades aportant solucions pràctiques i eficients en la direcció d'una millor comprensió dels processos de fabricació, permetent una millorara en el seu rendiment i la sostenibilitat.

Paraules Clau: *Models Substituts, Aprenentatge Automàtic, Intel·ligència Artificial, Indústria 4.0, Estampació en Calent, Fosa a Pressió, Injecció de Plàstic*

Acknowledgements

Firstly, I would like to express my gratitude to Dr. Oriol Pujol for his academic guidance and supporting me during all the months of work. His willingness to maintain a constant communication and provide crucial advice was instrumental in the development of this thesis. Seemingly, I would like to express my sincere gratitude to Francesc Bonada from Eurecat for his exceptional mentorship. His expertise, encouragement and dedication were essential to my success, and I am truly grateful for the time and effort he invested in guiding me.

I am also indebted to my colleagues at Eurecat, especially the ones who actively contributed in the publications derived from this thesis, for their helpful discussions, conversations and for creating a kind and stimulating work environment.

I would like to extend my sincere thanks to Dr. Daniel Casellas for his advice and support from Eurecat and for the opportunities he provided, which enriched my experience as a doctorate student. Also, I am very grateful to Dr. Jörgen Kajberg at Luleå University of Technology for making my research stay possible. I also extend my thanks to the entire Solid Mechanics Department at LTU for their generosity and assistance, which greatly enhanced my time there.

And last but not least, als meus pares, a la Lali, a les meves germanes i als meus amics, perquè sense el seu acompanyament i amor, no hauria arribat fins aquí.

Funding

Albert Abio was a fellow of Eurecat Vicente López PhD Grant Program.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

Abio Rojo, Albert

Date: 29/01/25

Contents

List of Figures

List of Tables

List of Abbreviations

Abbreviation	Definition
AI	Artificial Intelligence
AR	Augmented Reality
AdSam	Adaptive Sampling
CAD	Computer-Aided Design
CFD	Computational Fluid Dynamics
CNN	Convolutional Neural Network
CPS	Cyber-Physical System
CV	Cross-Validation
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
DES	Discrete Element Simulation
DNN	Deep Neural Network
DT	Decision Tree
FE	Finite Element
GAN	Generative Adversarial Network
GB	Gradient Boosting
GP	Gaussian Process
GPT	Generative Pre-trained Transformer
GRU	Gated Recurrent Unit
HFM	High-Fidelity Model
HPDC	High Pressure Die Casting
I4.0	Industry 4.0
I5.0	Industry 5.0
IoT	Internet of Things
KNN	K-Nearest Neighbors
KPI	Key Performance Indicator
LFM	Low-Fidelity Model
LLM	Large Language Model
LOF	Local Outlier Factor
LR	Linear Regression
LSTM	Long Short-Term Memory
MAE	Mean Absolute Error
MDP	Markov Decision Process
ML	Machine Learning

Continued on next page

Table 1 – *Continued from previous page*

MLP	Multilayer Perceptron
MSE	Mean Squared Error
NN	Neural Network
OEE	Overall Equipment Effectiveness
PCA	Principal Component Analysis
R&D	Research and Development
R^2	R-squared
RAG	Retrieval-Augmented Generation
RF	Random Forest
RL	Reinforcement Learning
RNN	Recurrent Neural Network
RUL	Remaining Useful Life
SL	Supervised Learning
SMod	Surrogate Model
SVM	Support Vector Machine
SVR	Support Vector Regressor
SaaS	Software as a Service
TL	Transfer Learning
UHSS	Ultra High Strength Steel
UL	Unsupervised Learning
VFM	Variable Fidelity Modeling
VR	Virtual Reality
XGBoost	Extreme Gradient Boosting

INTRODUCTION

Chapter 1

Introduction

1.1 Context

Industry 4.0 (I4.0) is understood as the introduction of a set of emerging technologies that have completely changed the manufacturing paradigm. Internet of Things (IoT), Artificial Intelligence (AI), Big Data, Cloud Computing, Augmented Reality (AR), robotics, among others, have been established as the fundamental technological pillars of I4.0. In the past few years, despite the costs that are required to update the old factories and processes, the traditional manufacturing has identified huge gain in the incorporation of these advanced innovations, which has supposed the evolution to this new industrial paradigm. The full potential of I4.0 lies in the combination and integration of several of the mentioned technologies, which has triggered a complete change in the way companies manufacture, improve and distribute their products. Thanks to the concept of a smart and digital manufacturing, the fabrication of high-quality products is enabled, with the advantage of an increase in the productivity, flexibility and a reduction of the economical and environmental costs in the overall manufacturing chain. To sum up, I4.0 framework provides not only a massive positive impact in the companies but also global benefits in the society.

The establishment of I4.0 has contributed in the identification of *data* as one of the most important assets in manufacturing. In this field, AI-based tools excel in the exploitation of data and can be applied in industrial tasks like process monitoring, fault detection, computer vision, decision-making or autonomous control. Among the branches of AI, Machine Learning (ML) has become a strong trend in the last few years and its implementation has reached many fields. Focusing on manufacturing, there has been an increase of the application of ML. In fact, there is a correlation between I4.0 paradigm shift and ML growth in popularity, as it is represented in Figure 1.1. On the one hand, I4.0 has provided the data availability, the computational resources and the innovation environment that has accelerated the development of ML algorithms and applications. On the other hand, the great capabilities of ML to handle high-dimensionality data, its ability recognizing highly non-linear patterns, extract valuable information from data or its adaptability to changes have enabled the implementation of ML solutions in several industrial problems. In this sense, ML has a significant impact in the I4.0 objectives regarding efficiency, productivity and sustainability. Thus, I4.0 and ML are two intertwined concepts, and the evolution of each one has driven the improvement of the other.

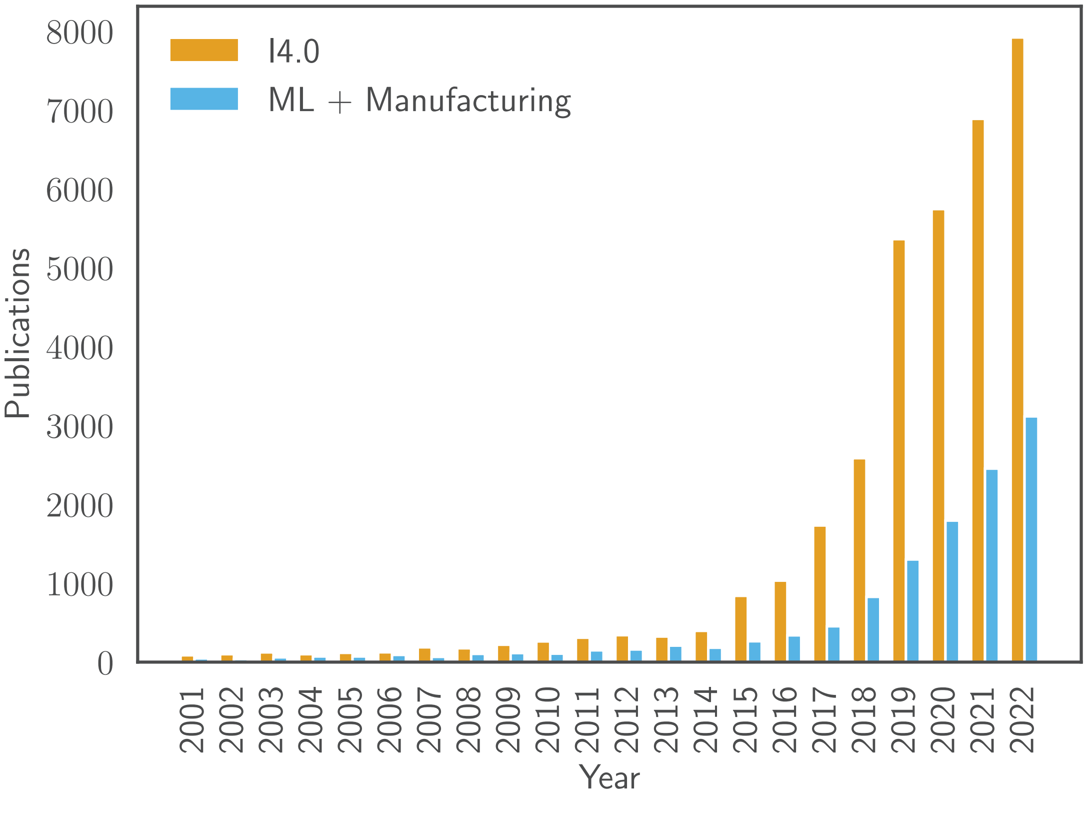


Figure 1.1: Number of publications related with the keywords "I4.0" and "ML + manufacturing" since 2001. Data obtained from Clarivate *Web of Science*. © Copyright Clarivate 2023. All rights reserved. (`https://www.webofscience.com/`)

However, the progress and successful implementation of AI data-driven solutions such as ML have limitations. Despite big companies have made important investments in I4.0 technologies, a majority of the small and medium sized companies have struggled in deploying sensors, robots or software due to its elevated economical cost. Also, these advanced technologies must be installed in the production plant in smaller companies, while big enterprises have testing environments and R&D departments. Consequently, the improvements in the data accessibility from industrial plants may not always be translated in a better understanding of the process or to develop data-driven solutions. In many cases, data acquired from processes performed in industrial environments are centered in production and the configuration parameters are not modified to avoid production disruptions in the machinery and the tools. The few changes in the configurations are a limiting factor for the knowledge inference and the potential of data-driven techniques. A more extensive exploration of the parameter space is needed in order to capture and comprehend the behavior of the system and the undelying physical phenomena of the manufacturing processes. The performance of tests to this purpose supposes a significant expense in human resources, raw material, time and the use of the production plant to non fruitful operations.

Historically, mathematical models of the manufacturing systems and the use of computer simulation methods have been the most common approach to acquire knowledge of this type of systems. This opens the possibility to the experimentation and the validation of the design and the process, providing insights and knowledge about the system without perturbing it. Thanks to its high fidelity, manufacturing simulation models act as alternative environments to explore new configurations and to generate data for AI-based solutions. Nevertheless, their detailed representation of the system implies high computational demands and an elevated time cost, leading to a very slow response. In addition, the mentioned drawbacks added to the problem of the curse of dimensionality in the exploration of process parameters often make the data generation process unfeasible. Thus, the exploration of a large parameter space with a simulation model is not sustainable in terms of time and computational resources. Finally, the adaptation to new scenarios demands new designs and calibrations, also supposing an additional time cost.

Therefore, as showcased in Figure 1.2, the development of AI-based tools for I4.0 is inherently linked to overcoming the challenges associated with data acquisition in industrial environments. While real-world data provides the most accurate representation of the system, it is often limited by practical constraints and cost. Simulation data offers a valuable alternative, allowing for broader exploration of the parameter space, but faces limitations in terms of computational resources and time. Successfully leveraging AI in I4.0 requires a strategic approach that balances the strengths and weaknesses of both data sources.

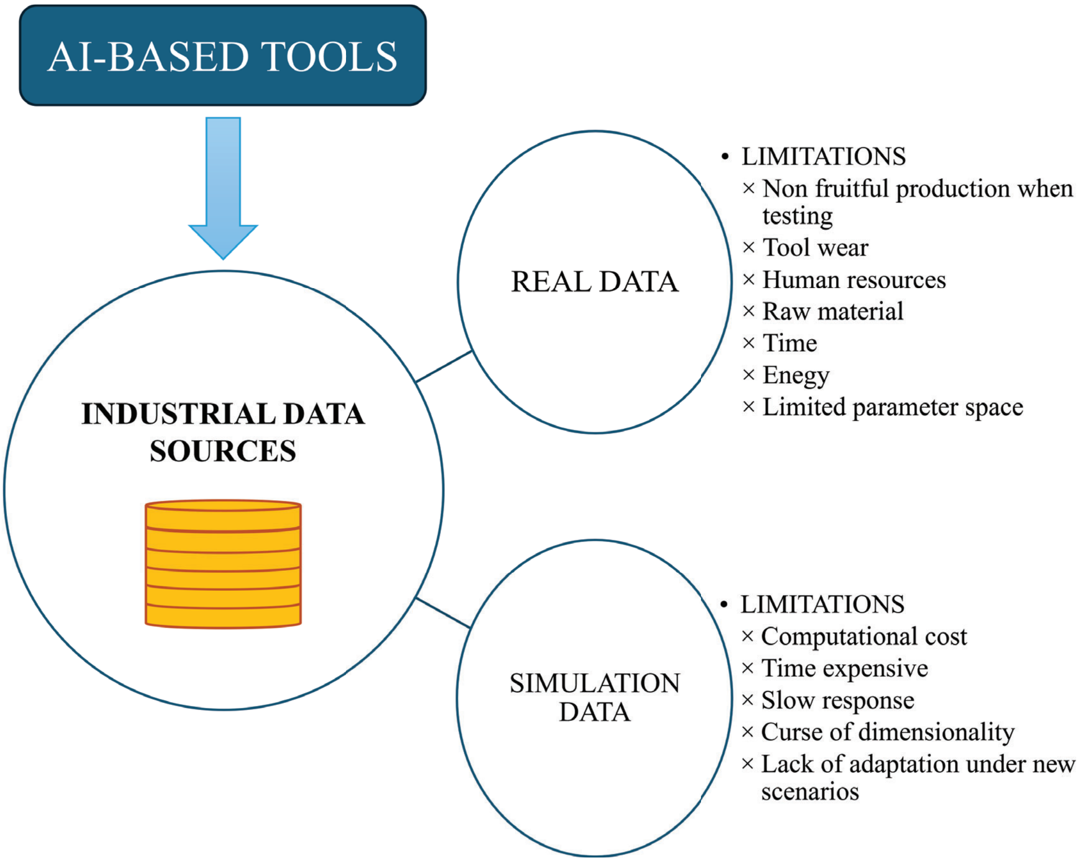


Figure 1.2: Main industrial data sources required for the development of AI-based tools and their principal limitations.

Some of the most promising and innovative AI-based technologies given by I4.0 are the Reinforcement Learning (RL) application in autonomous control agents and the complete virtual representation of the manufacturing system offered by the Digital Twin (DT) concept. RL is a type of ML algorithm based in an autonomous learning method in which an agent learns to make decision by interacting with an environment. With respect to DTs, they are representations of physical systems in a virtual environment, which are completely connected through the exchange of data in real-time. The transferred information enhances the knowledge of the operating status and can optimize the real-world performance. These technologies highlight the need to enhance the data generation in manufacturing processes, since they demand a large volume of data and a fast response models. Otherwise, the training phase of RL is not achievable unless a lot of time and resources are spent. Furthermore, the real-time process monitoring is not possible in

DTs, which are constraint to a limited time scale larger than the process time.

To sum up, numerous limitations exist to gather and exploit data in industrial and simulation environments: First, the elevated cost of the investment in the data acquisition and measuring equipment and the waste of production time and primary resources in industrial plant. Secondly, the computational and temporal cost and the unfeasibility of obtaining real-time responses from simulations. In this scenario, several questions arise: *How can we reduce the temporal and economical cost of data acquisition in industrial plants? How can the drawbacks of simulations regarding time be mitigated? Which are the constraints to deploy AI-based I4.0 tools like RL and DTs in industrial plants? How can they be mitigated?* ***How can we model a manufacturing process in a fast, sustainable and effective way?*** *How will we adapt the model to the desired environments?* ***How can we boost the data generation to obtain manufacturing data with a real-time response?*** For answering some of these and other questions, the present thesis is centered on a set of general methods to create fast and efficient ML-based metamodels known as Surrogate Models (SMods) applicable in different manufacturing use cases. This will allow to the manufacturing community to diminish the economical and temporal impact in the data generation process and the cost to implement avant-garde AI-based tools.

1.2 Objectives

The improvement of the productivity, efficiency and sustainability in manufacturing is a current high-interest topic within the I4.0 framework. The emergence of data-driven tools have resulted in a drift of the manufacturing paradigm towards data-driven systems and I4.0. Nevertheless, the most promising I4.0 technologies and AI-based tools that enable these upgrades have some requirements: large volume of representative data and fast data generation. However, the present data acquisition in industrial plants implies performing tests that may perturb the production line, leading to costly operations that suppose an important consumption of energy, human resources, raw material and time. Another suitable way to generate the required amount of data could be simulation, but they require an elevated computational and temporal cost. Moreover, they have difficulties in the design and calibration phase in the adaptation to new scenarios.

The main objective of the present work is to provide to companies and manufacturers a set of generals methods to boost their data generation process through the construction of Surrogate Models (SMods) that can model manufacturing systems in a fast and efficient way. This will enable the implementation of advanced AI-based industrial technologies and, at the same time, it will supply a method for high-speed predictions in manufacturing environments, accelerating the knowledge of the manufacturing system performance under different conditions and scenarios and thus providing a novel tool for flexible manufacturing in changing environments. In practice, the time impact of the manufacturing simulations and the cost of acquiring data from industrial plants will be diminished. In this thesis, the surrogate modeling methods will be based on ML algorithms and related techniques. We specially focus on the practical application of the SMods in industry, which are discussed in different levels of detail through the implementation of general methods in specific manufacturing use cases. Thus, the thesis objectives are three-fold:

- Provide a comprehensive definition of Surrogate Model in the industrial field, highlighting ML-based SMods as the most relevant tendency and identifying the potential advantages of implementing SMods in manufacturing problems.

- Develop general frameworks to create baseline ML-based SMods in industrial environments and validate them in real use cases, remarking the gains by SMod for each particular manufacturing process.

- Present potential improvements that enhance the capabilities the baseline ML-based SMods to build a complete general optimized procedure for surrogate modeling in manufacturing and validate it in the manufacturing process of hot stamping.

1.3 Contributions

In this thesis, we present some contributions to the acceleration of the data generation phase in manufacturing processes, through the reduction of the cost of the simulation methods by the implementation of data-driven ML-based SMods. The main contribution is a set of tools to construct these SMods, validated in different practical manufacturing use cases. Thus, the main contributions can be grouped into three:

- A comprehensive description about the potential role of the SMods in boosting the data generation process in manufacturing. We define an application framework where ML-based SMods are crucial to improve the efficiency of knowledge acquisition. Moreover, it is detailed how ML-based SMods provide ideal fast response environments that can be employed to ease the development of other optimization procedures or advanced data-driven tools for process control and monitoring, like DTs or RL agents.

- A set of methods to build ML-based SMods in three real-world industrial processes:

  - For plastic injection moulding, a node reduction method is applied. We construct a ML-based SMod employing Random Forest combined with a backward selection method. This method leads to cost savings and simplified system design by minimizing the number of simulated sensors required to generate accurate predictions in the whole geometry.
  - In the context of high pressure die casting , a mesh upscaling method is proposed. We utilize Random Forest to create a ML-based SMod that predicts fine mesh simulation results using coarse mesh simulations. This approach reduces computational time and enables a faster analysis.
  - For the hot stamping process, a parameter interpolation method is introduced. We generate a ML-based SMod using XGBoost that efficiently predict simulation results across a wide range of parameter values. This method allows rapid exploration of the design space and efficient data generation, facilitating the development of optimized hot stamping processes.

  These approaches enable enhanced efficiency in data generation through three distinct surrogate modeling techniques. The benefits are quantified through accuracy and temporal metrics compared to benchmarking high-fidelity data. Importantly, the proposed surrogate modeling methods demonstrate potential for generalization to other manufacturing systems by addressing scenarios and simulation models that are representative and typically encountered in real-world industrial challenges.

- An extensive study of surrogate modeling for the hot stamping process, using Deep Neural Networks as the baseline ML algorithm. Continuous improvements of the surrogate modeling method for this process are discussed, overcoming several limitations. Concretely, we introduce different strategies for these improvements:

- Importance of the sampling. We study the influence of different sampling techniques in the ML-based SMods of the hot stamping process. This includes a quantitative analysis of the impact in accuracy and efficiency of the different methods and a comparative study of their performance.
- Transfer Learning for efficient real plant modeling. We develop a transfer learning-based methodology for constructing accurate and efficient ML-based SMods in real industrial hot stamping plants. This method effectively addresses the sim-to-real gap by combining low-fidelity simulation data with limited high-fidelity experimental data, significantly reducing the need for extensive experimental testing.

1.4 Thesis Outline

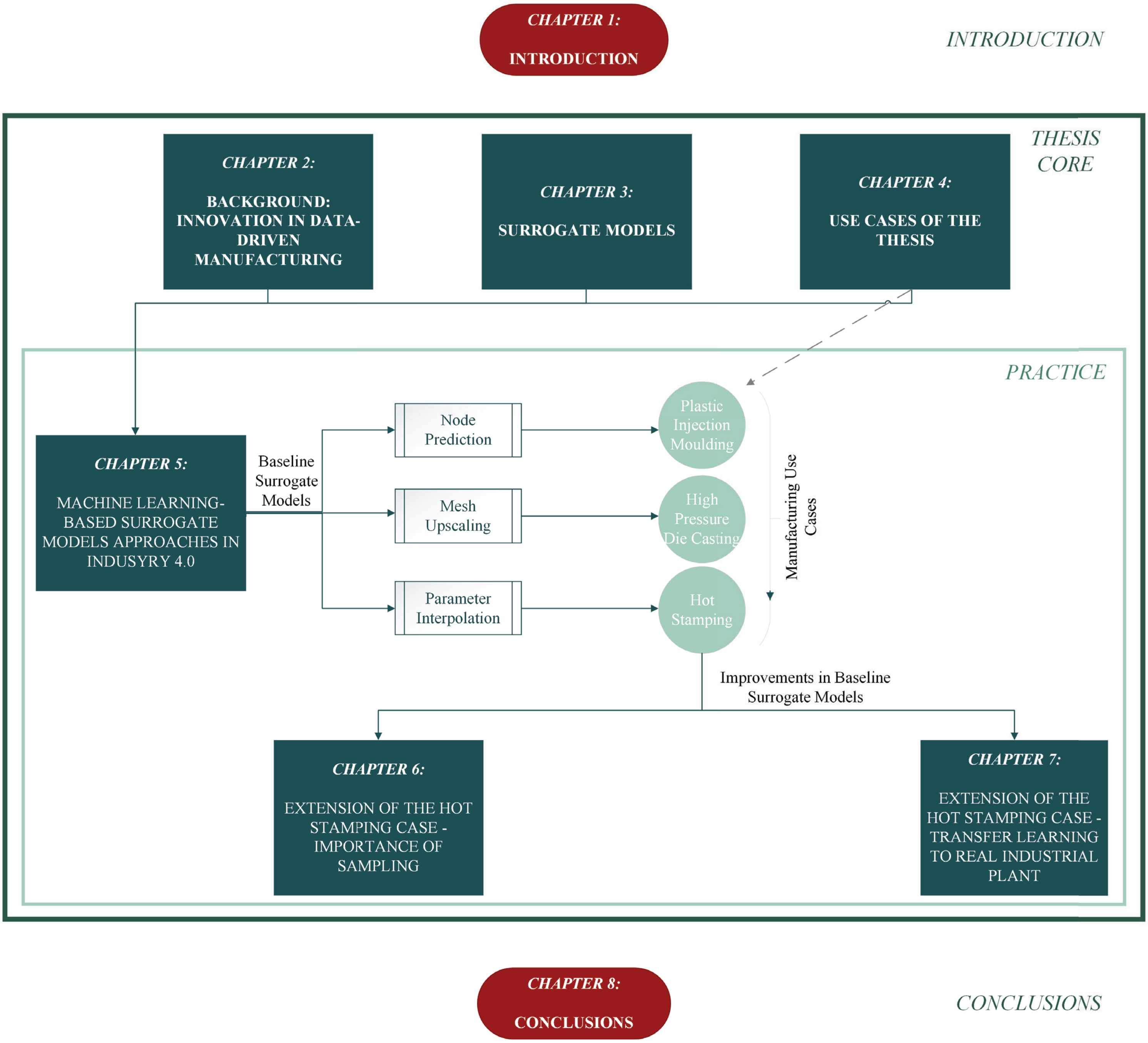

CHAPTER 1:
INTRODUCTION
INTRODUCTION
THESIS CORE
CHAPTER 2:
BACKGROUND: INNOVATION IN DATA-DRIVEN MANUFACTURING
CHAPTER 3:
SURROGATE MODELS
CHAPTER 4:
USE CASES OF THE THESIS
PRACTICE
CHAPTER 5:
MACHINE LEARNING-BASED SURROGATE MODELS APPROACHES IN INDUSYRY 4.0
Baseline Surrogate Models
Node Prediction
Mesh Upscaling
Parameter Interpolation
Plastic Injection Moulding
High Pressure Die Casting
Hot Stamping
Manufacturing Use Cases
Improvements in Baseline Surrogate Models
CHAPTER 6:
EXTENSION OF THE HOT STAMPING CASE - IMPORTANCE OF SAMPLING
CHAPTER 7:
EXTENSION OF THE HOT STAMPING CASE - TRANSFER LEARNING TO REAL INDUSTRIAL PLANT
CHAPTER 8:
CONCLUSIONS
CONCLUSIONS

Figure 1.3: Diagram of the thesis outline. It shows the relations between the chapters and the differentiation between the introduction and conclusion parts from the thesis core, and the practice part within the thesis.

The general outline for this dissertation is shown in Figure 1.3. This Industrial PhD thesis focuses on generating new knowledge with practical applications for industry. It aims to address real-world challenges by transferring research findings to relevant stakeholders. Therefore, the thesis is separated in parts, which contain several chapters. The order of the chapters tries to be a journey from general to specific. In the following, the chapters of the thesis are summarized, indicating their key contributions. Note that the chapters are distributed in a logical order with the aim of helping the reader to delve into the distinct contributions in a structured manner.

INTRODUCTION

- **Chapter 1: Introduction**. The current initial chapter aims to introduce the thesis. The readers are situated in the context, the main objectives are presented, along with the contributions and the thesis outline.

THESIS CORE

- **Chapter 2: Background: Innovation in Data-Driven Manufacturing**. The first part of the core of this thesis serves to explain the context and to define key concepts appearing in the thesis for the readers. In the first section, the current manufacturing paradigm of I4.0 is introduced, highlighting data as one of the key resources and the benefits of the use of data-driven tools in manufacturing. Concretely, a review of ML applications in manufacturing is provided and RL and DTs are presented and identified as key technologies to improve and optimize manufacturing processes, along with their implementation challenges in industrial environments. To sum up, the chapter presents the challenges that this thesis aims to address.

- **Chapter 3: Surrogate Models**. This chapter provides a definition of the main topic of the thesis, the concept of SMod, from a manufacturing point of view. A justification about the importance of SMods is discussed, with different proposals about the possible roles of SMods in the interaction with AI-based tools. A description of the existing surrogate modeling methods and applications is presented, while ML-based SMods are proposed to be applied in this thesis.

- **Chapter 4: Use Cases of the Thesis**. In this chapter, the manufacturing processes used during the thesis are described in an informative way.

PRACTICE

- **Chapter 5: Machine Learning-based Surrogate Models Approaches in Industry 4.0**. This part of the thesis proposes three different practical frameworks for surrogate modeling in manufacturing use cases. Particularly, each of the proposed approaches is applied to a specific manufacturing use case. This results in three different baseline SMods for the simulations of the processes of plastic injection moulding, high pressure die casting and hot stamping, respectively. Despite all these methods aim to reduce the computational burden from manufacturing simulations, the surrogate modeling approach varies depending on the defined industrial goal.

- **Chapter 6: Extension of the Hot Stamping Case - Importance of Sampling**. In this chapter, the importance of the sampling in surrogate modeling is remarked extending the approach presented in the previous chapter for surrogate modeling in the hot stamping process. The influence of the sampling in the creation of the SMod is quantified and a comparison in accuracy and time between different sampling methods is performed.

- **Chapter 7: Extension of the Hot Stamping Case - Transfer Learning to Real Industrial Plant**. This chapter is also considered an extension to the baseline SMod of the hot stamping process. In particular, it addresses the problem of implementing a SMod in a real industrial plant. A method to generate an efficient and effective SMod of a real hot stamping plant overcoming the sim-to-real gap problem. This method consists in the transfer learning concept, and its key concept lies in using low-fidelity simulation data and few real high-fidelity data to obtain a reliable SMod reducing a lot the number of experimental tests.

CONCLUSIONS

- **Chapter 8: Conclusions**. Finally, this chapter concludes the work, listing the academic and industrial contributions that have arisen from this dissertation and pointing out the future research lines that can be explored from this thesis.

1.5 Publications Derived from the Thesis

Most of the contributions and results explained in this thesis have been published in academic conferences and journals and developed within industrial research projects.

The following is a list of publications directly related to the thesis content, where the thesis author is the first author:

- Abio, A. *et al.* in. Chap. Combining Simulations and Machine Learning for Efficient Prediction of Process Parameters Evolution in Injection Moulding (IOP Press, 2021). doi:`10.3233/FAIA210135`
- Abio, A. *et al.* Machine Learning-Based Surrogate Model for Press Hardening Process of 22MnB5 Sheet Steel Simulation in Industry 4.0. *Materials* **15**.10. doi:`10.3390/ma15103647` (2022)
- Abio, A. *et al.* *Machine Learning Surrogate Model for Sensitivity Analysis in Hot Stamping* in *CHS^2 2024 Proceedings* (AIST, 2024), 21–27. ISBN: 978-0-930767-24-2. doi:`10.33313/512/A0201`

  An extended version of this publication is in progress to be submitted:

  Abio, A. *et al.* Study of Sampling Importance in Machine Learning Surrogate Model for Sensitivity Analysis in Hot Stamping. *Metals*. Manuscript in preparation.
- Abio, A. *et al.* A transfer learning method in press hardening surrogate modeling: From simulations to real-world. *Journal of Manufacturing Systems* **77,** 320–340. doi:`10.1016/j.jmsy.2024.09.012` (2024)

Additionally, here are publications that utilize some of the models presented in this thesis, where the thesis author is not the first author:

- Torres, P. *et al.* Millora de l'eficiència i la fiabilitat de sistemes industrials a través de la implementació de Digital Twins. *CIDAI-POC-2023-02.* `https://cdn.eurecat.org/PDF/CIDAI/PoC/POC-EURECAT-110523.pdf` (2023)
- Nievas, N. *et al.* A Reinforcement Learning Control in Hot Stamping for Cycle Time Optimization. *Materials* **15**.14. doi:`10.3390/ma15144825` (2022)
- Nievas, N. *et al.* A scalable Deep Q-Learning approach for hot stamping process under dynamic control environment. *International Journal of Production Research* **0**.0, 1–22. doi:`10.1080/00207543.2024.2411377` (2024)

- Nievas, N. *et al.* Offline Reinforcement Learning for Adaptive Control in Manufacturing Processes: A Press Hardening Case Study. *Journal of Computing and Information Science in Engineering* **25.** doi:10.1115/1.4066999 (1 2025)

Finally, there are some works that have been done during the duration of the doctorate programme, but that have not been included in this dissertation.

- Abio, A. *et al. Graph Neural Network-Based Surrogate Model of Hot Stamping Finite-Element Simulations* Conference Poster session. 2024. https://phantomsfoundation.com/AI4AM/2024/Abstracts/AI4AM2024_Abio_Albert_88.pdf. Conference Artificial Intelligence for Advanced Materials (AI4AM), Barcelona, Spain.

THESIS CORE

Chapter 2

Background: Innovation in Data-Driven Manufacturing

This part of the thesis presents an overview of current trends in manufacturing innovation and describes the manufacturing processes employed, providing a brief overview of key concepts that will be referenced throughout the next chapters. The aim is to ensure clarity and accessibility for readers who may not be familiar with certain specialized terminology or theoretical frameworks. Firstly, we provide an introduction about the idea of the Industry 4.0 (I4.0) and Industry 5.0 (I5.0) paradigms connected to data-driven manufacturing, which comprehends the application of Machine Learning in industrial framework, with a special focus on the Supervised Learning (SL) solutions. Next, we briefly describe the emergence of tools to improve manufacturing, such the use of Reinforcement Learning (RL) for autonomous control or the creation of virtual replicas of the manufacturing system through the concept of Digital Twin (DT), highlighting the principal challenges regarding their implementation in real-world industrial scenarios.

2.1 Industry 4.0 and Industry 5.0 Frameworks

Since its origins, manufacturing has passed through a lot of periods and paradigm shifts due to the need of the companies to remain competitive. In general, the implementation of new technology solutions is usually one of the main drivers of the paradigm shifts [10, 11]. Thanks to these solutions, the performance of the manufacturing systems is enhanced, with an increase of the quality of the products, lower operational costs and better decision making, which can be measured with the key performance indicators (KPIs) [12]. For instance, during the last decades, we have experienced the stages of agent-based manufacturing (2000s) [13] and cloud manufacturing (2010s) [14].

As mentioned in the introduction, the Fourth Industrial Revolution has transformed the manufacturing industry, leading to a digitalization process with the objective of complete connection between manufacturing elements [15, 16]. This new understanding is known as I4.0 or Smart Manufacturing and it is supported by a set of technological tools, the so-called pillars of I4.0 [17–19]. Figure 2.1 displays a representation of some of these fundamental pillars, and a brief explanation is given:

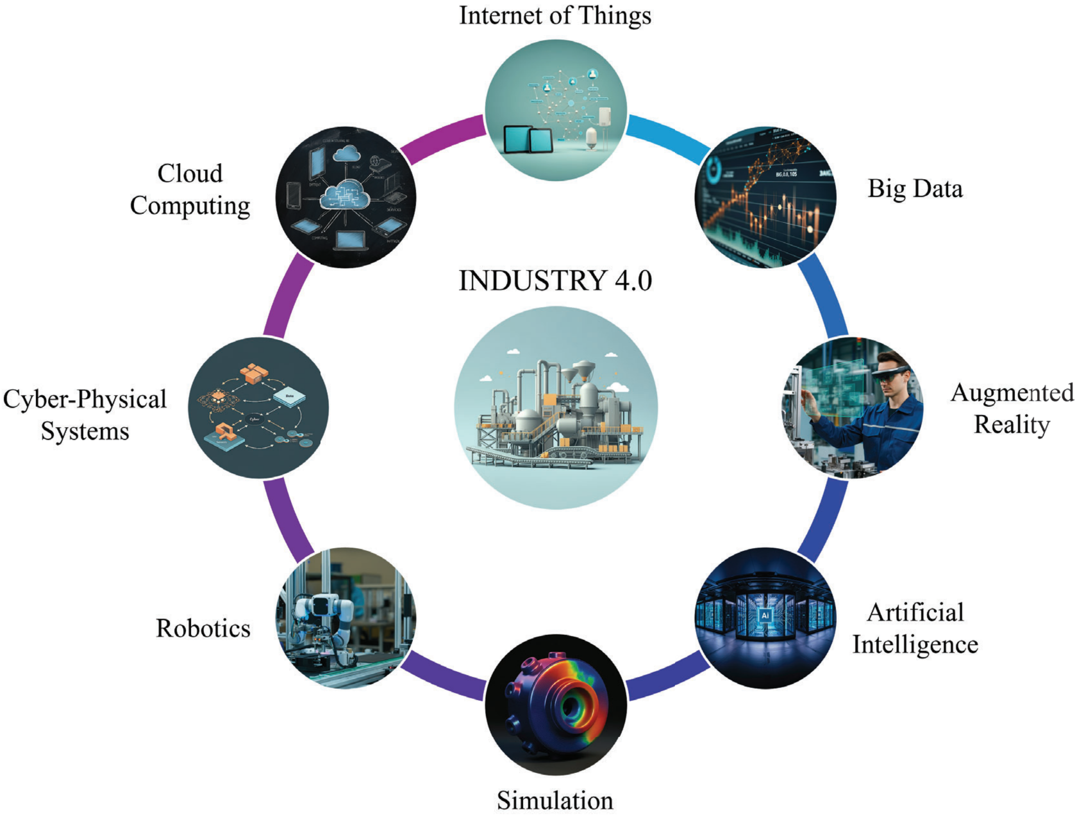


Figure 2.1: Fundamental pillars of the I4.0 paradigm.

- Internet of Things (IoT): Network of interconnected devices and physical assets that collect data and the services that enable the communication and exchange between them and the cloud [20, 21].

- Cloud Computing: On-demand supply of scalable and flexible computing services that provide efficient storage, servers, software or databases in the cloud according to the manufacturers needs [22].

- Artificial Intelligence (AI): Development of models or systems to realize tasks or actions usually related with human intelligence, including learning from data, reasoning and self-improvement [23, 24].

- Big Data: Large amounts of data that are generated at high velocity from various sources and with different structure are collected, processed and analysed [25].

- Simulation: Powerful tools and models able to provide a digital representation of the complex manufacturing systems, helping in the design, evaluation and optimization stage without the risks and costs of altering the real manufacturing system [26].

- Robotics: Industrial robots as autonomous agents in the performance of operations, detection of problems and reduction of repetitive and dangerous tasks [27].

- Cyber-Physical Systems (CPS): Integration of computational elements into a physical entity allowing real-time data acquisition and transfer, inducing a potential dynamic control and monitoring of the physical part from the digital perspective [28].

- Augmented Reality (AR) and Virtual Reality (VR): Virtual environments that combine real and digital information to enhance the interaction experience assisting and facilitating some manufacturing jobs [29, 30].

The integration, interaction and cooperation of the technological pillars assets is the core of I4.0 and it results in a collaborative manufacturing system that constitutes the concept of I4.0 Factory or Smart Factory [31, 32]. The I4.0 Factory can be understood as an integrated hyperconnected network, being a self-adaptive system continuously exchanging information, with the capability of identifying situations or issues, reacting and autonomously modifying operations achieved through merging the physical and digital environments. It encompasses the objectives of I4.0 of the fabrication of advanced and personalized high-quality products, improving the management, performance and productivity while reducing the economical and environmental costs in the entire manufacturing chain [16, 33]. The I4.0 Factory is the core and the meeting point of the technological pillars of I4.0, as shown in Figure 2.1.

Recently, accelerated by the COVID-19 pandemic, the term of I4.0 has evolved into I5.0, which besides the technological developments of I4.0 extends the paradigm to a more social and environmental dimension [34]. I5.0 promotes a shift to a human-centered point of view, taking into account the expertise and skills of the workers in combination with technology and robots for a fruitful human-robot interaction [35]. To sum up, the I5.0 paradigm provides a way to use the advantages given by I4.0 but in a step-wise direction to reach a human-based approach that focus on sustainability and resilience, with the objective of the welfare of people and the planet [36]. During the next chapters of this thesis, for simplicity, we will use the term I4.0, since we do not address the human-centered point of view of I5.0. In fact, I4.0 or smart manufacturing are still the most trending word to refer to the current manufacturing paradigm.

Both I4.0 and I5.0 have established the current data-driven manufacturing framework. This has reinforced the importance of *data* as one of the key resources in manufacturing [37, 38]. In fact, the term "data-driven manufacturing" is widely accepted to define the new era of industrial production. The advanced sensors and the connectivity offered by IoT services has supposed a big improvement in data acquisition, leading to large volumes of data from the industrial plants. Big data and cloud computing capability have allowed the management, processing and analysis of the obtained data. Consequently, the previous expertise-based manufacturing has evolved to this data-driven manufacturing, where both the knowledge from experts of the field and the insights from data exploitation are used in a balanced way to obtain the maximum profit [39]. In fact, some studies demonstrate that the productivity and the profitability increases in companies using data-driven approaches [40, 41]. In this area, AI-based tools excel in industrial tasks such as process monitoring, fault detection, computer vision, decision-making or autonomous control.

Concretely, within the context of I4.0/I5.0 and AI-driven tools for manufacturing, ML, RL and DT have significantly impacted industrial innovation in the last years. These technologies have enabled the drift towards achieving the goal of efficient, resource-sustainable, high-quality and environmentally friendly manufacturing systems.

2.2 Machine Learning in Manufacturing

ML is a branch of AI that is based on the learning capability of a machine or system from data, with the objective of making decisions or predictions without being explicitly programmed [42]. The use of ML has suffered an exponential growth during the last decade and it has been extended not only in academia or industrial environments, but in society fields such as healthcare [43] or transportation [44]. The availability of data, the improvement of the computational resources and the big technological companies interest and investment in ML innovation have been the main drivers of this popularization.

Focusing on manufacturing, I4.0 has emerged as the perfect scenario for ML application. It has induced an increment of the available process data at the production line, thanks to novel sensors, CPS, metrology equipment, etc. Thus, new algorithms, methods, technologies and software need to be put in place in order to collect, extract, save and create value from data. ML techniques have recently emerged as effective methods for predicting, classifying and controlling processes inside the production systems for decision support and the enhancement of process knowledge of the manufacturing industry. Extracting knowledge from the given data with the ability to learn from it makes ML able to achieve better performance in problem's resolution than humans. Due to the large number of variables and the complexity of the manufacturing processes, it is very difficult to establish cause-effect correlations or to establish analytical expressions governing a highly non-linear system. Moreover, each manufacturing process may require a tailored solution. For this purpose, the creation of data-driven models based upon ML algorithms is key for predicting the desired outcomes, such as the quality of the manufactured parts or the process stability.

The most classical classification of ML algorithms distinguishes between Supervised Learning (SL) methods, Unsupervised Learning (UL) methods and Reinforcement Learning (RL). In the following, SL and UL are grouped together and described in this section, while RL is explained in the next section. The main reason of this differentiation lies in the distinct tasks that these algorithms perform in the industrial framework. SL and UL are based on the concept of predictive modeling or pattern recognition algorithms and they have been widely used in manufacturing problems. RL is categorized as a decision making algorithm and it has recently started to show its potential. Furthermore, the training strategies of SL and UL rely on the learning from existing data. In contrast, RL is usually trained through the interaction with an environment, although it can learn from a pre-defined dataset that simulates this interaction of the environment. Finally, the goal of SL and UL is to realize predictions and find patters from new data whereas RL aims to make decisions to accomplish a given objective. Nevertheless, we have dedicated more attention to the SL techniques and its manufacturing applications, since it is the most used approach during the thesis. UL and RL methods are briefly described and some industrial use cases are also presented.

2.2.1 Supervised Learning Methods: Industrial Applications

Supervised algorithms [45] are trained on datasets that contain labeled samples. The patterns extracted serve to learn a function that related the input data to the output. Formally, given a dataset Z comprising n samples: $Z = \{(\mathbf{x}_i, y_i)\},\ i = 1 \dots n,\ x_i \in \mathbb{R}^d$ and y_i is the target label, the aim of a SL algorithm is to learn the unknown function $f : x \rightarrow y$. Each of the components of the vector $\mathbf{x}_i$ are usually called features. Depending on the nature of the target labels y, there are two types of SL problems:

- Classification problems [46]: The label set is discrete, $y \in [K]$[1]. The task is predicting the class label. A two-class classification problem is also called binary classification problem and a problem with more than two classes is called a multi-class classification problem. It is common for classification models to predict a continuous value as the probability of a given example belonging to each output class interpreted as the likelihood or confidence of it belonging to the class.
- Regression problems [47]: The labels are a continuous, usually real, value, i.e. $y \in \mathbb{R}$. These problems are often called multivariate regression problems.

In SL, the standard procedure involves using the algorithm on a set of training samples to learn a classifier or a regressor, which is an approximation of the function f. Afterwards, the classifier/regressor predicts y from new values of $\mathbf{x}$ and an evaluation of the prediction quality is done. Some of the most used evaluation metrics are accuracy, recall, F1-Score and AUC-ROC in classification [48] and Mean Absolute Error (MAE), Mean Squared Error (MSE) and R-squared (R^2) in regression [49].

[1] $[K] = \{1, \dots, K\}$

In literature, an elevated number of SL algorithms can be found. Many algorithms work for both classification and regression problems after slight modifications. In the industrial context, SL have been widely applied as a powerful tool to leverage production data, taking advantage of human expertise and quality control protocols. The most typical algorithms are Support Vector Machine (SVM) [50], K-Nearest Neighbour (KNN) [51], Decision Tree [52], Random Forest (RF) [53], Gradient Boosting (GB) [54], Gaussian Process (GP) [55] and Neural Networks (NNs) [56]. Within NNs, several types of NNs has been used to deal with manufacturing processes. Among the most important ones, multi-layer perceptron (MLP) [57], Convolutional Neural Networks (CNNs) [58] and Recurrent Neural Networks (RNNs) [59] have been widely employed. The term Deep Neural Networks (DNNs) is a more general way to refer to NNs. Concretely, it is usually employed for NNs with multiple hidden layers, that can have different activation functions and also can combine different types of networks within its architecture, such as convolutional or recurrent layers. However, for convenience, MLPs are often referred as DNNs, since the term "Deep" usually means that they have several hidden layers. From now on, during this thesis we will refer to MLPs as DNNs, while we will keep the nomenclature of CNNs and RNNs. Additionally, the term Deep Learning (DL) is also a common term referring to the field of study of the utilization of NNs that have several layers and also comprises the related techniques including data preprocessing, hyperparameter tuning, model evaluation, etc.

The following is a brief overview and classification of the most relevant SL applications and algorithms used in industry:

- Quality prediction: The SL algorithms are employed to predict defects or mechanical properties of the process or products which may to reduce the costs that are associated to the production of non-quality parts. Therefore, this includes applications such as tool wear monitoring, material identification, fault diagnosis and the classification or prediction of the final products quality.

  In [60], SVM algorithm effectively monitored the machining process of thin-walled parts, where machine tool wear and work-piece deformation always coexist and in [61], SVM has been also used for the monitoring of the wear of the tools in micromilling process is done from the vibration and sound signals. Regarding material identification, while some works also employed

SVM [62], K-Nearest Neighbours has been utilized to classify biomass based on its properties [63] and for polymer classification [64].

Focusing on fault diagnosis problems, SVM has served to address the prediction of various bearing defects in combination with preprocessing techniques [65, 66], the classification of the failures of steel plants [67] or the detection of the faults produced in photovoltaic arrays [68]. Moreover, KNN interpretability has lead to its use in fault diagnosis [69–72] and Gaussian process (GP) has been applied to real-time fault diagnosis of wind turbines operating with SCADA system data [73]. Neural Networks (NNs) have been successfully implemented for the same purpose. The application of DNNs ranges from the modeling of the quality of the injection-molded parts [74] to the prediction of the quality state from process alarm events in semiconductor fabrication [75]. For instance, CNN-based fault diagnosis is realized from wavelet transforms that can convert vibration signals into multiscale spectrograms [76] and RNNs like Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) architectures have also been proved for fault diagnosis in aircraft turbine engines [77]. CNNs have also been employed in other manufacturing scenarios to classify particle defects on semiconductor wafers from Scanning Electron Microscopy images and Energy-Dispersive X-ray spectroscopy data [78], to identify faulty and non-faulty categories from images captured by drones of wind turbine blades [79] and to distinguish pores in laser powder bed fusion [80].

In addition, for pure quality prediction and classification, SVM has also been used to in processes like laser additive manufacturing [81], carbon fiber production [82] or in the mining industry for the quality of the iron ore [83]. Likewise, a quality prediction system for steel manufacturing has been realized with GB algorithms in [84], including an additional explainable method SHAP [85] to understand the contribution of the features to the model prediction. In [86], a GP model combined with a Dirichlet Process to handle noise and outliers is presented to predict product quality from process parameters in continuous manufacturing, and the approach is proven in a use case from a UK foundation industry. Nevertheless, other works also apply GP to classify the quality of aluminum alloy foams produced via investment casting [87].

- Process monitoring and control: The SL algorithms serve to monitor and control manufacturing processes in real-time to detect anomalies, predict maintenance needs or estimate the remaining useful life (RUL) of the tools.

  In [88], SVM is used to predict the RUL and the procedure is validated in data from turbo fan jet engines. The study in [89] proposes an SVM-based model for accurately predicting bearing life. KNN algorithm has effectively worked under dynamic manufacturing environments like the monitoring of uneven-lenght batches comprising multiphase processes with different durations [90] or the recommendation of adjustments to operators based on similar past situations to improve the adaptation speed and system performance [91].

  Respecting the monitoring of the health of the tools and predictive maintenance, in [92] the benefits of using a kernel-based SVM approach to monitor the health condition of a marine diesel engine over time were demonstrated. Extreme Gradient Boosting (XGBoost) [93] has been used to forecast the fatigue lifetime of 3D-printed biomaterials in [94]. DNNs have been used to predict the motor failure times from vibration data in [95], to forecast faults in a centrifugal pump [96], to predict tram track gauge deviation [97]. Within the realm of health monitoring, LSTMs have proven effective in modeling long-range dependencies for tool wear prediction, handling both raw sequential sensory data [98] and processed wavelet-transformed data [99] to model long-range dependencies for tool wear prediction. As well, in another interesting work [100], GRU have been employed to predict the RUL of gears undergoing wear, and its gating mechanism allows it to selectively retain or discard information from previous time steps, mitigating the vanishing gradient issue and enabling more accurate predictions of gear RUL. Anomaly detection in aircraft flight data [101], and even online chatter detection in milling processes [102], where LSTMs process current signals applied to the ball screw drive. In a different approach, the RUL of ball bearings working under different conditions is predicted implementing a RF [103]. A similar work has been presented in [104], consisting in the application of RF combined with Park's Vector and Principal Component Analysis (PCA) [105] to detect stator winding short circuit faults in squirrel-cage induction motors. RF has also been employed for process control in multiple studies. In [106], from SCADA real-time data, the operational state of a cement production line has been predicted with RF. A control method of the tapping manufacturing process has been successfully implemented in [107], where RF is trained on historical signal data coming from the spindle motor current and it evaluates the quality of the tapped holes.

More complex methods such as the combination of NNs architectures have also been successfully implemented. For example, in [108] a CNN and a bi-directional LSTM are joined to eliminate the manual feature engineering and capture the temporal dependencies. In addition, stacked fully-connected layers process the LSTM output, and a linear regression layer predicts the tool wear. In a related approach, the RUL of machining tools are estimated through the extraction of the local temporal features from sequential sensor data by a 1D-CNN. Afterwards, LSTM layers learn the long-term dependencies and degradation patterns in the time-series data and a fully connected layer with a sigmoid activation function generates the RUL prediction [109].

- Predictive Modeling: The SL algorithms are used to predict key process parameters and outcomes, like surface roughness, pressures or temperatures.

  Within the applications, the work of [110] proposed a Decision Tree to predict the pellet sphericity from a set of input variables containing formulation and process parameters, allowing to identify the key factors that influence the pellet aspect ratio. In the study of [111], an estimation of the relative humidity is done through RF fed by data from IoT devices, which is an important factor for the presence of problems such as metal corrosion, moisture condensation or bacteria growth in places like semiconductor factories cleanrooms. A distinct scenario is presented in [112], where RF interacts with a Model Predictive Control framework to control the paste thickening process in mineral processing. RF is applied to forecast the underflow solids concentration and interface level based on measurements of flocculant dosage and underflow rate, while an optimization process is performed based on the RF predictions to select the best control actions. Further, RF has been important in predictive modeling works such as the modeling of the thermal error [113] and the surface roughness prediction [114], among others.

  Other algorithms have also proved their effectiveness in predictive modeling. In [68], GB has proven to be the best algorithm for the real-time prediction of rolling force in hot strip rolling, incorporating a self-training function to adapt the model to changing production conditions and maintain accuracy. In other works, XGBoost has employed to estimate the heat transfer efficiency of oscillating heat pipes [115] and to predict cutting forces in milling [116]. In another example, a spatial GP is implemented to estimate the porosity in metallic parts assembled through selective laser melting [117]. Moreover, GP is employed to model the relationship between process parameters and output metrics such as effective area ratio, height difference, and deposit angle in Wire Arc Additive Manufacturing in [118]. For predictive tasks, DNNs have been employed in the works [119] and [120], applied to Fused Deposition Modeling 3D printing process and free-form machining, respectively. Otherwise, the CNN-LSTM architecture has been implemented in different scenarios like for intrusion detection in industrial IoT devices [121] and to forecast melt pool temperature in wire-arc additive manufacturing [122].

2.2.2 Unsupervised Learning Methods: Industrial Applications

UL methods [123] focus on uncovering hidden patterns and structures within datasets that lack any pre-existing labels. These algorithms process datasets with numerous features, aiming to learn the underlying probability distribution of the data or identify its key characteristics. Through self-organized learning procedures, they discover previously unknown patterns or relationships, providing valuable insights into the data without relying on explicit guidance. In a more formal definition, given a dataset Z with n samples: $Z = \{\mathbf{x}_1, ..., \mathbf{x}_n\}$, the goal of an UL algorithm is to infer the properties of the underlying probability distribution $P(X)$, where X represents the random variable from which the samples are drawn and it is an array comprising all the features of the dataset. This inference is performed without any prior knowledge of the "correct" answers or labels.

Unsupervised algorithms are widely used in the context of outlier and novelty detection. Those anomalies that deviate sufficiently from most observations are called outliers and their number is significantly smaller than the proportion of nominal cases (typically lower than 5 %, but depends on the dataset and case). The anomalies are called novelties, instead of outliers, when the model has been trained on a dataset free of anomalies. Excluding outliers from a dataset is a task from which most data mining algorithms can benefit. For example, a heavy imbalanced class distribution in the dataset can affect the efficiency and robustness of supervised algorithms. Therefore, an outlier-free dataset allows for accurate modeling tasks, making outlier detection methods are extremely valuable for data cleaning. Examples of outlier detection algorithms and applications are One-class SVM, Local Outlier Factor (LOF) and Isolation Forest [124–126].

In addition to outlier detection, clustering is another significant task performed by UL algorithms. It is used to identify hidden patterns in the data and this is achieved separating the data into different groups. The groups and patterns are defined searching for similarity into the data. This is achieved through a metric of distance as the euclidean distance, Manhattan distance or another metric for distance. Regarding outlier detection, it is closely tied to clustering because outliers are often defined as points that do not fit well into any cluster, or that form their own very small clusters. Many clustering algorithms have built-in ways to identify or handle outliers. Respecting the algorithms, K-Means [127] and Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [128] are commonly implemented for clustering tasks [129, 130].

Furthermore, UL usually addresses dimensionality reduction tasks. The aim is to reduce the number of features in a dataset. It is a commonly used step in ML, especially when dealing with a high dimensional space of features. The original feature space is mapped onto a new reduced dimensional space. The dimensionality reduction is usually performed either by selecting a subset of the original dimensions or/and by constructing new dimensions. Feature extraction can also be understood as a type of dimensionality reductions. CNNs have been employed for feature extraction [131] although its main application lies in SL. The unsupervised role of CNN has already been mentioned in the combination of CNNs with other algorithms seen in Section 2.2.1, where the global task is supervised but the CNN performs unsupervised feature extraction [108, 109, 121, 122]. Other techniques such as Principal Component Analysis (PCA) [105] and Autoencoders (AEs) [132] have addressed dimensionality reduction.

Besides, other unsupervised techniques such as Generative Adversial Networks (GANs) focus on the generation of synthetic manufacturing data or manufacturing data augmentation [133, 134].

Finally, semi-supervised learning methods have emerged recently and they been used for data classification, regression and key performance indicators estimation in manufacturing processes. They are applied for similar purposes as supervised learning, but a reduced amount of labelled data and a large amount of unlabelled data are typically assumed for the leaning process. Nevertheless, they can be considered more related to the unsupervised methodologies rather than to the supervised ones since, in the vast majority of cases, a label is not provided per each sample during the training phase. The semi-supervised techniques are able to make use of this additional unlabeled data to better capture the shape of the underlying data distribution and generalize better to new samples. These algorithms can perform really well in those situations where a very small amount of labelled observations and a large number of unlabeled points are provided. This scenario may occur in the industrial domain where mainly bad parts are tagged. Some of the semi-supervised algorithms consider a so called contamination parameter. The percentage of outliers is provided to the algorithm, facilitating the clustering and establishing the models that are able to categorize and predict the belonging group of new samples. Thus, individual labels are not provided per sample, but the amount of parts per class is known or statistically estimated. Semi-supervised learning is particularly useful when the cost associated with labelling is too high to allow for a fully labelled training process. Instead, unlabelled data is much cheaper and takes less effort to acquire from the process. In the industry, for semi-supervised learning, there have been used generative model-based method, self-training, co-training, graph-based method, etc [135–138].

2.3 Methods for Improving Manufacturing Processes

This section overviews some of the methods that help to improve the manufacturing processes, focusing on tools enabled by I4.0, including RL and DTs. The most relevant applications of these technologies in manufacturing are displayed, highlighting its potential and its limitations in this context.

2.3.1 Autonomous Process Optimization with Reinforcement Learning Framework

Reinforcement Learning (RL) [139] is a computational approach to understanding and automating goal-directed learning and decision making. It is distinguished from other computational approaches by its emphasis on learning by an agent from direct interaction with its environment, without requiring exemplary supervision or complete models of the environment. The environment provides to the learner (the agent) the needed training information: the global system status and the reward, meaning the information on how well the system performed in the respective turn. Based on that, the agent has to uncover which actions generate the best results by trying instead of being told.

Therefore, RL is an efficient framework for optimal control and decision-making tasks in stochastic and sequential environments modeled as Markov Decision Process (MDP) [140]. An environment is said to be MDP when the state of the system contains all information the agent needs to act optimally. Namely, this means that the current state is enough to forecast the system behavior in the next step. Additionally, a reward is necessary, i.e., a scalar feedback signal to the agent the results of its actions.

Manufacturing is increasing its complexity. Production environments are in constant change and customers want personalized products. The traditional ways controlling and planning production are not practical anymore. We require smarter and more flexible methods. RL offers a powerful solution to these challenges, dealing with the common problems encountered in industrial processes. Specifically, this supposes managing resources and materials effectively, maximizing the efficiency of machinery, minimizing environmental pollution, reducing energy consumption and optimizing logistics systems. Summarizing, RL is a most promising, flexible and powerful approach regarding the industrial autonomous control and automation, enabling systems to learn, adapt, and optimize performance in complex and dynamic environments.

Figure 2.2 illustrates a conceptual model of how RL operates within an industrial setting. In this iterative learning paradigm, an intelligent agent interacts with the industrial process, which serves as its environment. The environment can also be a SMod, as it is explained in Section 3.1. At each step, the agent observes the current state of the process (encompassing control parameters and KPIs) and selects an action (such as adjusting setpoints or process parameters). The environment responds by transitioning to a new state and providing the agent with a reward signal that reflects the immediate outcome of the chosen action. The agent objective is to learn an optimal policy, π, a mapping from states to actions that maximizes the expected cumulative reward over time. In essence, the agent learns to control the industrial process by trial and error, guided by the feedback it receives in the form of rewards.

Hence, RL have been used in some key manufacturing tasks:

- Robotic manipulation: RL algorithms have been employed to train robots to perform complex tasks, such as assembly, welding, and painting, with high precision and efficiency. The ability of RL agents to adapt to variations in the environment and generalize to new situations makes them ideal for robotic applications in manufacturing. The works in [142–144] presentes different RL strategies to train robots in distinct industrial scenarios.

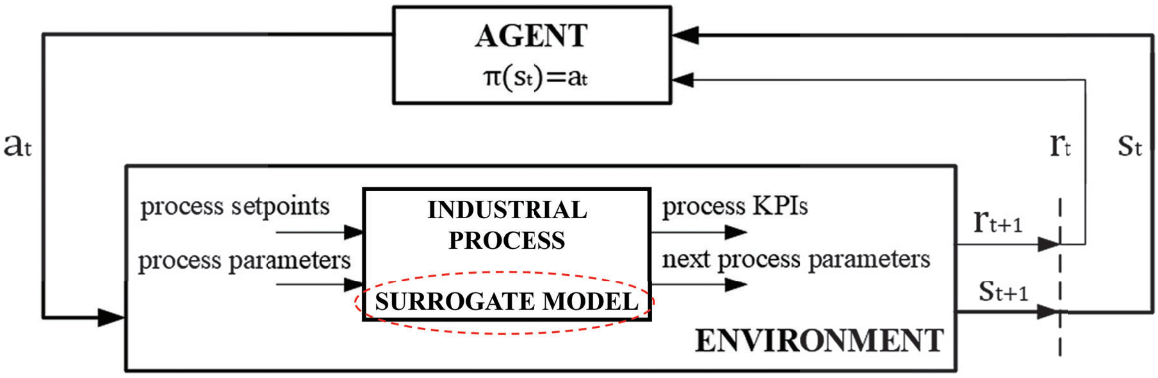


Figure 2.2: Schematic representation of RL operation in industrial processes, inspired from [141]. The RL training phase interation with a SMod-based environment is also remarked in red.

- Process control: RL has been used to optimize process parameters in various manufacturing processes, including injection molding, metal forming, and welding. By learning to control process variables in real-time, RL agents can improve product quality, reduce waste, and increase production efficiency. For instance, some practical applications of RL in process control are studied in [145–147].

- Planning and schedulling: RL algorithms have been applied to production scheduling and planning problems, where the goal is to allocate resources efficiently and meet production targets. The ability of RL agents to handle uncertainty and adapt to changes in demand makes them valuable for optimizing production schedules in dynamic manufacturing environments. Examples of this type of RL utilization can be found in [148–150].

- Fault diagnosis: RL has been implemented for fault diagnosis in manufacturing equipment, where the goal is to detect and diagnose problems early to prevent costly downtime. The ability of RL agents to learn from data and identify patterns makes them effective for fault diagnosis and predictive maintenance. In this sense, the studies in [151, 152] demonstrate the RL potential in fault diagnosis.

2.3.2 Overview of Digital Twin Framework

DTs are digital representations of a real-world physical entities that facilitates understanding the present behaviour and predict the future asset evolution and state. This accountable AI data-driven technology can enable the study of different test scenarios to improve efficiency while anticipating undesirable situations and risks. The idea of DT was introduced by Michael Grieves in 2003, but it kept evolving until the acceptance of the current concept [153]. Also, NASA introduced DT in their road map and tried to put the idea in practice [154]. DTs consist in a complete connection between the virtual part and real part of the system through the exchange of data in real-time. This implies the full integration and interconnection of the I4.0 technologies in a industrial environment. It aims and features the bi-directional integration of the physical and digital worlds. That is, the bi-directional exchange of information between the real device or system and its digital replica. Thus, the real system feeds data (in real time) from sensors, IoT devices, management systems (MES, SCADA) to the digital replica and the digital replica can provide recommendations for action, parameter changes and control of the physical system. DTs allow to simulate, monitor and control physical objects or systems, facilitating data analysis using AI and ML solutions, optimization and predictive maintenance. They enable live feedback loops and foster insights to improve performance, efficiency and reliability. Thus, there is a bi-directional interaction between the physical and digital environment, where the digital replica is able to recommend and change the way the physical entity operates. The digital twin allows us to introduce innovative approaches to asset use and management and allows us to develop control solutions towards the autonomous factory.

Regardless of the vast potential of DT solutions, there are still few developments beyond the theoretical or conceptual framework. According to [155], there is an evident lack of a universal definition, implementation framework, and protocol. However, we displayed the basic idea behind DT in Figure 2.3.

As mentioned, DT involves three principal aspects: the real system, the virtual model of the real system and the bidirectional communication between both. Additionally, the final user of the technology can also be included within the DT structure. The candidate real systems for implementing DTs can be processes, products, machines, structures or large-scale systems like factories or cities. Below, based on the Figure 2.3, a description of the DT parts is given:

1. The real system is determined by known and controllable input variables (system operation parameters such as velocity, pressure, etc) or non controllable variables (atmospheric conditions, noise, etc). Then, the real system is dynamic and changes over time, for example due to the wear and tear of its components. These changes are unknown variables of the system. The output variables of interest are extracted from the system. All these variables

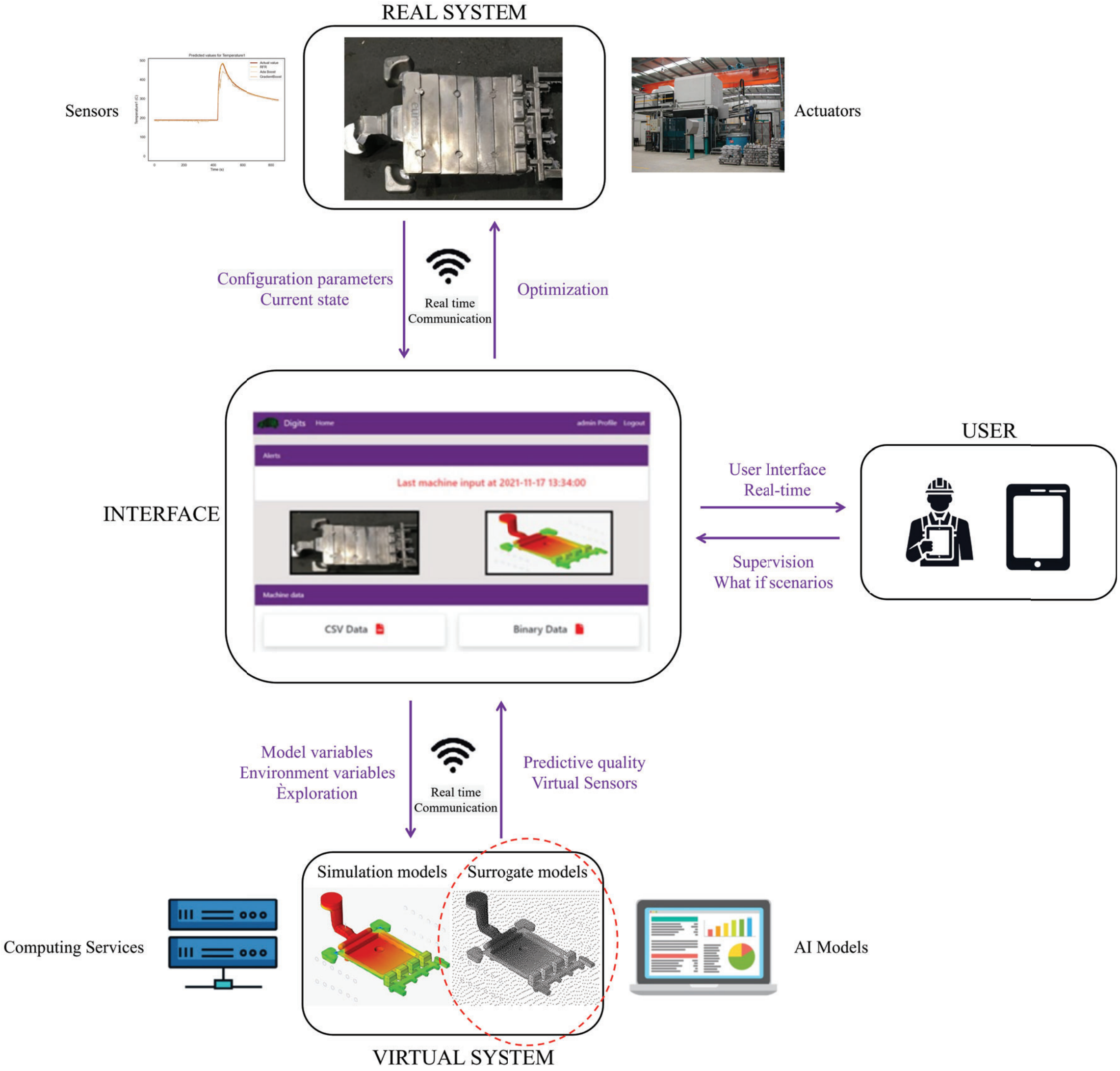


Figure 2.3: Schematic representation of DT framework in industry. The introduction of a SMod as a virtual representation of the real system is highlighted in red.

are monitored and collected through several sensors (temperature, pressure, vibration, etc) and along the configuration parameters, they define the current state of the real system. Moreover, the actuators are components that can modify and have influence on the real system, such as adjusting machine settings, turning on a valve, etc.

2. The virtual model, which is based on numerical or analytical simulation methods and simplified models. SMods can also be used as virtual models, as it is presented in Section 3.1. It allows the unknown variables of the system to be determined through a correlation of the virtual results with the

output variables of the real system. Once these variables are correlated and determined, the DT provides much more information than is available in the real system. It enables the forecasting of possible scenarios and to evaluate the system under possible but unseen scenarios. In this sense, DTs are practical tools for engineers and operators to better understand the systems present and future performance enabling real-time visualization of the real current state. Furthermore, AI models based on virtual and real data can be included in the virtual model, performing predictive quality, predictive modeling of virtual sensors, anomaly detection tasks. Both simulation and AI models are sustained by integrated computing services. This allows to support advanced predictive analytics solutions to address new challenges and opportunities in terms of efficiency, optimization and risk mitigation.

3. The interface is a platform to facilitate the connection between the real and the virtual system. The real-time data acquired from the sensors are sent to the virtual system and, in the opposite direction, the information provided from the virtual system models is sent back to the actuators, which control the real system to optimize its performance. The key aspect to achieve this is a real-time bidirectional communication with both real and virtual system. In the interface, the combination of innovative AI solutions and data-driven predictive algorithms combined with sensor data and interaction between various components enable the creation of digital replicas of real systems with enhanced functionalities. Besides, the platform may allow an easier interaction with the user, focused on tasks like monitoring, supervision, and running "what-if" scenarios.

Focusing on manufacturing, DTs are revolutionizing industries enabling engineers to speed up product design, detect malfunctioning and economize the costs. The DTs allows to explore scenarios and to extract information in a virtual environment, for instance through SL and UL algorithms mentioned Section 2.2, and directly transforming the acquired knowledge in an impact in the real world. Additionally, the twinning of the physical and digital worlds provides an enhanced way to manage resources and improve the customer and operators experience. Also, real-time modifications and the performance of virtual tests removes the number of prototypes to be created, reducing the temporal and economical costs.

DTs are a broad and very versatile technology that can be applied at different levels of the manufacturing value chain, i.e., for product design [156], for process design [157], production planning and control [158], for remote monitoring [159, 160] and for asset performance evaluation [161] in new scenarios or conditions. They enable key functionalities, for instance planning and scheduling optimization based on real time monitoring [162], predictive quality [163], predictive maintenance for increased asset availability [164], productivity improvement [165] or fault detection for risk mitigation [166]. Remarkably, one of the key steps for building digital replicas of real systems is to develop a DT framework that then can be instantiated or fine-tuned for each individual asset. The main benefit of the DT based solutions is that captures and models the individual particularities. Thanks to the data flow from the real system to the virtual model, the DT captures the differences in behaviours due to usage, maintenance or manufacturing tolerances of different individual assets of the exact same system.

2.3.3 Challenges in Real-World Industrial Scenarios

The implementation of these technologies in industrial environments is not an easy task. In fact, there are several limitations, not only in RL and DTs, also in classical optimization procedures in manufacturing.

Firstly, the development of accurate models that capture the complex dynamics and high-dimensionality of manufacturing systems is crucial for effective improvements, because reliable data is essential for sensitivity analysis, knowledge acquisition and optimization. However, acquiring sufficient quality data can be challenging in manufacturing environments. This affects classical optimization, RL training data and DTs data integration for the virtual model. These methods usually require a large number of iterations to obtain the necessary data. In real systems, this is not feasible, since it implies an important consumption of energy, human resources, raw material and time. To prevent these costs and ensure fruitful operations, the configuration parameters are usually limited to the conditions required for production. Also, directly training on real systems introduces safety concerns, because the configurations explored may lie outside the machinery recommendations. Then, the only possible option is a very efficient learning from limited data, which is unlikely to happen. As an alternative, simulations can be employed used for training to reduce costs and risks linked to real-world training, because they offer great capabilities of the FE simulations to reproduce real-world systems. Even though, these models are very demanding in terms of computational resources and time to produce high-fidelity results. A detailed representation supposes an elevated cost, leading to a tradeoff between the efficiency and the quality of the simulation results. Moreover, to perform a rigorous study of the manufacturing processes, executing a significant amount of simulations is necessary, which magnifies the aforementioned drawbacks of the simulations' computational time and resources requirements.

Next, and closely related, achieving real-time performance is crucial for many manufacturing applications. This poses challenges for RL, due to the need of enhancing training and response times. Besides, DTs also suffer a lot from a slow real-time response, since it is fundamental to maintain real-time synchronization between real and virtual systems and enabling immediate actuation.

Finally, manufacturing environments are dynamic, making it challenging to model the system under all the possible scenarios. Additionally, usually these methods are developed using simulation models. Therefore, there is a need to an adaptable and flexible modeling procedure, which also is capable to bridge the gap between simulation and reality. This is essential for successfully deploying optimization solutions, leading to a significant challenge for RL in transferring policies from simulated to real environments and DTs in ensuring that the virtual model accurately reflects the physical system.

To conclude, this thesis aims to accelerate the data generation phase to enhance the training phase of RL solutions, mitigate the temporal costs associated with modeling real systems for DTs or boost the pipeline of optimization methods. It achieves this by proposing alternative efficient and economical environments for developing RL agents and alternative modeling techniques for DTs that enable fast responses. These goals address the mentioned challenges of implementing advanced solutions that can improve the manufacturing processes.

Chapter 3

Surrogate Models

In this chapter we establish a framework for understanding SMods within the industrial context explained in the previous Chapter 2. We emphasize the importance of SMods as an alternative and efficient modeling technique in manufacturing. Also, we remark the benefits of SMods respect to manufacturing simulations and we outline their potential advantages both in enhancing AI-driven tools like RL agents and DTs for practical manufacturing applications. Therefore, in this chapter we identify SMods as the main research direction of the thesis and we aim to demonstrate to stakeholders and engineers why incorporating SMods into their simulations, data generation processes and AI strategies is crucial for efficient implementation and overcoming challenges in industrial settings. Besides, we advocate for the use of SMods to optimize the improvements offered by RL agents and DTs in real-world manufacturing scenarios. To support these ideas, we provide a description about different methods of SMods in industry, with a special focus on the ML-based SMods that we will apply during the thesis.

3.1 Understanding Surrogate Model in Industry 4.0

Surrogate Models (SMods), sometimes known as reduced order models or response surfaces, are metamodels that offer the capability to boost the response of a complex and expensive model or system. They provide an approximation of the function that relates the inputs with the outputs, being able to model the system in a simpler but representative way. The accuracy of the result of a SMod tries to be as close as possible to the high-fidelity simulation model or the real data. Their fast response and the quality of the predictions can break the well-known tradeoff between time and accuracy inherent to the simulation models. Furthermore, they can overcome the costs and temporal limitation in real data acquisition from industrial plants. A representation of the surrogate modeling concept is displayed in Figure 3.1.

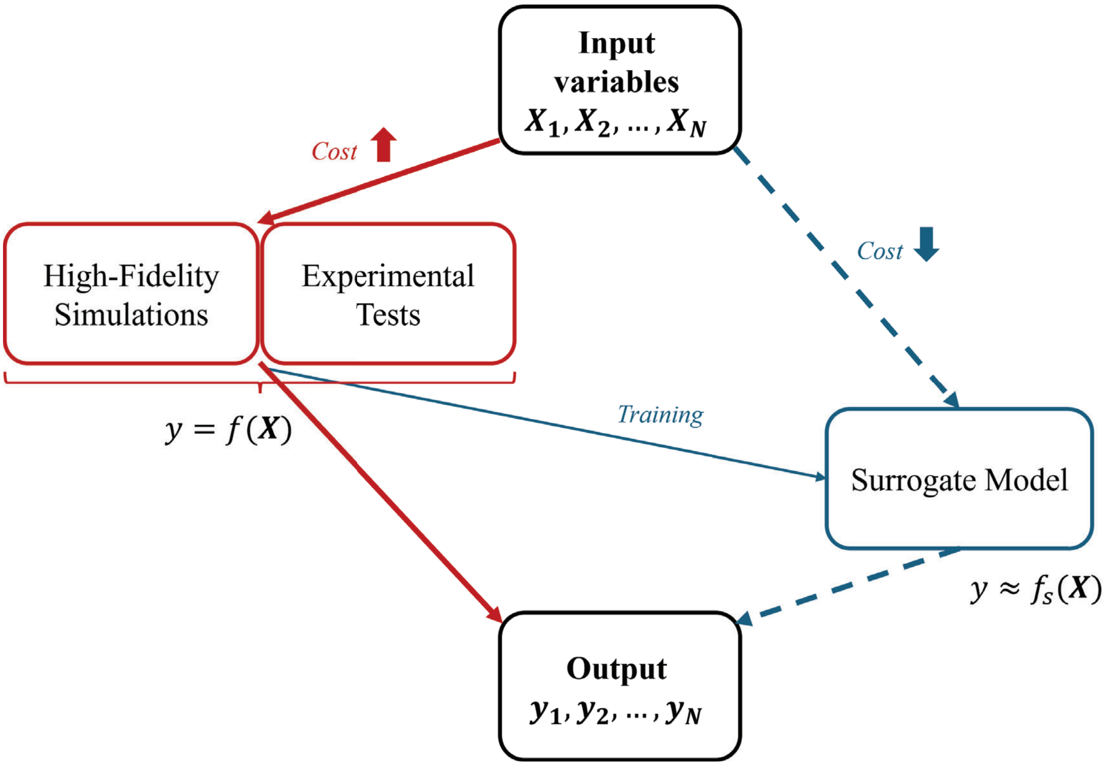


Figure 3.1: Outline of the surrogate modeling concept.

Formally, the concept of SMod can be understood as a function $f_s(x)$ that is an approximation of the true model $f(x)$, where x represents the input parameters to the model. The objective of a SMod is to minimize the difference between $f_s(x)$ and $f(x)$. Therefore, surrogate modeling is usually considered to be part of the category of regression problems. The main difference between SMods and regression models is their role as simplified representations of existing models or systems.

In the industrial context, the use of computer simulation methods to model the manufacturing systems have been the most common approach to acquire knowledge of this type of systems. Manufacturing simulations comprise an elevated number of methods applied in the different stages of an industrial plant, such as the design of the parts, the involved physical processes, the scheduling and the planning of the machinery and resources, among others. For instance, computer-aided design (CAD) or virtual reality (VR) are more oriented to the design stage, finite element (FE) analysis or computational fluid dynamics (CFD) are used in the physical processes simulation and discrete element simulation (DES) is typical in the scheduling and planning problems. The aforementioned methods and the manufacturing simulation tools enable the study of the complex manufacturing systems without perturbing the real system. This opens the possibility to the experimentation and the validation of the design and the process, providing insights and knowledge about the system. The understanding of the complex dynamics of the different manufacturing stages may lead to better decision-making, which is translated to an improvement of the key performance indicators (KPIs), like the product quality or the operational costs.

Thanks to its high fidelity, manufacturing simulation models act as alternative environments to explore new configurations [167]. Nevertheless, their detailed representation of the system implies high computational demands and an elevated time cost, leading to a very slow response [168, 169]. In addition, the mentioned drawbacks added to the problem of the curse of dimensionality often make the data generation process unfeasible. Then, the exploration of a high parameter space with a simulation model is not sustainable in terms of time and computational resources.

Moreover, the improvements in the data accessibility from industrial plants brought by the digitalization of I4.0 may not always be translated in a better understanding of the process or data-driven solutions. In many cases, the processes performed in industrial environments are centered in production and the configuration parameters are not modified to avoid undesired issues in the machinery and the tools. The few changes in the configurations are a limiting factor for the knowledge inference and the potential of data-driven techniques. A more extensive exploration of the parameter space is needed in order to capture and comprehend the behavior of the system and the complex physical phenomena of the manufacturing processes [170, 171]. The performance of tests to this purpose, supposes a significant expense in human resources, raw material, time and the use of the production plant to non fruitful operations [172–174].

To sum up, I4.0 has lead to the emergence of a data-driven manufacturing, where tools based on AI and ML are fundamental. However, the data generation phase through simulations and experimental tests in manufacturing plants is still a challenging task. Also, the most recent advances regarding decision support systems, sensitivity analysis or optimization techniques rely on fast responses for their viable application. Some of the presented issues regarding real industrial plants and simulations are depicted in Figure 1.2.

Similar data generation problems can be encountered in other disciplines that are dependent on simulation models of physical systems or phenomena, such as water management, biomedical engineering or architecture, among others. In fact, efficient and accurate output generation in real-time is a requirement to overcome the limitations in the exploration and the study of the physical systems.

Under the presented challenges for faster and reliable data generation methods, SMods have demonstrated outstanding potential [175–178]. The advantages of SMods have lead to a trending tendency to use them as a solution to face the curse of dimensionality problem, to speed up optimization techniques and to enhance the data generation process in a lot of environments, as shown in Figure 3.2.

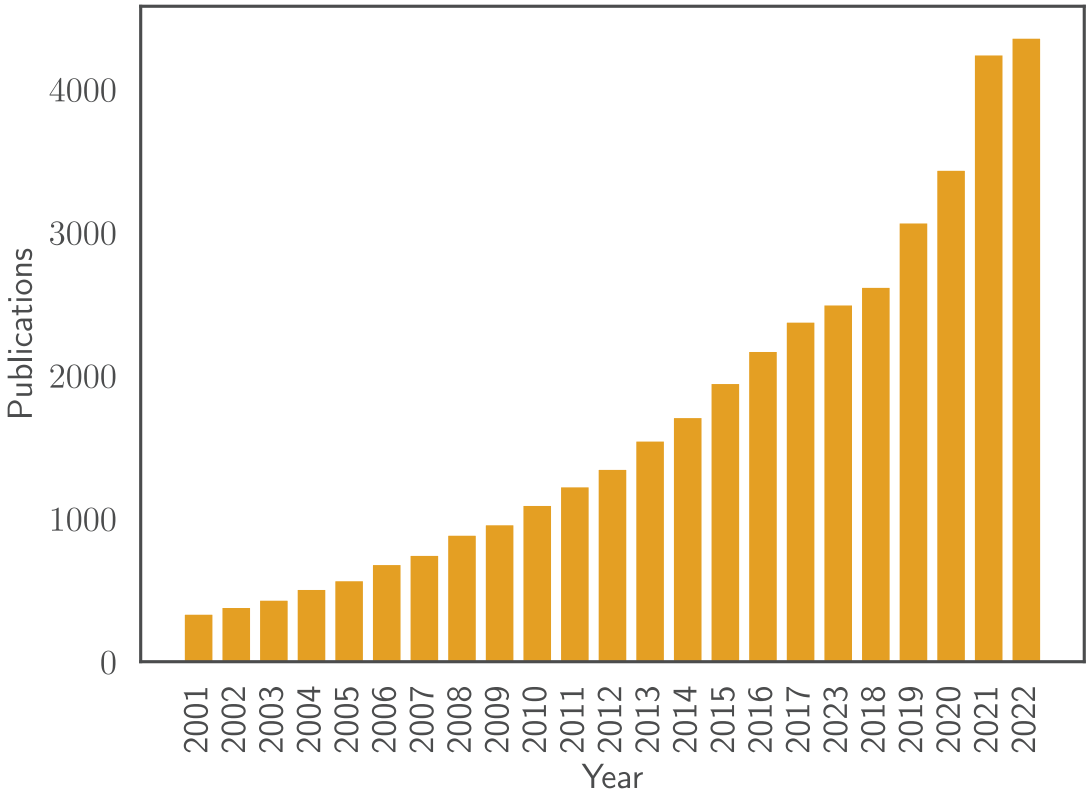


Figure 3.2: Number of publications related with SMods since 2001. Data obtained from Clarivate *Web of Science*. © Copyright Clarivate 2024. All rights reserved. (`https://www.webofscience.com/`)

Focusing on the industrial field, the use of SMods leads to crucial advantages that help to overcome the limitations in data generation from real plants or simulation models and to improve KPIs in industrial settings:

- Production efficiency: SMods significantly reduce the time required for analysis and exploration of complex manufacturing processes, leading to faster design iterations and optimization cycles. This translates to quicker turnaround times for new product development and process improvements.

- Cost reduction: By minimizing the need for extensive physical testing or time-consuming simulations, SMods help save on resources, materials and human effort. This can result in significant cost reductions in research and development, production and quality control.

- Process knowledge: SMods enable a more thorough exploration of the parameter space in manufacturing processes, facilitating a better understanding of system behavior, complex physical phenomena and the impact of process parameters on final product quality.

- Product quality and predictive maintenance: The insights gained from SMods lead to more informed decisions about process parameters and design choices, ultimately resulting in improved product quality, well-functioning of the tools and reduced defects.

- Working with limited data: In situations where real-world data from industrial plants is scarce or expensive to obtain, SMods offer a valuable tool for understanding, optimizing and controlling manufacturing processes offering a low-cost data generation alternative.

- Real-Time Response: The SMods can run in parallel to the production process, forecasting the output of the system in real-time. This enables to act to the production system immediately if needed.

In addition to the benefits that SMods directly bring to the data generation process, these tools can also be applied to the development of RL agents or DT technology. SMods features can help to deal with the challenges highlighted in Section 2.3.3. Concretely, the RL training phase requires several interactions with an environment. Instead of using the real manufacturing system or a simulation model, an efficient and low-cost environment based on SMods can be employed as an alternative boosting a lot the training phase, as shown in Figure 2.2. Similarly, SMods can replace the virtual representation of a DT done with simulations, mitigating the cost of modeling the real manufacturing system. They enable a real-time response from the virtual part of the DT, which improves the immediate actuation capabilities on the real system, accelerates the exploration of "what if" scenarios, permits the real-time exchange of data between real and virtual parts and increases the overall DT performance. This is displayed in Figure 2.3.

In this thesis we aim to rigorously quantifying the presented benefits of SMods in specific real-world manufacturing scenarios. Through detailed case studies and analysis in the following chapters, we demonstrate how SMods can lead to measurable improvements in efficiency, cost reduction and product quality. Also, the use cases of application serve us to identify improvements in the original SMods and to extend them for a more complete solution. Furthermore, in this work we consider how these SMod implementations can be used as a foundation for developing advanced AI solutions, such as RL agents and Digital Twins, tailored to the specific needs of the studied processes. Ultimately, we seek to provide a comprehensive framework for leveraging SMods to drive innovation and optimize performance in the I4.0 paradigm.

3.2 Methods and Applications of Surrogate Models

Creating a SMod involves a diverse range of methods, each with its own strengths and weaknesses depending on the specific application. Nonetheless, despite this variety, a typical pipeline for building an SMod shares common features. SMods are generally data-driven, constructed from a limited set of high-fidelity input-output data. This data is used to train the model, which then undergoes a validation process to assess its accuracy. Figure 3.3 illustrates this generic procedure for developing a SMod. The following paragraphs delve deeper into the various methods used in surrogate modeling and some of their applications in manufacturing.

In the origins, the initial approach in surrogate modeling was to develop response surfaces [179], which are based on polynomial functions that try to model the complex relationships between input variables and output responses [180]. This method is called Reponse Surface Method (RSM). For an unknown $f(x)$, RSM approximation is first-order if the true function takes a linear form

$$f(x) = a_0 + a_1x_1 + a_2x_2 + ... + a_ix_i \tag{3.1}$$

or second-order if the true function is non-linear

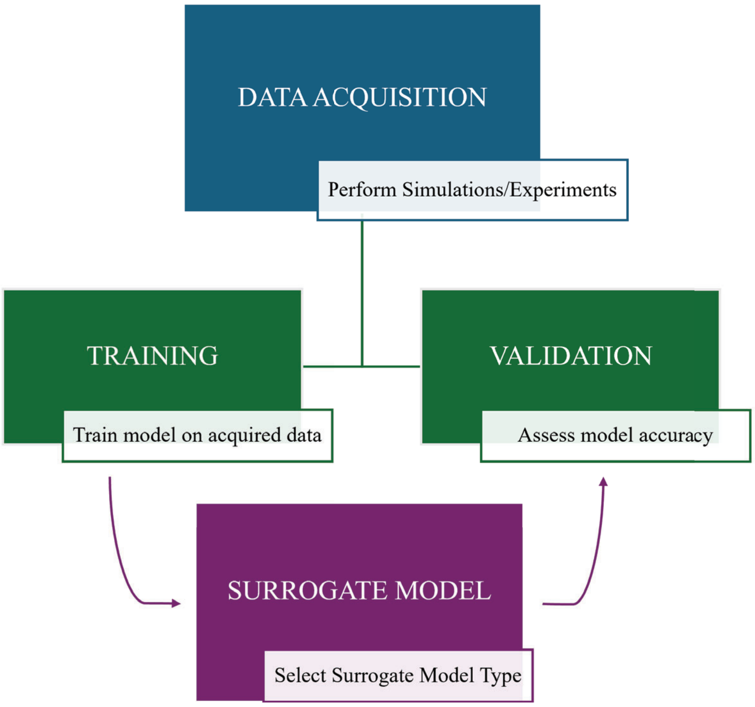

DATA ACQUISITION
Perform Simulations/Experiments
TRAINING
Train model on acquired data
VALIDATION
Assess model accuracy
SURROGATE MODEL
Select Surrogate Model Type

Figure 3.3: Generic pipeline to build a SMod.

$$f(x) = a_0 + \sum_{i=1}^{k} a_i x_i + \sum_{i=1}^{k} a_{ii} x_i^2 + \sum \sum_{i<j} a_{ij} x_i x_j \tag{3.2}$$

where x_i are the input variables, $i, j = 1, ..., k$ are indexes for the input variables and a_i, a_ij are coefficients. RSMs have been used to address engineering problems in applications such as curve fitting, empirical correlations and simplified physical models. In addition to the common polynomial approach, other functions can also be used in RSM. Nowadays, the term response surface is also considered to be a synonym of SMod.

Many studies from different areas have applied RSM to simplify and enhance their modeling. For instance, RSM have been used in chemical engineering industries [181], where the tests to develop a new product are very costly and expensive. RSM can help to optimize reaction yields, minimize impurities and understanding the effects of process parameters on product quality. In the aerospace industry, RSM has been utilized for aerodynamic design, optimizing aircraft wing shapes and controling surfaces to minimize drag and maximize lift [182] and to replace time expensive simulations from real-time responses [183]. Finally, in manufacturing machinery it can be used to optimize cutting parameters, to improve surface finish, tool life and material removal rates [184, 185].

Kriging [186], which is a specific case of GP regression originally designed for spatial data, was posteriorly used to build the next generation SMods, and it has been used until today. Kriging assumes that the underlying function being modeled is a realization of a Gaussian process, characterized by a mean function and a covariance function. More specifically, from the deterministic response $y = f(X)$ with $X = \{X_1, ..., X_M\}^T$ as the M-dimensional input vector, Kriging model considers a realization of a stationary Gaussian process $Y(X)$ with the following formulation:

$$Y(X) = \sum_{k=1}^{P} a_k w_k(X) + Z(X) \tag{3.3}$$

where $w(X) = \{w_1(X), w_2(X), ..., w_P(X)\}^T$ are regression functions typically represented by polynomial functions, $a = \{a_1, ..., a_P\}^T$ are the regression coefficients and $Z(X)$ is a Gaussian process. On the one hand, the Kriging model comprises the regression part that conveys the data mean trend. On the other hand, it includes a stationary Gaussian process with zero mean ($E[Z(X)] = 0$), constant variance σ_Y^2 and stationary correlation function $Cov[Z(X), Z(X')] = \sigma_Y^2 R(|X - X'|, \theta)$, where σ_Y^2 represents the uncertainty of the statistical inference and the Gaussian correlation function is used:

$$R(|X - X'|, \theta) = exp\left(-\sum_{i=1}^{M} \theta_i |x_i - x_i'|^2\right) \tag{3.4}$$

where $\theta = \{\theta_1, ..., \theta_M\}$ is the vector of hyper-parameters. The correlation between x and x' is primarily determined by their proximity. As the distance between these points decreases, their similarity increases, leading to a higher correlation. This implies that predictions made at nearby points will tend to be more alike. The hyperparameter θ_i quantifies the influence of the i-th input variable on the model output. Unimportant variables will have correspondingly small hyperparameter values.

Under these conditions, it is assumed that the Kriging predictor, denoted as $\hat{f}(x)$, is a function that estimates the unknown function $f(x)$ at a new input point x based on observed data points and their corresponding outputs. It is a weighted average of the observed outputs, where the weights are determined by the spatial correlation between the observed points and the new point. Mathematically, the Kriging predictor is expressed as:

$$\hat{f}(x) = w(x)^T a + r(x)^T R^{-1}(y - Fa) \tag{3.5}$$

where $w(x)^T$ represents the trend or mean component of the model, $r(x)^T R^{-1}(y-Fa)$ indicates the correlated deviations from the trend, R is the correlation matrix between observed data points, $r(x)$ is the correlation vector between the new point and the observed points, y is the vector of observed outputs and F is the matrix of regression functions w computed at the observed points. The Kriging predictor also provides an estimate of the prediction error or uncertainty. This uncertainty estimate is crucial for assessing the reliability of the prediction. The calibration of the Kriging model is done through the estimation of the hyperparameters θ maximizing a likelihood function.

The introduction of Kriging permitted a higher flexibility with respect to RSM to capture higher order non-linearities, deal with irregular data and having a more precise interpolation method [187]. The main drawback of Kriging with respect to RSM is that it can be computationally more expensive than RSM, especially for large datasets.

There are numerous examples of the uses of Kriging in surrogate modeling in manufacturing. In [188], Kriging method is utilized in the context of three-dimensional optical surface metrology and its is implemented to address the issue of invalid data points (artifacts) that arise in surface measurements due to various factors such as steep edges, material transitions, and environmental influences. The efficiency of Kriging in structural reliability analysis that involve computationally intensive FE numerical models is studied in [189], comparing the accuracy of failure probability predictions using Kriging models against traditional RSM regression models. The paper in [190] presents a Kriging SMod that establishes the relation between the input welding process parameters and the bead profiles in laser brazing, to boost a posterior particle swarm optimization to optimize the process parameters. In another work, Kriging is applied to predict bend deduction in sheet metal bending, and acts as a metamodel that offers a simplified representation of the complex bending process simulation [191].

The rise of ML that has occurred in the first decades of this century has led to the popularization of ML application in metamodeling, and their reliability and accuracy have been proved. Both traditional ML techniques such as SVM, RF and also DNNs (see Section 2.2.1) have gained participation in surrogate modeling and they have complemented the previously introduced methods of RSM and Kriging. A ML-based SMod can be defined as a data-driven approximation of a complex function or system, constructed using ML algorithms. It learns the underlying relationship between input parameters and output quantities of interest from a set of training data, enabling efficient prediction of system behavior without the need for computationally expensive simulations or experiments.

Formally, given a system represented by the function $f : \mathbb{R}^m \to \mathbb{R}^n$ that maps the input vector $x \in \mathbb{R}^m$ to an output vector $y \in \mathbb{R}^n$, a ML-based SMods is a predictor $\hat{f}$, which approximates this function $\hat{f} : \mathbb{R}^m \to \mathbb{R}^n$, where $\hat{f}(x) \approx f(x)$. In this case, the predictor is a ML algorithm performing a regression task. Therefore, the process of constructing a ML-based SMod requires from training and validation stages that are usually present in the development of ML algorithms:

1. Data: Initially, sufficient and representative training data is necessary for the ML regressor to perform a training stage and a posterior validation to check its generalization ability.

2. Loss function: The loss function is a crucial component of the training process for ML-based SMods. It quantifies the difference between the model predictions and the actual data. The goal of training is to find the model parameters that minimize the loss function. The choice of the loss function depends on the problem, the most common ones being MAE and MSE [49].

3. Training the ML model: The training of the model is an optimization procedure aiming to find the model parameters that minimize the loss function. The optimization algorithms iteratively adjust the parameters based on the gradients of the loss function with respect to the parameters. Among the optimization algorithms used to train ML models, some of the most popular are Stochastic Gradient Descent (SGD) [192] or Adaptive Moment Estimation (Adam) [193].

4. Validation of the ML model: Validation techniques are used to evaluate the performance of the trained SMod on unseen data. This helps assess its ability to generalize to new situations and avoid overfitting. Typical validation techniques include:

   - Hold-out Validation [194]: Splitting the data into training and validation sets, training the model on the training set and evaluating its performance on the validation set.

- Cross-Validation [195]: Dividing the data into multiple folds, training the model on different combinations of folds and averaging the performance across folds.

The evaluation of the models is done through the computation of error metrics. Error metrics are used to quantify the performance of the SMod. The choice of metric depends on the specific application and the type of problem. Common error metrics are MAE, MSE and R^2 [49].

Despite they require a certain amount of training data and that they are black-box models with difficult interpretability, ML-based SMods offer a higher capability in capturing nonlinear relationships in data than RSM, and they are usually more efficient than Kriging. They are able to handle a larger number of variables, they are flexible and adaptable to updates with new data and they very efficient to deal with large datasets and once trained the outputs are produced in real-time.

These attributes can be found in different works that implement ML-based SMod in manufacturing processes. In [196], a SMod based on a DNN is trained to predict the thermal history in wire and arc additive manufacturing using FE simulations. The trained SMod can accurately predict the temperature history in cases not used for training in only 38 seconds, compared to 5 hours for the FE model. A DNN-based SMod is also employed in [197] to quickly predict the wrinkling patterns that occur in a biaxial non-crimp fabric layup for any given tool geometry during forming. The DNN have the flexibility to incorporate various factors affecting formability and wrinkling within the same model by expanding the dataset and model inputs. In [198], the development of a DNN that can estimate the strength of additively manufactured steel based on microstructural images is described. This DNN that acts as a SMod is a more efficient alternative to traditional crystal plasticity FE models and it has been trained using a large database of input-output samples from FE models. A distinct approach is presented in [199], where a GP model that can predict the probability distributions of the geometrical features is developed from a previous feature engineering based on domain expertise, which allows them to create a low-dimensional and interpretable representation of the cup geometry. The potential of the proposed method to capture nonlinearities is explicitly highlighted. In [200], a ML-based SMod trained with FE simulations is used to monitor the quality of thermoplastic composites, predicting the temperature profile of the composite during manufacturing. The SMods are able to find nonlinear correlations between variables and produce real-time outputs to control the process, working with a large simulation dataset that includes all the nodes of the FE mesh. In other works, the combination of ML-SMods with optimization techniques or inverse design to find the optimal operational parameters is applied to optimize the design of photonic structures [201], the performance of microchannel heat sinks [202] or the synthesis of lithium-ion battery cathode materials [203]. Also, focusing on lithium-ion battery cathode, a calculator based on a NN potential trained on density functional theory data is created in [204]. It can accurately predict the energy and forces of different cathode materials, allowing for fast and efficient screening of different cathode compositions.

Additionally, several publications are found in literature describing and classifying surrogate modeling techniques in different application fields. In [205], an overview of the different types of SMods is provided along with examples of applications of these models to groundwater modeling. Similarly, the review in [206] highlights how SMods can improve the phases of building design and it describes steps for the development of SMods. The work in [207] includes an extensive discussion of the development of SMods with the objective to be used to perform a surrogate-based analysis and optimization in aerospace sciences. In [208], a literature study of the most common SMod algorithms and their applications to chemical process engineering is performed. Moreover, some recent reviews have focused on the improvement of the SMods performance considering the tradeoff between computational time, size and accuracy. For instance, in [176] a practical guidance for selecting appropriate surrogate modeling methods is presented and several types of algorithms are discussed, taking into account the mentioned trade-offs. As well, the review in [209] identifies the most common challenges in the development of SMods and it describes different strategies that have been used to overcome these limitations. Finally, the work in [210] has highlighted the value of the use SMods in the FE computations and the main categories of problems where they are employed. As commented, models such as FE and CFD play a very important role in the exploration and understanding the complexity of manufacturing systems. The successful application of ML-based SMods to face the computational and temporal limitations of manufacturing simulations has brought their identification as an effective solution. Despite the completeness and extension of the mentioned reviews, they are not specifically centered on manufacturing nor ML-based SMods. Although some of them propose a generic procedure to recognize the appropriate method for surrogate modeling, the solutions are specific and highly dependent on the studied manufacturing use case.

In this scenario, the next chapters of the thesis we extend the theoretical description of SMods with the support of real manufacturing applications. We described the methods in detail, complementing the information provided in this chapter. By harnessing the capabilities of ML-based SMods, we focus on developing general methods to generate ML-based SMod that can be applicable to different manufacturing processes. We validate these general approaches in real manufacturing use cases, remarking their advantages and identifying the limitations. In some cases, we introduce improvements to further enhance the performance of SMods or broaden their funcionalities.

Chapter 4

Use Cases of the Thesis

In this chapter, our objective is to briefly describe the primary aspects of the manufacturing processes we have used for surrogate modeling in the practical applications of this thesis, addressing the readers who are not familiar with these topics. Concretely, we have considered the processes of plastic injection moulding, high pressure die casting and hot stamping, also known as press hardening. It is important to note that the provided explanation about hot stamping will be extended during the thesis due to its relevance in the applications of the thesis.

4.1 Plastic Injection Moulding

Plastic injection moulding is the most used manufacturing process for the fabrication of plastic parts. Plastic is melted using shear and temperature inside a barrel and afterwards is injected into the mould where it cools and solidifies into the final part, taking the desired shape [211, 212].

The injection moulding process is divided into four main steps: plastification, injection, packing and cooling and ejection, represented in Figure 4.1.

In the plasticization step, the process begins with the preparation of the plastic material, which is usually in the form of pellets. These pellets are fed into the barrel of the injection machine. Then, the mould closes and the plastic is melted inside the barrel applying temperature, pressure and shear using a rotational screw. Once the plastic is completely melted, the injection phase begins, and a shot is carried out injecting a certain amount of plastic at high pressure into a hermetically sealed mold, which defines the final shape of the part. This mold, usually made of hardened steel, consists of two halves, one containing the main cavity (where the plastic is injected) and another creating the external shape of the part.

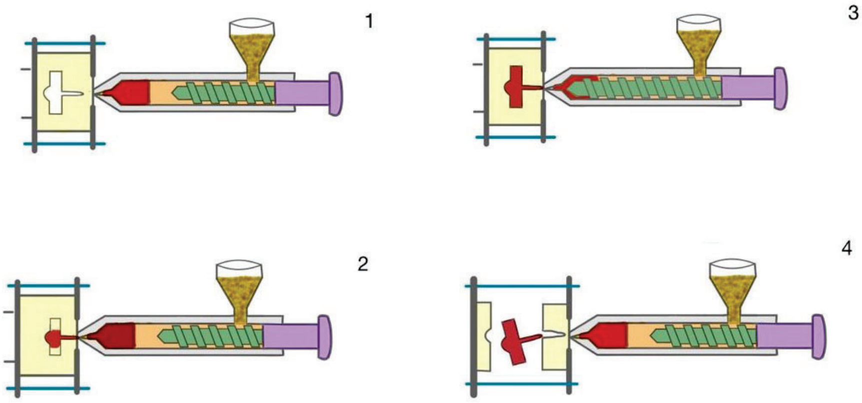


Figure 4.1: Representation of the different steps of the injection moulding process. 1. Plasticization. 2. Injection. 3. Packing. 4. Cooling and ejection.

Once the mold is filled, the pressure is maintained for a certain time to ensure that all cavities are filled correctly and to compensate for the shrinkage of the plastic as it cools. This stage is called as packing, and the pressure applied is usually known as holding or secondary pressure. Afterwards, the cooling and ejection step starts. The cooling is done by a cooling system that circulates through the mold, accelerating the solidification process. When the plastic has cooled sufficiently, the mould is opened and the part is ejected using an ejection system. Finally, excess material, such as feed channels or sprues, is removed and the part is ready for use or to move on to subsequent finishing stages.

Figure 4.2 shows a representative example about how pressure progresses during the injection moulding cycle inside the cavity, identifying the previously described phases of the process.

Plastic injection moulding is a highly efficient process that allows mass production of parts with great precision and repeatability. Automation of the process contributes to cost reduction and optimisation of production times. However, precise control of process parameters such as temperature, pressure and cooling time is essential to ensure part quality [213]. Proper selection of plastic material is also crucial, as each type of plastic has specific properties that make it suitable for different applications [214].

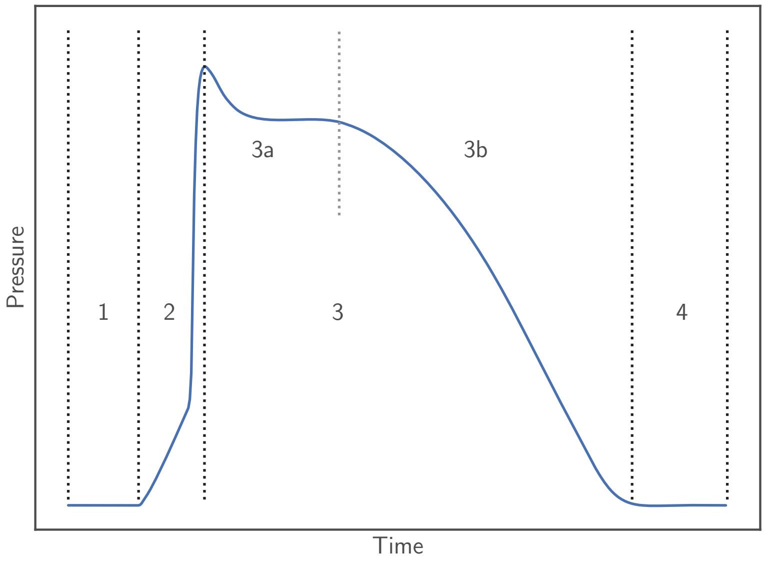


Figure 4.2: Example of the evolution of the pressure inside the cavity during the injection moulding process. The different stages are indicated: 1. Plasticization. 2. Injection. 3. Packing: 3a. Holding pressure, 3b. Decay. 4. Cooling and ejection.

4.2 High Pressure Die Casting

High Pressure Die Casting (HPDC) is a process in which a molten metallic alloy is forced under pressure into a locked metal die cavity, where it is held by a powerful press until the metal solidifies thanks to a set of cooling channels. After solidification of the aluminium, the die is opened and the piece ejected. Once the part is extracted from the cavity, the whole die surface is sprayed with a release agent to prevent the interaction between molten metal and tool steel. Finally, the die is closed again for the next cycle [215].

The die filling process is composed of different phases, which are displayed in Figure 4.3:

1. 1st phase or slow shot phase (prefill): the plunger advances at low speed until the metal starts to fill the die cavity.

2. 2nd phase or fast shot phase (filling): once the metal reaches the gate of the die the plunger speed should be sharply increased, between 4 and 10 times.

3. 3rd phase or solidification phase (consolidation): once the die cavity is filled about 95-98 % of their volume, the plunger reduces its speed and the controlling variable is switched from plunger position to pressure, inducing a high

pressure during the metal solidification process, also induced by the cooling channels.

4. Ejection of the final solidified part.

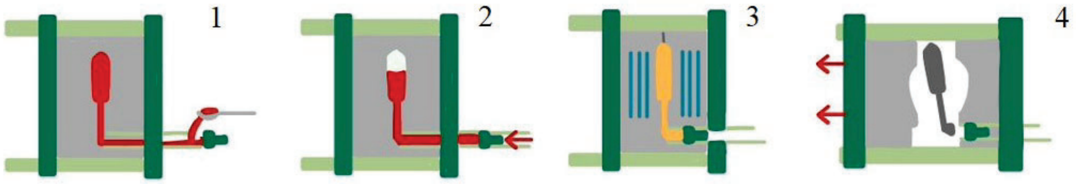


Figure 4.3: Representation of the different steps of the HPDC process. 1. Prefill. 2. Filling. 3. Solidification. 4. Ejection.

HPDC is a versatile and efficient manufacturing process used to produce high-quality metal parts with complex geometries and tight dimensional tolerances. This method, which relies on high-speed injection of molten metal into a steel mold, is widely used in various industries such as automotive [216], electronics [217] and aerospace [218]. However, it is important to note that HPDC is best suited for the production of small to medium-sized parts, although recently some big parts have been done with gigacasting [219], and the cost of manufacturing the molds can be high, especially for complex designs. Despite these limitations, high-pressure die casting remains a fundamental manufacturing technique in modern industry. Its ability to produce high-quality parts quickly and efficiently makes it an attractive option for a wide range of applications.

4.3 Hot Stamping

Hot stamping, also called press hardening, is a metal forming process where a steel blank is heated to austenitization temperature (typically over 900 °C) and then transferred to a press where it is formed and quenched in a single step. It is defined as a "non-isothermal forming process for sheet metals, where forming and quenching take place in the same forming step" [220]. The main advantage is the production of lightweight components with complex shape and very high mechanical properties [221], overcoming the common limitations in the cold forming of Ultra-High Strength Steels (UHSS), such as spring back on the component or damage on the tools [222]. Hence, the process of hot stamping is a staple in automotive lightweight design, being extensively used in safety cage body-in-white components such as A- and B-pillars.

Hot stamping involves a simultaneous forming and heat treatment, leading to thermal history of the parts being a key quality indicator: indeed, most of the process is designed around the compromise of achieving a successful quench while minimizing cycle time. These clear-cut aims, combined with the high responsibility and need for traceability of the parts produced, make hot stamping an interesting use case for digitalization and optimization.

In this use case, we have centered on a popular hot stamping technique called direct method. It consists in different steps, shown in Figure 4.4. In the first step, sheet steel blanks are austenitised at a temperature between 900 and 950 °C inside a furnace. Afterwards, an automated system transfers the austenitised blank to a set of cooled dies, where it is formed in a single stroke while its temperature is approximately within the 700 and 850 °C range (the transfer time decreases the temperature). After forming, the dies are kept close and pressure is applied for a short period of time (typically 5 to 15 s, depending on the plant characteristics). During this step, the cooled dies quench the formed component at a cooling rate between 50 to 100 °C/s to a temperature of 100-250 °C, ensuring full martensitic microstructure. The finished component is then extracted from the die [223]. Usually, this process is repeated sequentially several times until the batch size is accomplished.

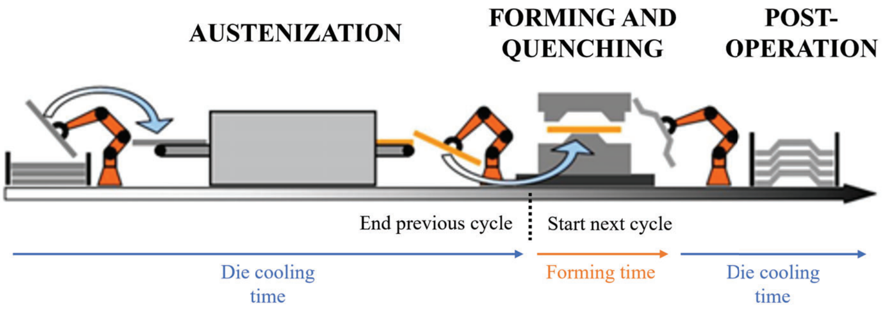


Figure 4.4: Schematic representation of the direct press hardening method. Source: Adapted from [224].

The resulting mechanical properties of the final hot stamping components are induced by the phase transition occurring during the hot stamping process. Initially, the steel has an austenite microstructure after the the furnace stage. As displayed in Figure 4.5, a fast cooling rate leads to martensite, which is usually the desired microstructure for the aforementioned hot stamping applications. The principal reason is that martensitic microstructure provides very hard and strong components that are able to absorb a high quantity of energy without being significantly deformed. While bainite and ferrite microstructures can be obtained with slower cooling rates, these phases are generally more ductile and less strong than martensite, making them suitable for distinct applications.

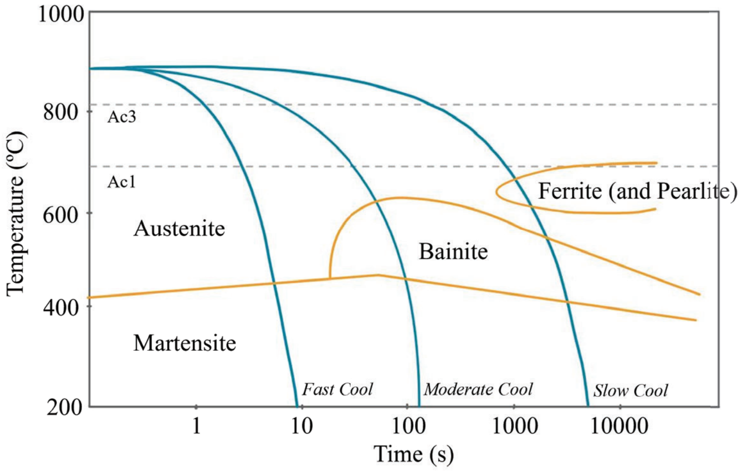


Figure 4.5: Different steel microstructures depending on the cooling rate during the hot stamping process.

Practice

Chapter 5

Machine Learning-based Surrogate Models Approaches in Industry 4.0

Building upon the theoretical foundation of surrogate modeling established in the previous chapters, in this chapter we delve into the thesis core of boosting the data generation process through the practical application of Machine Learning-based Surrogate Models (ML-based SMods) within the context of Industry 4.0. In this chapter we explore the development and implementation of three distinct SMod methods applied to real-world manufacturing challenges, including node reduction in injection moulding, mesh upscaling in HPDC and parameter interpolation in hot stamping. We introduce each method with a general description and then we validate it through its application in a specific Industry 4.0 use case.

5.1 Node Reduction

5.1.1 Description of the Method

The node reduction method consists in leveraging SMods to predict relevant variables in unobserved points within a certain geometric domain. Among the potential applications of this method, some of the most significant are the optimization of the sensor placement, the decrease of the mesh size in simulations and the reduction the cost of implementing expensive sensors in real processes.

The core concept of node reduction lies in training a SMod on a set of points within a geometry under different configurations, selecting some points as inputs and the rest as targets. Then, the SMod is utilized to predict values at the target points where data acquisition may be expensive or impractical. This approach reduces the reliance on high-fidelity data acquisition across the entire geometry, leading to significant gains in computational efficiency and resource optimization. A descriptive scheme of the method is displayed in Figure 5.1.

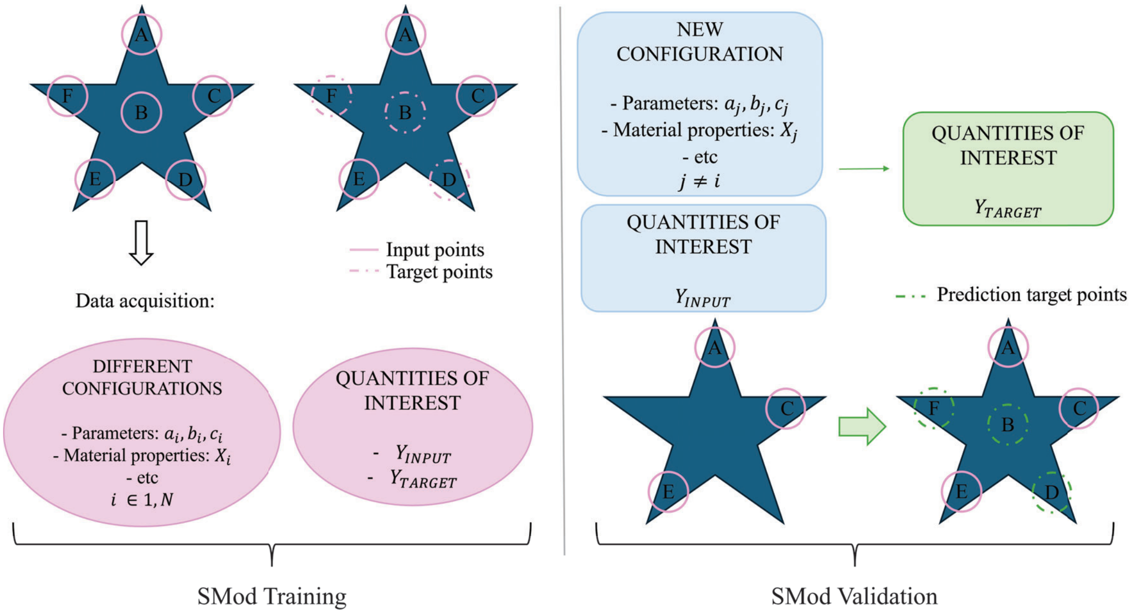


Figure 5.1: Schematic description of the node reduction method for surrogate modeling in a toy geometry.

The node reduction method can be broadly described in the following steps:

1. Data acquisition: Initially, it is required to obtain data from simulations (or experiments) at a set of points within the geometry under various configurations, boundary conditions, material properties or other relevant parameters that result in different values of the quantities of interest.

2. Data preparation: Designation of a subset of points as inputs and the remaining as target points where the variables of interest will be predicted. Furthermore, a preprocessing step may be required to prepare the dataset. The quantities of interest in the input points and other relevant features such as geometrical coordinates or configuration parameters are included as inputs for the SMod and the quantities of interest at the target points as outputs. Also, data should be separated into training and test sets containing different configurations.

3. SMod training: The prepared data is employed to train a suitable ML-based SMod. This model learns the complex relationship between the considered points, the geometry, configuration and the quantity of interest.

4. Prediction and validation of the SMod: The performance of model is evaluated against a separated test set. The unseen configurations are introduced and only data from the designated input points is available. The trained SMod predicts the quantity of interest at the target points.

Once the SMod is developed, the posterior implementation in a real system enable the use its predictions to represent the system behavior with fewer data points, to evaluate new configurations with minimal computational cost or to estimate quantities at location where direct measurements are difficult.

The node reduction method can be applied to multiple scenarios with a promising generalization potential. The idea of learning the intricate relationships between geometry, configurations and quantities of interest can be extended to a wide spectrum of problems and domains. Among its key advantages, it leads to an enhancement of data efficiency. By strategically selecting input points and training the SMod across a range of configurations, the method minimizes the need for extensive data acquisition for new configurations. This reduction in data requirements translates to significant savings in time, computational resources and experimental costs, making it particularly attractive for computationally expensive or experimentally challenging scenarios. In real-world systems, this supposes a reduced need for physical sensors, which can lead to substantial cost savings and simplified system design. By accurately predicting quantities at unobserved locations, node reduction allows for effective monitoring and control with fewer sensor deployments, minimizing the complexity and intrusiveness of the sensing infrastructure.

Moreover, node reduction provides a considerable boost in computational efficiency. Evaluating new configurations with the trained SMod is significantly faster than relying on full-scale simulations or conducting comprehensive experiments. This accelerated evaluation process facilitates rapid exploration of the design space, enabling more efficient design optimization and analysis. This is further amplified in simulation contexts where node reduction can allow a significant mesh coarsening. By accurately predicting quantities at unobserved points, the method enables the use of coarser meshes without compromising the overall accuracy of the simulation. This reduction in mesh density leads to substantial decreases in computational time and memory requirements, making complex simulations more tractable and efficient.

5.1.2 Practical Application

We have demonstrated the applicability of the node reduction method in the manufacturing process of plastic injection moulding, presented in Section 4.1. Concretely, in this practical use case, we explore the hybridization of physical phenomenological simulations of the plastic injection moulding process with ML prediction techniques to develop a methodology to increase the simulation efficiency. The goal of this SMod model is to combine ML predictions with a reduced number of simulation nodes with the goal of describing a more complete representation of the system. In the proposed study the best points to simulate in order to obtain a reliable description of the process are identified. Additionally, an adapted backward selection methodology is used for node selection task.

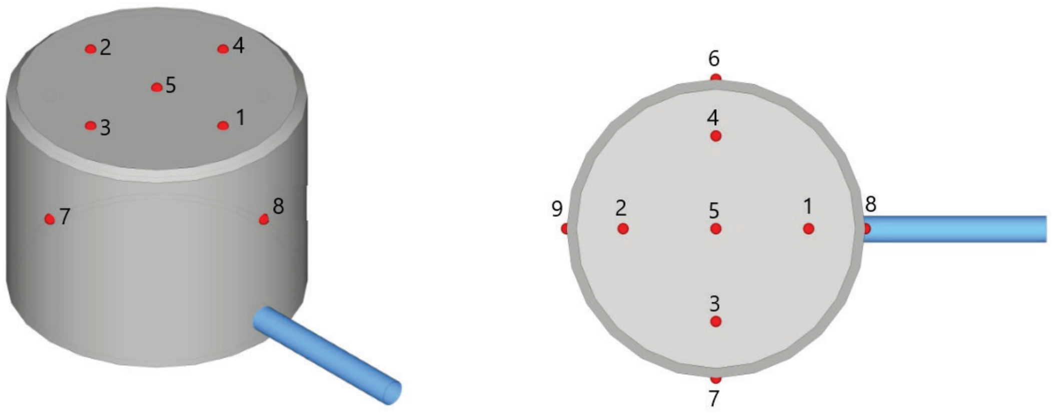
This practical application of the node reduction method is presented on the work in [1].

5.1.2.1 Plastic Injection Moulding Simulation

In this use case, the objective is to increase the efficiency of data generation in plastic injection moulding process. For this reason, simulations of the process have been performed with the Moldex3D® mold flow analysis commercial software [225], based on the FE method. The studied part is a cap injected in a mould cavity. In Figure 5.2, the geometry of the cap is illustrated. Moreover, nine virtual or simulated sensors have been displayed in the geometry: SN1 - SN9. The sensors measure the cavity pressure evolution in these nine selected points, since it has been identified in the literature as one of the most relevant variables for the quality of the final product [213].

To extend the study to different conditions, several configurations have been generated changing two parameters of the simulations: the injection speed (v) of the plastic into the mould and the value of the packing pressure (PP), as shown in Table 5.1. These parameters have been modified around their nominal working values v_{ref} and PP_{ref}. The result are 15 configurations representing the conditions of a real-environment. The simulations last in the range of 40 to $50min$ per configuration.

The output of the simulations is the evolution of the cavity pressure exerted by the melted plastic in the nine points where the virtual sensors are located. Initially, the sampling time in Moldex3D® for the pressure data is different for each simulated configuration. In order to homogenize them, we apply a resampling each

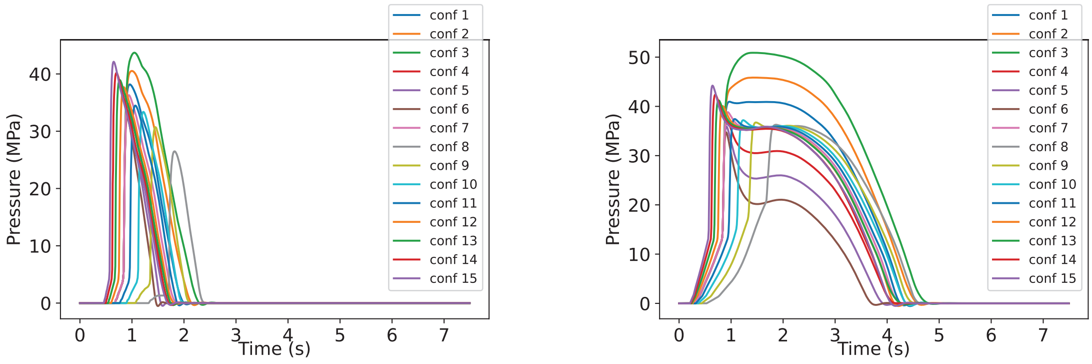


(a) (b)

Figure 5.2: Sketch of the cap form different perspectives. The melted plastic enters through the blue runner. The red dots are the sensed points and the numbers are used to identify the different pressure sensors.

Table 5.1: Values of the packing pressure (PP) and the injection speed (v) for the 15 generated configurations. The marked configurations (*) are used for testing.

Configuration	PP **(MPa)**	v **(mm/s)**
conf 1	$PP_{ref} + 0.1PP_{ref}$	v_{ref}
conf 2*	$PP_{ref} + 0.2PP_{ref}$	v_{ref}
conf 3	$PP_{ref} + 0.3PP_{ref}$	v_{ref}
conf 4	$PP_{ref} - 0.1PP_{ref}$	v_{ref}
conf 5*	$PP_{ref} - 0.2PP_{ref}$	v_{ref}
conf 6	$PP_{ref} - 0.3PP_{ref}$	v_{ref}
conf 7	PP_{ref}	v_{ref}
conf 8	PP_{ref}	$v_{ref} - 8$
conf 9	PP_{ref}	$v_{ref} - 6$
conf 10	PP_{ref}	$v_{ref} - 4$
conf 11*	PP_{ref}	$v_{ref} - 2$
conf 12	PP_{ref}	$v_{ref} + 2$
conf 13	PP_{ref}	$v_{ref} + 4$
conf 14*	PP_{ref}	$v_{ref} + 6$
conf 15	PP_{ref}	$v_{ref} + 8$



(a) (b)

Figure 5.3: Pressure evolution in the simulation of the process for all the configurations in (a) SN5 and (b) SN8.

0.01 s to homogenize all the configurations, obtaining 750 samples per configuration. In Figure 5.3, the pressure evolution behavior is shown for all the different configurations in two of the sensed points. It is worth noting the heterogeneity of the results for different sensors and configurations.

5.1.2.2 Experimental Setup and Methodology

In the following, we present the experimental set-up that aims to understand up to which extend simulation nodes can be replaced with ML predictions. To this end, two experiments are carried out: An individual assessment of each sensor and a global assessment for the complete set of sensors. In this practical application, we adopt a methodology that systematically replaces simulated sensors by predicted ones by means of a backward search. In particular, the sensors selected for testing purposes are such that they define the worst case scenario, ensuring that any other choice would achieve better scores.

Experimental Setup From the nine different simulated sensors, three of them will be used for assessing the quality of the prediction system. The three selected target pressure sensors will be predicted using the remaining simulated sensors data. Additionally, to ensure the generalization of the algorithm on independent test data, the set of 15 configurations is split in 11 configurations for training and a 4 configurations for testing (see values in Table 5.1 marked with a star). The test configurations have been chosen to be intermediate values of the simulation parameters (injection speed and packing pressure).

To select the target testing sensors, we use the concept of similarity between the samples of the time series of the pressure sensors. Then, we compute the mean similarity [226] between the pressure curves of the sensors $\mathrm{tsim}(X, Y)$ as follows,

$$\mathrm{tsim}(X, Y) = \frac{1}{n}\sum_{i=1}^{n} \mathrm{numSim}(x_i, y_i) \tag{5.1}$$

where $X = x_1, ..., x_n$ and $Y = y_1, ..., y_2$ are time series of two pressure sensors and $\mathrm{numSim}(x_i, y_i) = 1 - \frac{|x_i - y_i|}{|x_i| + |y_i|}$ is the similarity between two samples in the same instant of time. The operation range of $\mathrm{tsim}(X, Y)$ lies in the interval $[0, 1]$. $\mathrm{tsim}(X, Y) = 1$ refers to two identical pressure curves.

Figure 5.4 is the result of computing the mean similarity between the different sensors averaging for the 4 test configurations. This result drives the selection of

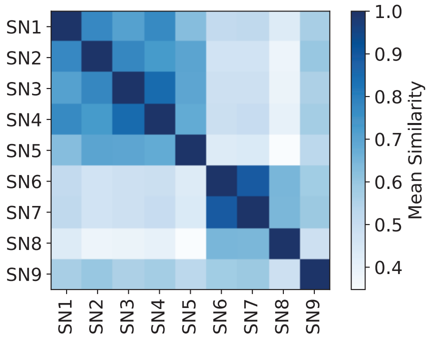


Figure 5.4: Mean Similarity Matrix averaged over the 4 test configurations. The spatial regions of the cap can be differentiated in this matrix. The first 5 sensors are in the superior part of the cap and the last 4 in the lateral.

the less similar sensors for the prediction, with the intention to use non-trivial cases to validate the presented methodology.

Comparing Figure 5.4 with the localization of the sensors in the cap (Figure 5.2), we can observe the symmetry relations displayed in the Mean Similarity Matrix between the sensors SN3 and SN4 and the sensors SN6 and SN7. These sensors will be discarded for prediction because they will not suppose a difficulty for the algorithm, that will use the corresponding symmetric sensor to obtain a very good prediction. The remaining sensors are not symmetric due to the position of the runner. The three sensors with less similarity are SN5, SN8 and SN9. Due to the central position of SN5 in the cap, we are interested in the real value of the cavity pressure in that point. We prefer not to include SN5 in the set of target sensors and replace it with SN2, which has no symmetries in the cap geometry. Summarizing, in a first approach, a SMod that predicts the sensors SN2, SN8 and SN9 using the values of the rest of pressure sensors will be developed. As mentioned, this defines a worst case scenario.

Backward Selection Methodology Previously, we have defined a set of three target sensors that will be predicted using the data from the six remaining sensors. This means that in future simulations six points will still have to be sensed. In order to minimize the number of sensed points for future simulations and explore to which extend these can be replaced by ML predictions we propose to use a methodology based on the technique of backward selection [227].

The technique consists in the elimination of the input features of a ML algorithm, using a metric that allows to decide which feature is the best to drop in a greedy manner. Starting from a set of $k = 1, ..., M$ input features and $i = 1, ..., L$ test configurations, the elimination of features is carried out through the following iterative process:

1. Use the current number of features M to predict the target.
2. Compute the error metric for each of the test configurations, the Mean Squared Error (MSE) in our case.
3. For all the $k = 1, ..., M$ current features:
   (a) Predict the target without using the feature k.
   (b) Compute the error metric for all the test configurations.
   (c) Calculate $\text{diff}_i^{M-1,M}(k)$, the error difference with and without feature k for each test configuration as follows,

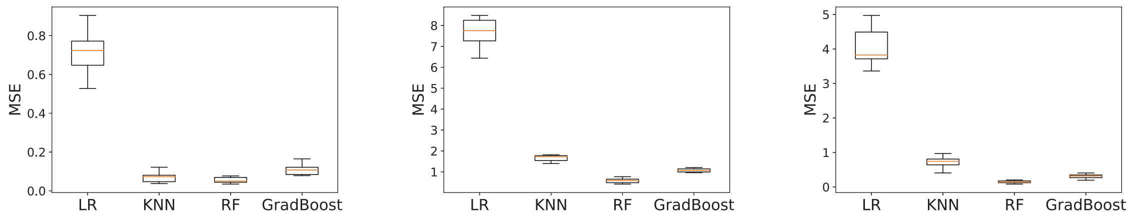
   $$\text{diff}_i^{M-1,M}(k) = \text{MSE}_i^{M-1}(k) - \text{MSE}_i^{M} \tag{5.2}$$

   (d) Perform the weighted average $t_{conf}^{M}(k)$ described in the following Equation 5.3 of the differences over the test configurations for the eliminated feature k.

   $$t_{conf}^{M}(k) = \sum_{i=1}^{L} c_i(k)\, \text{diff}_i^{M-1,M}(k) \tag{5.3}$$

   where $c_i(k) = \frac{\text{MSE}_i^M}{\sum_{j=1}^{L} \text{MSE}_j^M}$
4. Select the smaller value of $t_{conf}^{M}(k)$ and drop the corresponding feature k.
5. Repeat the process with the new set of features of size $M = M - 1$.

The process ends when the number of desired features is reached or when the error overcomes a given threshold. Observe that the value of $\text{diff}_i^{M-1,M}(k)$ may be negative if the elimination of the feature k improves the prediction algorithm performance. The proposed methodology takes into account the value of error metric for each test configuration to decide which is the best feature to drop, since it is preferable to optimize the predictions of the configurations that have a higher error.



(a) (b) (c)

Figure 5.5: Average Mean Squared Error (MSE) of a 10-Fold CV for algorithm comparison between LR, KNN, RF and GradBoost. Target predicted sensor: (a) SN2. (b) SN8. (c) SN9.

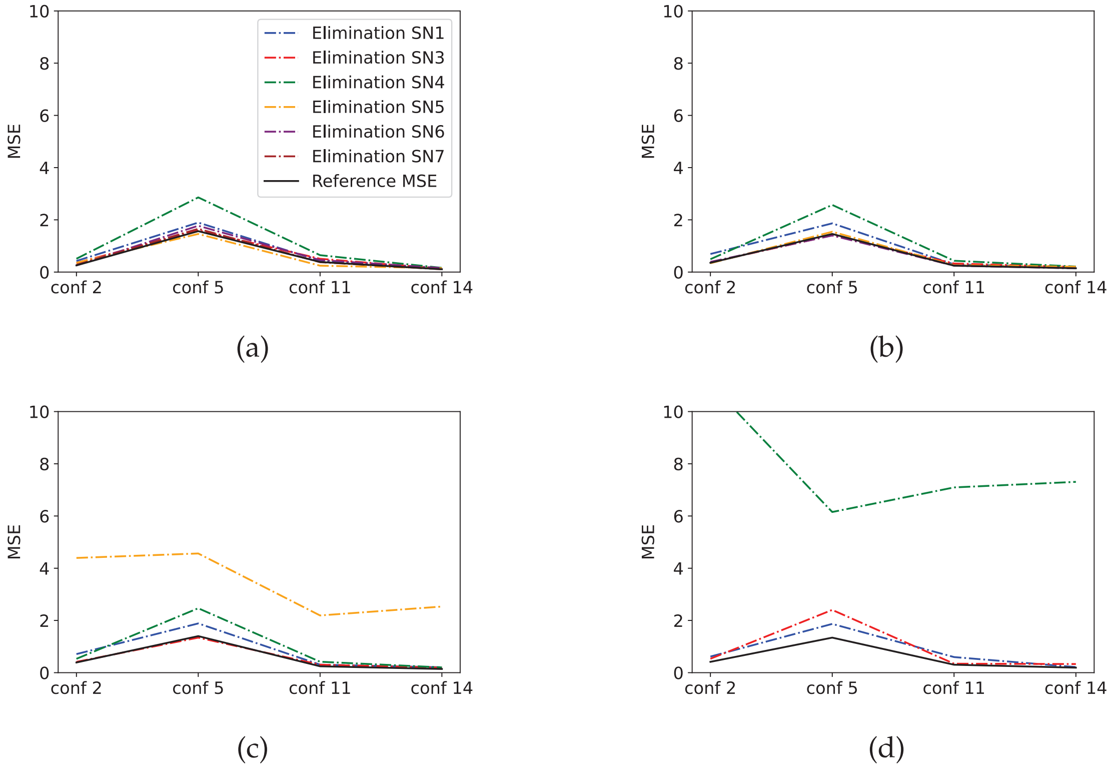
5.1.2.3 Results and Discussion

Baseline Prediction Results The reduction of the sensed points is realized with a ML-based SMod which is based on regression algorithm that uses as input data coming from a few locations to predict the rest of the points. Before that, we perform an algorithm comparison in order to know the prediction capability of some regression algorithms to all the available data. Therefore, we will randomly merge the data from all the configurations, obtaining a dataset composed by 750×15 samples and 9 features. Selecting a target sensor to predict and using all the others for training, we will implement a 10-Fold Cross-Validation (CV) [195] to choose a candidate algorithm.

Figure 5.5, shows the error performance comparison of four different regression techniques applied to the complete dataset. RF [228] achieves a lower error rate and will be used for the rest of the experiments as the regression algorithm of the SMod.

Individual Sensor Reduction Assessment The purpose of this study is to demonstrate the feasibility of achieving an important reduction of the number of sensed points without having a high impact in the prediction error. As explained in the experimental setup section, we will reduce the number input sensors used to predict the set of three target sensors, by applying the backward selection methodology presented in Section 5.1.2.2.

The different steps of the backward selection process are displayed in Figure 5.6, where the MSE for each test configuration is represented when we eliminate the input sensors. The curve of reference MSE does not suffer a relevant variation



(a)

(b)

(c)

(d)

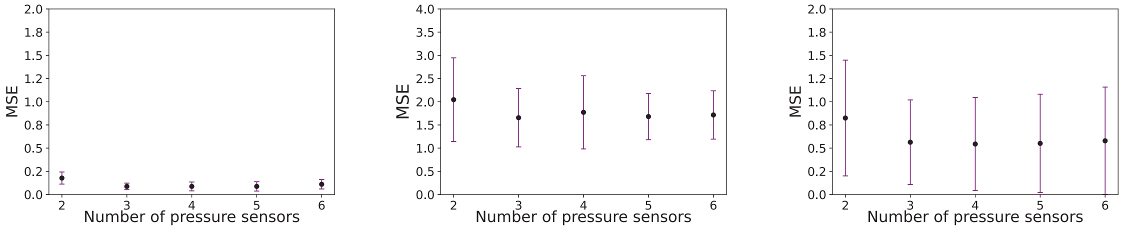
Figure 5.6: MSE in test configurations for target SN9. The black curve indicates the reference MSE from the previous step of the elimination process and the colored curves indicate the MSE with the drop of one of the sensors. (a) 5 sensor selection. (b) 4 sensor selection. (c) 3 sensor selection. (d) 2 sensor selection.

during the stages of the process, meaning that the prediction capability of the algorithm remains despite the discarded sensors. It refers to the MSE computed with M sensed points and it is used to evaluate the predictions with $M-1$ sensed points through the use of Equation 5.3.

Figure 5.7 shows the result of the selection process for each one of the target sensors. The evolution of the mean MSE of the 4 test configurations allows to identify a threshold in three input sensors. Below this threshold, the use of less input sensors induces the error metric to start having a relevant increase. By inspecting these values, Table 5.2 shows the three best input sensors for individually predicting each target sensor.

Table 5.2: Best input sensors for the corresponding target sensors.

Target sensor	**SN2**	**SN8**	**SN9**
Best input sensors	SN1, SN3, SN6	SN6, SN1, SN5	SN6, SN4, SN1



(a) (b) (c)

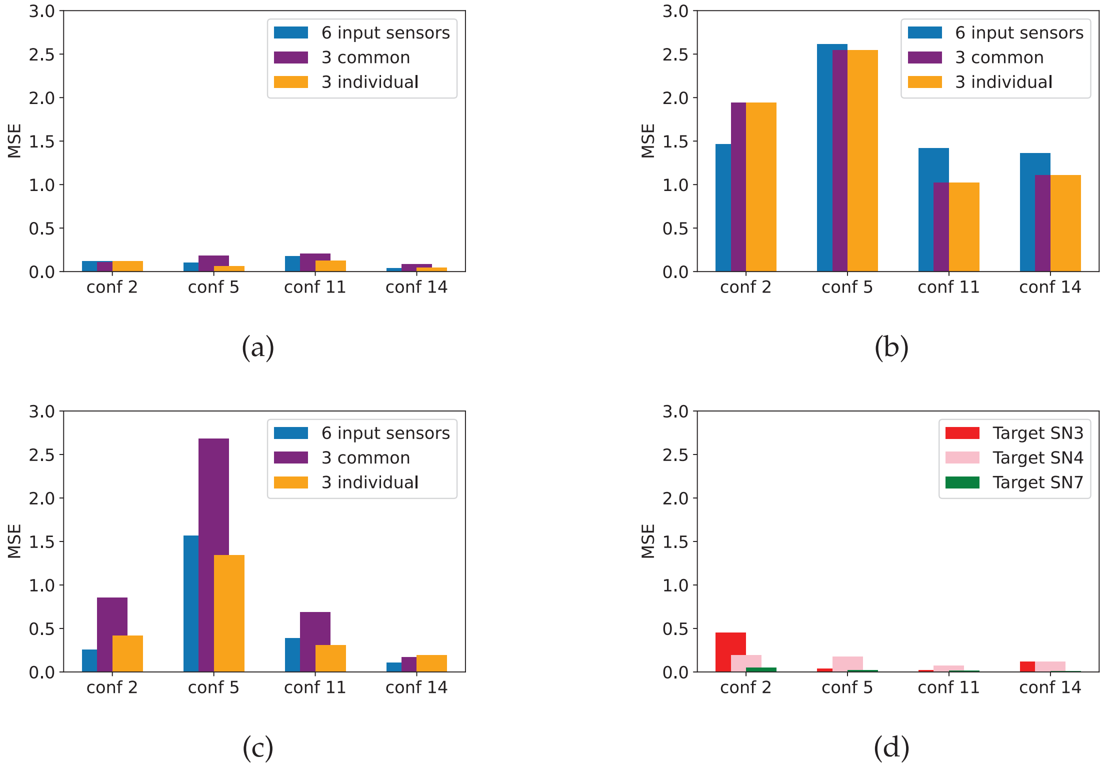
Figure 5.7: Evolution of the mean and the standard deviation over configurations with the number of input sensors used in the prediction. (a) SN2. (b) SN8. (c) SN9.

Global Sensor Reduction Assessment The last results open the possibility of reducing the number of sensed points, showing that each individual sensor can be predicted using 3 sensed points without a relevant effect in the error. However, the results show that different nodes require of different sensors for a good performance. In this subsection we consider whether a small common set of sensors may suffice for predicting all targets.

In order to do so, we will select the most repeated input sensors to predict all the target sensors. With the information of Table 5.2, we can identify that SN1 and SN6 are important for the prediction of the 3 target sensors. Additionally, we will choose the SN5, since it has a central position in the cap geometry (Figure 5.2). Accordingly, the final set of input sensors is formed by SN1, SN5 and SN6.

The defined final set of input sensors is used to predict the pressure of the target sensors. In Figure 5.8, the error metric MSE is compared when the prediction is done with 6 or 3 input sensors. If we use the individual set of 3 sensors of Table 5.2, we achieve a decrease of the error in most of the cases. Elseways, the use of the common set of sensors leads to a higher prediction error, but it allows to reduce the number of sensed points in the simulations. Moreover, the common set of sensors is not only able to predict the target sensors but also it yield good predictions for all the remaining sensors that the methodology has discarded. Figure 5.8d shows these results and it demonstrates the generalization capability of the proposed methodology. This remarks that, in the framework of industrial problems, it can be useful to include expertise knowledge combined with AI tools to help the global system performance, as shown including SN5 in the common set of input sensors.

As a final result, Figure 5.9 shows simulated and predicted pressure curves of the target sensors for a certain configuration. As observed in Figure 5.8, the vari-



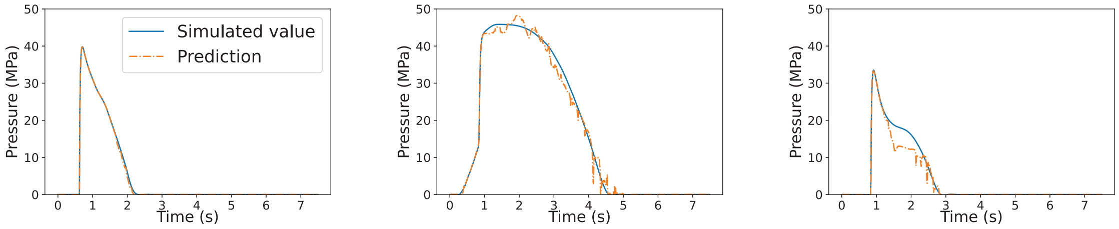
(a)

(b)

(c)

(d)

Figure 5.8: MSE comparison using 6 input sensors, the final set of 3 input sensors or the best 3 selected sensors for each individual target sensor. Target sensor: (a) SN2. (b) SN8. (c) SN9. (d) MSE of the prediction of the sensors SN3, SN4 and SN7 as target with the selected set of 3 input sensors.



(a) (b) (c)

Figure 5.9: Comparison example between the simulated and the predicted temporal evolution of the pressure using the 3 selected input sensors. (a) Target SN2, conf 14. (b) Target SN8, conf 2. (c) Target SN9, conf 5.

ability of the accuracy of the prediction is highly dependent on the target sensor and the test configuration. Figure 5.9c shows the worst case prediction. Regardless of not being a perfect prediction, some relevant features of the injection moulding process such as the maximum value of the curve or the duration of the different stages are correctly characterized [213]. This result is of high importance as in manufacturing processes, the control a few relevant process variables is enough to determine the global performance of the system.

5.1.2.4 Insights from Node Reduction Surrogate Model in Plastic Injection Moulding

This application case investigated the potential of integrating simulations with ML predictions through a SMod, specifically implemented in the plastic injection molding process. The primary objective was to enhance efficiency in the simulation process by strategically reducing the number of sensor nodes required to generate accurate predictions.

The key takeaway from this investigation is the successful demonstration that sensor nodes can be effectively replaced by predicted values using a minimal set of real simulated data. This approach proves robust even when process parameters deviate from those used in the training dataset, highlighting the generalization capability of the SMod. While reducing sensed nodes might occasionally lead to less accurate predictions in specific configurations, the critical process variable values such as maximum pressure and timing of process stages remain reliably preserved. This is crucial because these values heavily influence the overall system performance.

Our findings strongly suggest that ML-based SMods, particularly through the node reduction method, can significantly streamline data acquisition and analysis in manufacturing processes. Moreover, they can provide insightful recommendations on the sensorization of real processes. This aligns with the broader theme of this thesis, which emphasizes the value of SMods in Industry 4.0 settings. By reducing the reliance on extensive simulations and physical sensors, node reduction offers a pathway to more efficient and cost-effective process optimization and control.

This real-world application showcased several significant advantages of the node reduction approach. The investigation successfully reduced the number of simulated sensors required, highlighting the potential for significant cost savings in real-world manufacturing processes by minimizing the number of physical sensors needed. The method also allowed for the generation of accurate predictions with a reduced number of nodes, leading to faster data acquisition and analysis, which can improve the overall efficiency of manufacturing processes by decreasing simulation times through the use of coarser meshes. Finally, despite using fewer sensor nodes, the SMod created using the node reduction method maintained the accuracy of critical process variable predictions, ensuring reliable process monitoring and control.

The node reduction method is a promising approach for improving the efficiency and accuracy of manufacturing processes. The method has the potential to be applied to a wide range of industrial scenarios.

5.2 Mesh Upscaling

5.2.1 Description of the Method

The mesh upscaling method is based on the implementation of SMod to enhance the accuracy of simulations performed on coarse meshes. In many engineering and scientific applications, simulations involving complex geometries or intricate physical phenomena necessitate the use of fine meshes to achieve accurate results. However, there is a direct relation between the number of mesh elements and the computational resources and time required for the simulation. Mesh upscaling addresses this challenge by utilizing SMods to bridge the gap between coarse and fine mesh simulations, enabling accurate predictions while reducing the computational burden.

The fundamental idea of the method involves training a SMod on data generated from both coarse and fine mesh simulations, employing the same configurations and parameters but changing the mesh. Then, the SMod is able to learn the inner relationship between the quantities of interest obtained from the two mesh resolutions. Afterwards, once trained, the aim of the SMod is to predict the fine mesh solution based on the coarse mesh simulation results, effectively "upscaling" the coarse mesh solution to the desired accuracy. Hence, the main benefit of the proposed surrogate modeling approach is that permits to obtain high accuracy results through low-cost coarse mesh simulations, directly contributing to the thesis objective of accelerating data generation in industrial environments. An illustrative pipeline of the mesh upscaling method is shown in Figure 5.10.

The general pipeline of the mesh upscaling method is comprised in these different stages:

1. Multi-resolution data acquisition: At the first stage, simulations on both coarse and fine meshes are conducted for a set of configurations. In general, it exists a gap between the coarse and mesh results and a huge difference in computational efficiency. The fine mesh simulations serve as the ground truth for the SMod.

2. Data preparation: Extraction of the relevant quantities of interest from both coarse and fine mesh simulation results. These quantities could include physical measurements, derived quantities or any other relevant output variables. After, organize the data into a training dataset, where the input features are

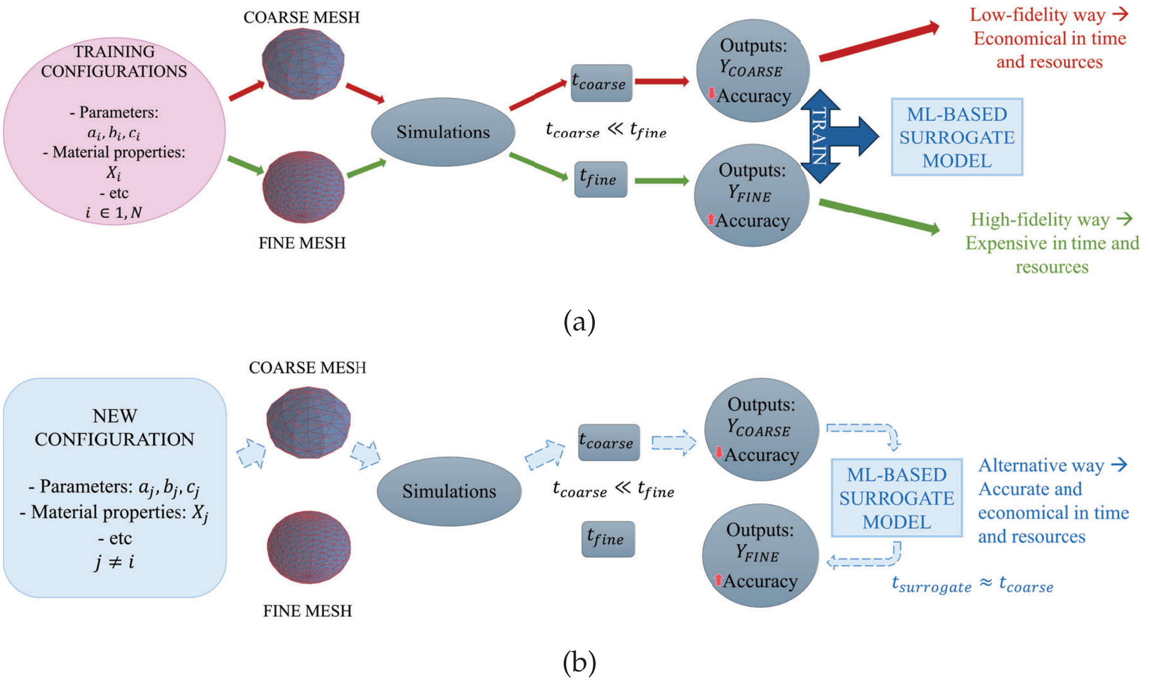

COARSE MESH
TRAINING CONFIGURATIONS
- Parameters: a_i, b_i, c_i
- Material properties: X_i
- etc
$i \in 1, N$
FINE MESH
Simulations
t_{coarse}
t_{fine}
$t_{coarse} \ll t_{fine}$
Outputs: Y_{COARSE} Accuracy
Outputs: Y_{FINE} Accuracy
TRAIN
ML-BASED SURROGATE MODEL
Low-fidelity way → Economical in time and resources
High-fidelity way → Expensive in time and resources
(a)
NEW CONFIGURATION
- Parameters: a_j, b_j, c_j
- Material properties: X_j
- etc
$j \neq i$
COARSE MESH
FINE MESH
Simulations
t_{coarse}
t_{fine}
$t_{coarse} \ll t_{fine}$
Outputs: Y_{COARSE} Accuracy
Outputs: Y_{FINE} Accuracy
ML-BASED SURROGATE MODEL
Alternative way → Accurate and economical in time and resources
$t_{surrogate} \approx t_{coarse}$
(b)

Figure 5.10: Schematic description of the mesh upscaling method pipeline. (a) SMod training stage. (b) SMod prediction and validation stage.

the quantities of interest obtained from the coarse mesh simulations and the output targets are the corresponding quantities from the fine mesh simulations.

3. SMod training: The constructed dataset is utilized to train a ML-based SMod that learns the mapping between the coarse and fine mesh simulations.

4. Prediction and validation of the SMod: To assess the accuracy of the trained SMod, test data containing an unseen configurations is selected. Then, simulations using the coarse mesh for these unseen configurations are performed. Next, the created SMod is utilized to predict the fine mesh solution based on the coarse mesh simulation results and the predictions are compared with the ground truth full-scale fine mesh simulations.

The posterior application of the resulting SMod supposes an important boost of the simulation time, the design phase and the fast analysis of a manufacturing process, since it permits to obtain fast accurate predictions for new configurations without the need of computationally expensive fine mesh simulations and optimize the exploration of design parameters using upscaled coarse mesh solutions.

Due to the wide range of processes that use meshes in their simulation models, the mesh upscaling method is a very versatile option for building SMods of different industrial applications. The method fulfills the aim reducing the computational burden in the simulation of complex manufacturing processes, achieved thanks to a significant reduction of full-scale fine mesh simulations. Instead, using less precise but faster simulations as a baseline, SMods are able to predict accurately the values of interest of the system for new scenarios. Actually, the apported benefits are similar than the node reduction method (see Section 5.1.1), but facing the problem from a different angle. These advantages include more efficient data generation, exploration and the diminish of the computational and time impact of the simulations. The current approach is more centered to be applied directly to simulation models, while the node reduction method can also be implemented in real-world scenarios.

5.2.2 Practical Application

Following the introduction of the mesh upscaling method in the previous section, we now explore its practical potential through an implementation in the HPDC process (described in Section 4.2). In particular, we test the method developing a ML-based SMod for HPDC simulations. The objective is to leverage a simplified and more efficient simulation model as a baseline for exploring the HPDC system, thereby significantly reducing computational costs and accelerating analysis. A ML-based SMod of HPDC simulations is created to correct the results of this low-fidelity simulation model, bringing the outputs closer to the ground truth obtained from high-fidelity simulations. This translates to a substantial acceleration in simulation time, enabling faster analysis and exploration of the HPDC process compared to the computationally expensive, high-fidelity model. In fact, this study has been realized with the final goal that the resulting ML-based SMod can support a DT of the HPDC process, boosting the response of the virtual part of the DT, as explained in [5].

5.2.2.1 High Pressure Die Casting Simulation

Characterization The current use case focus on reducing the temporal impact of the high-fidelity HPDC simulations. A numerical simulation model has been built using Altair® Inspire™ Cast [229], a specific software to simulate metal injection processes in liquid phase. It allows to obtain results during all the stages of the HPDC process: the prefill, the filling process of the die and the posterior metal solidification to get the profile of the final part.

The studied part is a tray of the $AlSi_{10}MgMn$ alloy, which geometry is displayed in Figure 5.11. The principal process parameters considered in the current simulation model are:

- Initial temperature of the die (T_{die}): It is the temperature of the die when the process starts.
- Initial temperature of the metal (T_{metal}): It refers to the temperature of the liquid metal at the start of the process.
- 1st phase velocity (v_1): Plunger initial speed until the metal starts to fill the die cavity.

- 2nd phase velocity (v_2): Plunger second speed, which is sharply increased with respect to v_1 when the metal starts to fill the die cavity.

Besides, the output of the simulation model and our quantity of interest is the temperature in the geometry of the part. Concretely, in the simulation model, we created 678 virtual sensors distributed in the geometry of the part and the die. Therefore, the simulation reproduces the temperature in these virtual sensors during the filling and solidification phases. The nomenclature used for the output temperatures is T_i, where $i = 1, ..., 678$.

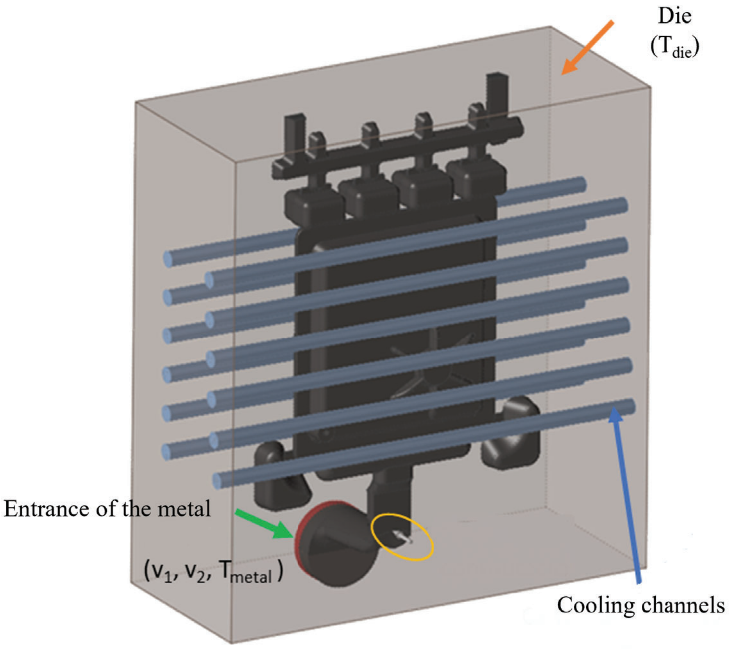
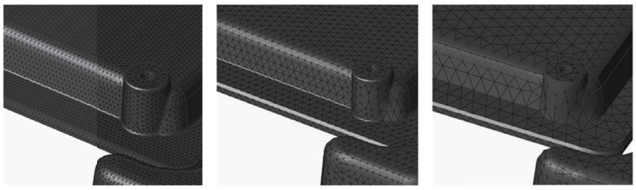


Figure 5.11: Geometry of the simulated part. The cooling channels, the mould and the entrance of the metal are indicated along with the most relevant process parameters (T_{die}, T_{metal}, v_1 and v_2.)

Finally, the simulations are separated in two phases: the filling phase, which includes the 1st (prefill) and 2nd (filling) stages explained in Section 4.2; and the solidification phase, which comprises the 3rd (consolidation) stage.

Study of the Mesh Sensitivity Numerical simulation models require designing the system to be characterized, as well as defining all its properties and the calculation mesh using appropriate simulation software. The mesh size is a determining parameter in the final quality of the result and in the computation time. The finer the mesh, the higher the accuracy, but also the longer the computation time.

To quantify this tradeoff between simulation time (or mesh density) and accuracy, a mesh sensitivity study has been carried out. The results presented in Figure 5.12, show that for this system the element size plays a fundamental role in the computation time and also in the accuracy of the results obtained. on the upper side, a comparison between the different meshes can be observed, where the lack of precision of the coarser meshes with respect to the fine one is reflected in the value of the temperatures of the part and the mold at specific instants of time. In the lower part of the figure, we can see the remarkable time reduction that using a coarser mesh implies. Moreover, an important difference is noticed in the evolution of the temperature of a given virtual sensor depending on the mesh used, as highlighted in Figure 5.13.



	Fine	Medium	Coarse
# Elements	2.30 M	0.94 M	0.50 M
Temperature part (2.2 s)	525.9 ºC	508.8 ºC (3.26%)	489.3 ºC (6.97%)
Temperature die (0.5 s)	278.0 ºC	253.9 ºC (8.67%)	231.8 ºC (16.62%)
Computation time	650.7 min	77.7 min (88.06%)	11.3 min (98.27%)

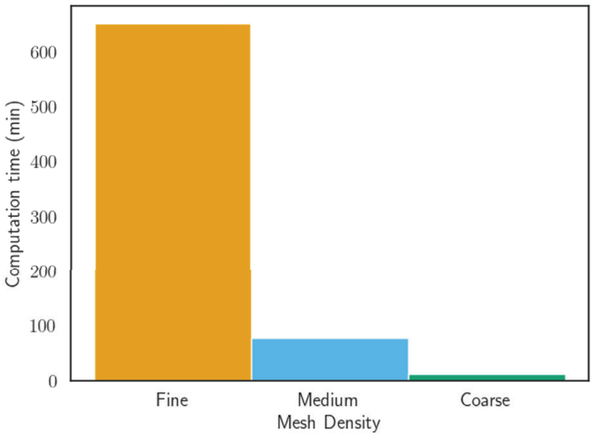
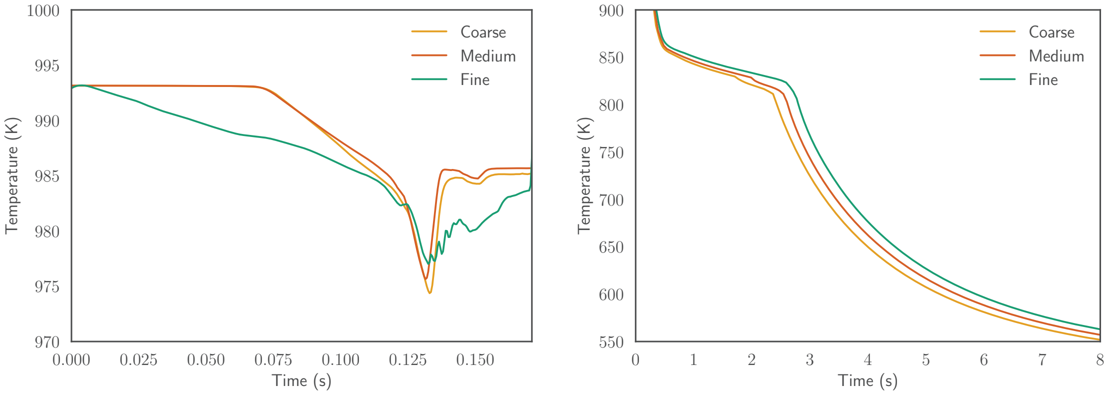


Figure 5.12: Study of the mesh sensivity.



(a) (b)

Figure 5.13: Temperature evolution evaluated in a given point T_{33} of the part for the differents mesh densities shown in Figure 5.12. (a) Filling phase. (b) Solidification phase.

5.2.2.2 Construction of the ML-based SMod

Methodology As observed in Figure 5.12, the size and number of elements used in numerical simulations have a significant impact on the time and accuracy of the simulations. Consequently, there is a trade-off between simulation speed and accuracy.

As mentioned, the final goal of this practical application is to integrate the final ML-based SMod into a DT of the HPDC process. Hence, regarding DTs, it is essential for the model response to be fast in order to know the current behavior in real-time, or at least, in the shortest possible time. In addition, a fast response favors the exploration of different scenarios through simulations in order to apply corrective measures if necessary. Unfortunately, we have checked that the simulation model offering the fastest calculation speed is the least accurate.

The simulation model reproduces the temperature and pressure values at 678 points in the mold geometry during the filling and solidification processes, and as shown in Figure 5.13, there are significant differences in the accuracy of the results between the fine mesh (the most accurate) and the medium and coarse meshes (the least accurate) in both filling and solidification phases.

To solve this problem, a ML-based SMod is proposed, with the aim of improving the accuracy provided by the simulation model while giving a fast response.

The SMod is based on several SL models, specifically regressors, which are capable of predicting the temperatures curves at the 678 points of the geometry with an accuracy similar to that of the fine mesh, using as inputs the values obtained with the simulations made with the coarse mesh.

In this way, ML models are trained using a dataset consisting of simulation data within a defined range of parameters. These simulations are performed using the coarse mesh as input data and the fine mesh as data to be predicted for each of the parameter configurations.

In Figure 5.14 we can see how the training of the ML models that form the SMod works in order to predict the evolution of temperature over time, as desired. For each of the 678 virtual sensors displayed in the geometry, the models use the speed configuration, time and temperature of same point in the coarse mesh simulation as input. The target variable to be predicted is the temperature given by the fine mesh simulation. In this way, the SMod comprises 678 regressors to predict the temperature of each virtual sensor.

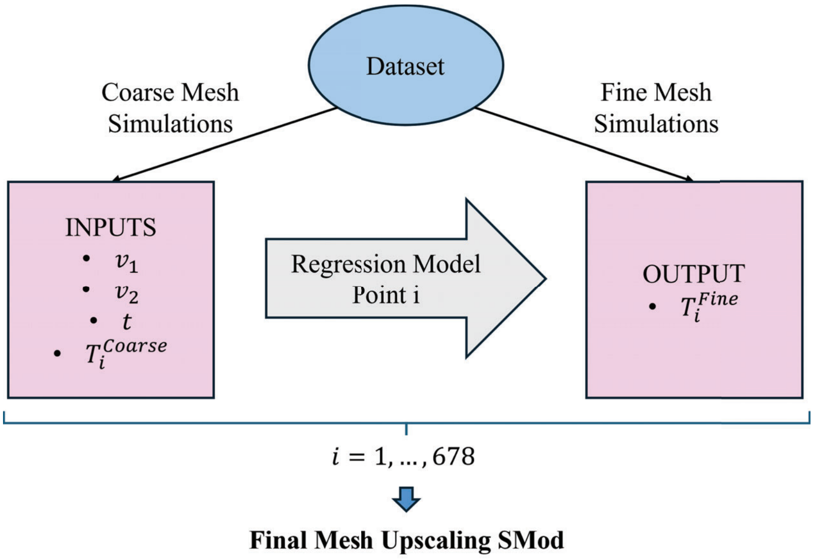


Figure 5.14: Schematic representation of the training process of the SMod to predict T_i.

Dataset Generation The simulations that form the training dataset have been chosen with the objective that the models can interpolate and predict results for unseen parameter configurations. The parameters that are modified are the 1st phase speed and the 2nd phase speed, since both time and temperature are results of the simulation itself. On the one hand, Table 5.3 displays the ranges of the velocities in the considered HPDC process, which have been defined based on the limitations of a real HPDC machine. On the other hand, T_{die} and T_{metal} are considered constant in this study. Additionally, a test dataset has been used to evaluate the models on new configurations.

The configurations used in the simulations that form these datasets are presented in Table 5.3 and Table 5.4. It is expected that the variability of v_1 and v_2 in the training set is enough to enable the SMod to cover the entire configuration domain.

Table 5.3: Range of values of the velocities in the simulations.

	Experiments
v_1 (m/s)	$0.3 - 0.5$
v_2 (m/s)	$1.5 - 2.5$

Table 5.4: Values of the velocities in the 15 configurations comprising the training dataset

Sim.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$\boldsymbol{v_1}$ **(m/s)**	0.3	0.35	0.4	0.45	0.5	0.3	0.35	0.4	0.45	0.5	0.3	0.35	0.4	0.45	0.5
$\boldsymbol{v_2}$ **(m/s)**	1.5	1.5	1.5	1.5	1.5	2.0	2.0	2.0	2.0	2.0	2.5	2.5	2.5	2.5	2.5

Table 5.5: Range of values of the velocities in the simulations.

Sim.	1	2
$\boldsymbol{v_1}$ **(m/s)**	0.37	0.42
$\boldsymbol{v_2}$ **(m/s)**	2.3	1.8

Generating these two datasets required approximately 8 days of simulation time, $\approx$ (650 min + 11 min) $\times 17$ simulations.

5.2.2.3 Results and Discussion

Model Choice To decide which type of regressor will be used for the ML learning models that compose the SMod, we have performed 5-Fold Cross-Validation [195], which allows us to evaluate the performance of several regression algorithms using data from the training set.

In Figure 5.15, the negative MAE (Mean Absolute Error) values of the different regression algorithms after the 5-Fold Cross Validation are represented. The MAE

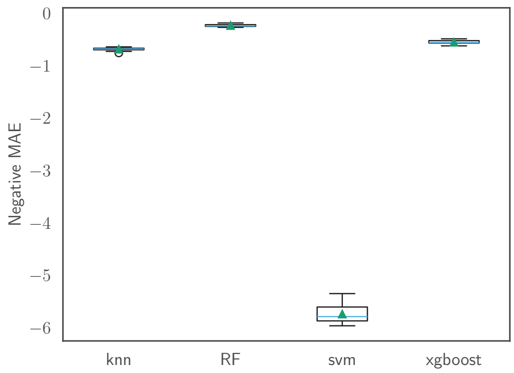


Figure 5.15: 5-Fold CV to evaluate 4 different regressions for the prediction of T_i.

is used as a metric to choose which is the best model. As a conclusion, the RF is established as the regression algorithm that will be used, since it is the one that has obtained the best metric.

Surrogate Model Performance Once the ML models are trained using the training dataset, the performance of the SMod is evaluated on the test dataset. The results exposed in Figure 5.16 prove that the implementation of the SMod supposed a decrease of the error done by the coarse mesh simulations. Concretely, the computation of the average of the MAE along the 678 virtual sensors results in MAE$_{SMod} = 1.01$ K and MAE$_{Coarse} = 3.82$ K. Therefore, the SMod reduces a factor of 3.7 the MAE of the coarse mesh simulations. This improvement can also be observed in Figure 5.17, where the comparison between the SMod prediction of a temperature curve for a specific point T_{33} and the simulated curve using the fine and the coarse mesh.

The fine mesh simulations for the test dataset required approximately 1300 min to run, while the SMod generated results in approximately 5.5 min. Including the 22 min required for the coarse mesh simulations, which serve as input to the SMod, the total time is approximately 27.5 min. This demonstrates that implementing a ML-based SMod can significantly reduce the computational expense while maintaining the accuracy of fine mesh simulations.

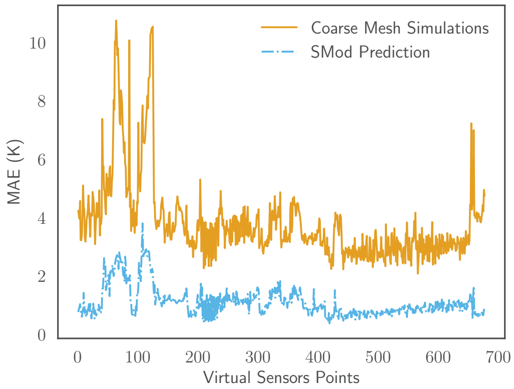


Figure 5.16: Comparison of MAE of the SMod prediction on the test dataset against the coarse mesh simulation results for all the 678 virtual sensors.

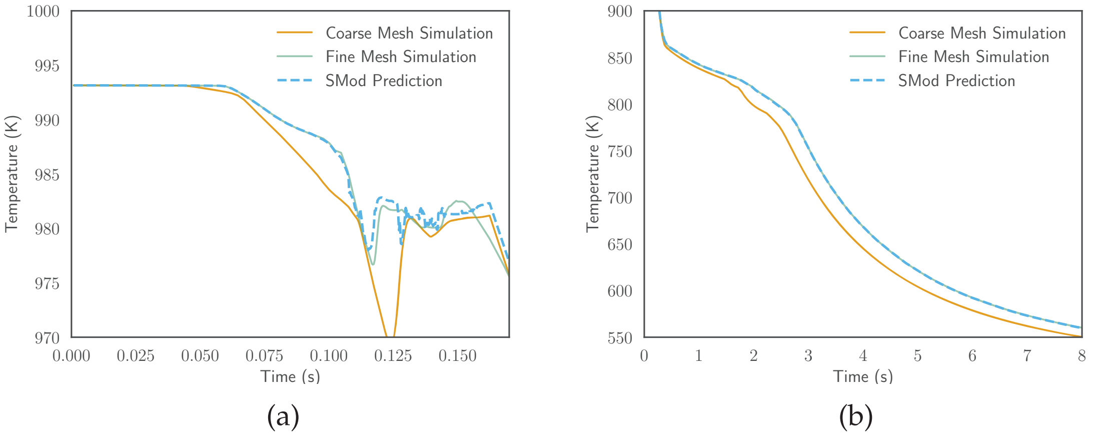


Figure 5.17: Comparison of the predictions of the SMod and the simulations of the coarse and fine mesh of the temperature evolution evaluated in a given point T_{33} of the part. The displayed case correspond to the test dataset ($v_1 = 0.37$, $v_2 = 2.3$). (a) Filling phase. (b) Solidification phase.

5.2.2.4 Insights from Mesh Upscaling Surrogate Model in High Pressure Die Casting

The current application case successfully demonstrated the effectiveness of the mesh upscaling method in the context of the HPDC process. By leveraging a SMod trained on data from both coarse and fine mesh simulations, the method enabled accurate predictions of quantities of interest while significantly reducing computational time. This approach has the potential to boost simulation workflows in industrial settings, allowing for faster analysis, design optimization and exploration.

The key insights gained from applying the mesh upscaling method for surrogate modeling in this case study include improvements in computational efficiency, accuracy, generalization and applicability. In particular, the method has proven to be an effective approach to diminish the mesh density and the simulation time while keeping the accuracy. The method generates a SMod able to correct the results given by a coarse mesh into high-fidelity results provided by fine mesh simulations. This is a crucial advantage in industrial scenarios, where fast response times are essential for real-time decision-making and computational resources are often limited. In this sense, the created SMod has shown the ability to generalize to unseen configurations, indicating its potential applicability across a range of scenarios within the HPDC process, enhancing then the exploration of various operation conditions and design parameters.

The successful implementation of mesh upscaling in the HPDC process showcases its practical relevance and potential for adoption in real-world industrial settings. This is aligned with the thesis goal of bridging the gap between theoretical surrogate modeling techniques and their practical applications in manufacturing, driven by the aim of accelerating data generation and analysis in manufacturing processes.

Finally, as mentioned, the generated SMod have been used to support the implementation of a DT for the HPDC process [5].

5.3 Parameter Interpolation

5.3.1 Description of the Method

This section introduces a surrogate modeling technique called parameter interpolation, which aims to construct SMods that efficiently predict simulation results across a wide range of parameter values. In many engineering and scientific applications, exploring the impact of different parameters on a system behavior often requires numerous simulations which, added to the curse of dimensionality problem, can be computationally expensive and time-consuming. Parameter interpolation addresses this challenge by utilizing SMods to approximate the relationship between parameter values and simulation outputs, enabling rapid and efficient exploration of the parameter space.

The parameter interpolation is based on the concept of training a SMod on a limited set of simulation data generated at specific parameter values within the domain of interest. The SMod learns the underlying relationship between the input parameters and the corresponding simulation outputs. Once trained, the SMod can predict the simulation results for any combination of parameter values within the domain, effectively interpolating the system behavior across the parameter space. A descriptive representation of the parameter interpolation method is depicted in Figure 5.18.

The generation of SMods through parameter interpolation relies on the following steps:

1. Data acquisition: A set of representative configurations within the domain of interest are selected. The idea is that the input parameters of the simulation form a domain of configurations limited by their ranges of values. Afterwards, the selected configurations are simulated through numerical models and the output quantities of interest are obtained.

2. Data preparation: The data acquired from the simulations is processed and organized to feed the posterior SMod training. The input features are the parameter values such as initial conditions, material properties, process time, among others, and the outputs are the simulation results, like temperatures, pressures, deformations, etc.

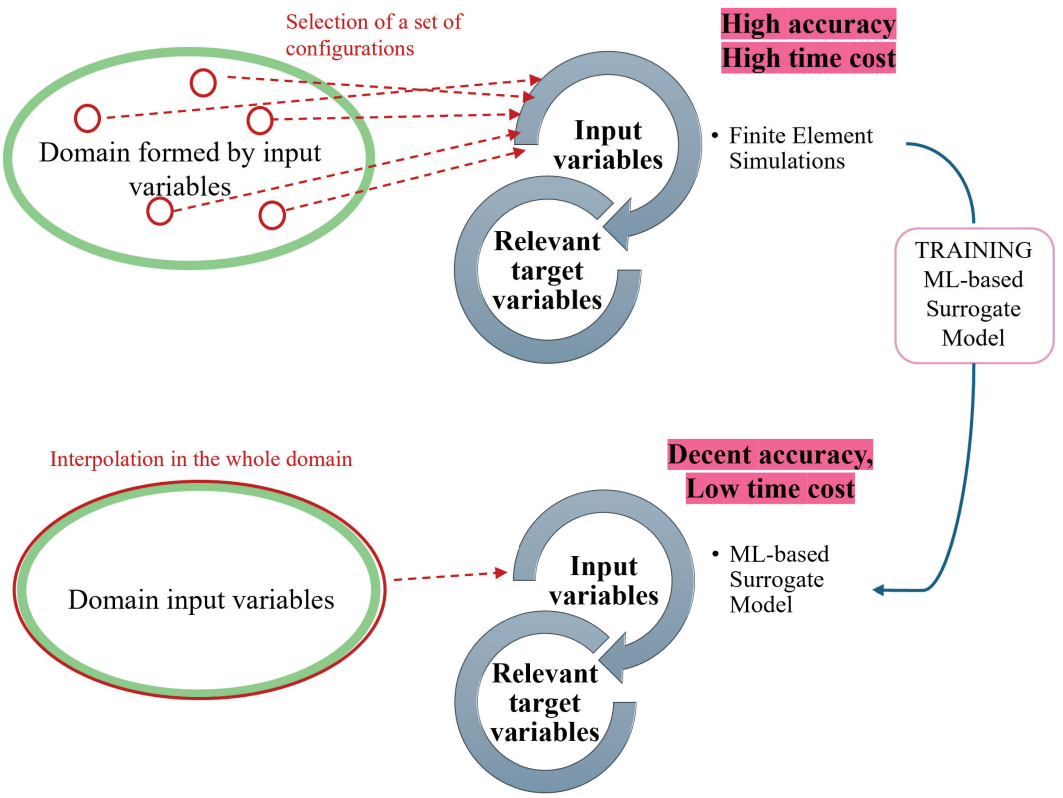

Selection of a set of configurations
High accuracy
High time cost
Domain formed by input variables
Input variables
Relevant target variables
• Finite Element Simulations
TRAINING ML-based Surrogate Model
Interpolation in the whole domain
Decent accuracy,
Low time cost
Domain input variables
Input variables
Relevant target variables
• ML-based Surrogate Model

Figure 5.18: Descriptive flowchart of the parameter interpolation method for surrogate modeling.

3. SMod training: The processed data is used to train a ML-based SMod that learns to map the input parameters to the outputs of interest from the simulation results.

4. Prediction an validation of the SMod: A set of unseen parameter values within the domain are chosen. Then, the trained SMod is employed to predict the outputs of these unseen configurations and the performance of the SMod is assessed comparing the predicted values with the simulation results.

The generated SMod can be utilized to efficiently explore the parameter space, predicting the system behavior for the whole domain of configurations without the need of computationally expensive simulations. This addresses a very important problem of manufacturing processes, which is the difficulty to generate data in a fast and sustainable way. By enabling rapid exploration of the parameter space, the method accelerates the process of gathering knowledge of the process and the impact of the parameter variations, opening the possibility of design optimization, process control and sensitivity analysis of the different parameters.

Again, the advantages of the method are analogous than the other presented methods and they follow the main objectives of this thesis dissertation, such as accelerated data generation, efficient and rapid prediction capabilities and reduction of the time and computational resources. However, it is important to highlight the fast response capability of the parameter interpolation method once the SMod model is trained. Unlike the previous methods, which might require additional data like geometry information or coarse mesh simulation results, parameter interpolation relies solely on the input parameter values. This independence democratize its application, making it a generic approach particularly well-suited for a wide of applications.

5.3.2 Practical Application

The digitalization of manufacturing processes offers great potential in quality control, traceability and the planning and setup of production. In this regard, process simulation is a well known technology and a key step in the design of manufacturing processes. However, process simulations are computationally and time-expensive, typically beyond the manufacturing-cycle time, severely limiting their usefulness in real-time process control. ML-based SMods can overcome these drawbacks, and offer the possibility to achieve a soft real-time response, which can be potentially developed into full close-loop manufacturing systems, at a computational cost that can be realistically implemented in an industrial setting. In this application case we explore the parameter interpolation method to build a SMod to analyze the case of the hot stamping of a steel sheet of 22MnB5. This hot sheet metal forming process involves a crucial heat treatment step, directly related to the final part quality. Given its common use in high-responsibility automobile parts, this process is an interesting candidate for digitalization in order to ensure production quality and traceability. We present a comparison of different data and model training strategies. We perform FE simulations for a transient heat transfer analysis with ABAQUS software [230] and we use them for the training data generation to effectively implement a ML-based SMod capable of predicting key process outputs for entire batch productions. The resulting final surrogate predicts the behavior and evolution of the most important temperature variables of the process in a wide range of scenarios, with a mean absolute error around 3 °C, but reducing the time four orders of magnitude with respect to the simulations. Moreover, the methodology presented is not only relevant for manufacturing purposes, but can be a technology enabler for advanced systems, such as DTs and autonomous process control with RL.

This practical application of the parameter interpolation method is presented on the work in [2].

5.3.2.1 Introduction

As previously discussed during this thesis the generation and accessibility of process data and its posterior treatment are crucial for gathering knowledge, optimizing and develop more advanced solutions in manufacturing processes. The possibility of performing real experiments of the processes to acquire data is often unfeasible, since they are costly and imply the waste of raw materials. In this context, manufacturing simulations, mainly finite element modeling, are the main source of data and knowledge without perturbing the real manufacturing system [26]. Simulations enable the exploration of new scenarios and configurations, as well as the modification of the experimental conditions.

In the specific case of press hardening, or hot stamping, FE modeling is a mature technology, commonly used in this type of industrial process. The application of FE modeling in hot stamping has been in widespread use since the early 2000s [231], and has steadily evolved to capture complex aspects such as thermal and mechanical interactions [231], plastic flows at different temperatures [232] and eventually phase transformations and the behavior of different microconstituents [233]. Currently, the simulation of hot stamping can be readily performed with commercial software [234] with industrially relevant results. However, despite their obvious advantage in front of real-world experiments, simulations are still a complex and time-intensive tool that cannot be realistically run in real time or used to generate very large libraries of data. In the previous Chapter 2 and 3 the problem of this inefficient data generation have been highlighted along with the limitation that it suppose for the development of technologies such as DTs or RL in industrial settings.

To overcome the mentioned limitations of the manufacturing simulations, it is possible to combine simulations and ML in a hybrid approach to build a highly efficient model which acts as a SMod of the simulations. Despite the simplicity of the model, the response is helpful for the understanding and the optimization of the process. Usually, in manufacturing, with a few relevant variables, it is possible to evaluate the performance of the manufacturing system through the key performance indicators (KPIs).

The possibility of implementing ML to process control in hot stamping has been proposed in the literature and specialized fora, with different approaches being considered [235–237]. The basic common ground tends to heavily lean into monitoring process temperature at different points, thus ensuring that the heat treatment and final part properties are controlled. Differently, the current application presents a novel approach to built a data-driven SMod of the hot stamping process of the UHSS 22MnB5. The aim is to demonstrate the model validity to predict the performance of a simplified hot stamping process reproduced in a finite element modeling environment, offering a much faster response than in the simulations. The model analyses a problem where an austenitized piece of sheet steel is quenched using water-cooled steel dies, reproducing the experimental setup of a real industrial plant described in reference [238]. The SMod focuses on the prediction of the most relevant target variables for the process, which determine the quality of the obtained sheet and the state of the hot stamping die.

The model consists of a supervised ML algorithm, which establishes relationships between the input variables of the process and the target variables. The training of the model is performed with a series of FE simulations performed inside a pre-defined parameter space. Concretely, two training methodologies are proposed: one built with simulations inside the typical operation framework and the other covering non-standard cases. Another purpose of the use case is to propose an efficient method to train the SMod in order to achieve the maximum generalization capability in the validation process. The validation scenarios are defined by adjusting the parameters to the facilities of the real industrial plants, but also with the capability to explore new operation possibilities towards dynamic process optimization. The results show that the SMod trained with non-standard cases is more suitable for the prediction in all the evaluated scenarios and it can be optimized with the objective to reduce the number of FE simulations required in the training phase. The key advantages of the SModing of the hot stamping process is that it provides a soft real-time response of the target variables of the process and enables the creation of time- and cost-efficient virtual environments for knowledge collection, overcoming the time and computational limitations of traditional FE simulations.

5.3.2.2 Simulations and Surrogate Modeling Methods

Hot Stamping Process In this use case, a simplified hot stamping process is analyzed by finite element modeling (FEM) using the ABAQUS software [230], based on the experimental layout discussed in reference [238]. The reproduced setup consists of a flat water-cooled die made out of steel DIN 1.2344 (roughly corresponding to AISI H13) tempered at 48 ± 1 HRC, with water channels 10 mm in diameter and located at 20 mm depth from the surface, with a separation of 50 mm between centers. On these tools, an austenitized 22MnB5 sheet 1.7 mm in thickness is introduced, and the dies are closed, resulting in the component being quenched. The chemical composition of the two materials is presented in Table 5.6.

Simulation 2D models have been created with a focus on economy of calculation, as the main aim of the study is to generate a very large amount of simulations to train and demonstrate the SMods capabilities. Transient heat transfer analysis is realized with a model meshed with four-node linear quadrilateral elements and

Table 5.6: Typical chemical composition in % of 22MnB5 sheet steel and 1.2344 tool steel.

Element	C	Si	Mn	Cr	B	Mo	V
22MnB5	0.20–0.25	0.15–0.35	1.1–1.4	0.15–0.30	0.002–0.004		
1.2344	0.39	1.00		5.40		1.35	1.00

using a slightly higher mesh density in contact boundary regions. Quadrilateral-shaped elements have been used instead of triangular to reduce the number of nodes involved in the model, consequently reducing the computational cost. A total of 783 elements and 995 nodes has been used to represent a transversal cut of the sheet and the die during the process, using the system symmetry to further simplify geometry, as shown in Figure 5.19. Plastic deformation and phase changes are not considered, reducing the scenario to a heat transfer problem representative of a local analysis of quenching of a press-hardened component. For the same reason, strategies to increase precision of results such as local mesh refinement are not employed, instead performing the whole batch of simulations using the basic-defined mesh.

For this model, the main material properties simulated have been density, estimated at 7800 kg/m^3 for both steels and heat conductivity, where values of 23 W/m·K for 22MnB5 steel and 27 W/m·K for 1.2344 have been used in accordance to references [232, 239]. Thermal contact conductance between dies and sheet metal has been set at 3000 W/m^2·K, as used in reference [220]. A boundary film condition has been applied on inner surfaces of die cooling channels. A 12000 W/m^2·K heat transfer coefficient and 25 °C of sink temperature were used regarding the turbulent flow of a water-cooled system, created with drilled channels.

Using this model, a series of heat transfer transient simulations are run sequentially. Then, the die temperature changes along the cycles. The initial die temperature is set to to 25 °C; from that point, each cycle uses the temperature distribution on the die resulting from the previous simulation. In this manner, die heating is reproduced in the simulation model as it is observed in the physical system. On each cycle, a new sheet is considered, with an initial temperature of 800 °C, a reasonable estimation of an industrial hot stamping process. A cycle simulation comprises two phases:

1. Forming phase: It represents the stage when the die is closed and there is a heat transfer between the hot sheet and the cold die: this phase is governed by the forming time;

2. Cooling phase: The sheet has already been extracted. It includes the recovery of the die after the forming phase and the transfer of the next metal sheet in the die. This phase is governed by the cooling time.

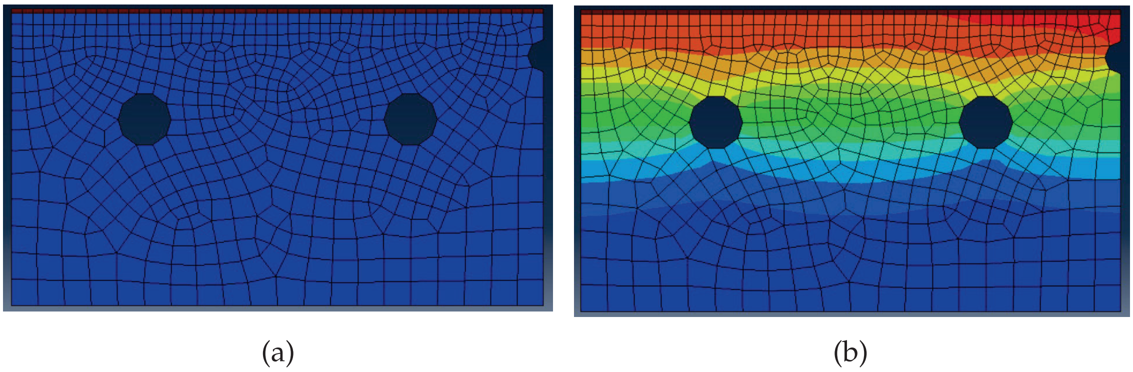


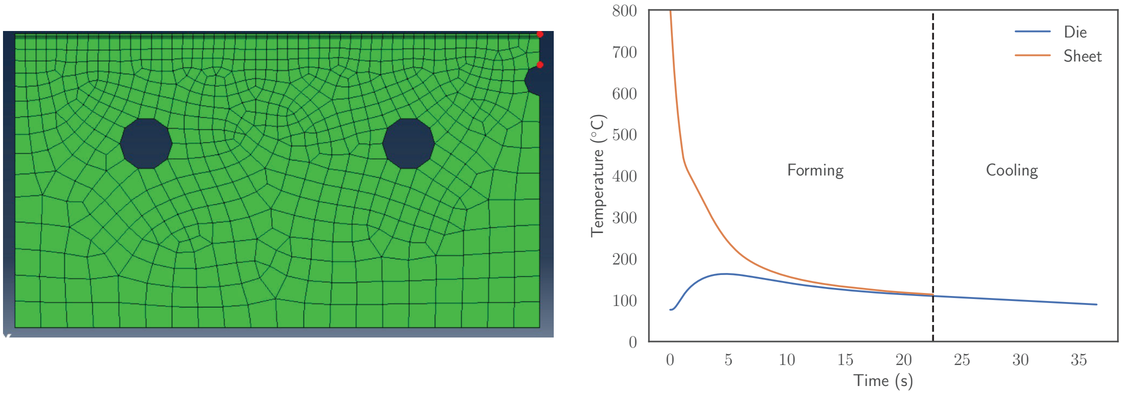
Figure 5.19: Representation of the mesh temperature profile in ABAQUS. Both the sensor of the sheet and the one of the die are highlighted in yellow. (**a**) Initial state. The hot sheet is displayed in red and the cold die in blue. (**b**) State after a forming phase. The temperature in the superior region of the mesh is similar between the sheet and the mesh and this makes difficult to differentiate the two parts.

Despite Figure 5.19, where the distribution of all the node temperatures is displayed, two nodes are taken as a reference for the sheet and die temperatures during the process. In Figure 5.20a, the location of these nodes in the mesh is indicated and it corresponds to the position of real sensors. As a result, the simulations allow to control the temperature of the sheet and the die of the hot stamping process from a similar point of view to the industrial plant. The behavior of the reference nodes in an example simulation cycle is shown in Figure 5.20b.

The hot stamping simulations demand some input values that are restricted to the conditions of the real industrial plant. The most relevant input variables are the following:

- Initial distribution of the sheet temperature. We focus on the temperature at the reference node T^S_{ini}. It is the temperature of the sheet at the start of the forming phase. It is assumed to be 800 °C in this study;

- Initial distribution of the die temperature. We focus on the temperature at the reference node T^D_{ini}. It is the temperature of the die at the start of the forming phase. It is assumed to be 25 °C in the initial cycle, but it keeps changing when sequential cycles are simulated. It represents the actual state of the hot stamping system at the start of the cycle;



(a) (b)

Figure 5.20: (**a**) Location of the reference nodes in the mesh. The upper red point is the sheet reference node (S) and the lower red point is the die reference node (D). (**b**) Evolution of the reference points' temperatures during a complete hot stamping simulation cycle.

- Forming time t_{form}. It refers to the duration of the forming phase. To ensure the quality of the sheet, the forming time has a minimum value of $t_{form} = 10$ s in the industrial plant;

- Cooling time t_{cool}. It refers to the duration of the cooling phase. The transfer cannot be immediate and a cooling of the die is required. Then, t_{cool} ranges from $[10 \; to \; 20]$ s in the current industrial plant;

- Cycle time t_{cycle}. It is the total duration of a simulation cycle. $t_{cycle} = t_{form} + t_{cool}$. The real plant restricts this variable in the interval of $[30, 40]$ s. In this plant, the sheets go through the furnace in a belt and cycle time depends on the furnace providing the next hot sheet. In more advanced industrial plants, the cycle time range is wider, because they use several furnaces at the desired temperature, and the hot sheet is available anytime.

At the end of a simulation cycle, there are three output variables which provide crucial information about the realized hot stamping cycle. These target variables determine the state of the system, and the quality and the good performance of the hot stamping process:

- Final distribution of the die temperature. We focus on the temperature at the reference node T^D_{fin}. It is the temperature of the die at the end of the cycle after the forming and cooling phases. In a sequential simulation of cycles it keeps evolving and it is the value for the T^D_{ini} of the next cycle. Therefore, it represents the actual state of the hot stamping system at the end of the cycle;

- Final distribution of the sheet temperature. We focus on the temperature at the reference node T^S_{fin}. It is the temperature of the sheet at the end of the forming phase when the sheet is extracted. This variable controls the quality of the final product. If it exceeds a threshold temperature, the sheet has not been able to acquire the martensitic microstructure due to a slow cooling;
- Distribution of the maximum die temperature. We focus on the temperature at the reference node T^D_{max}. It is the maximum temperature achieved in a cycle. This variable makes sure that die capacities are not exceeded and ensures that it is able to support the process.

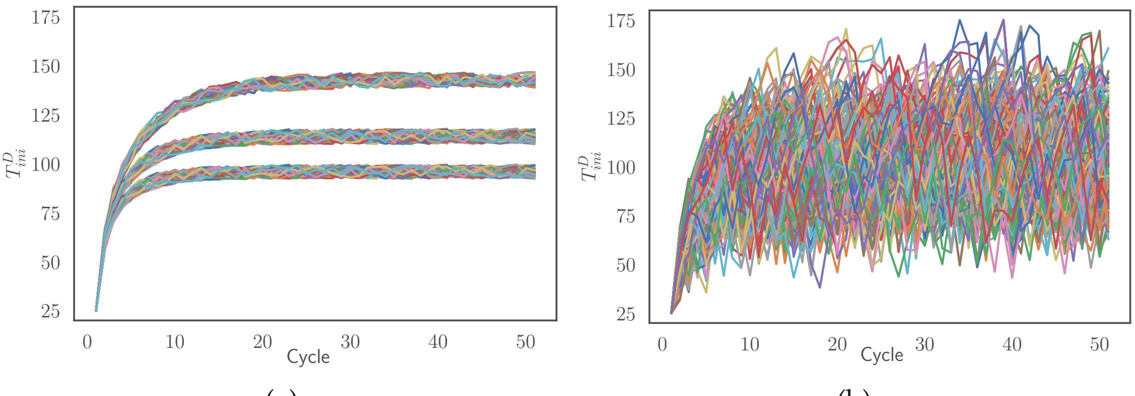
In general, in industrial plants, the execution of only a single cycle of hot stamping is not the usual way of operation. The demand requires the production of batches of more than one product, which implies carrying out several cycles. In the current study, 50 sequential cycles are equivalent to a batch. Therefore, to build the SMod, we generate batch simulations of 50 hot stamping cycles to mimic a possible real-demand case.

Although the industrial plant characteristics limit the cycle time to a range of values, the SMod aims to generate an environment to look for the optimization of this feature and the product quality in the batches. The modification of forming and cooling times could lead to a change in the cycle time, but also could imply the manufacturing of a defective sheet. Hence, there is a trade-off between the reduction of the total cycle time in the batches and the final quality of the metal sheets. Since the transference of the sheet from the furnace into the die is usually performed by an automated system, in the study we focus on the creation of a SMod able to reproduce scenarios where the cooling time is set constant, according to the possibilities of real industrial plants. Then, the forming time is modified, affecting the total cycle time. The forming time can be changed in a real plant, controlling the duration of the die closure. In Figure 5.20b, it is shown how a change of the forming time influences in the T^S_{fin} and the T^D_{fin}. Summarizing, in this use case, the SMod opens the door to explore a possible real-operation scenario where we try to optimize the time and the quality of the metal forming process, modifying the forming time of the cycles while keeping the cooling time constant.

Surrogate Modeling of the Process This study proposes a methodology for the development of a data-driven SMod, consisting of a supervised machine learning regression algorithm. The SMod is built using simulations from the high-fidelity FE simulations described in Section 5.3.2.2. The simulations consist of different batches of 50 cycles. Ideally, the SMod should be able to reproduce batches in the whole region of the parameter space. To achieve this generalization capability, we have generated two candidate training sets under different simulation conditions in order to evaluate which provides the most general surrogate model.

- **Training Set A:** The parameters of the simulations agree with real-operation conditions. Each cycle lasts randomly within $t_{cycle} = [30, 40]$ s, with a discretization of the interval each 0.5 s. The cooling time is set constant along the batches according to the scenario that we want to reproduce in this study. Therefore, the training set contains 90 batches of t_{cool} = 10 s, 15 s, and 20 s, respectively. The forming time oscillates depending on the random value of t_{cycle}, fulfilling the restriction of $t_{cycle} = t_{form} + t_{cool}$. This training set A it is used to feed the Surrogate Model A (SModA). Figure 5.21a shows the evolution of T^D_{ini} of the simulations of the Training Set A;
- **Training Set B:** The simulations do not correspond to normal operation conditions. In this case, the cycle time also has a random value within $t_{cycle} = [30, 40]$ s, discretized each 0.5 s. Moreover, the cooling time has a random values for each cycle ranged in the interval $t_{cool} = [0, 40]$ s, with a discretization of 0.5 s. The forming time is the result of the condition $t_{cycle} = t_{form} + t_{cool}$. This dataset is formed by 270 batches and it is used to train the Surrogate Model B (SModB). Figure 5.21b shows the evolution of T^D_{ini} of the simulations of the Training Set B.

Both SMods are based on supervised regression ML algorithms. In Section 5.3.2.2, we have determined the most important input variables of the process and the most relevant output variables. To mimic the simulation, the SMods have the same inputs and outputs. The inputs are T^S_{ini}, T^D_{ini}, t_{form}, and t_{cool} and the target variables are T^D_{fin}, T^S_{fin}, and T^D_{max}. However, there exist a slight difference between the SModA and SModB. The SModA also adds in its inputs a temporal window with the 3 previous values of the variables T^D_{ini}, because, in Figure 5.21a, the simulations are shown to have a temporal dependence. Several supervised regression algorithms are implemented using the Scikit-learn [240] and the XGBoost [93]



(a)

(b)

Figure 5.21: Representation of the evolution of T^D_{ini} for the simulated batches in the training sets (**a**) A and (**b**) B.

Python libraries. Concretely, 4 candidate algorithm are explored: KNN, based on Euclidean distance as similarity metric; SVM regressor (SVR), which works with hyperplanes in the dimensio Knal space defined by the input data; and XGBoost and RF, which are ensemble techniques. These candidate algorithms cover some of the most-used types of supervised ML algorithms. A 5-fold CV [195] is applied to check the performance of the algorithms in both training sets in order to determine the best one.

As observed, there is a big difference when comparing Figure 5.21a, whose simulations are performed under standard operation conditions, with Figure 5.21b. In Training Set B, the simulations do not cover standard cases, but they explore a wider region of possible states of the system. The SMods are validated with simulations that are analogous rather than the ones forming Training Set A, with distinct parameter values according to the facilities of our industrial plant. The FE simulations act as our ground truth. Finally, we identify which is the better option to built a SMod capable to generalize unseen simulation data.

5.3.2.3 Results and Discussion

Below, we evaluate the SMods created with the different methodologies proposed in previous sections, presenting the results of the accuracy of the models in the prediction of the target variables. In addition, we try to optimize the simulations that are required to feed the SMod, to achieve a reduction of the simulation time without having a significant impact in the quality of the models.

As emphasized in Section 5.3.2.2, the state of the system after a hot stamping process is determined by the final die temperature, T^D_{fin}. Consequently, the evolution of this variable governs the evolution of the whole system. Since it is used as input for the SMod, a poor prediction of this variable affects the next cycles prediction of the other target variables, which establish the metal sheet quality and the smooth operation of the system. For that reason, we give numerical results for all the target variables, but, as an insight, we display figures only for the temperature of the die. The figures for the rest of target variables are presented in Appendix A.

Baseline Prediction Results Both SMods are based on regression algorithms that predict the state of the system and the most relevant process variables after a complete cycle of hot stamping. To choose an algorithm we implement the technique of 5-Fold CV [195] to check the performance of various candidate algorithms on both training sets. The metric for algorithm evaluation used in this study is the mean absolute error [49]:

$$\mathrm{MAE} = \frac{\sum_{i=1}^{N} |x_i - \hat{x}_i|}{N} \tag{5.4}$$

where x_i are the actual values, $\hat{x}_i$ are the predicted values, and N is the number of samples.

As shown in Table 5.7, the best algorithms are able to obtain very good results for all the target variables. The values of MAE are less than 0.5 for Training Set A and less than 2 for Training Set B, without a relevant SD. Therefore, the XGBoost algorithm acts as the basis of the SMod during the rest of the study. The results of the five-fold CV of this regression algorithm in both training sets represent a baseline for the creation of a SMod. The next step is the evaluation of the SMod in validation sets corresponding to the real-plant framework, with the objective of generating a general SMod capable to predict the target variables in the all the regions of the parameter space.

Exploration of Validation Scenarios The validation of the SMods is performed according to different situations that may be encountered in a real industrial plant. Then, we have several validation scenarios to compare the performance of the two SMods and determine which is the best model able to generalize in various regions within the range of the parameter space. It must be noticed that some of the validation scenarios are built using the same simulation conditions, rather than Training Set A or B. In these cases, it is unfair to compare the SModA performing in a validation set with the conditions of Training Set A, and the same happens for SModB and Training Set B conditions. However, this issue has been introduced with the

Table 5.7: MAE and standard deviation (SD) results from the 5-Fold CV for the different target variables and the four candidate regression algorithms: KNN, XGBoost, SVR and RF.

SMod	**Target Variable**	**KNN**	**XGBoost**	**SVR**	**RF**
A	T^D_{fin}	$0.261(0.009)$	$0.161(0.006)$	$1.002(0.010)$	$0.184(0.006)$
	T^S_{fin}	$0.797(0.012)$	$0.157(0.004)$	$2.287(0.074)$	$0.203(0.006)$
	T^D_{max}	$0.205(0.005)$	$0.045(0.002)$	$0.127(0.026)$	$0.055(0.002)$
B	T^D_{fin}	$2.075(0.013)$	$1.674(0.021)$	$2.615(0.026)$	$1.853(0.015)$
	T^S_{fin}	$8.703(0.421)$	$1.433(0.035)$	$34.993(1.148)$	$1.769(0.022)$
	T^D_{max}	$1.796(0.064)$	$0.768(0.023)$	$5.141839(0.202)$	$0.871(0.029)$

purpose of validating the SModA in the training conditions of the SModB and vice versa. In this way, the generalization capability of a SMod in the prediction of unseen scenarios compared with an unfair prediction is remarked upon.

Single-Cycle Prediction

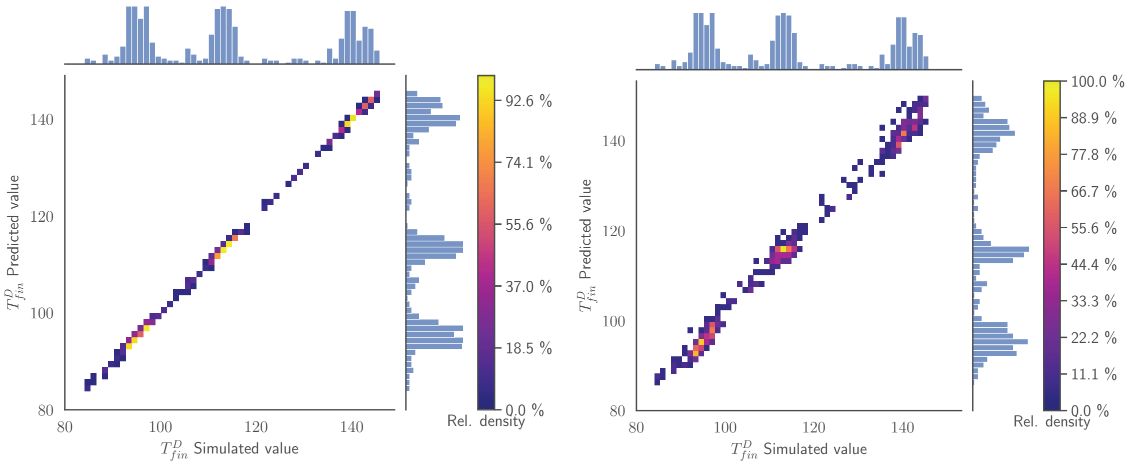
The final metal sheet quality and the die state are the most significant features after a hot stamping process. The resulting temperatures of the sheet T^S_{fin} and the die T^D_{fin} provide this information. Additionally,, the control of the maximum temperature of the die T^D_{max} during the process ensures that the die has not exceeded its operational window. The simulation of a single cycle of a forming process provides these target variables as outputs. Hence, we expect the SMod to accurately predict the same target variables after a process without the need of the simulation, under different input conditions. Each of the three validation sets consists of 500 samples, which gives a total ratio of ∼1:8 with respect to the training sets.

Validation Scenario 1: t_{cool} = 10 s, 15 s and 20 s.

The validation set is formed by 500 randomly input samples obtained from simulations under the conditions of Training Set A. We do not include samples from the first three cycles in the set, since the SMod fed with Training Set A requires information about the three previous cycles to work. In this scenario, the initial die temperature varies between 80 and 150 °C, the cooling time have values of t_{cool} = 10 s, 15 s, and 20 s, and the forming times have a random value with the restriction of the cycle time $t_{cycle} = t_{form} + t_{cool} = [30, 40]$ s.

Figure 5.22 presents the SMods' prediction values for the T^D_{fin} versus the simulated values obtained from the simulation using the same inputs. The axis of the plot are divided into 50 bins to build a histogram of the distribution of T^D_{fin} for both the predicted and simulated values, which act as our ground truth. This divides the space in the cells that are colored according the relative density of the samples compared to the cell with the maximum number of samples. For instance, 100% of relative density in a cell means that there are the same number of samples, rather than in the cell with the maximum number of samples. The figure additionally shows the empirical distributions of the simulated values (at the top of the figure) and the predicted ones (at the right side of the figure). In the ideal case, both distributions should be the same.

In Figure 5.22a, we appreciate a narrow line following the diagonal of the plot, implying an almost perfect prediction from the SModA. Observing the empirical distributions, we see that the zones with more density correspond to the values of the T^D_{fin} in the stationary regime for the cases t_{cool} = 10 s, 15 s, and 20 s, as can be seen in Figure 5.21a. The high prediction capability of the SModA in this validation scenario was expected, as training and validation sets share the same t_{cool} values. On the other hand, despite the randomness in its training, the SModB is able to approach the diagonal line and also correctly captures the zones with more density, as it is shown in Figure 5.22b. Nevertheless, the dispersion of the points indicates that the predictive power is lower than in the other case.



(a) (b)

Figure 5.22: Validation Scenario 1: Predicted values as function of the simulated output values of the T^D_{fin}. The histograms and the color map represent the relative counts as function of the temperature. (**a**) SModA and (**b**) SModB.

These features are repeated for the other target variables T^S_{fin} and T^D_{max}. The respective figures of these variables are displayed in the Appendix A. The results are condensed in Table 5.8, where it is evidenced how the SModA outperforms the

SModB in this particular validation case for all the target variables, although the SModB does not show very high values of the MAE. An error of about 2 °C is not unfeasible in experimental conditions, and can be often present due to systematic errors or calibration issues.

Table 5.8: MAE and SD results for the different target variables and the two SMods in the next cycle prediction for Validation Scenario 1.

Target Variable	SModA	SModB
T_{fin}^D	$0.172(0.200)$	$1.781(1.475)$
T_{fin}^S	$0.193(0.246)$	$1.893(1.417)$
T_{max}^D	$0.054(0.069)$	$1.055(0.778)$

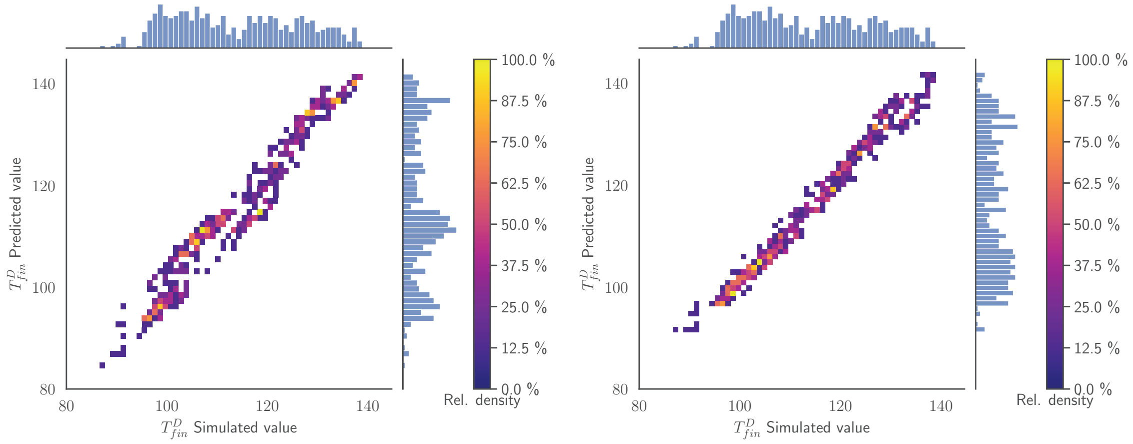
Validation Scenario 2: t_{cool} = Intermediate values.

In this case, the validation set contains 500 randomly selected input points from simulations with intermediate values of t_{cool}, rather than the ones in Training Set A. The same as before, we do not add points from the first three cycles in the set, taking into account the limitation of the SMod trained on Training Set A. Then, the initial die temperature ranges between 85 and 145 °C, the forming times have values of t_{cool} = 11 s, 12 s, 13 s, 14 s, 16 s, 17 s, 18 s, and 19 s, and the forming time has a random value with the restriction of the cycle time $t_{cycle} = t_{form} + t_{cool} = [30, 40]$ s.

In the current validation scenario, the SModA does not perform as well as in the previous case. The intermediate values of t_{cool} force the model to make an interpolation. In Figure 5.23a, the points are distributed around the diagonal, although they form a line with a significant width, meaning more prediction error and SD. The SModB presents a narrower line around the diagonal, as it can be observed in Figure 5.23b. We notice that the intermediate values of t_{cool} cause a more uniform density distribution along the range of temperatures.

Table 5.9 shows the commented results of the SModB for the variable T_{fin}^D. It must be noted that for the rest of the target variables, SModA has a lower MAE. Nonetheless, also taking into account Figure 5.23, we consider that SModB is better in the prediction of the T_{fin}^D of a next cycle than SModA in this parameter interpolation case, but observing the SD of both models, we observe that the overlap in the results makes it difficult to establish a clear option.

Validation Scenario 3: t_{cool} = Random.



(a) (b)

Figure 5.23: Validation Scenario 2: Predicted values as function of the simulated output values of the T_{fin}^{D} under. The histograms and the color map represent the relative counts as function of the temperature. (**a**) SModA and (**b**) SModB.

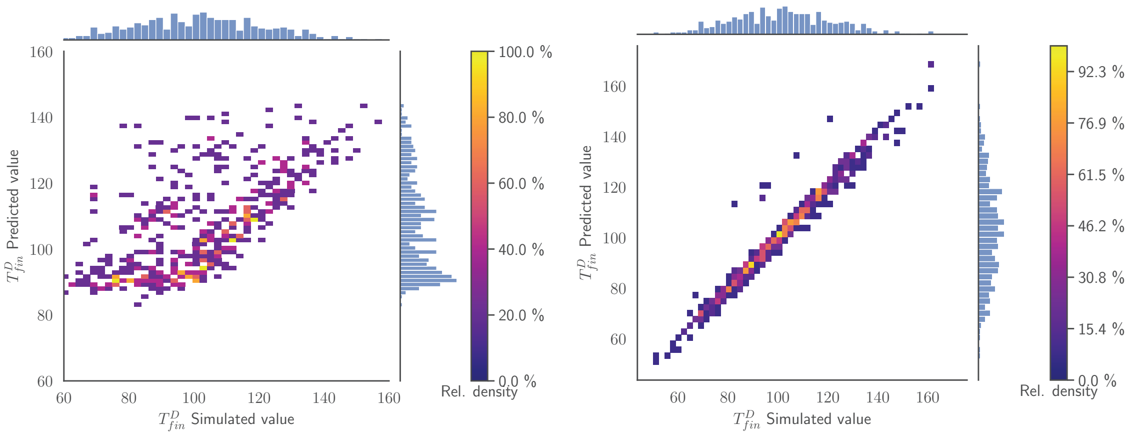
Table 5.9: MAE and SD results for the different target variables and the two SMods in the single-cycle prediction for Validation Scenario 2.

Target Variable	**SModA**	**SModB**
T_{fin}^{D}	$3.190(1.607)$	$1.939(1.559)$
T_{fin}^{S}	$1.179(0.912)$	$2.308(1.521)$
T_{max}^{D}	$0.506(0.376)$	$1.160(0.728)$

The validation set consists of 500 randomly sampled input points obtained from simulations under the conditions of Training Set B. Again, for the same reason as before, the first three cycles are not included in the set. In this case, the initial die temperature of the samples ranges between 50 and 165 °C, while the forming and cooling times range between $t_{form}\ and\ t_{cool} = [0, 40]$ s, with the restriction of the cycle time $t_{cycle} = t_{form} + t_{cool} = [30, 40]$ s.

In Figure 5.24a, we see that in a random scenario the SModA performs poorly due to its lack of information about some regions of the parameter space. We observe a high dispersion of the points and the diagonal has nearly disappeared. Otherwise, as expected, Figure 5.24b shows that the SModB maintains its good performance. With a few exceptions, almost all the points are condensed around the diagonal, meaning that the predictions are very close to the simulation values. The training under random conditions results in a high adaptability to any value of the input variables. Checking the other target variables in Table 5.10, we confirm that SModB outperforms SModA in this more general scenario.

Summarizing, we identify that the SModA is able to carry out good predictions of the next cycle target variables in the exactly same training regimes, specifically, cases when $t_{cool} = [10, 20]$ s. However, the SModB achieves reasonably good performances in all the validation scenarios, showing a constant and controlled behavior.



(a) (b)

Figure 5.24: Validation Scenario 3: Predicted values as function of the simulated output values of the T^D_{fin}. The histograms and the color map represent the relative counts as function of the temperature. (**a**) SModA and (**b**) SModB.

Table 5.10: MAE and SD results for the different target variables and the two SMods in the single-cycle prediction for Validation Scenario 3.

Target Variable	**SModA**	**SModB**
T^D_{fin}	$12.536\ (10.429)$	$2.153\ (3.274)$
T^S_{fin}	$60.157\ (117.197)$	$2.665\ (13.875)$
T^D_{max}	$8.023\ (17.931)$	$2.078\ (9.802)$

Batch Prediction

Usually, in industrial manufacturing, the demand requires several hot stamping processes to obtain a batch consisting of a specific number of parts. The simulation of this sequence of cycles is even more time demanding. Therefore, we evaluate the SMods in the prediction of the target variables for all the cycles in a batch. Since the objective is to effectively substitute the simulations, the SMod performs a sequence where the prediction of the next cycle is performed by taking as input the previous predictions.

For the reasons explained in Section 5.3.2.2, the validation sets have $t_{cool} = ctant$ along the whole batch, corresponding to real experimental cases where the transference of the sheet into the die is automatized and the forming time can be changed within the range of values given by the total cycle time. The validation sets consist of 14 batches for each one of the values of t_{cool}, which gives a total ratio of $\sim$1:2 with respect to the training sets.

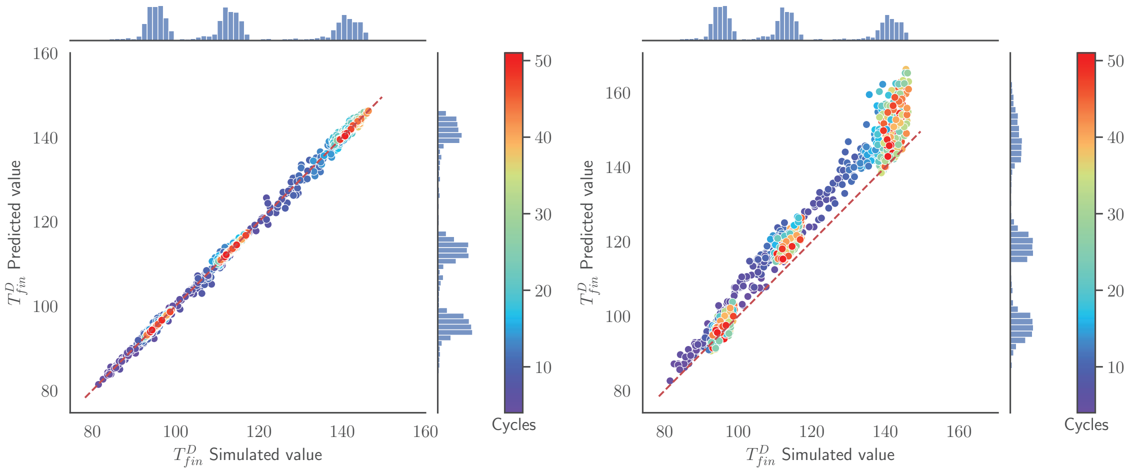
Validation Scenario 4: Batches of t_{cool} = 10 s, 15 s, and 20 s.

The validation set consists of batches of 50 cycles, where the cooling time is kept constant within the entire batch and it has values of t_{cool} = 10 s, 15 s, and 20 s, the same ones as Training Set A. The forming time has a random value for each cycle, with the restriction of the cycle time $t_{cycle} = t_{form} + t_{cool} = [30, 40]$ s. For each value of t_{cool}, we have 14 batches for validation.

In Figure 5.25, we compare how both SMods predict the target variables T^D_{fin}, which defines the state of our system. The diagonal line acts as a reference of the perfect prediction. We can also observe the distribution of the simulated values and the predicted values in the histograms. Moreover, since we are evaluating batches of 50 cycles, the colors indicate the cycle of the prediction. Notice that the SModA is able to have a very good performance in this scenario. The reason is that it has been trained and finely tuned to those particular settings. The SModB predictions are shifted to higher values of T^D_{fin} than our ground truth simulations, although the histograms are similar. The deviation from the diagonal becomes more evident in higher temperatures. In both Figure 5.25a and Figure 5.25b, the batches with different values of t_{cool} can be identified, as higher values of t_{cool} imply lower values of T^D_{fin}. Quantitatively, the MAE between the predictions and the simulated values for all the data of this validation scenario is presented Table 5.11, where the rest of target variables also have a very low value of MAE with the SModA.

Table 5.11: MAE and SD results for the different target variables and the two SMods in the batch prediction for the Validation Scenario 4 data.

Target Variable	**SModA**	**SModB**
T^D_{fin}	$0.359(0.492)$	$5.484(3.886)$
T^S_{fin}	$0.391(0.518)$	$6.335(4.470)$
T^D_{max}	$0.293(0.412)$	$5.362(3.586)$



(a)

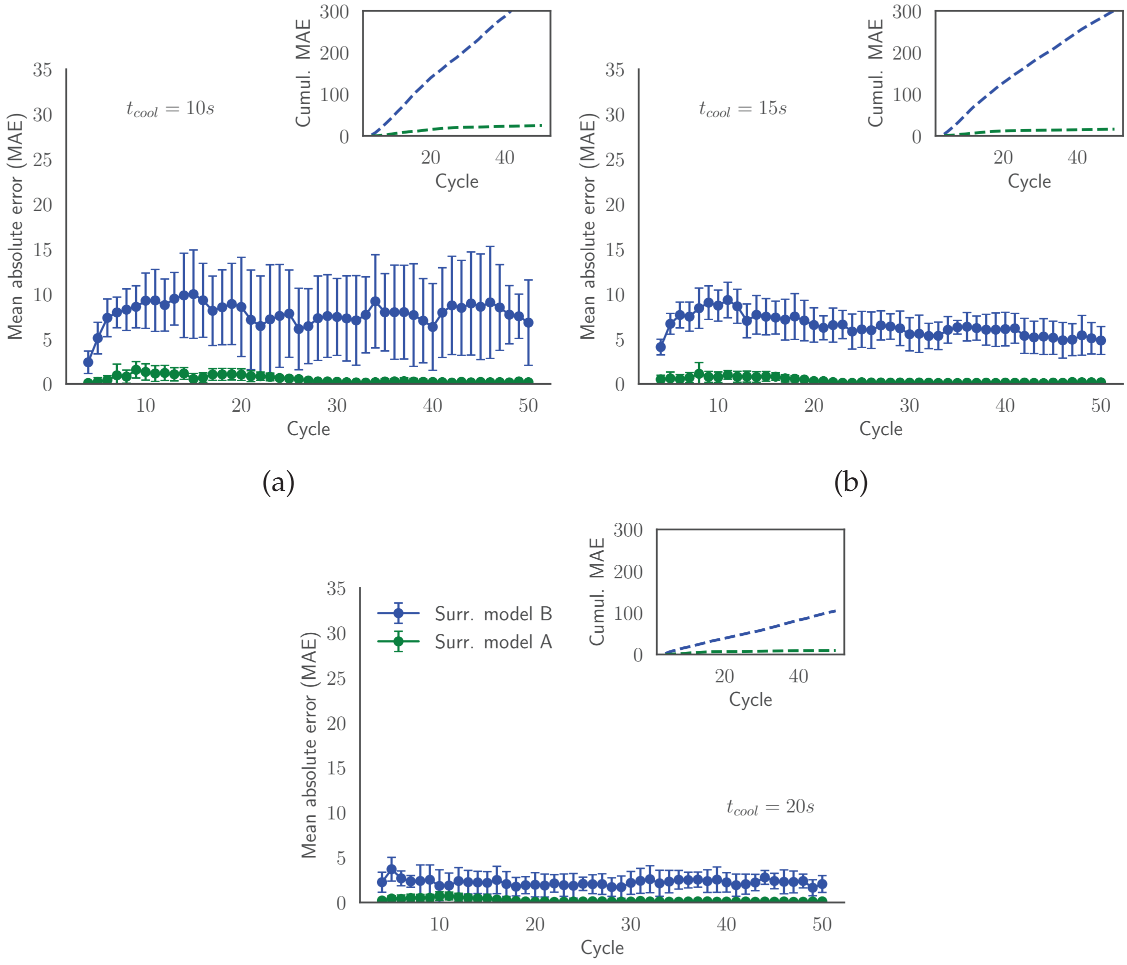
(b)

Figure 5.25: Validation Scenario 4: Predicted values as function of the simulated output values of the T^D_{fin}. The histograms represent the relative counts as function of the temperature and the color map indicates the cycle. (**a**) SModA and (**b**) SModB.

Figure 5.26 represents the MAE and the cumulative MAE of the predictions of T^D_{fin} for each cycle within the batches in the validation set. We observe how the SModA has a nearly perfect prediction for the previously commented reasons. Furthermore, the error of SModB is accumulated in the first cycles and after that it remains constant or even decreases. Additionally, we confirm that this model works better for higher values of t_{cool}, i.e., for lower temperatures. The explanation can be found in the training sets. In Figure 5.21a (where the parameters are the same as in the current validation set), the stationary region of the curves of t_{cool} = 10 s is not reached until cycle 15, in which the stationary region achieves temperatures around 140 °C. Figure 5.26 shows the important accumulation of error in the transient region for the SModB, and when t_{cool} = 10 s, the transient region lasts more cycles. Besides, looking at Training Set B, the interval of temperatures around 140 °C in Figure 5.21b is not very populated. These are the main causes of the loss in the predictive power of the SModB for low values of t_{cool}. Summarizing, the transient region is the main source of error for the SModB, since the mean absolute error increases in the first cycles, while in the stationary region it is kept constant.

Validation Scenario 5: Batches of t_{cool} = Intermediate.

In this case, the validation set is formed by batches of 50 cycles that have in-



(a)

(b)

(c)

Figure 5.26: Validation Scenario 4: Mean absolute error (MAE) of the model prediction of T^{D}_{fin} evaluated for batches with t_{cool} = (**a**) 10 s, (**b**) 15 s, and (**c**) 20 s. The inner plot shows the evolution of the cumulative mean absolute error.

termediate values of t_{cool} = 11 s, 12 s, 13 s, 14 s, 16 s, 17 s, 18 s, and 19 s, and that are kept constant along the cycles. Therefore, the SModA is not trained with the same values of cooling time. The forming time has a random value for each cycle, but it is restricted by $t_{cycle} = t_{form} + t_{cool} = [30, 40]$ s. For each value of t_{cool}, we have 14 batches.

In this scenario, Figure 5.27a evidences the lack of generalization of the SModA. We notice that the distribution of the predictions displayed in the vertical histogram has peaks in the same ranges of temperatures as the ones of Figure 5.25a. These ranges correspond to the stationary region of when t_{cool} = 10 s, 15 s, and 20 s, implying that the SModA is not able to interpolate for intermediate values.

In opposition, the predictions of the SModB present a similar temperature distribution to the simulated values. Additionally, comparing with Figure 5.25b of the previous validation scenario, we found an analogous behavior of the SModB in this case, as shown in the distribution of the points of Figure 5.27b. Focusing on Table 5.12, the comparison of the two SMods shows the lower values of MAE of the SModB with respect to the SModA for all the target variables for the intermediate values of t_{cool}. We notice that the values of MAE for the SModB are close to the ones in Table 5.11, which implies a comparable performance in both validation scenarios. Then, despite the small shift with respect to the diagonal line and the dispersion observed in high temperatures, the SModB is more convenient if we are seeking a model capable of generalizing within the defined range of the operational parameters.

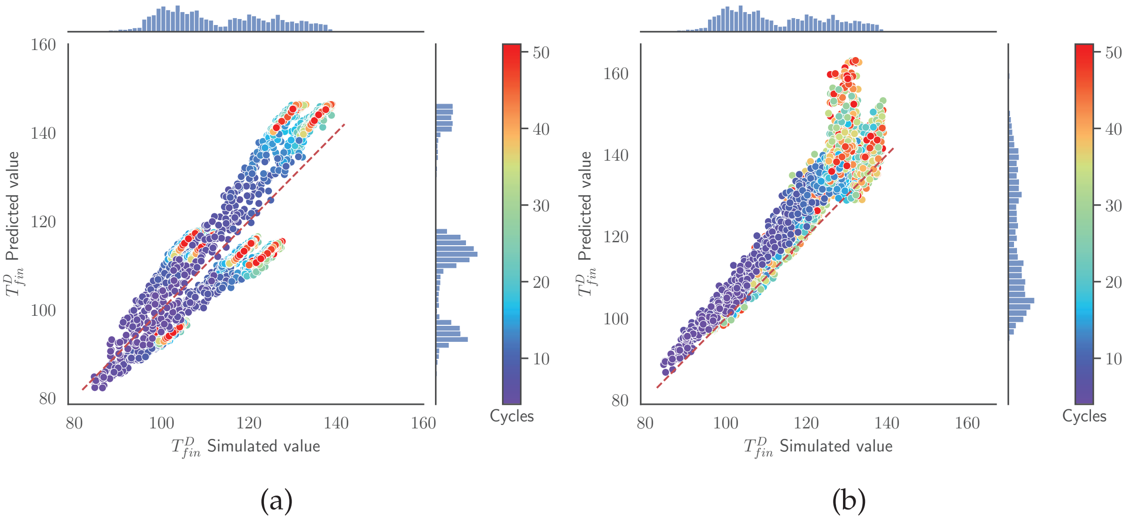
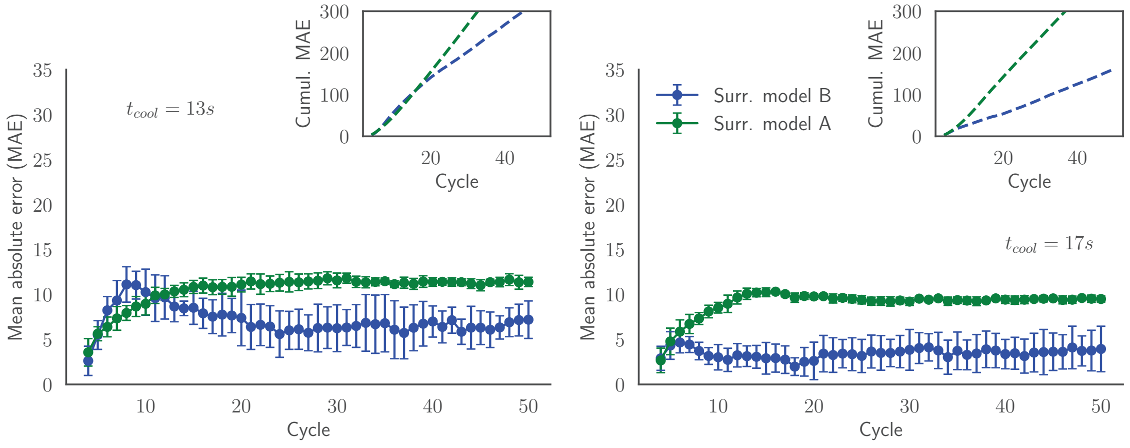


Figure 5.27: Validation Scenario 5: Predicted values as functions of the simulated output values of the T^D_{fin}. The histograms and the color map represent the relative counts as functions of the temperature. (**a**) SModA and (**b**) SModB.

Table 5.12: MAE and SD results for the different target variables and the two SMods in the batch prediction for the Validation Scenario 5 data.

Target Variable	**SModA**	**SModB**
T^D_{fin}	7.598(3.515)	5.605(4.760)
T^S_{fin}	8.434(2.961)	6.796(5.051)
T^D_{max}	7.044(2.909)	5.554(4.309)

The better performance of the SModB and its generalization potential are verified in Figure 5.28. Although the error increases in the first cycles, coinciding with the transient region, the SModB approaches the simulated values after that. On the contrary, the SModA error in the transient zone remains during the rest of the batch. As discussed, notice how the SModB works better for higher values of t_{cool}. After the evaluation of the model performance in the different scenarios, we choose the SModB over the SModA because it has shown to be a more general model. In spite of the remarkable generalization capability of the SModB, the model is far from being perfect, especially if we focus on Figure 5.28a. In the next section, we try to optimize the model performance and to reduce the computational time spent in the generation of the training set.



(a)

(b)

Figure 5.28: Validation Scenario 5: Mean absolute error (MAE) of the model predictions of T^D_{fin} evaluated for the batches in the validation set with t_{cool} = (**a**) 13 s and (**b**) 17 s. The inner plot shows the evolution of the cumulative mean absolute error.

Model Optimization Taking the SModB as the current baseline, in this section we aim to optimize this model, which has demonstrated a higher generalization capability than the SModA. However, the model has not been accurate enough in the prediction at high temperatures, i.e., low values of t_{cool}. Moreover, the current model is fed with the simulation of 270 batches of 50 cycles, which implies a lot of computational time. Therefore, there is a need to tune the model in order to reduce the computational time spent in the creation of the training set without a significant impact on the model accuracy. The performance of the model optimization criteria is evaluated in Validation Scenario 5, since it is the most adequate scenario for evaluating generalization in the batch prediction case.

Number of Cycles in Training

A reduction of the number of cycles of the batches of Training Set B implies training the SMod with less simulations. Table 5.13 shows that, in fact, this reduction not only decreases the time needed to create the training set but also it results in a better MAE. The SMod trained with only the first 10 cycles of the batches is the one that achieves the best MAE for all the target variables. This is verified in Figure 5.29, where for two arbitrary values of t_{cool}, the curves of MAE and the cumulative MAE of the model trained with batches of 10 cycles are displayed below the curves of the other models.

Looking at Figure 5.21b, we detect that once we reach the stationary region (around the 10th–15th cycle), the values of the temperatures are limited within an interval. Therefore, we have a biased training set with a lot of data inside the interval of temperatures of the stationary region. This bias leads to an overspecialization in the training phase, preventing good results in the transient state. Because errors at the transient state weigh a lot and are cumulative, the models trained with 10 or 15 cycles have better performances. In those training regimes, the model is fed with a better balance of both transient- and steady-state samples.

Table 5.13: MAE and SD results of the model predictions of T^D_{fin} for the different target variables in the batch prediction for the Validation Scenario 5 data as we decrease the number of cycles of the batches of Training Set B.

Target Variable	**Surr. Model B (50 Cycles)**	**20 Cycles**	**15 Cycles**	**10 Cycles**	**5 Cycles**
T^D_{fin}	5.605(4.760)	5.395(5.377)	3.398(2.983)	3.058(2.291)	5.316(4.324)
T^S_{fin}	6.796(5.051)	6.045(5.962)	4.360(3.592)	3.402(2.457)	4.686(3.781)
T^D_{max}	5.554(4.309)	4.855(5.084)	3.305(2.894)	2.903(2.013)	3.983(3.207)

Number of Simulated Batches in Training

Once the use of the first 10 cycles to train the SMod is determined , we focus on the number of simulated batches that form the training set. By reducing the number of batches in the training set, less simulations are required, saving a lot of computational time for the creation of the training set. In Figure 5.30, we explore how the number of batches affects the model accuracy. Concretely, a threshold is found at

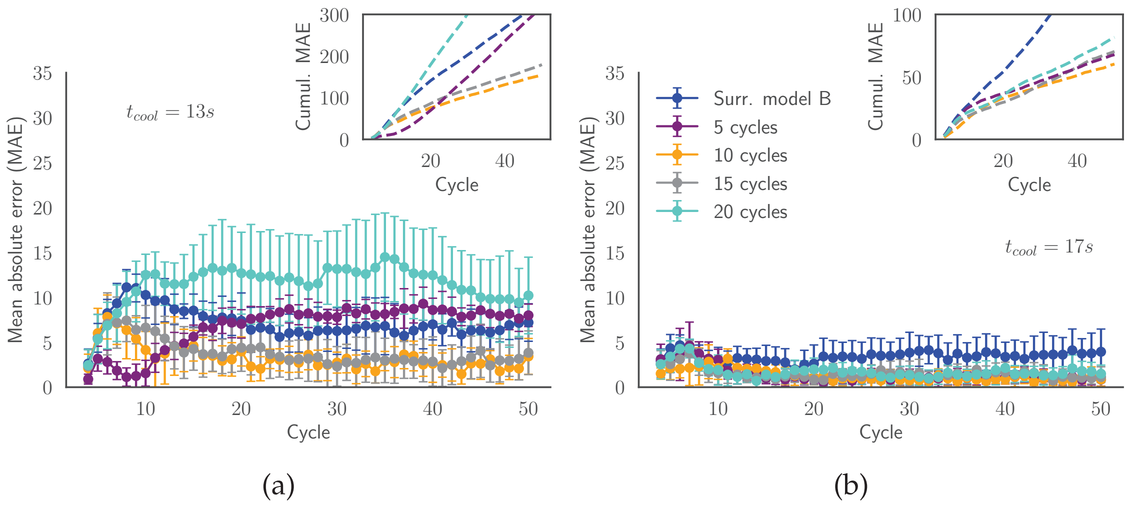


Figure 5.29: Validation Scenario 5: Mean absolute error (MAE) of the SModB and the models trained with less cycles. The predictions are evaluated for the batches in the validation set with t_{cool} = (**a**) 13 s and (**b**) 17 s. The inner plot shows the evolution of the cumulative mean absolute error.

around 220 batches in the training. As we decrease the number of batches down to this threshold, the MAE keeps increasing because the model does not have enough information to work properly.

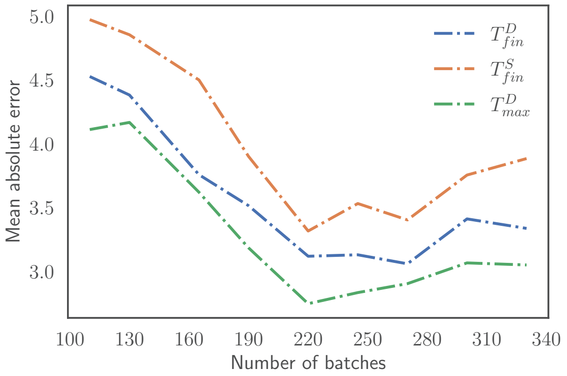


Figure 5.30: Evolution of the MAE for the three target variables evaluated in Validation Scenario 5 as we reduce the number of batches in the training set.

In addition, an increase of the simulated batches in the training set does not imply a better accuracy. The reason is that as we increase the number of data, the training set becomes more biased in the range of temperatures of the stationary region and the model is not able to capture the behavior well outside this range of temperatures. The effect of the bias is shown in Figure 5.31, where models trained with different numbers of batches perform a single-cycle prediction of different random cycles extracted from Validation Scenario 5. We define three intervals that classify the samples depending on the input T^D_{ini}: $T^D_{ini} < 80$ °C, $T^D_{ini} = [80, 130]$ °C, and $T^D_{ini} > 130$ °C, and each interval has 20 samples. We notice how the models with 220 and 270 batches have similar behaviors, although the MAE is lower for the one trained with 220 batches. The model trained with more samples struggles in the intervals of low and high temperatures and it has a lower MAE in the intermediate interval than the other models. This verifies the existence of a bias in the training set that affects the predictions when the number of batches is too large. Then, as it happens when we evaluated the number of training cycles, the model suffers from overfitting, reducing its generalization capability.

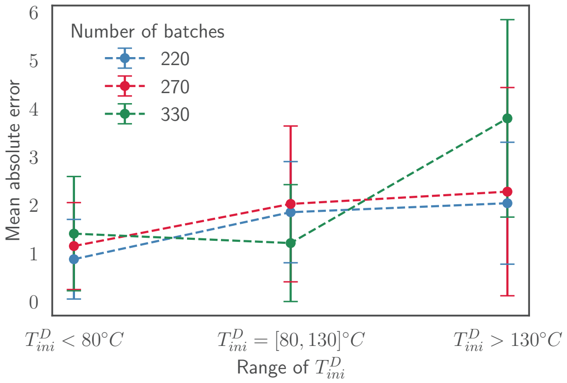


Figure 5.31: MAE for single predictions of T^D_{fin} of samples of models trained with different numbers of batches. The sample parameters have the conditions determined in Validation Scenario 5 and they are classified depending on the input temperature of the die T^D_{ini}.

Final Surrogate Model

From the previous results regarding the number of cycles and batches in training, we have obtained an optimized SMod that is trained with less simulations and it has a better performance than the baseline SModB. This final SMod is trained with 220 batches of 10 cycles under the same conditions as Training Set B: the cooling time has a random value for each cycle within the range $t_{cool} = [0, 40]$ and the cycle time also has a random value between $t_{cycle} = [30, 40]$. The forming time is restricted by the condition of $t_{cycle} = t_{form} + t_{cool}$. The randomness of the input variables is the main cause for the generalization power of the final surrogate model.

In Figure 5.32, we have compared several curves of T^D_{fin} from different simulated batches against the predictions of the model. The batches correspond to the case where $t_{cool} = ctant$, identified as a very interesting case to explore new possibilities in real industrial scenarios, as remarked in Section 5.3.2.2. We notice that the curves are very close and that the model is able to reproduce the simulation inside the whole range of $t_{cool} = [10, 20]$.

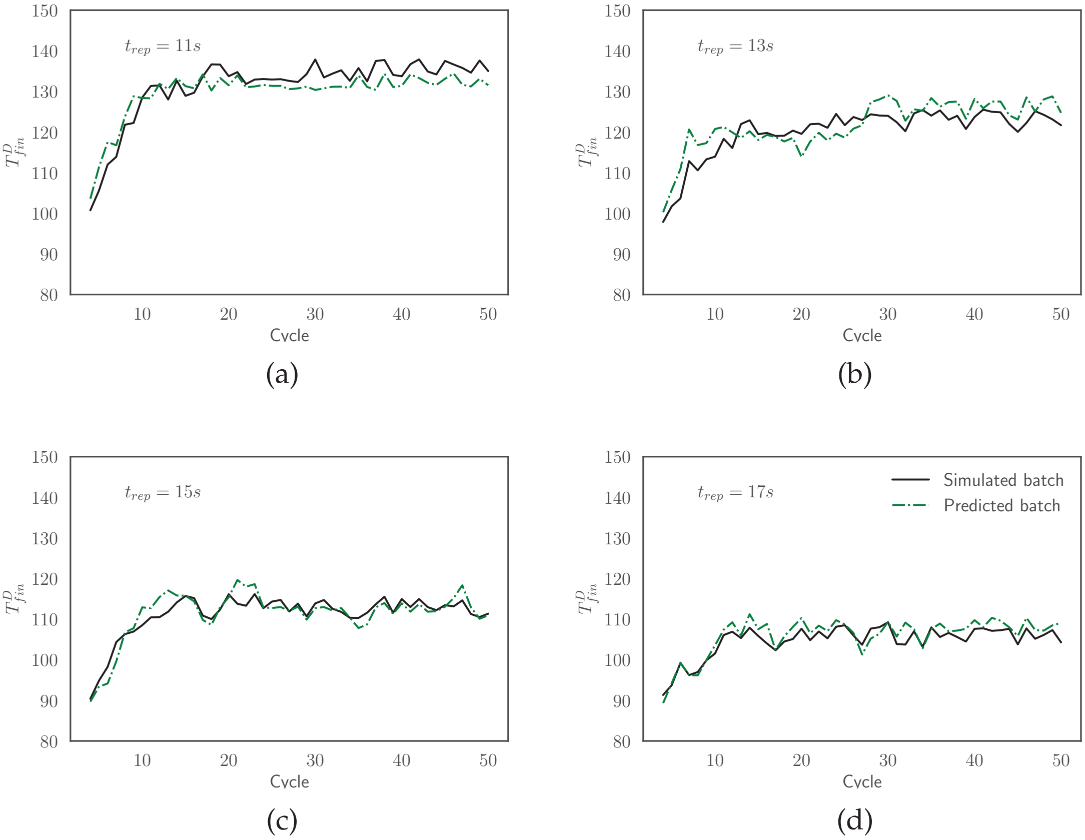


Figure 5.32: Comparison between the simulated curves and the predicted curves for the final SMod of T^D_{fin} evaluated for batches with t_{cool} = (**a**) 11 s, (**b**) 13 s, (**c**) 15 s and (**d**) 17 s.

Furthermore, the optimization of the baseline SMod to this final SMod has supposed a gain in the simulation time required to generate the training set. As mentioned, the CPU simulation time for a cycle is about 40 s. Initially, our baseline SMod was trained with 270 batches of 50 cycles, which are generated in approximately 150 h of CPU time. The training of the optimized model is performed with 270 of 10 cycles, decreasing the CPU time spent in simulation to $\sim$ 30 h. In addition, the 270 batches of 10 cycles have been reduced to 220 batches of 10 cycles for the training of the final SMod. Hence, we need less simulations to feed the model. This implies a reduction of almost ∼6 h in simulations, resulting in ∼24 h of CPU time spent in simulation. We have achieved a total reduction of $\sim 84\%$ of the CPU time.

Despite the time spent in the simulations for the training, the major benefit of the final SMod is that it can generate a cycle or a batch four orders of magnitude faster than the simulations, as it is displayed in Table 5.14. It must be noticed that the geometry in this study is a metal sheet, and that a more complex geometry will enhance the need of surrogate modeling due to the increasing difficulty in FE simulations. Therefore, this creates a powerful tool to explore new operation scenarios, with a low computational and temporal cost. The aim of this exploration is to optimize the batch production with a direct effect in the improvement of the KPIs of the process. For instance, OEE could be improved by defining the optimal forming time for each cycle, reducing then the total cycle time and preventing the defects in the products, which leads to a better scrap rate. Moreover, this environment is very efficient to train promising data-driven technologies, such as digital twins or reinforcement learning agents, because it allows to reproduce a lot of batches under different parameter conditions. The remarkable soft real-time response that the SMod provides could be used by digital twins to predict the optimal operational configuration of the systems. In addition, reinforcement learning agents could be trained inside this efficient environment that mimics the real manufacturing system, and afterwards they could act in the real scenario, applying the learned policies and optimizing the process in the industrial plant. To sum up, the SMod opens the possibility to the introduction of self-autonomous systems in the hot stamping process in industrial plants.

Table 5.14: Comparison between the simulation times and the final SMod times in cycle and batch generation.

Method	Cycle Time	Batch Time
Simulation	$\sim$40 s	$\sim$2000 s
Final SMod	$\sim 3 \times 10^{-3}$ s	$\sim 1.5 \times 10^{-1}$ s

5.3.2.4 Insights from Parameter Interpolation Surrogate Model in Hot Stamping

The wide range of applications of the hot stamping process in the automotive industry and the extensive use of 22MnB5 in safety-related components highlight the importance of the process and the interest in ensuring the quality of the final products and a good manufacturing performance. In the actual paradigm of Industry 4.0, in this use case we propose a novel ML-based SMod to predict the most relevant results of the hot stamping process of 22MnB5. We have validated the model in several scenarios and it is capable to provide a much faster response than simulation models. This enables the possibility to explore the parameter space and configurations in an efficient environment without the time limitations of the simulations. Hence, we have proven that the parameter interpolation approach for surrogate modeling is a compelling solution in this manufacturing use case. Moreover, the current approach is innovative, since in previous works the applications of ML in the hot stamping process have been based on the process monitoring and control at different points.

The model is trained with FE simulations and it is based on the XGBoost regression algorithm, which establishes relations between the input variables of the simulations with the most relevant process variables. The validation is performed in various feasible operational scenarios of a real plant, consisting in series of simulated batches of 50 cycles. In terms of the generalization of the SMod, it has been demonstrated that the training with batches with non-standard parameter conditions, which cover more regions of the parameter space, outperforms the training with batches with the standard parameter conditions.

Furthermore, the SMod has been optimized, decreasing the number of FE simulations required for its training. First, the number of cycles of the training batches has been reduced from 50 to 10 cycles. Remarkably, the SMod trained with batches with less cycles is able to reproduce larger batches more accurately in the validation scenarios. Next, the number of batches that feed the SMod has been decreased, and we have found a threshold of 220 batches before the accuracy starts to fall. This optimization has supposed an important decrease of 84% of the CPU time and has minimized the computational resources spent in the simulations needed to create the surrogate model.

Finally, the final optimized SMod is able to reproduce reasonably well the simulations inside the whole range of parameters of the real industrial plant. In fact, the models have the ability predict the target variables in the validation scenarios with a MAE of around 3 °C from the simulations, which is considered an acceptable error in an experimental context. The key advantage of the SMod is that it is four orders of magnitude faster than the simulations, triggering the exploration of new operation scenarios in an efficient environment. This opens the door to the setting of the optimal parameter values of the hot stamping process, improving the KPIs of the batch production of the process. In addition, the SMod provides a soft real-time response, which is crucial for the development of tools such as DTs or RL agents. Actually, the current model has been employed in [6, 8] as a fast environment to train a RL agent that controls the hot stamping process with the objective of decreasing the batch process time while keeping the quality of the stamped parts. To recap, the SMod methodology proposed in the study enables the self-autonomous system's presence in the hot stamping process industrial plants, with the possibility to be expanded to other manufacturing conditions or processes.

5.4 Lessons Learned

- In the context of I4.0, the development of ML-based SMod in real-world industrial scenarios is an instrumental way to accelerate data generation and analysis for manufacturing processes. The application of surrogate modeling methods in the use cases demonstrates their practical relevance and potential for adoption in real-world industrial settings. This aligns with the thesis goal of bridging the gap between theoretical surrogate modeling techniques and their practical applications in manufacturing.

- The implementation of ML-based SMods in real-world manufacturing challenges highly depends on the type of process and problem. However, we proposed three distinct ML-based SMods methods applied to real-world manufacturing challenges which can be extended to other manufacturing processes. The methods include node reduction in injection molding, mesh upscaling in HPDC and parameter interpolation in hot stamping.

- The node reduction method generated a ML-based SMod that can effectively reduce the number of sensor nodes required to generate accurate predictions in the plastic injection molding process. This leads to cost savings and simplified system design in real-world manufacturing processes by minimizing the number of physical sensors needed.

- The mesh upscaling method leaded to a ML-based SMod that can accurately predict fine mesh simulation results using coarse mesh simulations in the (HPDC) process. This significantly reduces computational time and enables faster analysis and exploration of complex manufacturing processes.

- The parameter interpolation method resulted in a ML-based SMod that can efficiently predict simulation results across a wide range of parameter values in the hot stamping process in soft real time. This addresses the challenge of generating data in a fast and sustainable way and enables the exploration of new operation scenarios.

Chapter 6

Extension of the Hot Stamping Case - Importance of Sampling

In this chapter we showcase another practical application of surrogate modeling in the hot stamping process. However, in the current use case we explore an extension of the parameter interpolation approach presented in 5.3.2. Concretely, we focus on investigating the influence the significant impact of different sampling methods on SMod performance, taking the parameter interpolation as the frame of reference to build baseline ML-based SMods. We compare the accuracy and time efficiency of various sampling methods, providing valuable insights into their effectiveness for constructing SMods in hot stamping applications.

Despite the similarity of both cases, the hot stamping scenario is distinct, which allows us to prove the validity of the parameter interpolation approach in other situations. In particular, the geometry of the hot stamping part and the considered input variables and output variables are not the same. This allows to cover a wider range of knowledge within the hot stamping process in the thesis.

The findings of this investigation complement the previous study in surrogate modeling in hot stamping in two directions: Firstly, they contribute to a deeper understanding of the role of sampling in surrogate modeling and its implications for optimizing hot stamping processes. Additionally, they also provide an improved version of the ML-based SMod, which is capable to predict two critical process variables in several regions of a hat-shaped geometry.

As mentioned, the role of simulation in the sensitivity analysis studies is crucial due to the need of exploring the parameter space without perturbing the real manufacturing system. In the particular case of hot stamping, the initial conditions, the setup parameters and the materials properties form a wide domain. The configurations within the domain have a direct impact in the thermomechanical phenomena and the material phase transformations occurring during hot stamping. Therefore, many simulations are required for extensive sensitivity analysis and optimization studies during the design stage to ensure the desired mechanical properties in the final parts. However, the elevated cost in time and computational resources of these simulations and the high dimensionality of the domain is an important limitation.

The aim of this chapter is to present a pipeline to overcome this drawback, with a ML-based SMod of the simulation of the hot stamping of a hat-shaped part of boron steel. Concretely, we specially focus on the importance of the sampling methods to build the training set of the surrogate model. Therefore, a comparison between Latin Hypercube Sampling and Forward Selection method is implemented for this purpose. The proposed methodology provides a pipeline to use a reduced number of simulations to generate a trustworthy SMod that is an enabler to boost the sensitivity analysis and optimization procedures, due to the fast response of its estimations. The proof-of-concept results show high potential in the soft-real time prediction of unseen configurations within the domain, focusing on important variables regarding the mechanical properties and the quality of the final part, such as the temperature and the martensite content.

The methods and results displayed in this chapter are presented on the work in [3].

6.1 Introduction

Simulations in hot stamping play a fundamental role in the design stage and prototype testing, where they offer an effective way for the product development, decreasing the waste of materials, the wear of the tools and the overall costs. Commonly, FE method has been employed to model the mechanical, thermal and microstructural phase transformation phenomena taking place during the hot stamping process [232]. Despite the increase in computing resources, the high-fidelity simulation models have also increased their complexity. This still supposes a high time expense, being unfeasible to apply these models during the real-time production.

In this use case, we define a general pipeline for the creation of a ML-based SMod of the hot stamping process of a hat-shaped part from LS-Dyna® simulations. Our approach consists in a ML-based SMod that not only predicts the final temperature, but also the martensite content in different regions of the final hat-shaped stamped part geometry inside a domain of configurations. Therefore, it extends the approach presented in Section 5.3.2 [2]. Once trained, it removes the need of simulations to perform the predictions and it uses input parameters measurable in a hot stamping plant, differing from [241]. Two different sampling strategies to build the training datasets of the SMod are implemented, Latin Hypercube Sampling and Forward Selection.Their impact in the development time and performance of the SMod is evaluated and compared. Concretely, the use case highlights the reduction the number of required simulations to create a reliable SMod thanks to the selection of a proper sampling strategy. a complete low-cost pipeline is presented, with a resulting SMod which predicts variables that act as important indicators of the process related to the quality and the mechanical properties of the final part. It provides a soft real-time framework to enhance the sensitivity analysis and the optimization procedures, and it eases the autonomous control and monitoring of the hot stamping plants through tools like DTs.

Summarizing, the objectives of the study of this chapter are three-fold:

1. Creation of a ML-based SMod that predicts in soft-real time the final temperature and the martensite content in the hot stamping process of a hat-shaped part.
2. Study the effect of the sampling in the performance of the SMod and propose the best sampling strategy to reduce the time and resources for the implementation of the SMod and increase its accuracy.
3. Provide a comprehensive guide for the development of SMods of hot stamping simulations procuring a complete pipeline of the surrogate modeling process.

6.2 Methodology

6.2.1 Description of the baseline pipeline

The creation of the SMod is done through the training of DNNs with LS-Dyna® hot stamping simulations of a hat-shaped part. The aim of the SMod is a temporal reduction in the exploration of the domain of configurations in sensitivity analysis or optimization studies. The proposed baseline pipeline is shown in Figure 6.1. In the first place, we define a configuration domain formed by the variables that will be included in a sensitivity analysis study. Generally, the variables implicated should be easily modifiable in real experiments and their range of values should be consistent with the possible values in a hypothetical hot stamping plant. The LS-Dyna® software has been used to create a simulation model that acts as a benchmark. A sampling strategy is used to select configurations within the domain. Accordingly, LS-Dyna® simulations are performed using as input the chosen configurations and obtaining the desired outputs, the temperature and the martensite fraction in different points of the hat-shaped part. This leads to the generation of two sets of data, a train set and a test set. The train set is employed to train the ML model that will constitute the SMod. In our case, we use a DNN with fully connected layerd architecture along with a 5-Fold CV method to tune the DNNs hyperparameters. Finally, the performance of the SMod in the unseen configurations of the test set shows the quality of the predictions of the SMod and its capability to be used in sensitivity analysis studies.

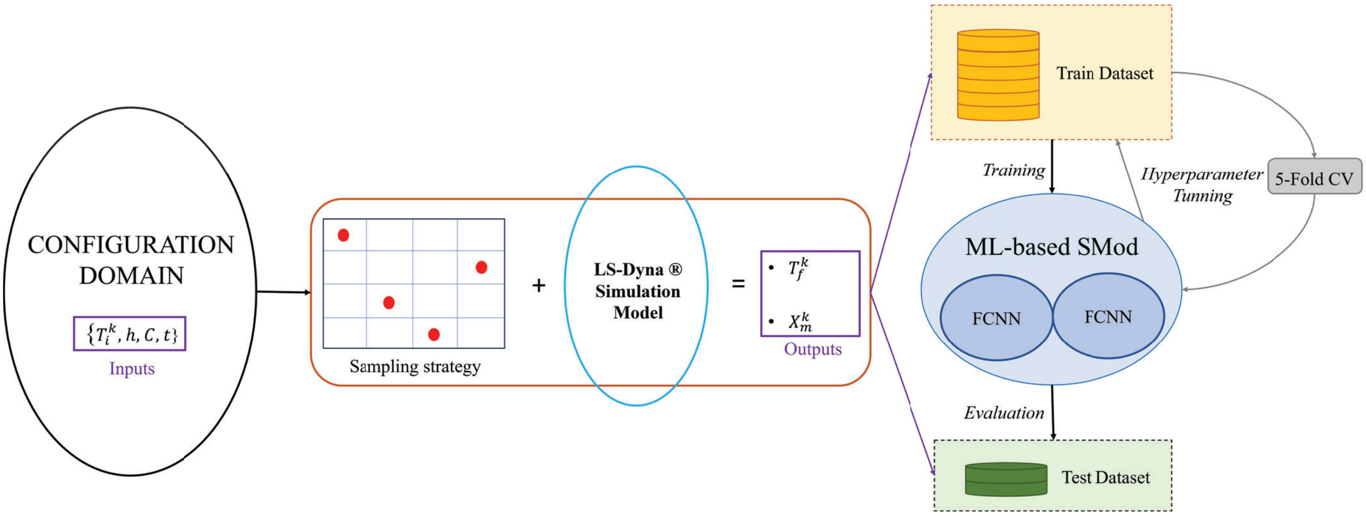


Figure 6.1: Flowchart of the proposed baseline pipeline to create a SMod of the hot stamping process from LS-Dyna® simulations.

6.2.2 Definition of the Configuration Domain

The considered domain consists in configurations of different values of 4 input variables of the hot stamping process. The variables need to have certain impact in the outputs for a posterior sensitivity analysis and being modifiable in the simulations and the real environment. In our scenario, the domain is formed by the variables in Table 6.1.

Table 6.1: Variables of the configuration domain and their corresponding ranges and units.

Variable	Range of values
Initial blank temperature (T_i)	$[1023 - 1173]$ K
Heat transfer coefficient (h)	$[1000 - 6000]$ W/m^2 K
Carbon content (C)	$[0.20 - 0.25]$
Holding time (t)	$[5 - 25]$ s

6.2.3 Simulation Model

The LS-Dyna® thermomechanical simulation model consists of a half model of a punch, die and blank which represent hot stamping of a hat-profile section. The punch and die are discretized with 2 mm 8-noded constant stress solid elements. The 1 mm thick blank is discretized with 2 mm Belytschko-Tsay shell elements with 5 integrations points throughout the thickness. The material model used for the punch and die are MAT_020_RIGID, and for the blank, MAT_244_UHS_STEEL. Thermal material properties are defined using the thermal material card MAT_T10-THERMAL_ISOTROPIC_TD_LC. A selection of the boundary conditions used are convection sets between the blank and the tool surfaces, temperature sets to set a fixed temperature on the punch and die external surfaces. The punch movement is controlled with a prescribed motion. The model contact type is CONTACT_AUTOMATIC_SURFACE_TO_SURFACE_MORTAR.

The parameters are modified in the simulations within the ranges presented in Table 6.1. 4 elements are chosen to represent different regions of the blank, as described in Figure 6.2. The element histories of these elements are obtained and final temperature (T_f^k) and final martensite fraction (X_m^k) in the last time step are used as target outputs of the process, where $k = A, B, C, D$ are the considered elements. These variables are valuable indicators of the quality and the mechanical properties of the final part.

6.2.4 Machine Learning-based Surrogate Model

The described simulation model is the source of data to develop the ML-SMod. Building efficient SMods has become a fundamental tool to reduce computational costs without sacrificing accuracy. They are simplified representations of a computationally expensive simulation model. Its goal is to emulate the behavior of the

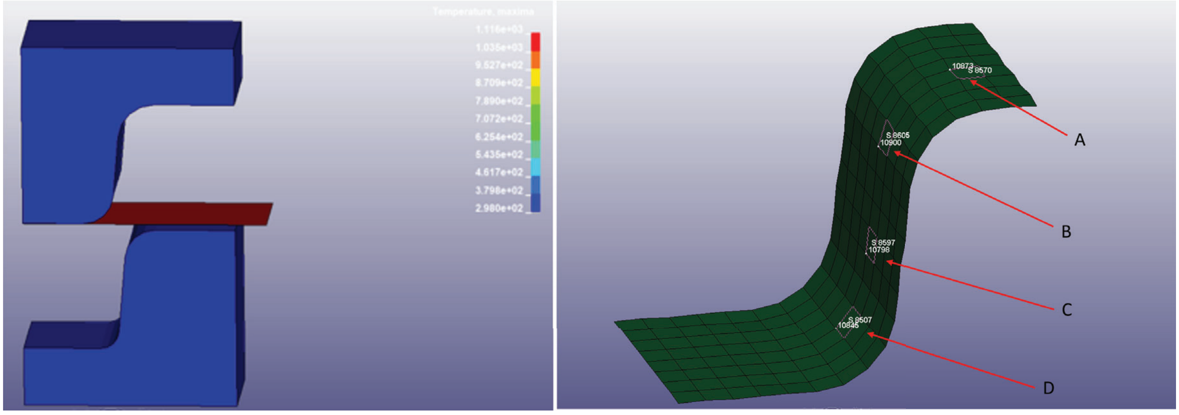


Figure 6.2: Left: Initial state of the LS-Dyna simulation. Right: Deformed part at the end of the simulation.

original model with sufficient fidelity to allow design space exploration, parameter optimization or sensitivity analysis, among other tasks.

6.2.4.1 Sampling Strategies and Data

The efficiency of a SMod is closely linked to the number of sample points used for its construction. A higher number of sample points generally yields higher accuracy, but also higher computational cost in the construction phase. Therefore, the viability of a SMod resides in the ability to find an optimal balance between accuracy and cost. In other words, the total cost of building the SMod and using it for the purpose of interest (optimization, reliability analysis, etc.) should be significantly lower than the cost of performing the same analysis using the original simulator. Therefore, selecting an adequate number of sample points is crucial. A SMod with too few points may be inaccurate and lead to erroneous conclusions. On the other hand, a SMod with too many points may result in excessive computational cost, negating the advantages of its use. Ultimately, building efficient SMods requires careful analysis of the accuracy requirements and computational limitations of the problem at hand. The goal is to minimize the total cost of the analysis, including building and operating the SMod, without compromising the accuracy needed to obtain reliable results. In this context, the sampling stategies come to play to select carefully the required simulations come to play. In this use case, we have compared three different sampling strategies to build the SMod of the hot stamping process:

Latin Hypercube Sampling Latin Hypercube Sampling (LHS) [242] is a popular technique for the design of experiments, which uniformly distributes the samples within the domain. Its name comes from the assumption that each of the input parameters of your system is a dimension of an hypercube. LHS distributes the sampling points within this hypercube strategically, ensuring that they cover the entire parameter space evenly. This is crucial when you have no prior knowledge of how the parameters influence the outcome, as it allows you to explore the full range of possibilities. For the sake of clarity, Figure 6.3 provides a visual example of the LHS distribution of samples in a given domain of 2 input variables with values inside a defined range [0,1]. In this case, the input variables form a 2D space (a square) and 10 points must be sampled. Then, in each dimension of the space, a 10 uniform subdivisions are done. Afterwards, 10 random points are selected with the condition that, taking into account all the dimensions, each subdivision must contain a single point.

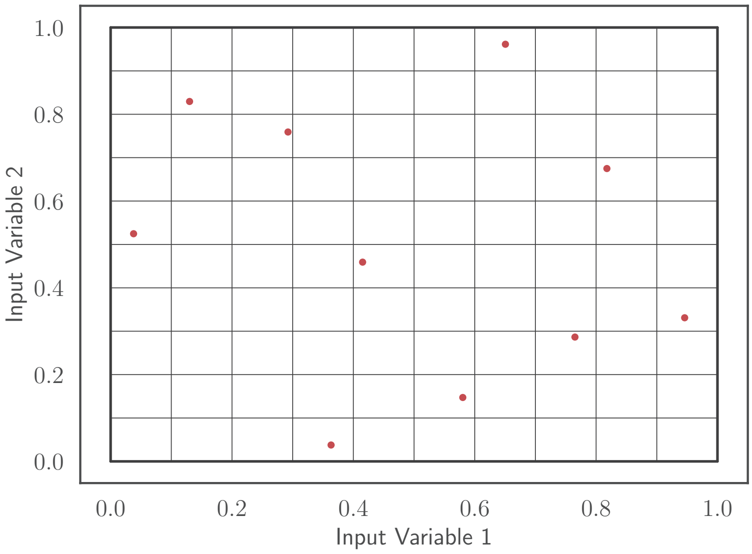


Figure 6.3: Example of LHS method in 2D.

In addition to effectively covering the parameter space, LHS prevents the sampling points from clustering in the same area. This is important because some SMods can have numerical problems if the data are too close to each other. Hence, LHS is a very useful technique thanks to its “space-filling” and “non-collapsing” properties which allow to build accurate and reliable SMods.

In this use case, as a baseline, we use LHS to generate 3 sets of 25, 50 and 100 samples. The samples are formed by a vector of dimension 4, corresponding to the variables in the domain. Next, we have used these sets of input samples to perform LS-Dyna simulations to get the temperature and the martensite fraction in the pointed elements indicated in Figure 6.2. Adding the corresponding outputs to the samples leads to the input-output datasets LHS_25, LHS_50 and LHS_100 detailed in Table 6.2.

Forward Selection Method The Forward Selection (FS) method [243] consists in the stepwise addition of samples to train an empty model choosing the ones that improve its performance. The process continues until a stopping criterion is reached, which can be given by a maximum number of samples or a threshold in the evaluation metrics. For FS method, we use the available samples of LHS_25, LHS_50 and LHS_100. First, we randomly select 150 samples for training and 25 samples for validation. The 150 training samples are split in groups of 10 samples, i.e., we obtain 15 subsets of 10 samples. Then, we train 15 models with the 15 subsets of 10 samples and we evaluate them in the 25 validation samples. The subset that has generated the best model is added in an empty training dataset. In the following iteration, we do the same with the 14 remaining subsets, and the best subset is included into the training dataset. This procedure is repeated until the training dataset contains 100 samples. The resulting training dataset is called FS_100 (Table 6.2).

Table 6.2: Datasets obtained from the LHS and LS-Dyna simulations and their purpose.

Dataset	Inputs	Outputs	Number of samples	Purpose
LHS_25			25	Train
LHS_50			50	Train
LHS_100	T_i^k, h, C, t	T_f^k, X_m^k	100	Train
LHS_25_test			25	Test
FS_100			100	Train

6.2.4.2 Model Details

The SMod is based on two DNNs with fully- connected layers. As a starting point, both DNNs are formed by an input layer of 4 neurons corresponding to the 4 input variables, 3 hidden layers and an output layer of 4 neurons and a linear activation function, trained with the Adam algorithm [193]. A hyperparameter tunning process is used to determine the DNNs parameters. The DNNs aim to perform a regression to predict the target outputs in the 4 points of the part geometry. Due to the range difference of the outputs, the T_f^k and the X_m^k are predicted separately and each DNN focus on one output variable. The inputs are scaled according to $x_s = \frac{x - x_{min}}{x_{max} - x_{min}}$, to ensure that their contribution is the same during the model fitting.

6.2.4.3 Training and Hyperparameter Tunning

The training of the DNNs of the SMod is done with the LS-Dyna® simulations datasets LHS_25, LHS_50 and LHS_100, obtaining 3 SMods. Also, the dataset FS_100 is used to train 10 DNNs increasing the number of training samples by 10 in each DNN. A hyperparameter tunning process based on a 5-Fold Cross Validation (CV) [195] is carried out in each case to select the best hyperparameters of the DNNs, including the batch size, the epochs, the number of neurons and the activation function of each hidden layer and the learning rate. The CV method allows to use the training data for tunning the model. After the hyperparameter selection, all samples are used to train the DNNs models.

6.2.4.4 Evaluation and Generalization Metrics

To evaluate the performance of the SMods and prove their generalization capability in unseen configurations of the domain, we use the dataset of LHS_25_test as test. The generalization metrics used in this use case are the Mean Absolute Error for each k element, $MAE_k = \frac{\sum_{i=1}^{N} |x_i - \hat{x}_i}{N}$, where k is the element, x_i are the simulation values, $\hat{x}_i$ are the predicted values and N are the number of samples in the test dataset. Besides, a global metric for all the elements predictions is calculated: $MAE_{glob} = \frac{\sum_k MAE_k}{4}$, where $k = A, B, C, D$.

6.3 Results and Discussion

6.3.1 Baseline Machine Learning-based Surrogate Model

The training and the hyperparameter tunning have been done using the datasets generated with the LHS strategy, LHS_25, LHS_50, LHS_100, giving rise to 3 SMods. The evaluation of the resulting SMods is done in the test set, LHS_25_test. The test set acts as an example on how the models perform in predicting unseen configurations within the whole domain. The quality assessed in the predictions will prove the validity of applying the SMod for a sensitivity analysis study.

Table 6.3: Metrics of the predictions of the SMods in the test dataset. The MAE_k is computed for both temperature and martensite fraction predictions for the $k = A, B, C, D$ elements considered.

Model	T_f^A	T_f^B	T_f^C	T_f^D	X_m^A	X_m^B	X_m^C	X_m^D
	$MAE_A(K)$	$MAE_B(K)$	$MAE_C(K)$	$MAE_D(K)$	MAE_A	MAE_B	MAE_C	MAE_D
SMod LHS_25	4.652	8.024	6.295	8.613	0.054	0.048	0.042	0.034
SMod LHS_50	4.253	4.386	4.187	6.605	0.030	0.019	0.022	0.023
SMod LHS_100	1.671	4.536	1.828	3.277	0.010	0.018	0.017	0.014

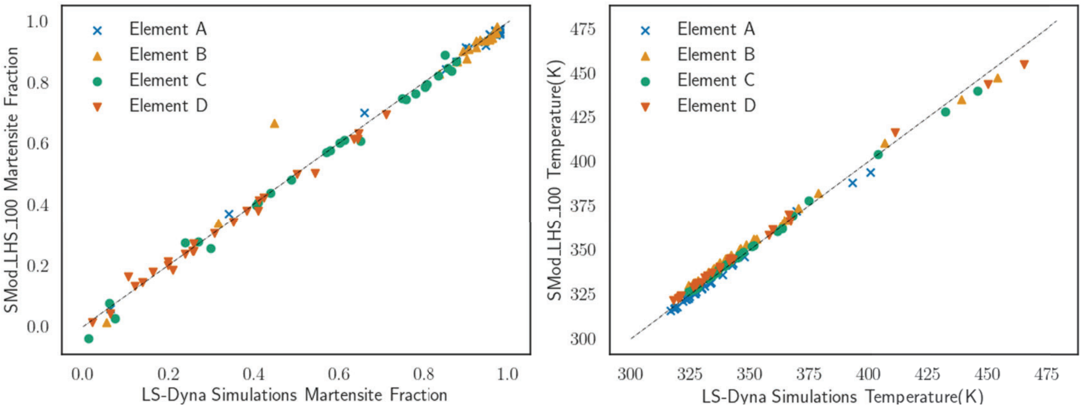


Figure 6.4: Comparison between the predictions of the SMod_LHS_100 and the simulation outputs. Left: Temperature. Right: Martensite fraction.

In Table 6.3, we present the metrics of the predictions of the SMods in the test set. As we increase the number of training samples the SMods improve their predictions. The best model SMod_LHS_100 performance is represented in Figure 6.4, comparing the predicted values with the simulation outputs. The SMod captures the behavior of the hot stamping outputs for the configurations of the test set and the DNNs identify the patterns in different regions of the part. For instance, focusing on the region of the element A, the contact with the tool is good and, for that reason, in almost all the configurations $X_m^A > 0.8$ and $T_f^A < 400K$. On the contrary, the contact in the region of the element D is not optimal. The curvature produces a higher gap between the part and the tools, achieving high temperatures for some configurations and $X_m^D < 0.7$ for most of the cases. Moreover, the approximated CPU time (varies with the inputs) of an Intel Core i7–10610 CPU using 4 processors is:

- LS-Dyna® Simulations: ~ 900 s (1 cycle) and ~ 22000 s (LHS_25_test, 25 cycles).
- SMod_LHS_100: ~ 0.02 s (1 cycle) and ~ 0.50 s (LHS_25_test, 25 cycles)

These CPU times remark the soft-real time response of the SMod predictions and the huge gain respect to the simulations. Then, the SMod accomplish the objective to provide a soft-real time reliable response for sensitivity analysis or optimization routines inside the configuration domain.

6.3.2 Forward Selection Strategy

The dataset generated with the FS method leads to 10 SMods. Each SMod has a different number of training samples from $N = 10, 20, \ldots, 100$, corresponding to the iterations of the described FS method. The evaluation has been done in the test set LHS_25_test. The results are shown in Figure 6.5. We observe that with 70 training samples, the SMod trained with the FS method achieves a lower MAE_{glob} compared to the SMod_LHS_100. Usually, the SMod benefits more from some regions of the domain and the uniform distribution of LHS is not optimal. Also, a higher number of samples does not always mean more accuracy, since problems like overfitting may arise. This proves that a proper sampling strategy is crucial to reduce the number of training samples of the SMod without impacting its performance. Hence, in this case, the SMod keeps the accuracy decreasing the training samples from 100 to 70, i.e., reducing the simulation time in the data generation phase a 30 %.

The results show the succesful creation of a ML-based SMod to predict the temperature and material properties of a complex part during a hot stamping process. This model offers valuable insights and methods to improve the manufacturing process. Because the model can predict in real-time, it can be used to fine-tune the process and make it more efficient, leading to better control and monitoring of hot stamping.

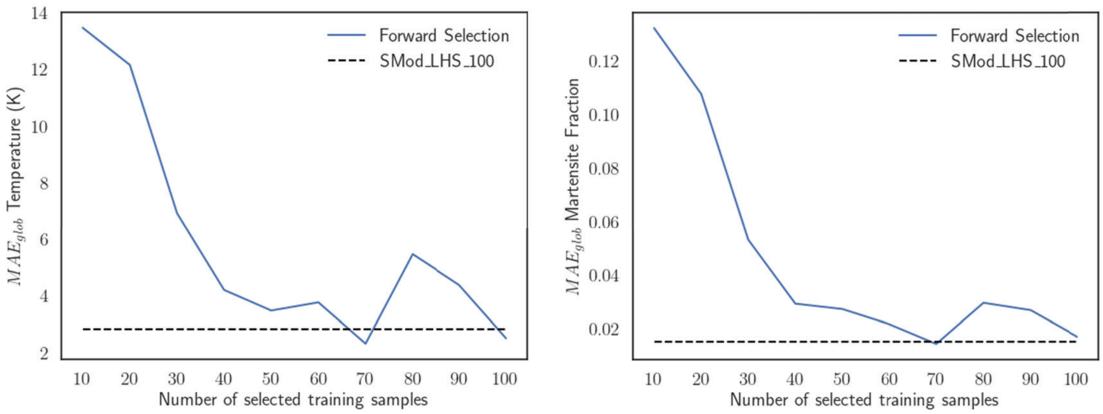


Figure 6.5: MAE_{glob} evolution as function of the number of training samples. Right: Temperature. Left: Martensite fraction.

6.4 Ongoing work

The previous results have proven that a good sampling strategy can reduce the size of the training dataset and, consequently, the simulation time of the overall pipeline. The approach have shown how a proper selection of the data can reduce the number of necessary training data to achieve a trustworthy SMod. Based on this idea, we aim to extend the study proposing an improved method where only the optimal simulations for training the SMod are done. Namely, there is no need to generate a prior dataset to construct the SMod. Then, the current section presents a method that addresses this using an Adaptive Sampling (AS) method. While the full implementation and analysis of this method are still in progress, this thesis provides an descriptive overview of its development and some promising results. The resulting work will focus on the full implementation of the method in the current hot stamping scenario.

Adaptive Sampling (AdSam) is a technique used to efficiently construct accurate SMod. Instead of relying on a predefined set of sample points, AdSam refines the SMod iteratively. This method starts with an initial set of sample points, often generated by LHS, which is used to evaluate the computationally expensive simulation and build a preliminary SMod. Subsequently, an analysis of the initial SMod is performed to identify the regions where the model presents greater uncertainty or prediction error due to the scarcity of information.

Based on this uncertainty assessment, new sample points are strategically selected in these regions. The expensive simulation is evaluated at these new points, and the SMod is updated accordingly. This iterative cycle of uncertainty assessment, selection of new samples and updating the SMod continues until a predefined stopping criterion is met, such as reaching a desired level of accuracy or reaching the maximum number of samples.

The main advantage of AdSam lies in its ability to concentrate computational resources on areas where the model requires further refinement. This strategic focus minimizes the number of costly simulations required to reach the desired level of accuracy. Furthermore, the iterative nature of the process allows for flexibility and adaptation based on observed model behavior, resulting in more accurate SMods. AdSam has proven its effectiveness in a wide range of applications, including engineering design, environmental modeling and optimization of complex systems thanks to its ability to generate accurate predictions with limited data.

To extend the previous study, an AdSam approach is proposed to create a SMod of the hot stamping LS-Dyna® simulations of a hat-shaped geometry. This demands a continuous interaction and a connection between the surrogate modeling framework and the simulation model. In what follows, we describe the proposed AdSam method, which is showcased in the flowchart displayed in Figure 6.6:

Initially, an initial set of samples of $m = 10$ is generated through the LHS strategy, using the configuration domain conformed by the input variables defined in Section 6.2.2. From this initial domain, LS-Dyna® is used to create a training set, which will be updated in each iteration of the AdSam process. Each iteration consists in:

1. Generation of a SMod with the current training set. The AdSam process ends if the evaluation of the SMod fulfills the stopping criterion. In our case, the stopping criteria are that the size of the training set is $m \leq 70$.

2. If the stopping criteria is not fulfilled, the current training set is divided in 5 subsets, that are used construct 5 different SMods. Each subset is formed by different samples. For instance, imagine that S1, S2, S3, S4 and S5 are the subsets and each one contain the same number of samples. Then, the first SMod1 is trained using S1, S2, S3 and S4. The second SMod2 uses S2, S3, S4 and S5. In this way, each of the generated SMods is built using different data.

3. 100 candidate samples from the domain are chosen ramdomly and predicted by the 5 SMods. This means each candidate sample is a configuration of the input variables. Then, the SMod1, SMod2, SMod3, SMod4 and SMod5 predict the outputs in each of the candidate samples.

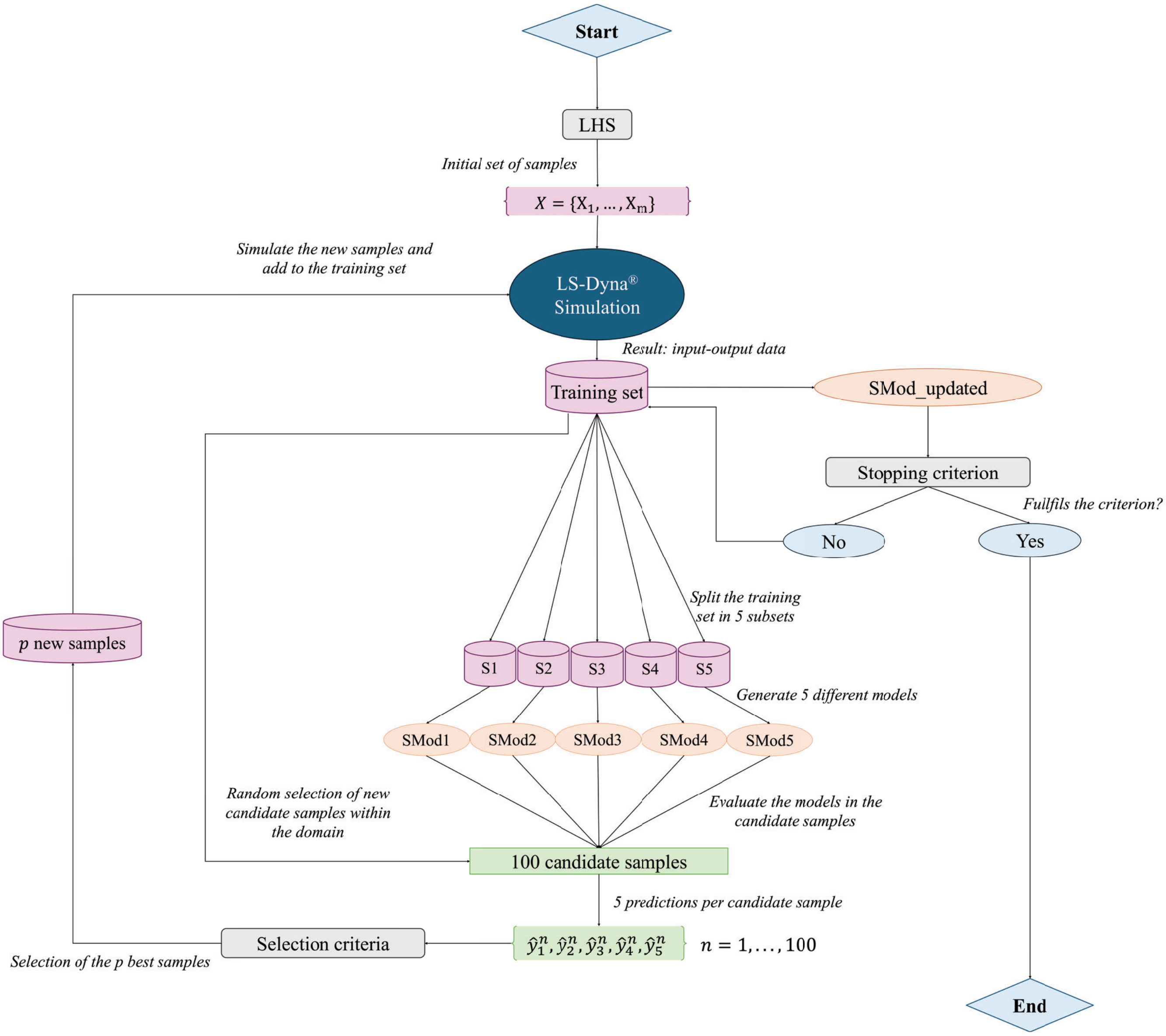

Start
LHS
Initial set of samples
$X = \{X_1, \ldots, X_m\}$
Simulate the new samples and add to the training set
LS-Dyna® Simulation
Result: input-output data
Training set
SMod_updated
Stopping criterion
Fulfils the criterion?
No
Yes
Split the training set in 5 subsets
p new samples
S1
S2
S3
S4
S5
Generate 5 different models
SMod1
SMod2
SMod3
SMod4
SMod5
Random selection of new candidate samples within the domain
Evaluate the models in the candidate samples
100 candidate samples
5 predictions per candidate sample
Selection criteria
$\hat{y}_1^n, \hat{y}_2^n, \hat{y}_3^n, \hat{y}_4^n, \hat{y}_5^n$
$n = 1, \ldots, 100$
Selection of the p best samples
End

Figure 6.6: Flowchart of the AdSam method proposed in this study.

4. Afterwards, from the results of the predictions, a selection criteria is applied to select the best samples to be added in the training set. In our case, the variance between the predictions of each SMod is computed in each candidate sample for the temperature and the martensite fraction. Since we have 4 elements in the geometry to predict, actually we consider the variance σ_T^2 (temperature) and σ_X^2 (martensite fraction) as the mean of the variances of the predictions in each of the elements. Then, we define a global variance for each candidate sample n:

$$\sigma^2(n) = \frac{\sigma_T^2(n)}{\max(\sigma_T^2(n))} + \frac{\sigma_X^2(n)}{\max(\sigma_X^2(n))} \tag{6.1}$$

that considers equally both output variables, temperature and martensite fraction, by normalizing each one of the terms in Eq. 6.1. The selection criteria is based on selecting the p candidate samples with the highest value of σ^2. In the considered case $p = 5$. The hypothesis is that if the 5 SMods predictions differ a lot between them, this means that there is a lack of information about this area of the domain in the training set.

5. Once the best $p = 5$ candidate samples are selected to be added in the training set, the corresponding LS-Dyna® simulations are performed. In this sense, we obtain the outputs of these new 5 of the domain, and they can be included in the updated training set. The process is repeated from 1 until the stopping criterion is fulfilled.

6.4.0.1 Preliminary Results and Next steps

The implementation and evaluation of this AdSam method is still in progress. We expect to compare the prediction metrics of the ML-based SMod created using the AdSam with the benchmark SMods constructed through LHS and FS methods.

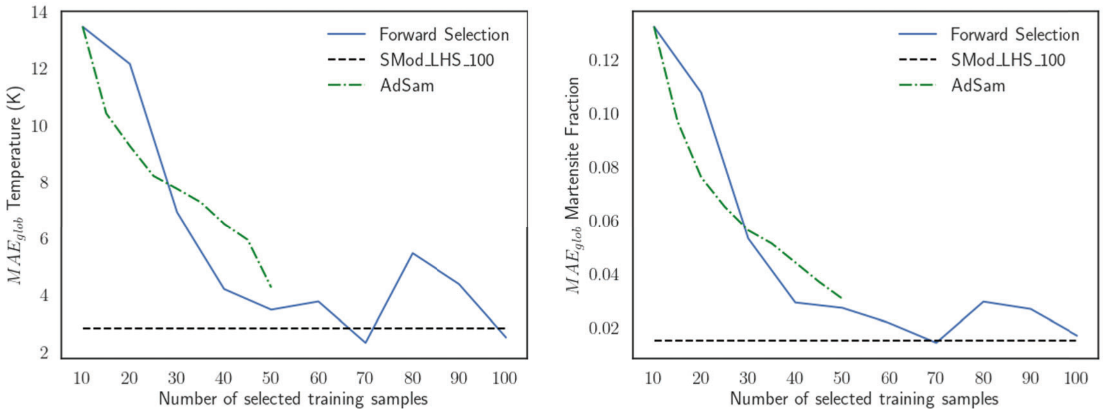


Figure 6.7: MAE_{glob} evolution as function of the number of training samples comparing the FS method, the AdSam strategy and the benchmark SMod_LHS_100 results. Right: Temperature. Left: Martensite fraction.

The preliminary results are displayed in Figure 6.7, where we can observe the evolution of the metrics of non-optimal SMods build with the AdSam procedure up to 50 training samples. The tendency of the AdSam curve shows promising results, with still a lack of hyperparameter tunning of the models and the addition of more training samples.

The effectiveness of the method will be demonstrated if the resulting ML-based SMod generated with AdSam achieves lower metrics than the SMod generated through the FS strategy using the same or less training samples, confirming our hypothesis.

6.5 Lessons Learned

- An important consideration when building ML-based SMods is the significant influence of sampling techniques on the accuracy and efficiency of surrogate models (SMods).

- By building upon the parameter interpolation approach presented in Section 5.3.2, this study provides a deeper understanding of how different sampling methods can impact the performance of SMods in predicting critical process variables in hot stamping simulations.

- The successful development of a ML-based SMod capable of predicting final temperature and martensite content in different regions of a hot-stamped hat-shaped part provides further valuable knowledge and methodologies in the manufacturing process of hot stamping. The SMod soft real-time prediction capability can enhance sensitivity analysis and optimization procedures, leading to improved process control and monitoring in hot stamping applications.

- The investigation performs a comparative analysis of Latin Hypercube Sampling and Forward Selection methods, highlighting the importance of the sampling strategy in surrogate modeling. This finding is directly related to the thesis objective of accelerating data generation and analysis in manufacturing processes. Ongoing work about the application of an Adaptive Sampling to this use case indicated a strong interest to keep the investigation in that direction and in the creation of an optimized sampling strategy for the hot stamping process.

Chapter 7

Extension of the Hot Stamping Case - Transfer Learning to Real Industrial Plant

This chapter investigaties a SMod extension of the baseline SMod of the hot stamping process introduced in Section 5.3.2 is investigated. The study in this chapter aims to transfer the ML-based SMod in a real industrial plant. While in the practical application of Section 5.3.2, all the data has been obtained from simulation models, in the current study the process is tested in a real hot stamping plant. Hence, we address the sim-to-real problem in surrogate modeling, focusing in a highly efficient method based on Transfer Learning (TL) to leverage the knowledge learned in the simulation environment into the real manufacturing environment.

Usually, data acquisition in real manufacturing plants is a very expensive task and FE simulations are employed to train ML-based SMods. However, the approximations of the FE models may induce a deviation from reality that is transferred to the SMods. This chapter proposes a methodology to combine AI-based surrogate modeling and Transfer Learning (TL) to create a trustworthy and efficient SMod of a real manufacturing process, using a low-fidelity FE model as data source. In particular, the methodology has been demonstrated in a study involving hot stamping of boron steel sheet in a pilot plant. Two DNNs have been trained with low-fidelity ABAQUS simulations [230], forming a baseline SMod that predicts the key outputs of the process. The use of few experimental real data of the process to perform TL and adapt the original baseline SMod to the real environment shows remarkable results, surpassing other Variable-Fidelity Modeling (VFM) approaches. The final TL SMod provides fast and accurate predictions of the most relevant outputs of the real process with little training, and it removes completely the calibration stage or the need of a high-fidelity simulation model. Additionally, the presented methodology can be a trigger for creating efficient virtual manufacturing environments that can enable developing DTs or process optimization tools like RL.

The investigation performed in this chapter can be found in the article in [4].

7.1 Introduction

As it has been mentioned in Chapter 2, there are several limitations of both industrial environments and simulations for the acquisition of manufacturing data. To minimize the impact of the simulations drawbacks, the development of surrogate models (SMods) has become a trend in the last few years [175–178]. The introduction of SMods is an enabling technology in manufacturing, since they are able to enhance the data generation and the fast response, overcoming the high computational and time cost of FE simulations.

Regardless of the advantages of the ML-based SMods, their training usually implies an important volume of simulation data and the number of samples have a direct impact in the accuracy [172, 174, 244]. Also, the fidelity of the simulations respect to the real process is affects the SMod performance [245]. Then, ideally, high-fidelity simulation models should be used. Nevertheless, in addition to the commented issues, working with high-fidelity manufacturing simulation models requires:

- A continuous and intensive calibration of their parameters through experimental validations in order to adjust the real process [246]. As commented, the acquisition of real data has a very high temporal and economical cost.
- In the case of FE simulations, there exists a direct relationship between accuracy and time. As the mesh density increases so does the accuracy but the simulation time also increases, and vice-versa. The dependence on the complexity of the mesh to increment the precision and fidelity aggravates the temporal limitations [168, 247].

Therefore, the construction of a SMod is also affected by the temporal constraints of the simulations and the training process must be time efficient to avoid suffering the same problem that it aims to solve. For that reason, several approximations are generally applied in the simulation models that reduce the time cost but may affect the reliability [248, 249]. Although behaviors and patterns of the process can be correctly captured, some discrepancies between the simulation results and the real measurements of the process variables may arise, called sim-to-real gap [250, 251]. Consequently, SMods trained on simulations inherit the same prediction deviations present in those simulations. This limits their potential for boosting the data generation in manufacturing process or the creation of DTs and RL agents which intend to act directly on real-world environments [252, 253]. To sum up, the scarcity of data from real industrial plants enhances the need of the SMod being a reliable representation of the real manufacturing process.

VFM has emerged as a surrogate modeling technique to address the above drawbacks of high-fidelity simulations in manufacturing and engineering. By strategically employing numerous low-fidelity simulations, which are computationally less demanding, VFM enables the extraction of crucial patterns and insights even when high-fidelity data is scarce. This approach significantly reduces computational costs and time without impacting in the accuracy of the resulting SMod [254–256]. VFM has been successfully implemented in various domains. For instance, in aerodynamic design optimization, VFM has been used rapidly exploring the design space with low-fidelity models, while relying on high-fidelity simulations to iteratively improve the SMod [257]. In the context of uncertainty quantification, VFM has been employed to propagate uncertainties from low-fidelity models to high-fidelity predictions, providing a more comprehensive understanding of the system behaviour [258, 259].

In this chapter, we present a pipeline that introduces Transfer Learning (TL) to construct a SMod of a real manufacturing plant, where the data acquisition is a very costly process. TL is an established machine learning technique for transferring the knowledge of a model trained in a source domain to a target domain, with the aim to reuse, adapt, and boost the model learning process in the new domain [260–262]. Accordingly, the presented methodology for building an AI-based surrogate model of a real manufacturing environment uses a simple and fast FE model as the main source of data. The methodology includes surrogate modeling and the application of TL to leverage the knowledge from the low-fidelity simulations to the real environment and to solve the sim-to-real gap. In the considered case, a reduced amount of high-fidelity data is obtained from a real environment while the low-fidelity data comes from a simplified FE simulation model. Concretely, the methodology has been implemented in the use case of a hot stamping pilot plant. Initially, we perform a random exploration of the entire configuration domain within the feasible range of variables, as defined by the pilot hot stamping plant. This exploration is conducted using simplified low-fidelity FE simulations of the hot stamping process. Next, the high amount of data generated from the simulations is used to train two Deep Neural Networks (DNNs) that predict the most important temperatures of the hot stamping process. The DNNs form a baseline surrogate model (SModBase) of the FE simulations. Finally, we apply TL to fine-tune the DNNs of the baseline SMod with a very reduced quantity of data from the real industrial plant. The final TL-SMod shows very accurate results in the prediction of the relevant variables of the hot stamping process in the real plant, outperforming a surrogate model trained only with experimental data and the standard VFM approach that merges low-fidelity and high-fidelity data.

Summarizing, the main contributions of this chapter are highlighted in the following points:

1. The use of the transfer learning (TL) method to solve both VFM problem and the sim-to-real gap problem at the same time generates an efficient and effective surrogate model of a real hot stamping plant. Contrary to current literature, as far as we know, our methodology addresses two steps at the same time: First, the correction of low-fidelity simulations to high-fidelity models, which is commonly addressed in engineering through VFM techniques. Second, the adjustment of the sim-to-real gap from the simulation environment to the real environment. Instead of solving the problems separately, the proposed approach uses TL to merge both problems and it shortcuts and boosts the process to create a SMod of the real environment.

2. We thoroughly explore the influence of different TL options and relevant parameters as well as analyze the methodology robustness in terms of the amount of real data required for a successful application of the technique. Also, the effect of the simulation data fidelity level is studied.

3. The resulting TL-SMod is applied and showcased in a real hot stamping plant showing the viability and effectivity of the method and allowing for quick exploration of critical parameters of the manufacturing process.

7.2 Methodology

In this section, we introduce the general overview of the proposed SMod-TL approach and we provide a comprehensive explanation of the applied methods, VFM and TL.

7.2.1 Overview of the Surrogate Modeling-Transfer Learning Process

As previously mentioned, in the specific case of hot stamping, FE modeling is a mature, well-established technique when used for the design and analysis of production processes [263]. The main reason for the use of this technique is that experimental tests imply an important waste of materials, wear of the tools and the misspend of resources. The fidelity of these simulations is directly proportional to the complexity of the used FE model. Thus, a large computational time and resources are needed in order to perform a well-grounded analysis. Then, it is usual to assume approximations and simplifications assumed the FE models which may lead to deviations between the real hot stamping process and the simulations. This use case presents a methodology to efficiently obtain accurate data of the hot stamping process without perturbing the real manufacturing system by means of combining FE simulations, real experiments and ML techniques.

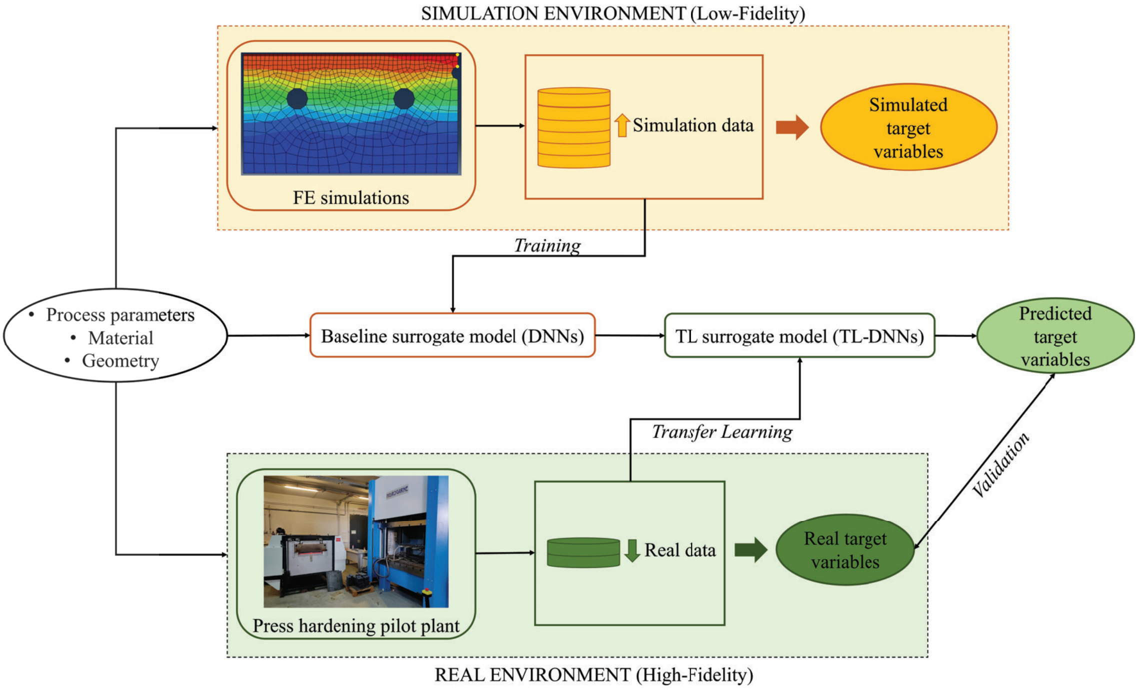

SIMULATION ENVIRONMENT (Low-Fidelity)
FE simulations
Simulation data
Simulated target variables
Training
Process parameters
Material
Geometry
Baseline surrogate model (DNNs)
TL surrogate model (TL-DNNs)
Predicted target variables
Transfer Learning
Validation
Press hardening pilot plant
Real data
Real target variables
REAL ENVIRONMENT (High-Fidelity)

Figure 7.1: Flowchart of the pipeline to build a SMod using FE simulations and TL to predict the target variables in the hot stamping pilot plant.

Figure 7.1 shows an schematic description of the proposed SMod-TL process. First, we create a baseline surrogate model, SModBase, using two DNNs that are trained with low-fidelity FE simulations adjusted to the process parameters of the target hot stamping environment with the goal of predicting the two relevant variables of the process, the temperature of the die and the sheet, respectively. From the real environment, high-fidelity data from the actual hot stamping process are collected by means of two pyrometers and a thermocouple. To reduce the discrepancy between the prediction and the real process, TL is applied. This allows to fine-tune the SModBase with the reduced data from the real manufacturing plant. As a result, an efficient and accurate predictor of the real environment key variables is produced.

7.2.2 Variable-Fidelity Modeling and Transfer Learning

Variable-Fidelity Modeling (VFM), also known as multi-fidelity modeling, is understood as the construction of surrogate models using data from different fidelity levels of the same process [254]. The idea is to employ the low-fidelity models to perform a cheaper and faster analysis of the parameter space overcoming the curse of dimensionality, obtaining an approximate representation of the system dynamics and reducing the volume of high-fidelity data required to build a SMod. However, the accuracy of a SMod build only with low-fidelity data is not desirable. Hence, the high-fidelity model is still fundamental to obtain a precise response in a few points of the parameter space. The resulting data from distinct sources and fidelity is combined leading to a variable-fidelity SMod (VFSM). The computational efficiency, the time optimization and the accuracy in the design space exploration of this strategy can even surpass the traditional surrogate modeling methods [264]. The main problem of the traditional approaches to build SMods is that the prediction accuracy directly depends on the number and the confidence level of the training samples. Therefore, the sample generation process and the SMod creation can be very costly when working with single high-fidelity FE simulations. Otherwise, VFSM can be used as a time and computationally efficient alternative of the conventional surrogate modeling.

The most common approach in VFM is the bridge function method [256]. Assuming a high-fidelity model (HFM) and a low-fidelity model (LFM), the corresponding surrogate models are a high-fidelity surrogate model (HFSM) and a low-fidelity surrogate model (LFSM), respectively. As commented, the development of a HFSM only from the samples of HFM is very expensive. Therefore, the bridge function method consists in the introduction of an additive or multiplicative correction that link the different fidelity models to construct the variable-fidelity surrogate model (VFSM). Depending on the relation considered between the two fidelity models, the VFSM approach is expressed in terms of Eq. 7.1 for the additive case and Eq. 7.2 for the multiplicative case.

$$y_{\text{VFSM}}(x) = y_{\text{LFSM}}(x) + \epsilon(x) \tag{7.1}$$

$$y_{\text{VFSM}}(x) = \rho(x) y_{\text{LFSM}}(x) \tag{7.2}$$

where $\epsilon(x)$ and $\rho(x)$ are SMods obtained from the difference or the division between the available HFM samples and the LFSM predictions of these samples. A combination of the two approaches is the comprehensive bridge function expressed in Eq. 7.3.

$$y_{\text{VFSM}}(x) = \rho y_{\text{LFSM}}(x) + \epsilon(x) \tag{7.3}$$

where ρ is a constant scaling function in this case and $\epsilon(x)$ is known as the discrepancy function. The VFM approach with bridge functions have been successfully modeled in literature using popular ML techniques. For instance, in [264] proposes an optimization framework applying a sampling strategy, a fuzzy clustering algorithm to reduce the design space and a Gaussian Processes (GP) regression to generate a VFSM from the multiplicative bridge function approach. The resulting VFSM-based environment serves as a tool to find optimal solutions very quickly with low computational cost. Also, in [265] a multi-fidelity DNN is employed to overcome the curse of dimensionality in uncertainty quantification methods.

Furthermore, an improvement of kriging called co-kriging has been very popular in modeling VFM problems. In [266], the method is explained in detail and it is extended to multiple fidelity levels. To sum up, it assumes the relation of the Eq. 7.3 between low and high-fidelity data and the correlation between data from different fidelity levels is taken into account. The idea is to incorporate the covariance between data into the kriging equations, allowing the model to balance the effect of both data sources. Thus, co-kriging has been applied in [267] to construct fast SMods from variable-fidelity CFD simulations to help to accelerate the optimization of aerodynamic surfaces. Another example is the use of co-kriging to overcome the curse of dimensionality in the surrogate modeling of electromagnetic simulations in [268].

The problem addressed in this study can be understood as an analogy of VFM. In our case, a hot stamping plant acts as a source of high-fidelity data, playing the same role as the HFM. Moreover, the elevated cost of data acquisition in industrial environments has already been mentioned, resulting in a very limited set of real data. On the other side, FE simulations are the source of low-fidelity data, namely LFM. This simulation environment allows to build SMods of the most relevant variables of the hot stamping process within a domain of configuration parameters, as demonstrated in Section 5.3.2 [2]. Despite this analogy with VFM, the proposed methodology differs from previous VFM works due to the introduction of TL, which not only solves the problem of having two-fidelity data sources, but also solves the called sim-to-real gap problem at the same time.

From its origins, Transfer Learning (TL) has proven to be a useful approach to handle problems where the target domain has a size-limited data set [262]. TL is a well-known learning framework based on the idea that a model trained to do a specific task in a specific domain can be adapted to perform another related task, or the same task but in another target domain of interest. This transfer of knowledge will improve the generalization performance of the model without the need of large amounts of data of the new task or the new domain [260, 261]. Therefore, TL is highly beneficial in domains where the data is limited and the available training data is insufficient or inaccurate. A more formal definition of TL is provided in [261]:

Given a source domain D_S and learning task T_S, a target domain D_T and learning task T_T, transfer learning aims to help improve the learning of the target predictive function $f_T(\cdot)$ in D_T using the knowledge in D_S and T_S, where $D_S \neq D_T$, or $T_S \neq T_T$.

DNNs have been established as the most usual framework for the application of TL, and several techniques have been implemented in this context, classified in [269]. According to this classification, the current study adopts a Network-Based Deep Transfer Learning method, detailed in Section 7.4.2.

TL has been widely used in fields like computer vision [270] and natural language processing [271], due to the possibility to train general initial models in accessible huge data sets such as ImageNet [272] or WordNet [273], and afterwards adapt them for specific tasks through TL. In the manufacturing context, the expertise knowledge or AI-tools working on a process may not be effective when the configuration parameters, the materials or the sensors are modified. To handle the data scarcity suffered due to the computational cost of new manufacturing simulations and the difficulty to acquire new data from industrial environments, TL has been introduced and its capabiltiy to adapt to unseen scenarios only using few data samples has shown promising results [274, 275].

Some recent works have used similar ideas of applying TL to face the VFM problem. In [276], a demonstration of the TL effectiveness with respect to other VFM methods is presented for the buckling prediction of variable-stiffness composite shells. Further, an approach for aerodynamic shape optimization that uses data from both LFM and HFM through a TL-CNN is proposed in [277]. Nevertheless, these studies do not focus on the real-world environment, and both the HFM and LFM are derived from FE models with varying mesh densities.

In some cases, the problem is not the use of less accurate LFM, but the mismatch between simulations and reality, called sim-to-real gap, even when using HFM in simulation. In addition, the small amount of available of data from the real processes is an important limitation for developing data-driven models. The sim-to-real gap may cause bad performance of ML models or the creation of non-representative SMods of the real scenarios. TL offers a solution to the sim-to-real gap problem, because it can take advantage of the knowledge of a simulation model which differs from reality (source domain) and transfer it to the real process (target domain). For instance, in [251] it is shown how TL can correct the discrepancy between simulation and real data and reduce significantly the samples of real data used to train a DNN for an accurate prediction the part weight in injection moulding.

Our study focus on creating a SMod of a hot stamping industrial plant. The methodology includes the VFM idea of employing a LFM as the main source of data to explore the configuration domain at a low cost and create a baseline low-fidelity SMod. Instead of a HFM, the high fidelity data comes from experiment data from a hot stamping plant and a sim-to-real gap exists between simulation (source) and real (target) data. The application of TL is the key point in the proposed method, since it allows to solve both VFM and sim-to-real gap problems, merging the ideas from the previously mentioned studies. The approach can achieve an important reduction of the number of required experiments and a trustworthy and efficient SMod that provides a very fast and accurate response can be generated.

7.3 Use Case Description

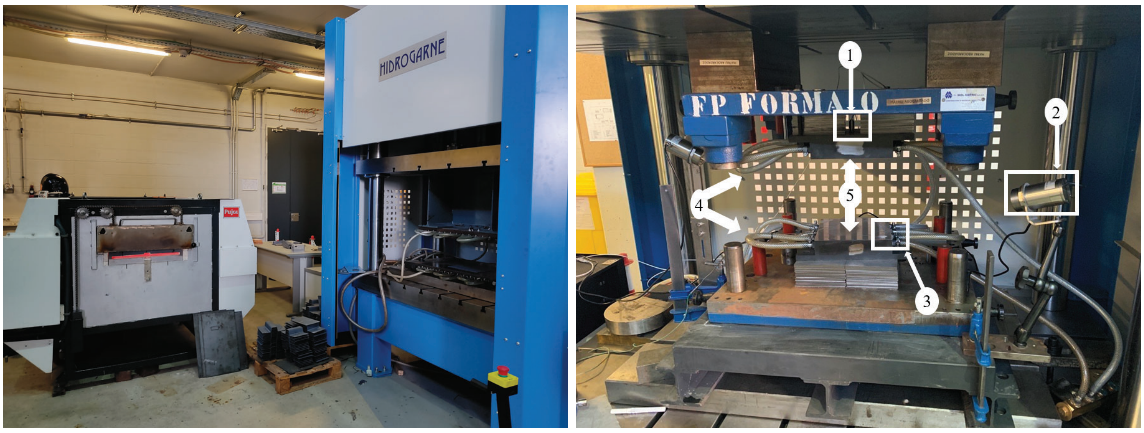
In the current section, we present the hot stamping process, briefly describing the pilot plant and identifying the process parameters and variables. The characteristics of the real environment and the simulation environment are also presented, along with the details regarding the collection data from both environments.

7.3.1 Hot stamping Process and Data Collection

The hot stamping process used in this chapter is identical than the one presented in Section 5.3.2. However, in the current study, we have realized the hot stamping process using both simulations and experiments in a real plant.

7.3.1.1 Pilot Plant Description

The data of the current study have been obtained reproducing a reduced-scale hot stamping line pilot plant located in Eurecat Manresa, described in [278]. Figure 7.2a illustrates this plant. It comprises a 3 m long continuous roller-hearth convection-radiation furnace and a set of water-cooled dies mounted on a hydraulic press system. All data have been collected within the standard operation conditions of this plant and under its spatial limitations, temperature ranges, etc.



(a) (b)

Figure 7.2: (**a**) The pilot plant at Eurecat Manresa where the tests have been performed. The furnace is located on the left-hand side of the image and the hydraulic press that contains the die is on the right-hand side. (**b**) Cooling die with cooling channels inside the hydraulic press. Localization of the three temperature sensors in the setup. 1) S1. 2) S2. 3) S3. 4) Cooling channels. 5) Cooling die.

The die is a flat water-cooled die made out of steel DIN 1.2344 (AISI H13) tempered at 48 ± 1 HRC, with water channels 10 mm in diameter and located at 20 mm depth from the surface, with a separation of 50 mm between centers.

These tools are used to process 200 mm × 100 mm blanks of commercial AlSi-coated 22MnB5 sheet, 1.7 mm in thickness. The corresponding chemical compositions of the used materials are exhibited in Table 5.6. More details on the geometry and materials can be found in Section 5.3.2 [2].

7.3.1.2 Process Parameters, Sensors and Variables

The process parameters of the tests performed in the pilot plant were given by the standard operational conditions of the pilot plant:

- Furnace temperature: 900 - 930 °C
- Cooling water temperature: 14.5 - 23.5 °C. It is the temperature of the cooling of the die.
- Cycle time (t_{cycle}): (30 ± 10) s. As it is clarified in Figure 4.4, a cycle is understood as the time between the moment a sheet is introduced into the die until the next one is transferred to the die. This time is determined by the distance between the sheet steel blanks into the continuous furnace, which has been done each 30 s. The human intervention to perform this action and the transfer of the hot blank into the die, induces an uncertainty of ± 10 s.
- Forming time (t_{form}): 5 - 15 s. During this time, the die is closed and the forming and quenching processes occur. In this case, the hydraulic press configuration allows to fix the desired forming time. However, it exists an minimum time required because below this threshold the forming and the quenching of the steel sheet is not executed correctly. However, the superior limit is given by the minimum time that the operator requires to extract the part from the die before the next steel sheet comes from the continuous furnace.
- Cooling time (t_{cool}): 5 - 35 s. It is the time when the die is not closed. Then, according to Figure 4.4, it can be expressed as $t_{cool} = t_{cycle} - t_{form}$. As can be noticed, this interval includes the uncertainty of the cycle time, which implies that in the worst case the operator only has 5 s to complete the cycle.

In addition of the mentioned process parameters, we have monitored the process through the deployment of temperature sensors in the simulation and real environments. Three temperature sensors have been placed in the experimental setup, as it is indicated in Figure 7.2b. We observe how the cooling die with the cooling channels is mounted on the hydraulic press and the distribution of the sensors. The sensors measure the temperatures of the steel sheet and the die, enabling the acquisition of the temperature profiles during the hot stamping process of all the cycles of a batch. The features of these sensors are presented in Table 7.1.

Table 7.1: Brief description of the three temperature sensors.

Sensor	**S1**	**S2**	**S3**
Type	Pyrometer	Pyrometer	Thermocouple
Temperature limits	Yes $> (148.9 \pm 1)$°C	Yes $> (253.0 \pm 1)$°C	No

According to its characteristics and its location, each sensor has a specific measurement role during a hot stamping cycle:

1. The pyrometer S1 is located above the die. It captures the radiation emitted by the lower part of the die surface, thanks to a hole in the superior part of the die. Since the temperature of the die itself is lower than the inferior limit of 150°C of the pyrometer, this sensor only reacts when the steel sheet is present. Then, its role is to measure the evolution of the steel sheet temperature during the forming process and it also determines the instant when the die opens.

2. The pyrometer S2 points also to the lower part of the die surface and it has an inferior limit of 250°C. Therefore, like S1, it only responds when the steel sheet is in the die, but in this case its measurement drops when the die closes since it is situated in an external spot. This enables to determine the moment of the closure of the die.

3. Finally, the thermocouple S3 is directly in contact with the die. It is inserted horizontally, measuring a spot in the center of the die and 8 mm under the die surface. It measures the temperature evolution of the die during the all the cycle steps of forming and cooling.

Therefore, using the sensor data, additionally to the process parameters, we will define two important input variables at the start of a cycle:

- Initial temperature of the die at the closure instant (T_{ini}^{D}): The range goes from the water temperature at the initial cycle up to a limit operation temperature that depends on the die characteristics. Nevertheless, in the studied process this variable is limited by the other process parameters and does not overcome the threshold of 150° C. It refers to the starting temperature of the die in the forming and quenching stage and its value is conditioned on the previous cycle parameters.

- Initial temperature of the sheet at the closure instant (T_{ini}^{S}): $\sim 680-880$°C. The range of values of this variable is highly dependent on the transfer of the part from the furnace to the die. This variable is the starting temperature of the posterior cooling of the sheet caused by the forming and quenching stage.

Once a cycle ends, it is important to know the quality of the final part and to understand the state of the setup to continue the batch production. As it has been mentioned in Section 5.3.2 [2], there are some output variables that are crucial in order to evaluate the hot stamping cycle. In this use case, we will focus on two of these target variables:

- Final temperature of the die at the opening instant (T_{fin}^{D}). It is the temperature of the die after the forming and cooling phases and it represents the state of the hot stamping setup after a cycle.

- Final temperature of the sheet at the opening instant (T_{fin}^{S}). It is the temperature of the final part and depending on its value ensures that the objective martensitic microstructure has been achieved.

Summarizing, the input variables for each cycle of the hot stamping process will be the forming time (t_{form}), the cooling time (t_{cool}), the initial temperature of the die (T_{ini}^{D}) and the initial temperature of the sheet (T_{ini}^{S}). The other mentioned parameters like the furnace temperature, the cooling water temperature and the cycle time (t_{cycle}) will remain within the established range during all the experiments. They are not considered input variables themselves since they are related on the real inputs and they are indirectly taken into account. For example, the furnace temperature regulates T_{ini}^{S}, the water temperature exerts influence on T_{ini}^{D} and t_{cycle} is the sum of $t_{form} + t_{cool}$. The output target variables will be the final temperature of the die (T_{fin}^{D}) and the final temperature of the sheet (T_{fin}^{S}).

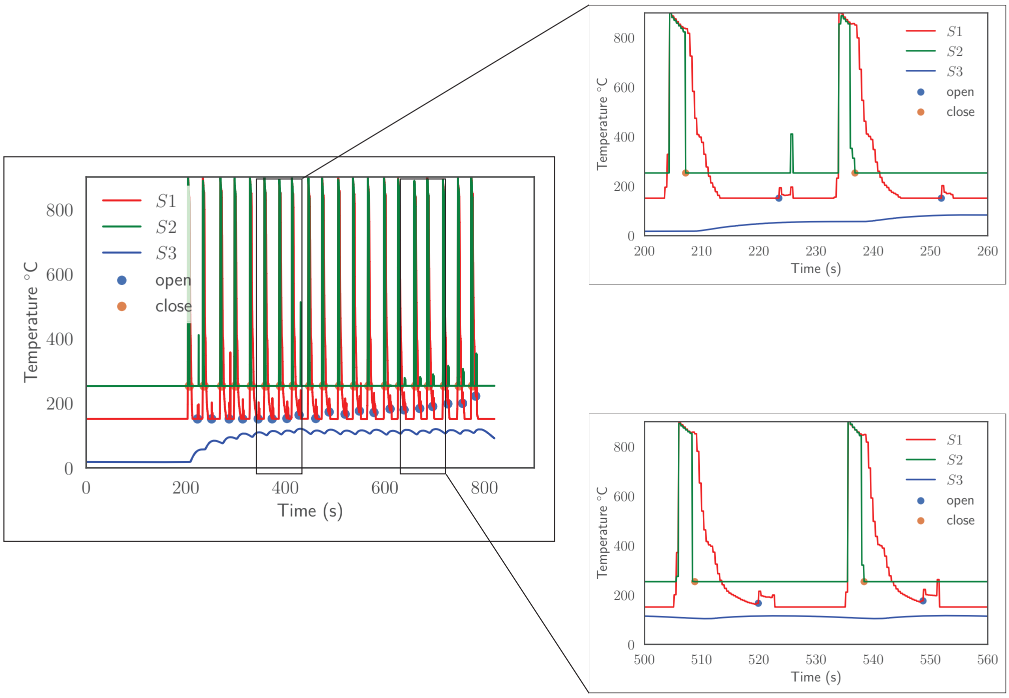
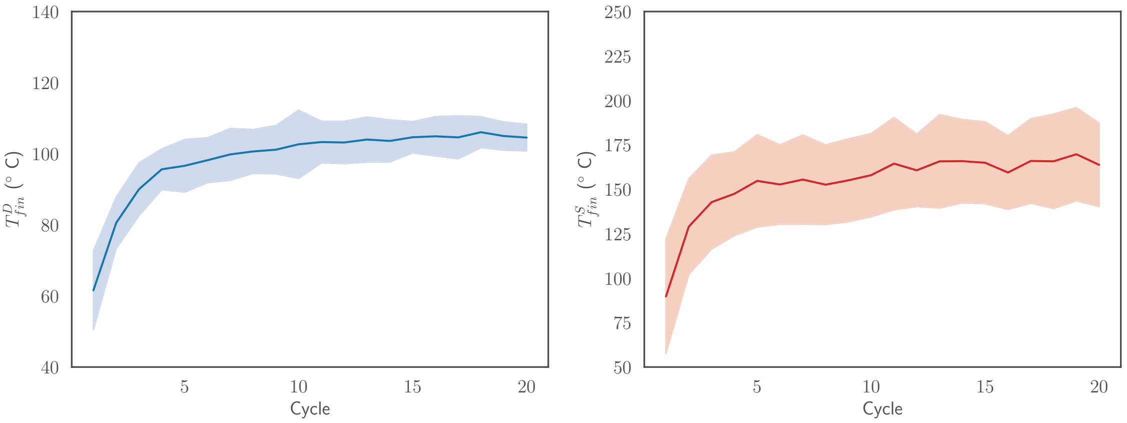


Figure 7.3: Temperature evolution measured by the three sensors. The moments of opening and closure of the die are pointed out. Left-hand side: Evolution of the sensor outputs for a whole batch of 20 cycles. Right-hand side: Zoom in of the first cycles (upper figure) and the intermediate cycles (lower figure).

The described behaviors of the sensors can be observed in Figure 7.3, where the evolution of the measured temperatures is shown for a whole batch and for individual cycles. The lower temperature limits of each sensor are identified in the first 200 s before the start of the first cycle. Once the first steel sheet emerges from the furnace and it is transferred to the die, we notice a sudden increase of S1 and S2, which detect the hot sheet. The posterior sudden drop of S2 to its ground temperature determines the closure of the die, and the values of S1 and S3 at this instant are T^S_{ini} and T^D_{ini}, respectively. During the forming stage, the heat transfer process between the die and the sheet induces a decrease of S1 and an increase in S3 until the die opens. This is captured by a small spike in the sensor S1, due to a change in the radiation received by the pyrometer in the opening process that ends when the sheet is extracted. At that moment, the measures of S1 and S3 correspond to T^S_{fin} and T^D_{fin}, respectively. The difference between the opening time and the closure time is the real forming time (t_{form}) of the cycle. Finally, in the last step S3 temperature decreases slowly and keeps cooling until the next cycle. It is important to mention that the detection of the peaks and sudden changes has been done with the derivatives of the sensor signals. Due to the difficulties caused by the temperature limit of the sensor S1, a procedure to measure T^S_{fin} is detailed in B.



(a)

(b)

Figure 7.4: Representation of the mean temperatures and the corresponding standard deviation of REAL$_{data}$. (**a**) T^D_{ini}. (**b**) T^S_{fin}.

In the setup of the pilot plant environment, we have been able to perform 40 experimental batches of 10 - 20 cycles. The resulting high-fidelity data set, REAL$_{data}$, is described in Table 7.2. The temperature evolution of the targets T^D_{fin} and T^S_{fin} is displayed in Figure 7.4. As noticed before, despite the variability along cycles, the repetition of the process causes a continued augment of the die and sheet temperatures, until a stationary regime is achieved around the eighth or tenth cycle.

7.3.2 Simulation Environment and Data

The simulation environment used in this study is the same than the one presented in Section 5.3.2.2, consisting in a 2D finite element model analysed by the ABAQUS software [230]. As a difference, in accordance with the variability of the real hot stamping plant, the sink temperature of the cooling channels was defined randomly for each cycle within the water temperature range mentioned in Section 7.3.1.2. Further details of the process can be found in Section 5.3.2 [2].

As commented in the previous Section 7.3.1.2, the input variables of the simulation model are t_{form}, t_{cool}, T^D_{ini} and T^S_{ini} within its given range of values. The variables of t_{form}, t_{cycle} and T^S_{ini} will take random values inside the established ranges. Hence, t_{cool} has also a randomness associated to t_{cycle} and t_{form}. Otherwise, T^D_{ini} has a initial value equal to the water temperature, but it evolves depending on the batch dynamics. The two sensors specified in Figure 5.19 have been situated strategically to mimic the positions of the sensors in the real system and

Table 7.2: Characteristics of the all the generated data sets in both environments.

Environment:	**Real**	**Simulation**	**Simulation**
Batches	40	500	40
Cycles per batch	10-20	20	10-20
Total samples	628	10000	628
Input variables	t_{form}, t_{cool}, T^D_{ini}, T^S_{ini}	t_{form}, t_{cool}, T^D_{ini}, T^S_{ini}	t_{form}, t_{cool}, T^D_{ini}, T^S_{ini}
Target variables	T^D_{fin}, T^S_{fin}	T^D_{fin}, T^S_{fin}	T^D_{fin}, T^S_{fin}
Application	Training and validation	Training	Validation
Abbreviation	REAL$_{data}$	SIM$_{data}$	SIM$_{valid}$
Fidelity	High	Low	Low

they capture the temperature profiles of the sheet and the die, shown in Figure 7.5. Depending on the input forming and cooling time, it is possible to determine the closure and opening instants of the die. In these points, we can measure T^D_{ini}, T^S_{ini}, and the process target variables T^D_{fin} and T^S_{fin}.

Figure 7.5 provides an insight about the dynamics of the simulated process. When the die closes, a decrease of the sheet temperature is induced thanks to the contact with the die of the forming and quenching phase, while the die temperature increases up due to the heat transfer between the two parts until the cooling of the sheet temperature slows. The opening of the die allows the extraction of the final steel sheet and the die enters into a cooling stage caused by the environment temperature until the next cycle starts.

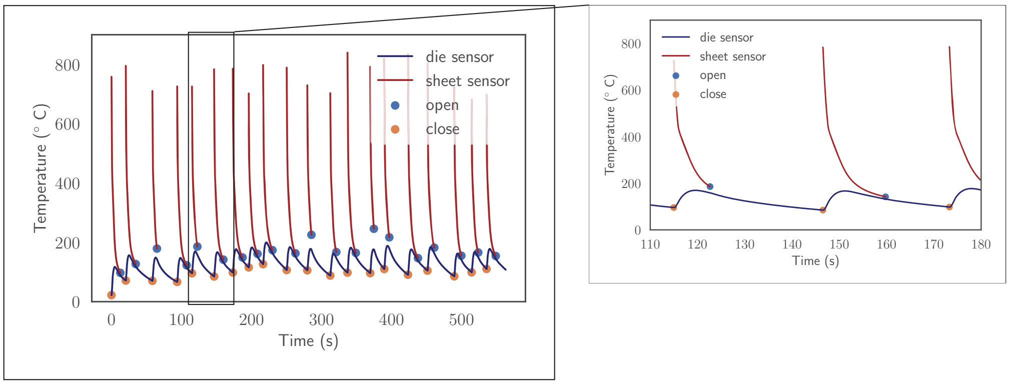


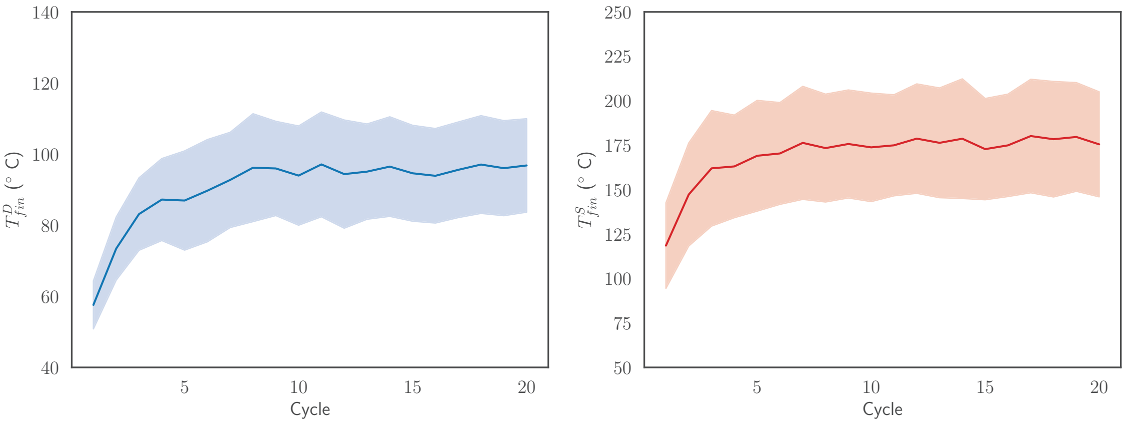
Figure 7.5: Temperature evolution for a simulated batch measured by the two sensors. The moments of opening and closure of the die are pointed out. Left-hand side: Evolution of the sensor outputs for a whole batch of 20 cycles. Right-hand side: Zoom in of the individual cycles of the batch.

The randomness in the input variables is used to avoid inductive biases when building the ML-based SModBase that will try to reproduce the hot stamping process at Eurecat pilot plant. In Section 5.3.2 [2], it has been demonstrated that a

ML-based SMod of the hot stamping process generalizes better when it is trained with a data set that explores the whole parameter space randomly than when it is trained with standard operational conditions.

Under these conditions, we have simulated 500 batches of 20 cycles for the training of the ML-based SMod, which form the low-fidelity SIM$_{data}$ data set. The most relevant aspects of this training data set are outlined in Table 7.2.

The outputs of the simulation data set T^D_{fin} and T^S_{fin} are presented in Figure 7.6, where we can see the their evolution along the cycles. It can be noticed that despite the randomness, both temperatures evolve in a increasing tendency until a stationary region is reached. After cycle 10, the global trend is constant and the temperatures keep oscillating inside a wide range of values. The reason of this behavior is that under the conditions given by the input variables, in that region the output value of T^D_{fin} is not too far from T^D_{ini}. Since T^D_{fin} will be used as input for the next cycle, despite the values of the other input variables, the situation is repeated. Considering that T^S_{fin} is directly related with the input T^D_{ini}, the periodic dynamic of the input variable has the same effect on T^S_{fin}.



(a)

(b)

Figure 7.6: Representation of the mean temperatures and the corresponding standard deviation of SIM$_{data}$. (**a**) T^D_{fin}. (**b**) T^S_{fin}.

Additionally to the training data set, we have also simulated the 40 batches produced in the real environment for comparison and validation purposes which compose an extra low-fidelity data set, SIM$_{valid}$ (see Table 7.2).

7.4 Application of the Surrogate Model-Transfer Learning Pipeline

In this section, we describe the initial SModBase, the details of the DNNs, the training procedure, the used performance metrics and the validation in both environments. Afterwards, the proceeding of applying TL in the SModBase is reported. The results of the validations lead to a discussion about the contributions and advantages of the employed methodology. Furthermore, we realize a study of the effect of the simulation data fidelity level. Finally, we summarize the importance of the modules of the pipeline with an ablation study.

7.4.1 Baseline Surrogate Model

The existence of a simulation environment opens the possibility of testing different scenarios and the identification of potential issues in the process of hot stamping, leading to the optimization of the process in the real industrial plant. Moreover, the simulation environment can give an insight about the impact of the changes in process parameters and variables, as well of the dynamics of the relevant target temperatures during the production of a batch, as seen in Section 7.3.2.

Nonetheless, drawbacks of manufacturing simulations, such as slow response times and high computational costs, can hinder data acquisition and exploitation within the simulation environment. In addition, the exploration of various possible configurations inside the capabilities of the hot stamping industrial plant will suppose a huge amount of simulations and time. To enhance the response of the simulation environment and to allow a faster exploration of the parameter space, a ML-based SMod of hot stamping simulations has been proposed in Section 5.3.2 [2].

The same idea is applied in this study, but in this case the simulation environment serves as a low-cost source of data that is used only to build a baseline SMod: SModBase. The main advantage of training the SMod in a simulation environment is the easiness to produce a huge amount of data of the process without perturbing the real system. Moreover, if the simulation model complexity is reduced in purpose, the cost of generating data is mitigated. Then, due to the model simplifications of the real process, we do not expect a good performance of SModBase in the prediction of the real plant. However, this initial SMod already has some information of the process dynamics and it is the starting point of the proposed methodology. Different to Section 5.3.2 [2], where XGBoost was used to develop the SMod, in this chapter we employ DNNs.

7.4.1.1 Deep Neural Networks Features

Preprocessing: Before the training, the inputs are normalized using Eq. 7.4 to the range (0,1). This is known as Min Max Scaler and it ensures that the inputs contribute equally to the model fitting:

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{7.4}$$

Model details: This SModBase consists in two Deep Neural Networks (DNNs) built with the Python package Tensorflow [279]. Each DNN will perform a regression to predict one of the target variables of the process, T^{D}_{fin} or T^{S}_{fin}. The employed DNNs are composed by an input layer of 4 neurons corresponding to the 4 input variables of Table 7.2, hidden layers to be determined in the hyperparameter tuning process, and an output layer of a single neuron with linear activation function.
Experimental setup and parameters: The training of the SMod is done with the data of SIM$_{data}$ obtained from the simulation environment of the hot stamping pilot plant, detailed in Table 7.2. First, the Keras Tuner API [280] is used to perform a 5-Fold Cross-Validation (CV) [195] combined with Bayesian Optimization (BO) [281] to select the best network architecture and hyperparameters. The networks have been trained using the Adam algorithm [193] implemented with the Python Keras package [282]. The results of the hyperparameter tuning are displayed in Table 7.3. Both DNNs are 3-layer fully-connected neural networks, with different number of neurons, activation functions ("SeLU"[1] [283], "sigmoid" [284] and "tanh" [285]) and batch size. For the sake of clarity, in this case, the batch size refers to the samples used for training in each epoch.

The evaluation of the performance of the model is done in two sets of previously unseen data: SIM$_{valid}$ and REAL$_{data}$. As is it stated in Section 7.3.2, we have generated an additional data set in the simulation environment for validation purposes. This validation set of simulations SIM$_{valid}$ consists in the simulation of the 40 batches produced in REAL$_{data}$ and its characteristics are indicated in Table 7.2.

[1] SeLU refers to Self-normalizing linear units. This unit implicitly controls internal covariate shift improving convergence properties during the learning process while avoiding the necessity of using tricks such as skip connections or batch normalization.

Table 7.3: Results of the hyperparameter tunning process.

Hyperparameters	**Result** T^D_{fin}	**Result** T^S_{fin}
Number of hidden layers	2	2
Neurons 1st hidden layer	32	64
Neurons 2nd hidden layer	64	16
Act. Function 1st hidden layer	SeLU	SeLU
Act. Function 2nd hidden layer	sigmoid	tanh
Learning rate	0.001	0.001
Batch size	8	4
Epochs	200	200

Performance metrics: The metric used in this study is the Mean Absolute Error (MAE) [49] for each cycle:

$$\mathrm{MAE_k} = \frac{\sum_{i=1}^{N} |x_i - \hat{x}_i|}{N} \tag{7.5}$$

where k is the evaluated cycle, x_i are the actual values, $\hat{x}_i$ are the predicted values, and N is the number of samples corresponding to the cycle k. Additionally to the MAE for single predictions, the considered global metric is the mean value of the MAE$_k$:

$$\mathrm{MAE}_{glob} = \frac{\sum_{k=1}^{N_{cycles}} \mathrm{MAE}_k}{N_{cycles}} \tag{7.6}$$

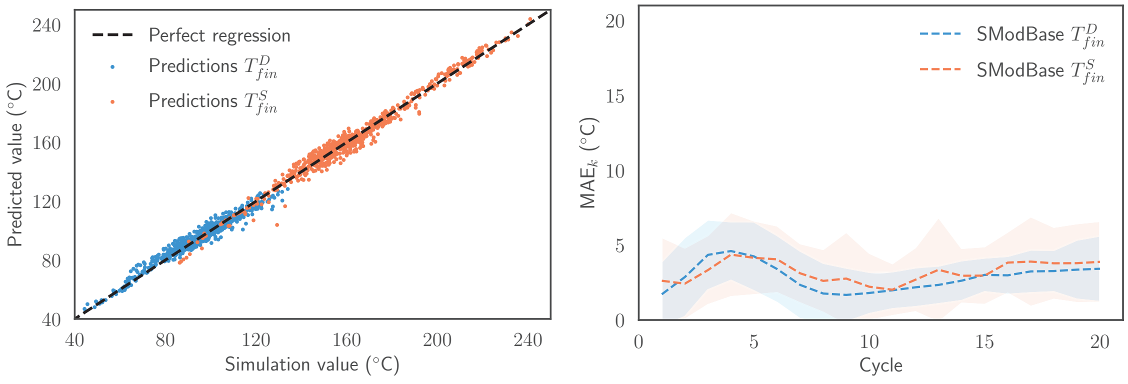
where MAE$_k$ is the MAE of the cycle k and N_{cycles} is the maximum number of cycles per batch, namely 20.

7.4.1.2 Results

In this subsection the results of the SModBase are shown for the simulated data created for validation purposes and for the real data.

Evaluation in simulated validation data.

The results of using the SModBase on the validation simulated data set are exhibited in Figure 7.7. Visually, the model represents the validation data SIM$_{valid}$ successfully. In particular, Figure 7.7a shows how the predictions of both temperatures are close to the perfect prediction line, indicating that the model is able to adequately achieve good prediction performance in the whole range of temperatures of SIM$_{valid}$. Besides, we have calculated the MAE$_k$ for each cycle computing the average absolute error over the samples of the cycle k in the batches. We obtain the results shown in Figure 7.7b. We observe that the MAE_k is always lower than 5 °C.



(a) (b)

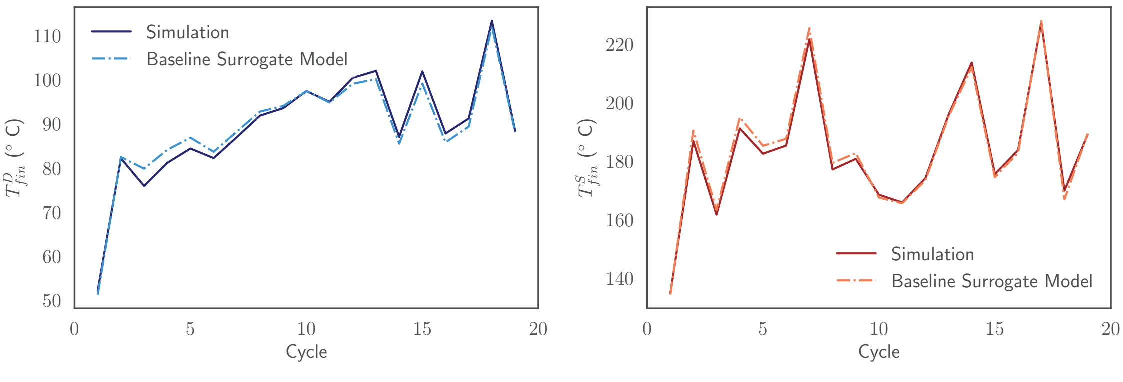
Figure 7.7: Validation of the baseline surrogate model SModBase on the data set SIM_{valid}. (**a**) Comparison between all the predicted values and the corresponding simulated values of SIM_{valid}. (**b**) MAE_k metric evolution with the corresponding standard deviation along cycles of the predictions of the batches of SIM_{valid}.

In this validation, $\mathrm{MAE}_{glob}(T^D_{fin}) = 2.8526$ °C and $\mathrm{MAE}_{glob}(T^D_{fin}) = 3.3085$ °C. It is important to remark that SModBase does predictions using the value of T^D_{fin} predicted in the previous cycle, implying that a discrepancy of the model in the past cycles affects the following predictions of the considered batch.

In Figure 7.8, the evolution of both target temperatures for a batch is closely reproduced by the SModBase, highlighting the good performance of the SModBase in the validation data set SIM_{valid}. This confirms that the usage of a data set that randomly explores the domain of the input variables is a favorable training strategy for the posterior generalization of the model to the standard operational conditions within the limits of the domain. However, the SModBase in this case is formed by two DNNs in comparison to the XGBoost approach presented in the previous study in Section 5.3.2 [2]. The SModBase only acts as the starting point for the generation of a SMod of the real environment.

Evaluation in real data

We will use the SModBase trained with SIM_{data} to try to reproduce the batches of the real environment data set REAL_{data}. Training a representative SMod of the hot stamping pilot plant is difficult since the data originated from the real environment are usually scarce and only covers a limited part of the possible configuration domain. This limitation is evidenced in the number of batches of REAL_{data} and



(a) (b)

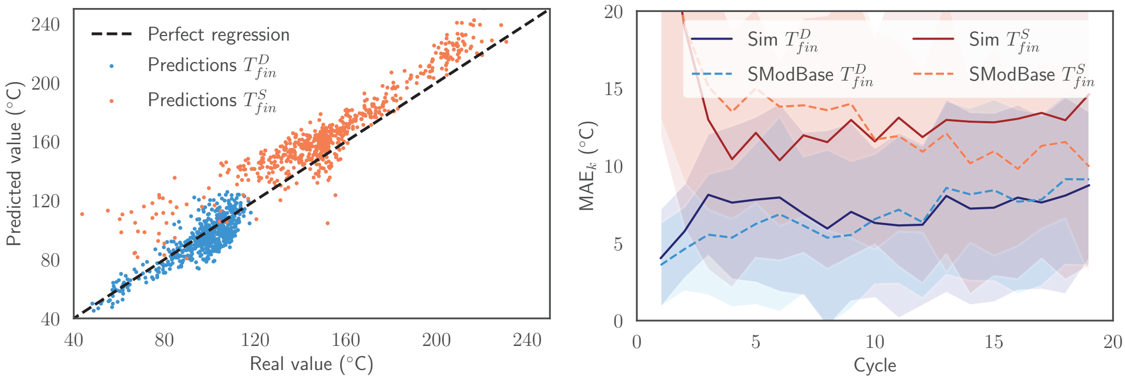
Figure 7.8: Example of the evolution of the target temperatures of a simulated batch compared to the prediction given by the baseline surrogate model SModBase. (**a**) T^D_{ini}. (**b**) T^S_{fin}.

the range of values that they enclose, shown in Figure 7.4. Data acquisition in a real industrial plant environment is more demanding because it supposes a waste of raw material, erosion of the tools, energy consumption, etc. Then, the idea is that the previous training in the simulation environment has allowed to feed the SModBase with a huge amount of simulated data which embrace an extensive set of configurations of the parameters.

The behavior of SModBase in the prediction of REAL$_{data}$ is shown in Figure 7.9. On the left-hand side, Figure 7.9a predictions are spread, indicating a lack of precision of the SModBase. The dispersion is amplified in the intermediate cycles for T^D_{fin} and for low first cycles for T^S_{fin}, as displayed in the right-hand side Figure 7.9b. In the metrics we can observe an important deviation of the model from the real curve along cycles, with MAE$_k$ along cycles greater than 5°C for T^D_{fin} and 10°C in the case of T^S_{fin}. The global metrics are MAE$_{glob}(T^D_{fin}) = 6.5216$ °C and MAE$_{glob}(T^S_{fin}) = 14.0385$ °C.

Discussion of the results using surrogate models

In the light of the former results, we observe that the SModBase has a poor performance in the real environment data obtained from the hot stamping pilot plant. Figure 7.10 presents an example of a batch from the data set REAL$_{data}$ compared to the analogous simulation from SIM$_{valid}$ and the prediction from SModBase. We identify a clear disagreement between the simulations and the SModBase respect to the real temperature curves. Due to the difference between the low-fidelity simulations and the real data, the SModBase is not able to predict the real curve, al-



(a) (b)

Figure 7.9: Validation of the baseline surrogate model SModBase on the data set REAL$_{data}$. (**a**) Comparison between all the predicted values and the corresponding values of REAL$_{data}$. (**b**) MAE$_k$ metric evolution with the corresponding standard deviation of the predictions of SModBase and the simulations of SIM$_{valid}$ respect to the batches of REAL$_{data}$.

though it faithfully represents the simulations. This justifies the difference between the simulations of SIM$_{valid}$ and the real environment data from REAL$_{valid}$ which are built under the same inputs and conditions. For that reason, the SModBase is a very good representation of SIM$_{valid}$, but is not a reliable option to choose for the prediction of hot stamping batches of the pilot plant of REAL$_{data}$. Nonetheless, developing the SModBase from a low-fidelity simulation environment allows boosting the exploration of the parameter space and gaining insights into the principal behaviors occurring in the hot stamping plant. Additionally, SModBase already produces a soft-real time response when trained and it is a good starting point to keep improving the surrogate modeling of the real process.

7.4.2 Transfer Learning to Real Environment

As seen in the previous section, there exists a divergence between the simulation environment data and the real environment data. The values of the relevant temperatures of the process in the experiments performed in the real plant do not coincide with the simulations. Hence, the SModBase has been trained in a different domain than the target domain. Therefore, this original baseline model SModBase is no longer useful to generalize and perform predictions of the real environment.

Then, to obtain a accurate surrogate model of the real pilot plant, the data of REAL$_{data}$ have to be used in the training. However, in most cases, data from real environment are not enough to train a SMod. In this use case, although the 40

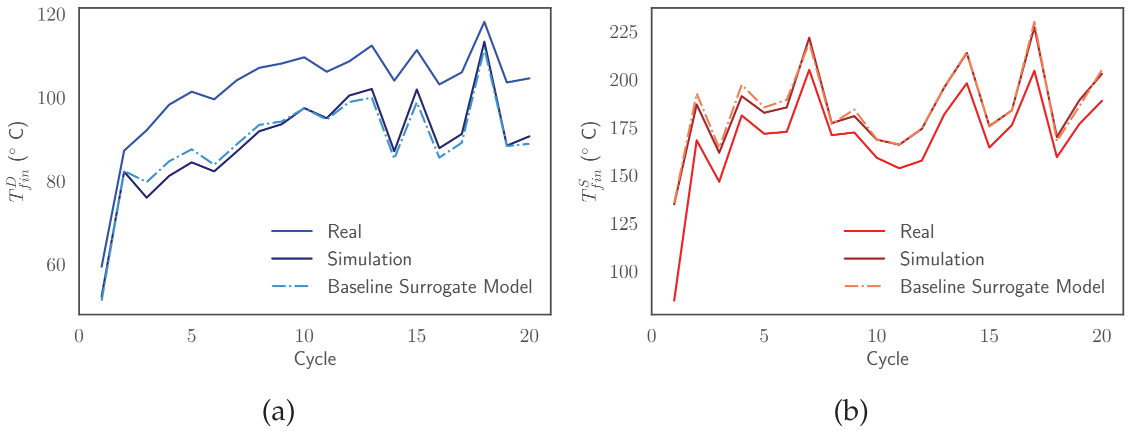


Figure 7.10: Example of the evolution of the target temperatures of a real batch compared to the prediction given by the baseline surrogate model SModBase and the corresponding simulation. (**a**) T_{ini}^{D}. (**b**) T_{fin}^{S}.

batches of REAL$_{data}$ may provide enough information to the SMod, this is not the typical situation. The data sets generated in industrial plants are usually small and some data have low variability, since in production the exploration of the input variables is limited and the processes work under similar configurations. To solve this problem, we propose a methodology to create a representative SMod of the real environment, enhancing the precision of a model trained in the simulation environment.

7.4.2.1 Transfer Learning Surrogate Model

The starting point is the low-fidelity SModBase. In Figure 7.10, we can identify that regardless of the divergence between the SModBase and simulation curves and the real temperature curves, some of the patterns are correctly captured, such as the peaks. This implies that SModBase has learned behaviors present in the real environment from the huge simulation data set SIM$_{data}$.

To overcome these differences between the SModBase and reality, we propose the application of TL to adapt the SModBase to the domain of the real environment, creating a new transfer learning surrogate model (TLSMod). The SModBase is a model specialized in prediction of the hot stamping temperatures in the simulation environment, as demonstrated in Section 7.4.1. When used to the real environment, the task of the model is the same and the domain is different, but still similar to the SModBase training domain. The result is the lost in performance of SModBase in the real environment shown in Figure 7.9. Thus, our TL strategy will be a Network-Based Deep Transfer Learning, which is based on using part of the data from REAL$_{data}$ to fine-tune the SModBase and obtain a new TLSMod.

The **preprocessing**, **model details**, **performance metrics**, and the hyperparameters of the DNNs used for the TL are the same as described in the previous Section 7.4.1. The main reason of not performing hyperparameter tunning in the network fine-tuning is the limited amount of data in REAL$_{data}$. However, different fine-tuning strategies to apply TL are explored.

Experimental setup and training parameters

From the initial two DNNs of SModBase, we will introduce the batches of REAL$_{valid}$ to a new training (or fine-tuning) stage, where the networks will modify the weights of some of the layers, whilst the remaining layers will be frozen during this retraining. With this approach, we conserve the layers of SModBase which contribute to the modelization of the process dynamics of the hot stamping process, learned in the first training in the simulation environment. Additionally, the other layers are able to adapt their weights to learn the particular features of the real environment.

The presented methodology is used on both 3-layer fully connected neural networks that form the SModBase. The training is performed with the same conditions than in the first training of SModBase commented in Section 7.4.1, excepting the batch size, which is 16. Note that the combination of frozen and retrained layers will impact on the resulting final TLSMod. All the possible combinations and their the corresponding models developed from this application of TL in the original DNNs of SModBase will be analysed and they are reported in Table 7.4. The models are named using three digits that encode with a 1 the specific layer that is unfrozen for fine-tuning. Notice that the TLSMod 000 is equivalent to the SModBase, since all layers parameters are frozen.

As discussed, the fine-tuning of SModBase requires data from the real environment. However, the evaluation of the performance of the developed TLSMod also needs data from REAL$_{data}$. While in the previous sections we have in disposition a whole set to validate the surrogate model, in this case the data from training and validation come from the same data set. Hence, we have implemented a CV method to train and test the new TLSMod. The cross-validation consists in the division of the data into folds. Some folds are used for the training and the others for model testing of the models. This is repeated a number of iterations equal to the

Table 7.4: Possible combinations in the transfer learning retraining of the DNNs of SModBase to generate the final TLSMod.

TLSMod:	**Hidden Layer 1**	**Hidden Layer 2**	**Output Layer**
000	Frozen	Frozen	Frozen
100	Retrained	Frozen	Frozen
010	Frozen	Retrained	Frozen
001	Frozen	Frozen	Retrained
110	Retrained	Retrained	Frozen
101	Retrained	Frozen	Retrained
011	Frozen	Retrained	Retrained
111	Retrained	Retrained	Retrained

number of folds changing the training and test folds, to diminish the dependence of the metrics on the chosen folds, before computing an average of the metrics.

We have used a 5-Fold CV applied to REAL$_{data}$ to train and evaluate the TLSMod models. Two different scenarios have been explored:

1. Firstly, 80% of data has been used in training and 20% in testing, with a high percentage of the data of REAL$_{data}$ employed in the the TL fine-tuning.
2. Afterwards, on the contrary, the 20% of data has been used in training and 80% in testing, reducing the number of batches utilized in the TL fine-tuning.

Benchmarks

With the purpose of comparing the effectiveness of the TL approach, two additional surrogate models are considered:

- A SMod have been generated with the architecture of the DNNs of SModBase, without any predefined weight, abbreviated as SModExp. This model is trained from scratch using only data of REAL$_{data}$. As commented before and following the same criteria than when we apply TL, due to the scarcity of data from REAL$_{data}$ it is not possible to do a hyperparameter tunning process. The SModExp will provide a benchmark about how a SMod can perform with the few available data from the real pilot plant, highlighting the need of alternative surrogate modeling strategies such as the application of TL.

- As stated in Section 7.2.2, the current problem of data sources of different fidelity levels is known as VFM. In order to check the viability of the TL method for VFM problems, a co-kriging technique based on the work in [286] has been implemented and compared to TL performance, leading to the SMod VFM.

7.4.2.2 Results and discussion

The results of the two introduced scenarios are shown in Figures 7.11 and 7.12, respectively, where the metrics shown are the results of averaging the MAE$_k$ over the folds of the 5-Fold CV for the target temperatures T^D_{fin} and T^S_{fin}. The different curves are the evaluated models: The SModBase trained only with the simulations of SIM$_{data}$, the SModExp trained only with real data of REAL$_{data}$, the VFM co-kriging SMod and the other TLSMod models that are built using the presented TL methodology. In fact, the SModBase performance in this data set has been already checked in the previous section 7.4.1, and its representation is used to remark the improvement provided by the other methods.

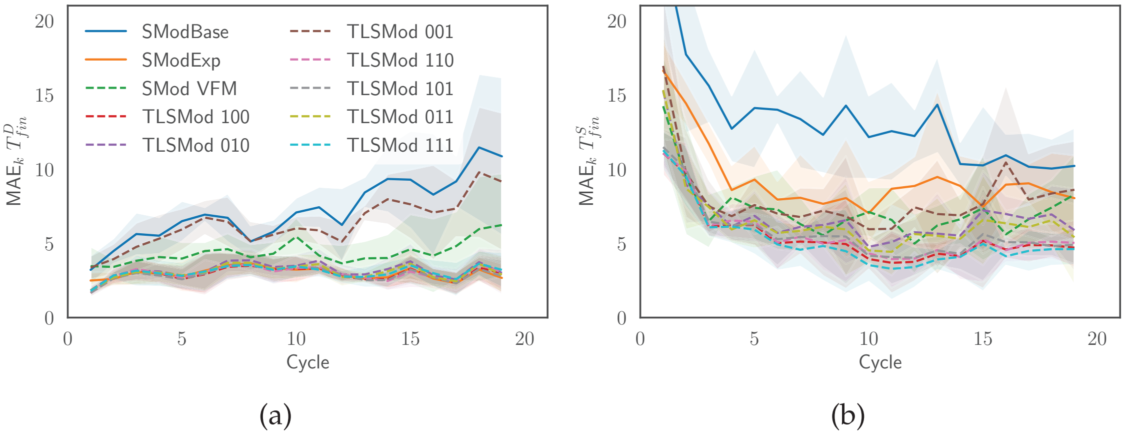
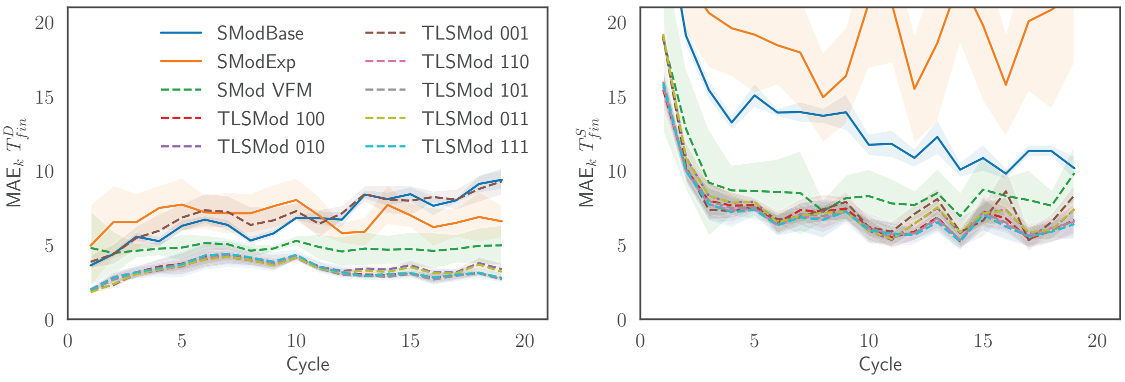


Figure 7.11: MAE$_k$ metric evolution and its corresponding standard deviation along cycles averaged over all the folds in a 5-Fold CV using 80% of the data of REAL$_{data}$ in training and 20% in testing. The different models evaluated are indicated in the legend. (**a**) T^D_{ini}. (**b**) T^S_{fin}.

In the first scenario of Figure 7.11, we observe that the TLSMod models enhance the prediction quality of the SModBase. In spite of this improvement, there in no difference between some TLSMod and SModExp results in Figure 7.11a, while in Figure 7.11b the transfer learning-based models achieve lower metrics than SModExp. Also, most of the TLSMod overcome the SMod VFM, that is only able to



(a)

(b)

Figure 7.12: MAE$_k$ metric evolution and its corresponding standard deviation alon cycles averaged over all the folds in a 5-Fold CV using 20% of the data of REAL$_{data}$ in training and 80% in testing. The different models evaluated are indicated in the legend. (**a**) T^D_{ini}. (**b**) T^S_{fin}.

achieve better metrics than SModExp in Figure 7.12b. The second scenario highlights the success of data fusion methodologies in both temperatures, where both the SMod VFM and most of the TLSMod reach values of MAE much better than SModExp and SModBase. Therefore, we can observe how the size of the training set affects differently the surrogate models. The curves of SModExp suffer an important penalty in the metrics passing from the partition of $80\% - 20\%$ in the training-test to the $20\% - 80\%$ one. Otherwise, both SMod VFM and TLSMod curves are similar in both scenarios, implying that they are less influenced by the reduction of the training size. Nevertheless, the TLSMod method shows better metrics than SMod VFM.

Accordingly, we can justify that the SModExp is a model trained from scratch with batches from REAL$_{data}$ and decreasing the number of training batches induces a more limited learning. Contrarily, the SMod VFM and the TLSMods have been already trained from the simulation data set and even the scarcity of the training data from REAL$_{data}$, they are able to adapt to the real environment. Moreover, the TL method based on the fine-tuning is better than SMod VFM based on co-kriging proving to be an effective tool for problems with data sources of different fidelity. Moreover, to extend the obtained result, we have computed the global metrics of several scenarios of with distinct train-test splits in CV, varying the number of batches of REAL$_{data}$ in the retraining stage of the TLSMods models and in the training of SModExp and VFM SMod. The studied scenarios are described in Table 7.5 as well as the global metrics calculations.

Table 7.5: Description of the different cross-validations realized to evaluate the surrogate models and the corresponding MAE$_{glob}$ results of T^{D}_{fin} and T^{D}_{fin}.

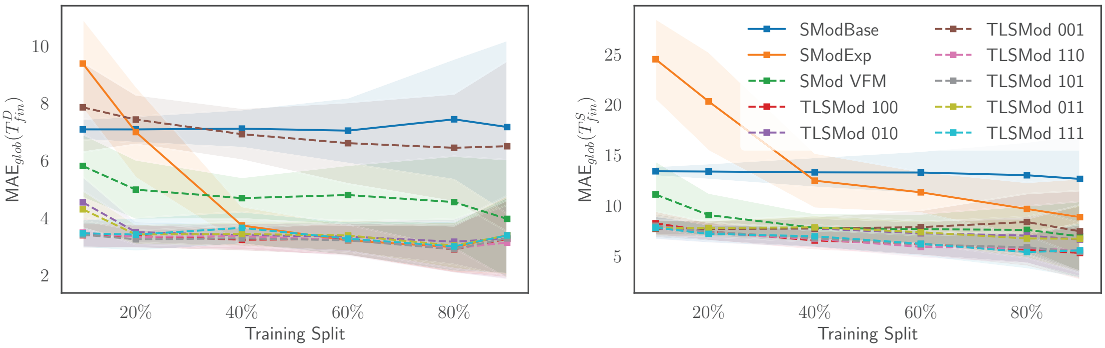
	Train-test split	$90\% - 10\%$	$80\% - 20\%$	$60\% - 40\%$	$40\% - 60\%$	$20\% - 80\%$	$10\% - 90\%$
CV	CV Folds	10	5	5	5	5	10
	Batches per fold	4	8	8	8	8	4
	Training batches	36	32	24	16	8	4
	Test batches	4	8	16	24	32	36
MAE$_{glob}(T^{D}_{fin})$	Simulation	7.3270	7.4303	7.2582	7.3736	7.3697	7.3671
	SModBase	7.1617	7.4303	7.0354	7.1053	7.0780	7.0793
	SModExp	3.3932	2.9753	**3.2122**	3.7309	6.9865	9.3725
	TLSMod 100	3.2643	**2.8976**	3.2585	**3.2345**	3.3612	**3.3913**
	TLSMod 010	3.2778	3.1654	3.3439	3.3920	3.5011	4.5375
	TLSMod 001	6.4932	6.4375	6.5991	6.9129	7.4210	7.8484
	TLSMod 110	**3.1343**	2.9383	3.2230	3.3543	3.3354	3.4807
	TLSMod 101	3.2820	2.9057	3.2148	3.2811	**3.2500**	3.4108
	TLSMod 011	3.3861	3.0296	3.3837	3.4321	3.4183	4.2957
	TLSMod 111	3.3845	3.0024	3.2718	3.6478	3.4145	3.4586
	SMod VFM	3.9598	4.5464	4.7923	4.6843	4.9794	5.8048
MAE$_{glob}(T^{S}_{fin})$	Simulation	12.6159	12.4692	13.3642	13.4238	13.5250	13.5495
	SModBase	12.6369	13.0032	13.2716	13.2877	13.3699	13.3969
	SModExp	8.8591	9.6649	11.3117	12.4649	20.3060	24.4823
	TLSMod 100	**5.3115**	5.6259	6.1936	**6.5476**	7.4420	8.2511
	TLSMod 010	6.6413	7.0307	7.2565	7.7305	7.7661	7.8394
	TLSMod 001	7.4552	8.3713	7.8648	7.7315	7.6860	**7.6919**
	TLSMod 110	5.5665	5.7821	**5.9125**	6.6962	7.2718	7.7769
	TLSMod 101	5.4262	5.8742	6.1414	6.7656	7.4176	8.0002
	TLSMod 011	6.7208	6.7432	7.3625	7.8431	7.7837	7.7780
	TLSMod 111	5.5713	**5.3997**	6.2233	6.9624	**7.2002**	7.8153
	SMod VFM	6.9547	7.5939	7.6607	7.7986	9.0622	11.0996

The metrics of the surrogate models performance on the several train-test splits scenarios considered in Table 7.5 have been calculated using either a 5-Fold CV or a 10-Fold CV, which are the most common types of cross-validation. The number of batches in training and test sets determines if a 5-Fold CV or the 10-Fold CV is implemented. The decision is taken ensuring that all the batches participate in the training or test at least once. For instance, a 10-Fold CV for the $90\% - 10\%$ and $10\% - 90\%$ splits allows that the each distinct fold of 4 batches is present in the testing or training, respectively, of each iteration. For the remaining splits, we have applied a 5-Fold CV. In the case of $80\% - 20\%$ and $20\% - 80\%$ partitions, we have the same situation than before but with folds of 8 batches, and we have reduced the number of folds to avoid the repetition of the testing or training batches in the different iterations. Finally, for the intermediate divisions of $60\% - 40\%$ and $40\% - 60\%$, we also have used a 5-Fold CV because it is enough to guarantee that all the folds are at least once in the test or training set.

The global metrics values show that, in general, all the surrogate models have a lower MAE in the prediction of T_{fin}^{D} than in the case of T_{fin}^{S}. The reason is that in both simulation and real environments, the temperature of the sheet has a wider range of values and it reaches higher temperatures respect to the temperature of the die, as shown in Sections 7.3.1.2 and 7.3.2. Since we address the variables separately it is not necessary to use a relative metric.

The variation of the number of training batches in the fine-tuning stage has relevant consequences in the metrics of the different surrogate models. The results of MAE$_{glob}$ in Table 7.5 have been represented in Figure 7.13 for the sake of clarity, where we can see the dependence of the global metrics of the studied surrogate models on the training split, i.e, on the number of training batches of REAL$_{data}$ used in the fine-tuning. The curves manifest the benefits of using the proposed transfer learning methodology, given that the lowest values of MAE$_{glob}$ are achieved for the some of the TLSMods.

As expected, MAE$_{glob}$ of SModBase do not change with the training split, since we do not apply the retraining stage to this model in cross-validation and we only evaluate its performance in the test split. The stationary values of MAE$_{glob}$ of SModBase are in agreement with the ones found in Figure 7.9b. In opposition, the SModExp faces a significant penalization when the training data is scarce, as we have already noticed in Figures 7.11 and 7.12. The increase of the number of batches in training supposes an improvement of the MAE$_{glob}$. This model is completely dependent on the number of batches of REAL$_{data}$ for its training



(a) (b)

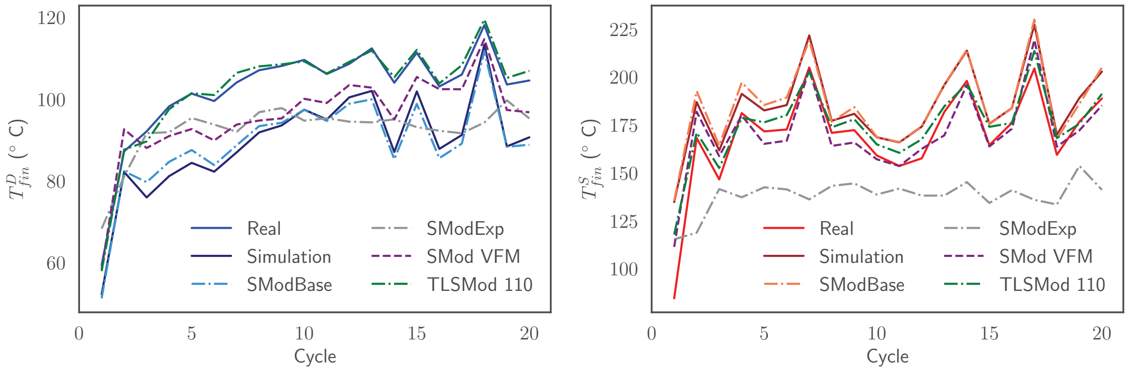
Figure 7.13: MAE$_{glob}$ of all the studied surrogate models as function of the training split, i.e., the number of training batches of REAL$_{data}$ in the fine-tuning. (**a**) T^{D}_{ini}. (**b**) T^{S}_{fin}.

stage, which is translated into this behavior. The SMod VFM does not suffer an important change in the metrics with the training set size. This method overcomes the SModExp performance in $T^{S}_{f}in$ and in the T^{D}_{fin} only when training data is scarce. Remarkably, despite the modification of the training split, the MAE$_{glob}$ of the TLSMods is neither relevantly affected, only suffering a soft decrease with the increment of the number of training batches. In addition, the TLSMods display the lowest values of the global metrics, enhancing the prediction capability of the SModExp trained only with REAL$_{data}$ and other VFM techniques such as co-kriging.

Concretely, in Figure 7.13, we can observe that SModExp has a poor performance for small training splits. This surrogate model is fed only with experimental data of REAL$_{data}$ and it loses predictive power as we decrease the size of its training set. SModExp can accomplish the same MAE$_{glob}$ than the TLSMods as the training set size increases, i.e, it requires a lot of experimental batches to have a decent performance. Also, SMod VFM metrics are the same under different training sizes, showing to be resilient when only few data is available. Although their performance is closer to the TLSMods than SModExp, they are not able to reach the performance of the TLSMods.

In opposition, Figure 7.13 also show that models built with the surrogate model-TL methodology generally have the lowest metrics, with the exception of TLSMod 001 which fails in the prediction of T^{D}_{fin} in Figure 7.13a. Nevertheless, despite some observed differences between the TL strategies, such as TLSMod 010 and TLSMod 011 presenting a higher MAE$_{glob}$ for the 10% training split, the TLSMods are evidenced to be the best option to predict T^D_{fin}. The same result is applicable to the prediction of T^S_{fin}. In this case, Figure 7.13b shows that the SModExp cannot reach the values of MAE$_{glob}$ given by the TLSMods even when the training data increase. The TLSMods have a more noticeable decrease as we increase the number of training batches, with the TLSMod 100, TLSMod 110, TLSMod 101 and TLSMod 111 achieving the lowest global metrics in this situation.

Remarkably, we can say that, in general, the surrogate models generated using the TL methodology outperform the other surrogate models, trained either with simulation or real data, respectively. Further, they have shown to be an effective alternative to the VFM approaches. Based on the obtained global metrics, we have selected the TLSMod 110 as the surrogate model to represent the real environment, because it has low MAE$_{glob}$ and small variation along the train-test split scenarios. The presented results demonstrate the gains that the application of the transfer learning methodology can provide for the generation of a surrogate model of a real environment. Impressively, even when substantially decreasing the size of the real environment training data set REAL$_{valid}$, with the proposed transfer learning-based technique we are able to still develop a reliable surrogate model of the real environment. In Figure 7.14, analogously to the previous sections, we have compared the prediction of a batch of the real environment using the 10% of the data for retraining the surrogate models. This implies that only 4 batches from REAL$_{valid}$ have been used in the fine-tuning process. The TLSMod 110 outperforms the rest of the surrogate models and it captures very well the temperature evolution of T^D_{fin} and T^S_{fin}.



(a) (b)

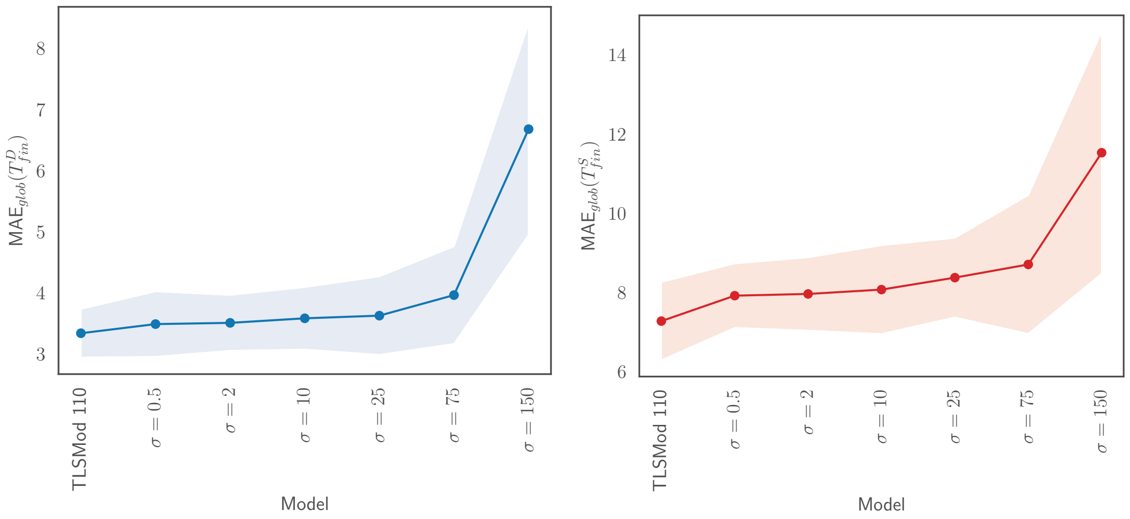
Figure 7.14: Example of the evolution of the target temperatures of a real batch and the corresponding simulation compared to the prediction given by the SModBase, SModExp and TLSMod 110. (**a**) T^D_{ini}. (**b**) T^S_{fin}.

7.4.3 Study of Simulation Data Fidelity Level

The low-cost source of data of the simulation environment defined in Section 7.3.2 boosts the low-fidelity training data generation process to develop the SModBase. Despite we have proven the effectiveness of this strategy, depending on the fidelity level of the low-fidelity simulation data SIM$_{data}$, the performance of the resulting SModBase is affected and, consequently, it can have an impact to the surrogate models developed with the proposed TL methodology. For that reason, we perform an study to analyse the effect of the fidelity level of SIM$_{data}$ on the final TLSMod.

Concretely, the SIM$_{data}$ previously used to create the SModBase has been modified including an additive term of Gaussian White Noise [287] to the simulation results. Hence, we have added the noise term, η, to T^D_{fin} and T^S_{fin} values from the SIM$_{data}$, where η follows the Gaussian distribution with mean $\mu = 0$ and standard deviation, σ, that determines the noise strength, i.e., the variation of the noise signal.

Therefore, we assume that as as we increase the noise level by modifying σ, we reduce the fidelity of the simulation data. In fact, the use of Gaussian White Noise perturbs data introducing instantaneous deviations while keeping the statistical expected behavior of the system.



(a) (b)

Figure 7.15: MAE$_{glob}$ depending on the noise strength (fidelity level) of the SIM$_{data}$, using TLSMod 110 as a benchmark and evaluated in a 5-Fold CV using a $20\% - 80\%$ train-test split. (**a**) T^D_{ini}. (**b**) T^S_{fin}.

To assess the effect of diminishing the fidelity of SIM_{data} in the TL methodology, we have used different values of σ to modify SIM_{data}. Afterwards, for each σ, we have applied the proposed TL pipeline to build a surrogate model of the real plant. Based on the previous section results, the architecture of the employed DNNs for the SModBase and the posterior TL process is equal to the TLSMod 110 and the metrics are computed through a 5-Fold CV using a $20\% - 80\%$ train-test split.

In Figure 7.15, we display how the increase of σ impacts the MAE_{glob} of the surrogate models built with the proposed approach. The benchmark TLSMod 110 exhibit the lowest value of MAE_{glob} and as σ augments the MAE_{glob} values worsen. This means that the fidelity level of the initial source of data to create the SModBase determines the prediction accuracy of the final TLSMods. However, it is important to notice the resilience that TLSMods show under the decrease of fidelity of the SIM_{data}, thanks to the TL pipeline and introducing few real data. Until an elevated noise level, the changes in MAE_{glob} are not very significant because SModBase is still transferring knowledge, learning low-fidelity data patterns and TL handles to adapt the model to real environment with scarce data. Once the noise is too high, despite the TL implementation, the TLSMod for $\sigma = 150$ converges to similar metrics to SModExp in Figure 7.13 for the $20\% - 80\%$ partition.

7.4.4 Summarizing System Ablation Results

The TL-surrogate modeling pipeline utilized in the use case consists in several modules represented in Figure 7.1. To sum up the different steps comprised in the approach and to wrap up the results obtained in previous sections, we present them in the format of ablation study with the objective of highlighting the contribution of each module of the pipeline.

The TLSMod 110 is used to perform the ablation study and the evaluation is done with the 5-Fold CV using a $20\% - 80\%$ train-test split. A schematic representation of the steps and the modules included in the creation of this TLSMod using the proposed methodology is shown in Figure 7.16.

Due to the specific dependencies between the modules, the ablation study consists in the sequential walk across the modules in Figure 7.16 as follows:

1. First, we consider Module 1, removing Module 2A, Module 2B and Module 3. Module 1A and Module 1B correspond to the Simulation Model and the Real Plant, respectively. In this first stage, we have the primary data sources:

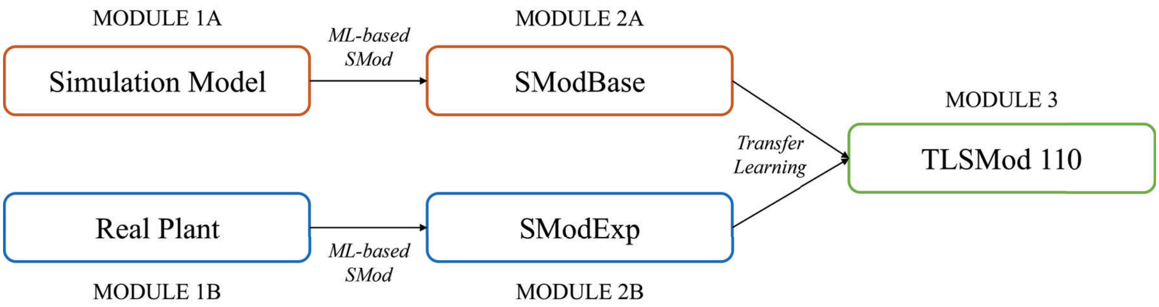


Figure 7.16: Schematic representation of the modules for the process to build the TLSMod 110 with the TL-surrogate model method.

Table 7.6: Summarized results of the ablation study.

Features	**Module 1**		**Module 2**		**Module 3**
	Module 1A: Simulation Model	Module 1B: Real Plant	Module 2A: SModBase	Module 2B: SModExp	
MAE$_{glob}$ (T^D_{fin})	7.3697	-	7.0780	6.9865	3.3354
MAE$_{glob}$ (T^S_{fin})	13.5250	-	13.3699	20.3060	7.2718
Response	Simulation Time	Process Time	$\sim$ Real Time	$\sim$ Real Time	$\sim$ Real Time
Costs	Temporal Computation	Temporal Human Material Energy	Low	Low	Low

- Simulation Model: To evaluate the simulation model, we perform the simulations of in the test split of the 5-Fold CV. The metrics results are displayed in Table 7.5, MAE$_{glob}(T^D_{fin}) = 7.3697$ and MAE$_{glob}(T^S_{fin}) = 13.5250$. Moreover, the evaluation implied running the simulations of the test set, which is costly in terms of time and computational resources, as noted in Section 7.1. The response time of the simulations is far from being real-time.
- Real Plant: In this case, since the real plant is our target environment, the evaluation of the metrics makes no sense. Nevertheless, acquiring the results of the experiments of the test set supposes an expense in time, human resources, raw material, energy, etc, as remarked in Section 7.1.

Also, the response of the manufacturing process is the process time.

2. Next, we focus on Module 2, without taking into account Module 3. In this scenario, we skip the TL fine-tuning phase. Module 2A is the SModBase trained with data from the Simulation Model. Module 2B is the SModExp trained with data from the Real Plant.

    - SModBase: This ML-SMod provides a boost in the response time to soft-real time. However, the training with the Simulation Model leads to a deviation from the target environment, the Real Plant. In fact, the metrics exhibit in Table 7.5 are $\text{MAE}_{glob}(T^D_{fin}) = 7.0780$ and $\text{MAE}_{glob}(T^S_{fin}) = 13.3699$, which are very similar to the ones of the Simulation Model.
    - SModExp: This ML-SMod also offers a soft-real time response. However, though the training data comes from the target environment, the utilization of a small amount harms the metrics and the SMod is not able to generalize. The metrics are $\text{MAE}_{glob}(T^D_{fin}) = 6.9865$ and $\text{MAE}_{glob}(T^S_{fin}) = 20.3060$, as shown in Table 7.5. We have seen in Figure 7.13 how the addition of more training data improves the performance of this SMod, but this supposes the penalization of realizing expensive experiments in the Real Plant.

3. Finally, we include all the pipeline modules when building Module 3. This module is the TLSMod 110 generated using the whole TL-surrogate modeling methodology of the study. The outcome is a soft-real time response SMod that performs very well using only few data from the Real Plant to train. In Table 7.5, we can observe the resulting metrics: $\text{MAE}_{glob}(T^D_{fin}) = 3.3354$ and $\text{MAE}_{glob}(T^S_{fin}) = 7.2718$.

A summarized version of the above results of the ablation study is presented in Table 7.6.

7.5 Lessons Learned

- Given the difficulties of data acquisition in manufacturing environment, this chapter demonstrates the feasibility and effectiveness of using TL to develop accurate SMods for real industrial hot stamping processes with limited experimental data. The study aligns with the broader thesis objective of accelerating data generation and analysis in manufacturing processes by providing a general practical and efficient approach for constructing accurate SMods in real-world industrial settings.

- The proposed method is based on leveraging low-fidelity simulation data and applying TL, addressing the variable-fidelity and sim-to-real gap problems at once, enabling the creation of a SMod that accurately predicts key process variables in a real hot stamping plant. The TL approach offers significant advantages over traditional surrogate modeling methods, particularly in scenarios where acquiring extensive real-world data is expensive or time-consuming.

- The presented investigation of the implementation of a TL-SMod method showcases the importance of careful consideration of TL strategies and parameters, as well as the robustness of the methodology in terms of the amount of real data required for successful implementation.

- The hot stamping process surrogate modeling pipeline has been extended in the current chapter, complementing the previous studies of Section 5.3.2 and Chapter 6. This enhances the upgrade done during this thesis in the surrogate modeling tools applied to hot stamping, which may lead to the efficient exploration of critical parameters and new production scenarios in this process, contributing to process optimization and the development of advanced data-driven tools. In fact the models developed in this chapter have been used in [7], where it has been used to develop RL agents that optimize the batch time and quality in the hot stamping plant presented in 7.3.1.1.

CONCLUSIONS

Chapter 8

Conclusions

This dissertation is the final output of an industrial doctoral thesis. Therefore, its main topics are tackled from a point of view that is in the intersection between academy and industry. The aim is to offer theoretical and practical solutions which can be transferred to the corresponding stakeholders and that arise as fruitful solutions for specific industrial needs as well.

In the actual I4.0 paradigm, the efficient development and deployment of advanced data-driven tools and AI-based solutions is hindered by the limited data availability and high computational costs associated with traditional simulations and experimental testing in manufacturing. the lack of a low-cost and efficient method to generate data suppose a limitation in the development and deployment of optimization methods and other advanced solutions like RL or DTs that rely on process data. Moreover, these tools could directly act on the manufacturing lines, leading to a more efficient, productive, autonomous and sustainable manufacturing.

In this thesis, we aim to address this challenge of enhancing the data generation phase in manufacturing scenarios by providing theoretical methods based on ML to build efficient and reliable SMods under different conditions. For this purpose, we provide both a general framework for the practitioners that want to implement these solutions as a fast-modeling alternative for their manufacturing processes and we delve in the use and combination of different well-known techniques practical with the goal of building SMods in several practical manufacturing use cases, such as plastic injection molding, high pressure die casting and hot stamping.

The main contributions of the thesis from the respective academic and industrial perspectives are highlighted in the two following sections. The last section focus on the potential research directions that could be explored from the outputs and ideas of this work.

8.1 Academic Contributions

We have formalized the surrogate modeling concept in manufacturing, providing a comprehensive definition of SMods in the industrial field, highlighting ML-based SMods as the most relevant trend and identifying the potential advantages of implementing SMods in manufacturing problems. Further, we have presented general frameworks to create baseline ML-based SMods in industrial environments. The main body of this dissertation have consisted in the validation of these theoretical frameworks in practice. The practical applications have highlighted the gains achieved through SMods for each particular manufacturing process. Despite the used methods are based on known ML-related techniques, the novelty resides in their adaptation to the current needs of the considered manufacturing scenarios. The methods and their contributions include:

- Node reduction: A method to effectively reduce the number of sensor nodes required in a geometry to generate accurate predictions, leading to cost savings and simplified system design.
- Mesh upscaling: A method to accurately predict fine mesh simulation results using coarse mesh simulations, significantly reducing computational time and enabling faster analysis and exploration of complex manufacturing processes.
- Parameter interpolation: A method to efficiently predict simulation results across a wide range of parameter values, addressing the challenge of generating data in a fast and sustainable way and enabling the exploration of new operation scenarios.
- Extension and improvement of SMod techniques: Potential improvements that enhance the capabilities of the baseline ML-based SMods are introduced, with the goal to build an improved general procedure for surrogate modeling in manufacturing. The validation of these improvements is done in the hot stamping process. These improvements comprise:
  - Importance of sampling: A study of the influence of different sampling techniques on the accuracy and efficiency of ML-based SMods in the hot stamping process.

  - TL for real plant modeling: A transfer learning-based methodology for constructing accurate and efficient SMods in real industrial hot stamping plants, effectively addressing the sim-to-real gap and focusing on the reduction of the required experimental tests to build the SMods of the real industrial plant.

8.2 Industrial Contributions

We have demonstrated the potential of SMods to accelerate data generation and analysis in manufacturing processes, enabling faster design cycles, more efficient analysis and improved productivity. In fact, the practical application in manufacturing use cases have showcased how SMods can lead to significant cost reductions and efficiency improvements in manufacturing by minimizing the need for extensive physical testing or time-consuming simulations. Moreover, the practical examples of implementing SMods in various manufacturing processes, including plastic injection molding, high pressure die casting and hot stamping, have proved their versatility and applicability across different industrial use cases. Nevertheless, this research extracts valuable knowledge about the considered processes, particularly the hot stamping process, providing insights that can benefit both industry practitioners and researchers. Actually, we expect that the generated knowledge about possible surrogate modeling tools in the hot stamping process may serve for industries and experts in this process.

Additionally, we have provided useful methods that can be deployed in real industrial plants. In this line, we have addressed the sim-to-real gap challenge by developing and validating SMods in real-world industrial settings. Also, some of the thesis results have been used in works centered in the development of RL agents or DTs that effectively monitor and control manufacturing processes [5–8], thanks to the benefits of the presented SMods. Overall, this thesis contributes to the advancement of data-driven manufacturing by providing practical and efficient solutions in the direction of enhancing the performance and sustainability of manufacturing processes.

8.3 Future Research

Building upon the foundation laid by this thesis, we see several promising avenues for future research. In the first place, one can explore the applicability of the developed SMods methods and techniques to other manufacturing processes beyond the use cases considered in this thesis, further validating their generalization and potential impact.

Furthermore, we would like to focus on the development of other algorithms and techniques to be applied in surrogate modeling. The most immediate action will be to implement the AdSam technique that we already presented as ongoing work in the thesis. Additionally, the popularizaiton of generative models may be useful to introduce deviations and stochasticity in the SMods, which mimics the non-controlled parameters effects in real industrial plants.

Moreover, additional mechanisms should be devised with the objective of improving the explainability and interpretability of the models. In this sense, the flow of knowledge can be bidirectional, from experts to data-driven modeling techniques and from the results of data-driven models to experts. The understanding of the phenomena occurring during manufacturing processes is the key to address the objectives of I4.0 and I5.0, leading to a more sustainable, resilient, efficient and human-centered manufacturing that will improve the life standard of the whole society.

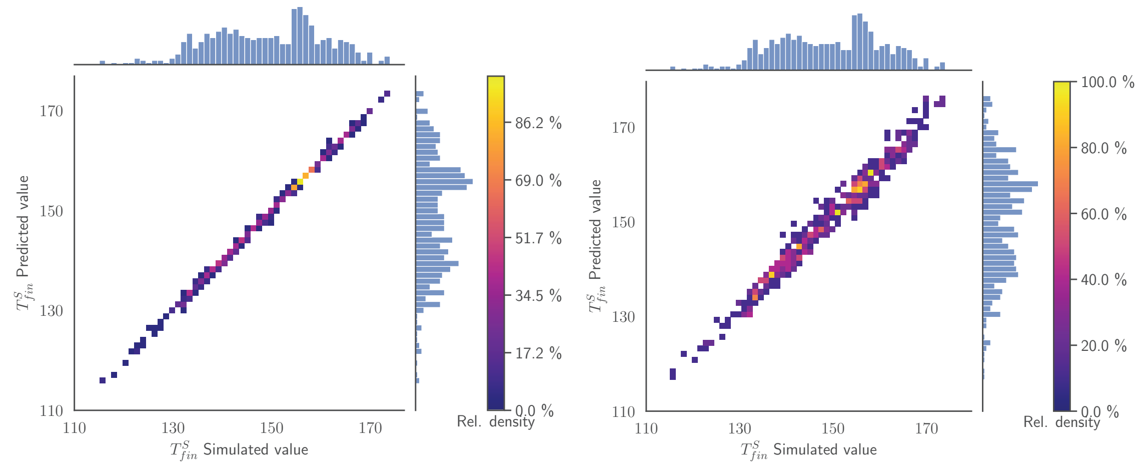
Despite some of the potential in using SMods combined with other advanced AI tools has been already unveiled in this dissertation, further investigation about integration of SMods these tools, such as DTs and RL agents, may enhance their capabilities and enable more sophisticated applications in manufacturing.

Also, an important improvement can be the industrialization of the proposed solutions, with the automatization of the surrogate modeling step in the industrial field. By integrating general surrogate modeling frameworks in simulation softwares or offering surrogate modeling as a SaaS solution can be a very interesting way to simplify the use of the proposed methods in industry. Finally, we think that the emergence of cutting-edge technologies like Large Language Models (LLMs), Retrieval-Augmented Generation (RAG) and Generative Pre-trained Transformer (GPT) models, can help to the generation of friendly-user APIs that include real data managment, simulation softwares and surrogate modeling capabilities in the back-end. Both the industrialization of the SMods and the creation of user-friendly APIs can facilitate the use of these technologies in the factories, with a consequent democratization of the access to this type of AI solutions for manufacturing.

Appendix A

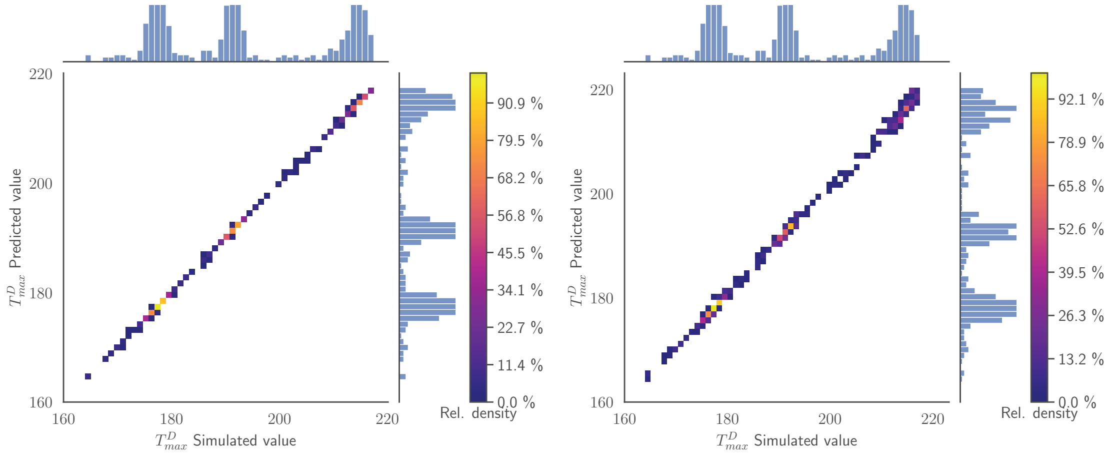
Practical Application of Parameter Interpolation in Hot Stamping: Supplementary Material

In this part, additional figures of the study presented in Section 5.3.2 predictions of the SMod of the target variables of the hot stamping process T^S_{fin} and T^D_{max} are displayed.



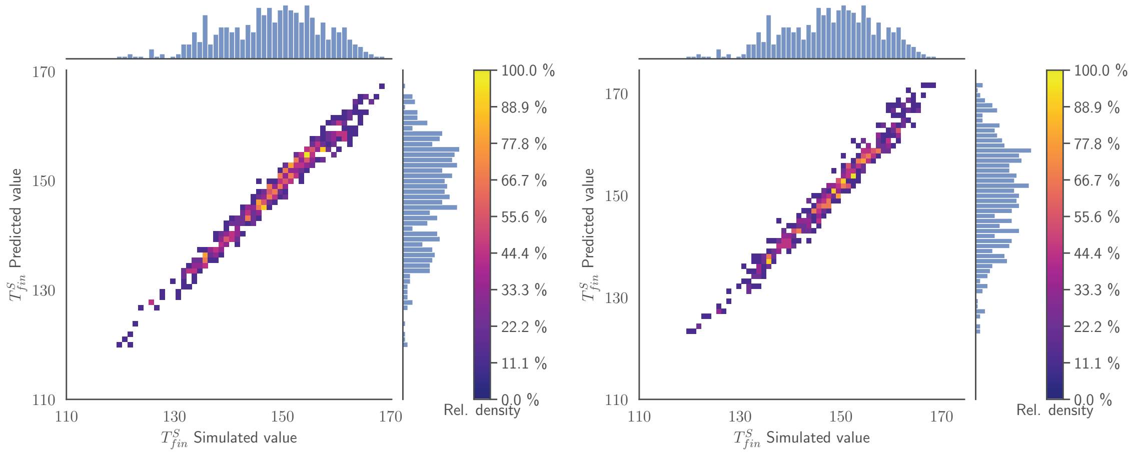
(a) (b)

Figure A.1: Validation scenario 1: Predicted values as function of the simulated output values of the T^S_{fin}. The histograms and the color map represent the relative counts as function of the temperature. (**a**) SModA and (**b**) SModB.



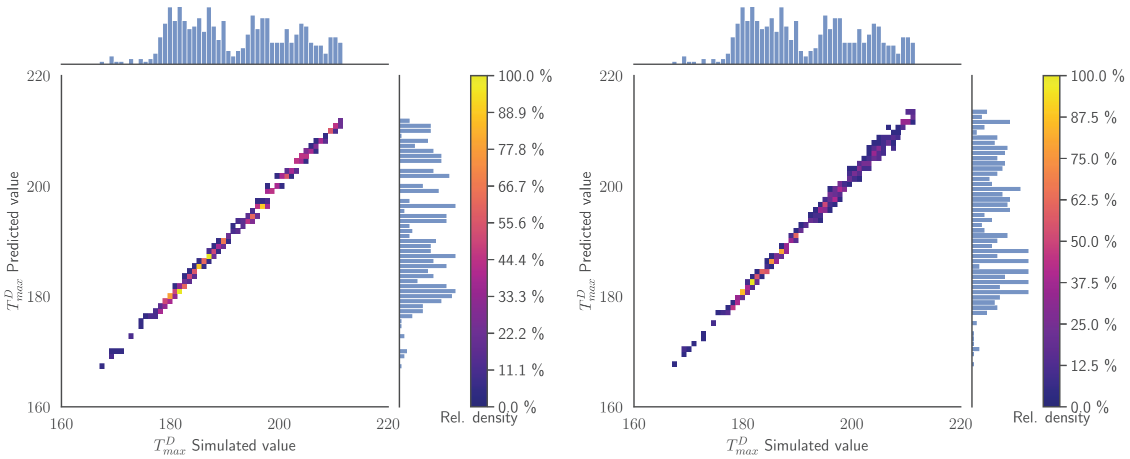
(a) (b)

Figure A.2: Validation scenario 1: Predicted values as function of the simulated output values of the T^D_{max}. The histograms and the color map represent the relative counts as function of the temperature. (**a**) SModA and (**b**) SModB.



(a) (b)

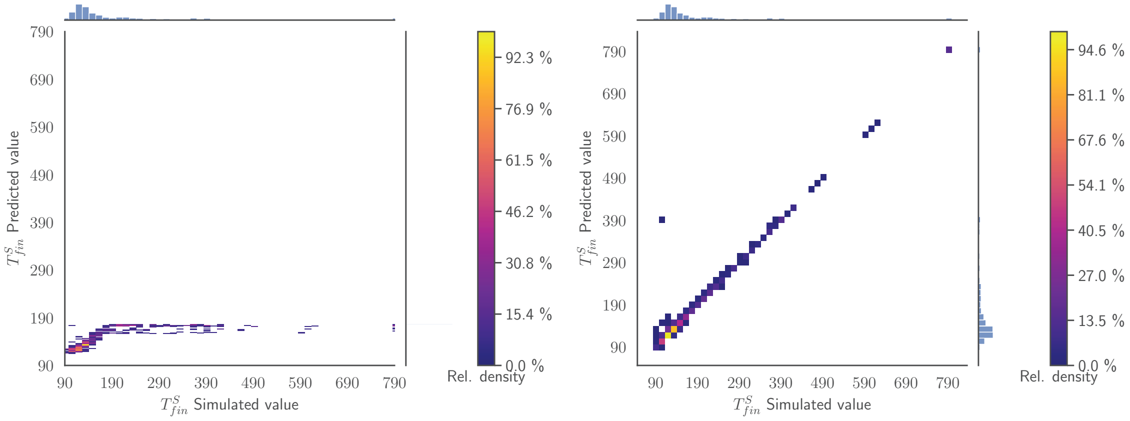
Figure A.3: Validation scenario 2: Predicted values as function of the simulated output values of the T^S_{fin} under. The histograms and the color map represent the relative counts as function of the temperature. (**a**) SModA and (**b**) SModB.



(a)

(b)

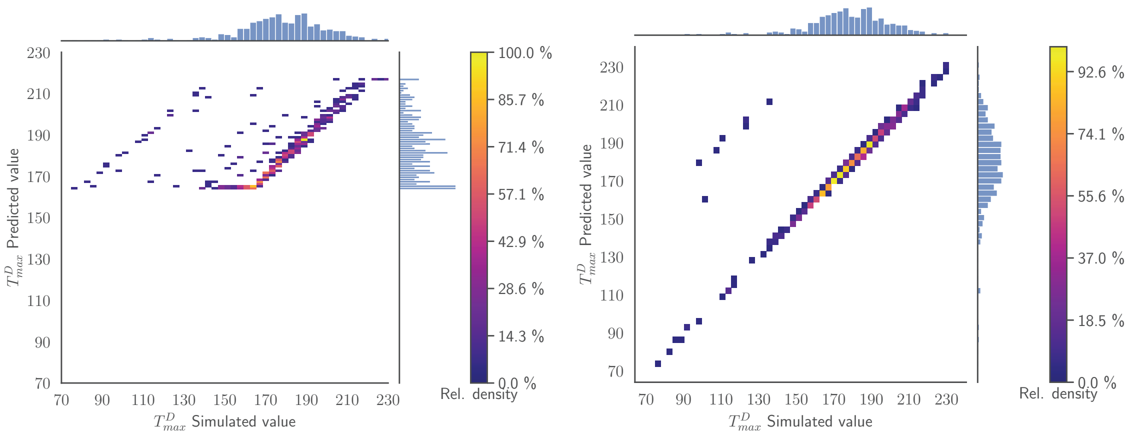
Figure A.4: Validation scenario 2: Predicted values as function of the simulated output values of the T^D_{max} under. The histograms and the color map represent the relative counts as function of the temperature. (**a**) SModA and (**b**) SModB.



(a)

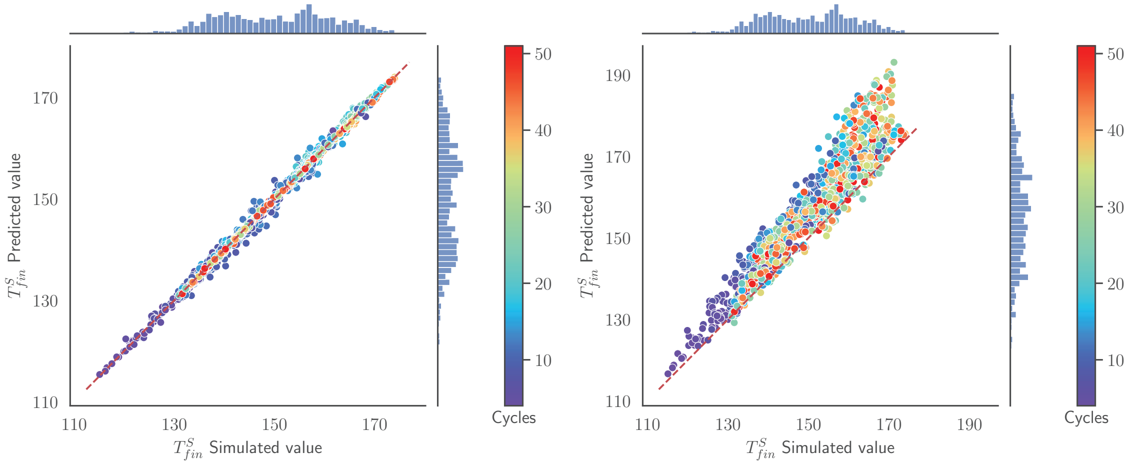
(b)

Figure A.5: Validation scenario 3: Predicted values as function of the simulated output values of the T^S_{fin}. The histograms and the color map represent the relative counts as function of the temperature. (**a**) SModA and (**b**) SModB.



(a) (b)

Figure A.6: Validation scenario 3: Predicted values as function of the simulated output values of the T^D_{max}. The histograms and the color map represent the relative counts as function of the temperature. (**a**) SModA and (**b**) SModB.



(a) (b)

Figure A.7: Validation scenario 4: Predicted values as function of the simulated output values of the T^S_{fin}. The histograms represent the relative counts as function of the temperature and the color map indicates the cycle. (**a**) SModA and (**b**) SModB.

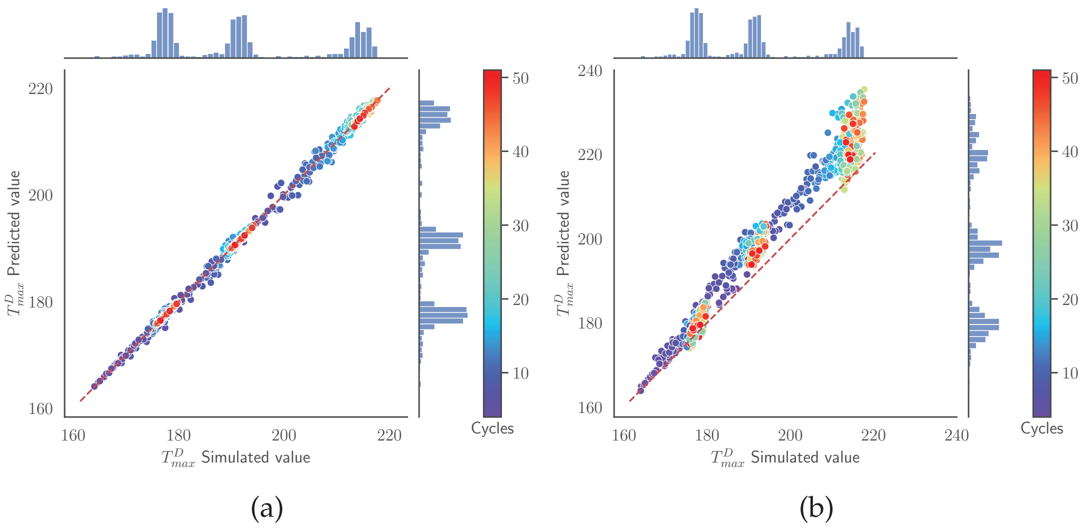


Figure A.8: Validation scenario 4: Predicted values as function of the simulated output values of the T^D_{max}. The histograms represent the relative counts as function of the temperature and the color map indicates the cycle. (**a**) SModA and (**b**) SModB.

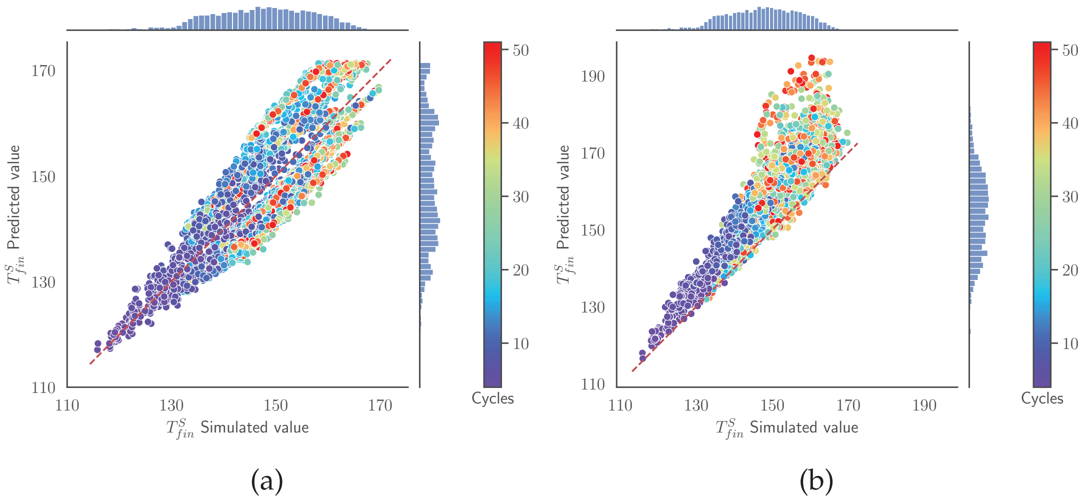
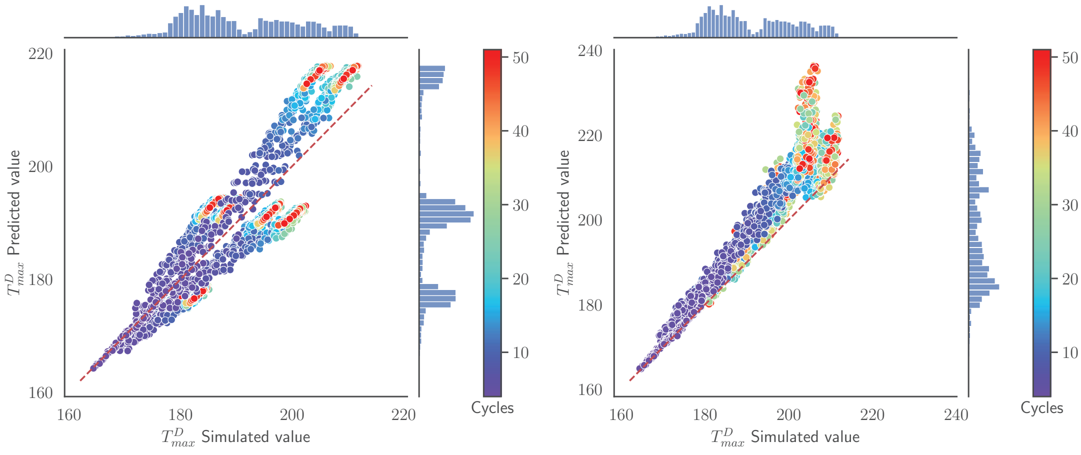
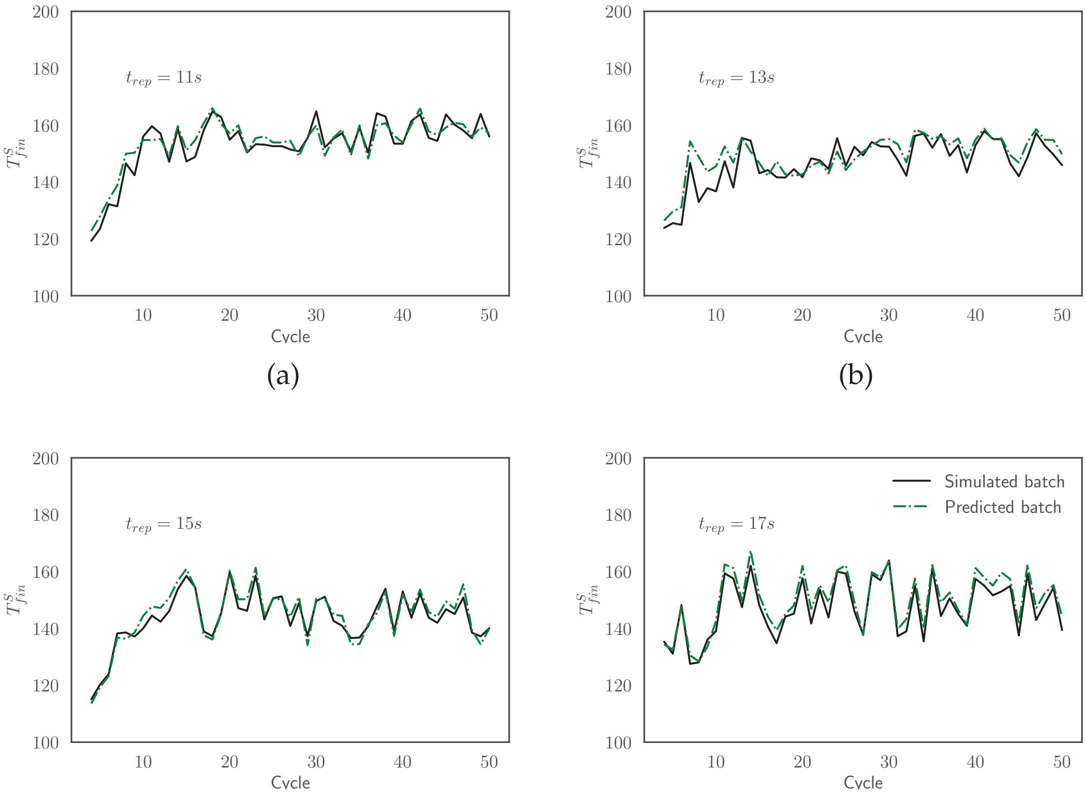


Figure A.9: Validation scenario 5: Predicted values as function of the simulated output values of the T^S_{fin}. The histograms and the color map represent the relative counts as function of the temperature. (**a**) SModA and (**b**) SModB.



(a) (b)

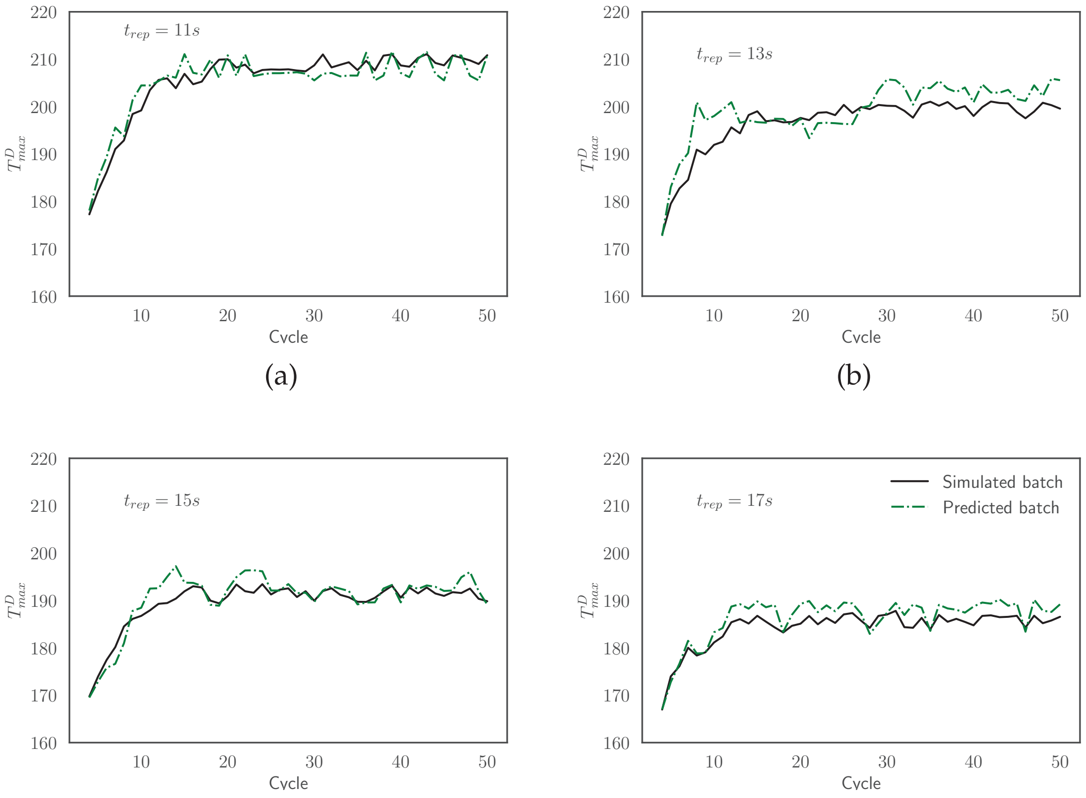
Figure A.10: Validation scenario 5: Predicted values as function of the simulated output values of the T^D_{max}. The histograms and the color map represent the relative counts as function of the temperature. (**a**) SModA and (**b**) SModB.



(a) (b)

(c) (d)

Figure A.11: Comparison between the simulated curves and the predicted curves for the final SMod of T^S_{fin} evaluated for batches with t_{cool} = (**a**) 11 s, (**b**) 13 s, (**c**) 15 s, and (**d**) 17 s.



(a)

(b)

(c)

(d)

Figure A.12: Comparison between the simulated curves and the predicted curves for the final SMod of T^D_{max} evaluated for batches with t_{cool} = (**a**) 11 s, (**b**) 13 s, (**c**) 15 s, and (**d**) 17 s.

Appendix B

Extension of the Hot Stamping Case - Transfer Learning to Real Industrial Plant: Supplementary Material

In this part, complementary explanations about the measurement of S1 data from the study in Chapter 7 is presented.

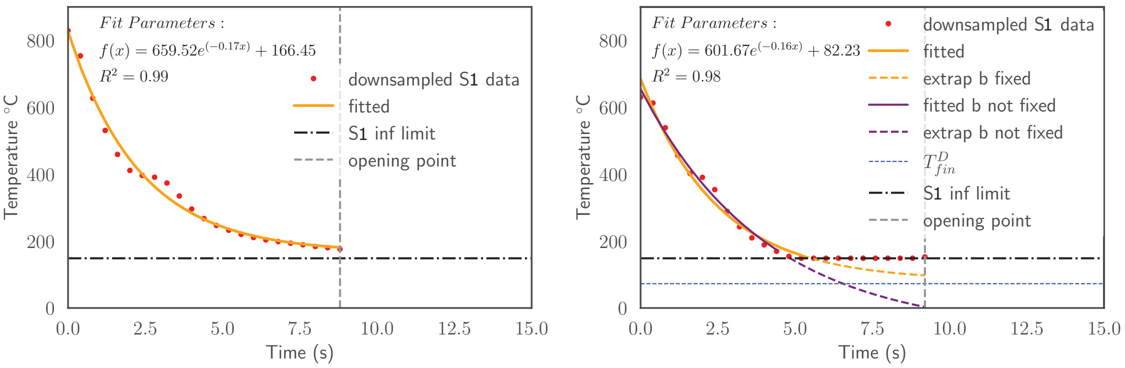
Despite the monitoring of the process with the sensors, some drawbacks have been present in the measurement of T^S_{fin}. Comparing the first cycles and the intermediate cycles in the right-hand side of Figure 7.3, we notice that in the upper figure the opening point S1 has already reached its inferior limit and we cannot obtain the real temperature. On the other hand, the output of S1 in the intermediate cycles is higher than the threshold and we can measure T^S_{fin}. Hence, depending on the process parameters and the input variables, in some cases we will not be able to acquire directly the value of T^S_{fin}.

To overcome this problem, we have assumed that the temperature in the sheet during the forming stage decreases exponentially following an expression of the form

$$T(t) = ae^{-bt} + c \tag{B.1}$$

The idea is to fit this expression adapting the parameters a, b and c to each cycle of the experiments. Then, although the inferior limit is reached, it will be possible to extrapolate the value of T^S_{fin} in opening instant. In fact, there is an accepted good estimation of the behavior of the temperature of the part during the press hardening process [221] based on the Newton's cooling law, which fulfils the analytical expression of Eq. B.2:

$$T(t) = (T_0 - T_\infty)e^{(-h(A/c_p\rho V)t)} + T_\infty \tag{B.2}$$



(a) (b)

Figure B.1: The data of S1 have been downsampled for fitting since the sensor repeated the same measure during 4 acquisitions. (**a**) Procedure to obtain the coefficient b for a given cycle. In this cycle the temperature decrease of S1 during the forming is completely captured, since our target variable T^S_{fin} is higher than the sensor limit indicated in black. Then, we can use all the data to fit the Eq. B.1. The fit is displayed in blue and we get the fit parameters. The value of R^2 is nearly 1, implying that the fit is good and we pick up the value of $b = 0.17$, which will be used in the computation of $\bar{b}$. (**b**) Procedure to obtain the value of T^S_{fin} in the opening point for a cycle where it cannot be acquired due to the temperature threshold. The fit is performed with data higher than the inferior limit of S1. The fit with the b fixed gives us a reasonable result and the fit parameters are indicated. Otherwise, the fit without a fixed value of b leads to an extrapolation without any physical meaning ($T^D_{fin} > T^S_{fin}$).

where A is the contact surface, c_p is the heat capacity, h is the heat transfer coefficient, V is the volume, t is the time, T_0 is the initial temperature, T_∞ is the environment temperature and ρ is the density.

Consequently, we can relate the coefficients of Eq. B.1 with Eq. B.2. It can be noticed that, theoretically, the b coefficient is constant along the cycles, since it only depends on geometrical parameters and physical constants associated to the materials properties. To compute this parameter, we will gather all the cycles where T^S_{fin} is above the inferior limit of S1 and we will fit Eq. B.1 in order to acquire a value for the coefficient b, as it is explained in Figure B.1a. We use only these cycles because the temperature of S1 is not interrupted by the threshold of the sensor and all data can be employed to do the fit. Afterwards, we perform the average over all the studied cycles and a mean reference value of the coefficient b is obtained: $\bar{b} = 0.16$.

The estimation method of T^S_{fin} in the cycles where it cannot be directly measured is shown in Figure B.1b: The available data above the temperature limit of

S1 are used to carry out the fit of Eq. B.1 with the fixed value of $b = \bar{b} = 0.16$. Once the fit parameters are calculated, we can extrapolate the exponential decrease until the instant when the die opens, obtaining the value of T^{S}_{fin} of the current cycle. The previous determination of the specific value $\bar{b} = 0.16$ for all the cycles prevents non-physical scenarios in extrapolation like having $T^{D}_{fin} > T^{S}_{fin}$, as it happens in the case of Figure B.1b when we fit Eq. B.1 without a fixed b parameter.

Bibliography

1. Abio, A., Bonada, F. & Pujol, O. in. Chap. Combining Simulations and Machine Learning for Efficient Prediction of Process Parameters Evolution in Injection Moulding (IOP Press, 2021). doi:10.3233/FAIA210135.

2. Abio, A. *et al.* Machine Learning-Based Surrogate Model for Press Hardening Process of 22MnB5 Sheet Steel Simulation in Industry 4.0. *Materials* **15**.10. doi:10.3390/ma15103647 (2022).

3. Abio, A. *et al. Machine Learning Surrogate Model for Sensitivity Analysis in Hot Stamping* in *CHS^2 2024 Proceedings* (AIST, 2024), 21–27. ISBN: 978-0-930767-24-2. doi:10.33313/512/A0201.

4. Abio, A. *et al.* A transfer learning method in press hardening surrogate modeling: From simulations to real-world. *Journal of Manufacturing Systems* **77,** 320–340. doi:10.1016/j.jmsy.2024.09.012 (2024).

5. Torres, P. *et al.* Millora de l'eficiència i la fiabilitat de sistemes industrials a través de la implementació de Digital Twins. *CIDAI-POC-2023-02.* https://cdn.eurecat.org/PDF/CIDAI/PoC/POC-EURECAT-110523.pdf (2023).

6. Nievas, N. *et al.* A Reinforcement Learning Control in Hot Stamping for Cycle Time Optimization. *Materials* **15**.14. doi:10.3390/ma15144825 (2022).

7. Nievas, N. *et al.* A scalable Deep Q-Learning approach for hot stamping process under dynamic control environment. *International Journal of Production Research* **0**.0, 1–22. doi:10.1080/00207543.2024.2411377 (2024).

8. Nievas, N. *et al.* Offline Reinforcement Learning for Adaptive Control in Manufacturing Processes: A Press Hardening Case Study. *Journal of Computing and Information Science in Engineering* **25.** doi:10.1115/1.4066999 (1 2025).

9. Abio, A., Bonada, F. & Pujol, O. *Graph Neural Network-Based Surrogate Model of Hot Stamping Finite-Element Simulations* Conference Poster session. 2024. https://phantomsfoundation.com/AI4AM/2024/Abstracts/AI4AM2024_Abio_Albert_88.pdf. Conference Artificial Intelligence for Advanced Materials (AI4AM), Barcelona, Spain.

10. Betz, F. *Managing Technological Innovation* ISBN: 9780470547823. doi:10.1002/9780470927564 (John Wiley & Sons, Inc., Hoboken, NJ, 2011).

11. Dosi, G. & Nelson, R. R. in *Handbook of The Economics of Innovation, Vol. 1* (eds Hall, B. H. & Rosenberg, N.) 51–127 (North-Holland, 2010). doi:10.1016/S0169-7218(10)01003-8.

12. Ahmad, M. & Dhafr, N. Establishing and improving manufacturing performance measures. *Robotics and Computer-Integrated Manufacturing* **18**.3. 11th International Conference on Flexible Automation and Intelligent Manufacturing, 171–176. doi:10.1016/S0736-5845(02)00007-8 (2002).

13. Shen, W., Hao, Q., Yoon, H. J. & Norrie, D. H. Applications of agent-based systems in intelligent manufacturing: An updated review. *Advanced Engineering Informatics* **20**.4, 415–431. doi:10.1016/j.aei.2006.05.004 (2006).

14. Wu, D., Rosen, D. W., Wang, L. & Schaefer, D. Cloud-based design and manufacturing: A new paradigm in digital manufacturing and design innovation. *CAD Computer Aided Design* **59.** doi:10.1016/j.cad.2014.07.006 (2015).

15. Machado, C. G. *et al.* Industry 4.0 readiness in manufacturing companies: challenges and enablers towards increased digitalization. *Procedia CIRP* **81.** 52nd CIRP Conference on Manufacturing Systems (CMS), Ljubljana, Slovenia, June 12-14, 2019, 1113–1118. doi:10.1016/j.procir.2019.03.262 (2019).

16. Ghobakhloo, M. Industry 4.0, digitization, and opportunities for sustainability. *Journal of Cleaner Production* **252,** 119869. doi:10.1016/j.jclepro.2019.119869 (2020).

17. Kusiak, A. Smart manufacturing. *International Journal of Production Research* **56**.1-2, 508–517. doi:10.1080/00207543.2017.1351644 (2018).

18. Frank, A. G., Dalenogare, L. S. & Ayala, N. F. Industry 4.0 technologies: Implementation patterns in manufacturing companies. *International Journal of Production Economics* **210,** 15–26. doi:10.1016/j.ijpe.2019.01.004 (2019).

19. Oztemel, E. & Gursev, S. Literature review of Industry 4.0 and related technologies. *Journal of intelligent manufacturing* **31**.1, 127–182. doi:10.1007/s10845-018-1433-8 (2020).

20. Wollschlaeger, M., Sauter, T. & Jasperneite, J. The Future of Industrial Communication: Automation Networks in the Era of the Internet of Things and Industry 4.0. *IEEE Industrial Electronics Magazine* **11**.1, 17–27. doi:10.1109/MIE.2017.2649104 (2017).

21. Borgia, E. The Internet of Things vision: Key features, applications and open issues. *Computer Communications* **54,** 1–31. doi:10.1016/j.comcom.2014.09.008 (2014).

22. Tao, F., Cheng, Y., Xu, L. D., Zhang, L. & Li, B. H. CCIoT-CMfg: Cloud Computing and Internet of Things-Based Cloud Manufacturing Service System. *IEEE Transactions on Industrial Informatics* **10**.2, 1435–1442. doi:10.1109/TII.2014.2306383 (2014).

23. Peres, R. S. *et al.* Industrial Artificial Intelligence in Industry 4.0 - Systematic Review, Challenges and Outlook. *IEEE Access* **8,** 220121–220139. doi:10.1109/ACCESS.2020.3042874 (2020).

24. Zeba, G., Dabić, M., Čičak, M., Daim, T. & Yalcin, H. Technology mining: Artificial intelligence in manufacturing. *Technological Forecasting and Social Change* **171,** 120971. doi:10.1016/j.techfore.2021.120971 (2021).

25. Kusiak, A. Smart manufacturing must embrace big data. *Nature* **544**.7648, 23–25. doi:10.1038/544023a (2017).

26. Mourtzis, D. Simulation in the design and operation of manufacturing systems: state of the art and new trends. *International Journal of Production Research* **58**.7, 1927–1949. doi:10.1080/00207543.2019.1636321 (2020).

27. Goel, R. & Gupta, P. in *A Roadmap to Industry 4.0: Smart Production, Sharp Business and Sustainable Development* (eds Nayyar, A. & Kumar, A.) 157–169 (Springer International Publishing, Cham, 2020). ISBN: 978-3-030-14544-6. doi:10.1007/978-3-030-14544-6_9.

28. Liu, Y., Peng, Y., Wang, B., Yao, S. & Liu, Z. Review on cyber-physical systems. *IEEE/CAA Journal of Automatica Sinica* **4**.1, 27–40. doi:10.1109/JAS.2017.7510349 (2017).

29. Mourtzis, D., Zogopoulos, V. & Xanthi, F. Augmented reality application to support the assembly of highly customized products and to adapt to production re-scheduling. *The International Journal of Advanced Manufacturing Technology* **105**.9, 3899–3910. doi:10.1007/s00170-019-03941-6 (2019).

30. Liagkou, V., Salmas, D. & Stylios, C. Realizing Virtual Reality Learning Environment for Industry 4.0. *Procedia CIRP* **79.** 12th CIRP Conference on Intelligent Computation in Manufacturing Engineering, 18-20 July 2018, Gulf of Naples, Italy, 712–717. doi:10.1016/j.procir.2019.02.025 (2019).

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

32. Büchi, G., Cugno, M. & Castagnoli, R. Smart factory performance and Industry 4.0. *Technological Forecasting and Social Change* **150,** 119790. doi:10.1016/j.techfore.2019.119790 (2020).

33. Vaidya, S., Ambad, P. & Bhosle, S. Industry 4.0 – A Glimpse. *Procedia Manufacturing* **20.** 2nd International Conference on Materials, Manufacturing and Design Engineering (iCMMD2017), 11-12 December 2017, MIT Aurangabad, Maharashtra, India, 233–238. doi:10.1016/j.promfg.2018.02.034 (2018).

34. Xu, X., Lu, Y., Vogel-Heuser, B. & Wang, L. Industry 4.0 and Industry 5.0—Inception, conception and perception. *Journal of Manufacturing Systems* **61,** 530–535. doi:10.1016/j.jmsy.2021.10.006 (2021).

35. Demir, K. A., Döven, G. & Sezen, B. Industry 5.0 and Human-Robot Coworking. *Procedia Computer Science* **158.** 3rd World Conference on Technology, Innovation and Entrepreneurship" Industry 4.0 focused Innovation, Technology, Entrepreneurship and Manufacture June 21-23, 2019, 688–695. doi:10.1016/j.procs.2019.09.104 (2019).

36. Commission, E. *et al. Industry 5.0 – Towards a sustainable, human-centric and resilient European industry* doi:doi/10.2777/308407 (Publications Office of the European Union, 2021).

37. Tao, F., Qi, Q., Liu, A. & Kusiak, A. Data-driven smart manufacturing. *Journal of Manufacturing Systems* **48.** Special Issue on Smart Manufacturing, 157–169. doi:10.1016/j.jmsy.2018.01.006 (2018).

38. Cerquitelli, T. *et al.* Manufacturing as a Data-Driven Practice: Methodologies, Technologies, and Tools. *Proceedings of the IEEE* **109**.4, 399–422. doi:10.1109/JPROC.2021.3056006 (2021).

39. Liu, J., Huang, J., Wang, T., Xing, L. & He, R. A Data-Driven Analysis of Employee Development Based on Working Expertise. *IEEE Transactions on Computational Social Systems* **8**.2, 410–422. doi:10.1109/TCSS.2020.3046726 (2021).

40. McAfee, A., Brynjolfsson, E., Davenport, T. H., Patil, D. & Barton, D. Big data: the management revolution. *Harvard business review* **90**.10, 60–68 (2012).

41. Dilda, V., Mori, L., Noterdaeme, O. & Schmitz, C. Manufacturing: Analytics unleashes productivity and profitability. *Report, McKinsey & Company* (2017).

42. Samuel, A. L. Some Studies in Machine Learning Using the Game of Checkers. *IBM Journal of Research and Development* **3**.3, 210–229. doi:10.1147/rd.33.0210 (1959).

43. Nayyar, A., Gadhavi, L. & Zaman, N. in *Machine Learning and the Internet of Medical Things in Healthcare* (eds Singh, K. K., Elhoseny, M., Singh, A. & Elngar, A. A.) 23–45 (Academic Press, 2021). ISBN: 978-0-12-821229-5. doi:10.1016/B978-0-12-821229-5.00011-2.

44. Bhavsar, P., Safro, I., Bouaynaya, N., Polikar, R. & Dera, D. in *Data Analytics for Intelligent Transportation Systems* (eds Chowdhury, M., Apon, A. & Dey, K.) 283–307 (Elsevier, 2017). ISBN: 978-0-12-809715-1. doi:10.1016/B978-0-12-809715-1.00012-2.

45. Suthaharan, S. in *Machine Learning Models and Algorithms for Big Data Classification: Thinking with Examples for Effective Learning* 183–206 (Springer US, Boston, MA, 2016). ISBN: 978-1-4899-7641-3. doi:`10.1007/978-1-4899-7641-3_8`.

46. Sen, P. C., Hajra, M. & Ghosh, M. *Supervised Classification Algorithms in Machine Learning: A Survey and Review* in *Emerging Technology in Modelling and Graphics* (eds Mandal, J. K. & Bhattacharya, D.) (Springer Singapore, Singapore, 2020), 99–111. ISBN: 978-981-13-7403-6.

47. Hoffmann, F., Bertram, T., Mikut, R., Reischl, M. & Nelles, O. Benchmarking in classification and regression. *WIREs Data Mining and Knowledge Discovery* **9**.5, e1318. doi:`10.1002/widm.1318` (2019).

48. Vujovic, Ž. Đ. Classification Model Evaluation Metrics. *International Journal of Advanced Computer Science and Applications* **12**.6. doi:`10.14569/IJACSA.2021.0120670` (2021).

49. Naser, M. Z. & Alavi, A. H. Error Metrics and Performance Fitness Indicators for Artificial Intelligence and Machine Learning in Engineering and Sciences. *Architecture, Structures and Construction* **3**.4, 499–517. doi:`10.1007/s44150-021-00015-8` (2021).

50. Cortes, C. & Vapnik, V. Support-vector networks. *Machine Learning* **20**.3, 273–297. ISSN: 0885-6125. doi:`10.1007/BF00994018` (1995).

51. Cover, T. & Hart, P. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory* **13**.1, 21–27. doi:`10.1109/TIT.1967.1053964` (1967).

52. Suthaharan, S. in *Machine Learning Models and Algorithms for Big Data Classification: Thinking with Examples for Effective Learning* 237–269 (Springer US, Boston, MA, 2016). ISBN: 978-1-4899-7641-3. doi:`10.1007/978-1-4899-7641-3_10`.

53. Liu, Y., Wang, Y. & Zhang, J. *New Machine Learning Algorithm: Random Forest* in *Information Computing and Applications* (eds Liu, B., Ma, M. & Chang, J.) (Springer Berlin Heidelberg, Berlin, Heidelberg, 2012), 246–252. ISBN: 978-3-642-34062-8. doi:`https://doi.org/10.1007/978-3-642-34062-8_32`.

54. Natekin, A. & Knoll, A. Gradient boosting machines, a tutorial. *Frontiers in Neurorobotics* **7.** doi:`10.3389/fnbot.2013.00021` (2013).

55. Seeger, M. Gaussian Processes for Machine Learning. *International Journal of Neural Systems* **14**.02, 69–106. doi:`10.1142/S0129065704001899` (2004).

56. Gurney, K. *An Introduction to Neural Networks* ISBN: 9781482286991. doi:`10.1201/9781315273570` (CRC Press, 2018).

57. Ramchoun, H., Amine, M., Idrissi, J., Ghanou, Y. & Ettaouil, M. Multilayer Perceptron: Architecture Optimization and Training. *International Journal of Interactive Multimedia and Artificial Intelligence* **4**.1, 26. doi:10.9781/ijimai.2016.415 (2016).

58. O'Shea, K. & Nash, R. An Introduction to Convolutional Neural Networks. *CoRR* **abs/1511.08458.** arXiv: 1511.08458. http://arxiv.org/abs/1511.08458 (2015).

59. Schmidt, R. M. Recurrent Neural Networks (RNNs): A gentle Introduction and Overview. *CoRR* **abs/1912.05911.** arXiv: 1912.05911. http://arxiv.org/abs/1912.05911 (2019).

60. Liu, C., Li, Y., Zhou, G. & Shen, W. A sensor fusion and support vector machine based approach for recognition of complex machining conditions. *Journal of Intelligent Manufacturing* **29,** 1739–1752. doi:10.1007/s10845-016-1209-y (8 2018).

61. Gomes, M. C., Brito, L. C., Bacci da Silva, M. & Viana Duarte, M. A. Tool wear monitoring in micromilling using Support Vector Machine with vibration and sound sensors. *Precision Engineering* **67,** 137–151. doi:10.1016/j.precisioneng.2020.09.025 (2021).

62. Penumuru, D. P., Muthuswamy, S. & Karumbu, P. Identification and classification of materials using machine vision and machine learning in the context of industry 4.0. *Journal of Intelligent Manufacturing* **31,** 1229–1241. doi:10.1007/s10845-019-01508-6 (5 2020).

63. Olatunji, O. O., Akinlabi, S., Madushele, N. & Adedeji, P. A. Property-based biomass feedstock grading using k-Nearest Neighbour technique. *Energy* **190,** 116346. doi:10.1016/j.energy.2019.116346 (2020).

64. Vaghefi, M., Barforoushan, A., Nejabat, G.-R. & Tavallali, M. S. A Machine Learning Approach for Polymer Classification Based on the Thermal Response under Data ScarcityTested on PMMA. *Industrial & Engineering Chemistry Research* **62**.27, 10711–10720. doi:10.1021/acs.iecr.3c00220 (2023).

65. Goyal, D., Choudhary, A., Pabla, B. S. & Dhami, S. S. Support vector machines based non-contact fault diagnosis system for bearings. *Journal of Intelligent Manufacturing* **31,** 1275–1289. doi:10.1007/s10845-019-01511-x (5 2020).

66. Saidi, L., Ben Ali, J. & Fnaiech, F. Application of higher order spectral features and support vector machines for bearing faults classification. *ISA Transactions* **54,** 193–206. doi:10.1016/j.isatra.2014.08.007 (2015).

67. Tian, Y., Fu, M. & Wu, F. Steel plates fault diagnosis on the basis of support vector machines. *Neurocomputing* **151,** 296–303. doi:10.1016/j.neucom.2014.09.036 (2015).

68. Wang, J., Gao, D., Zhu, S., Wang, S. & Liu, H. Fault diagnosis method of photovoltaic array based on support vector machine. *Energy Sources, Part A: Recovery, Utilization, and Environmental Effects* **45**.2, 5380–5395. doi:10.1080/15567036.2019.1671557 (2023).

69. He, Q. P. & Wang, J. Fault Detection Using the k-Nearest Neighbor Rule for Semiconductor Manufacturing Processes. *IEEE Transactions on Semiconductor Manufacturing* **20**.4, 345–354. doi:10.1109/TSM.2007.907607 (2007).

70. Yadav, A. & Swetapadma, A. *Fault analysis in three phase transmission lines using k-nearest neighbor algorithm* in *2014 International Conference on Advances in Electronics Computers and Communications* (2014), 1–5. doi:10.1109/ICAECC.2014.7002474.

71. Madeti, S. R. & Singh, S. Modeling of PV system based on experimental data for fault detection using kNN method. *Solar Energy* **173,** 139–151. doi:10.1016/j.solener.2018.07.038 (2018).

72. Verdier, G. & Ferreira, A. Adaptive Mahalanobis Distance and k -Nearest Neighbor Rule for Fault Detection in Semiconductor Manufacturing. *IEEE Transactions on Semiconductor Manufacturing* **24**.1, 59–68. doi:10.1109/TSM.2010.2065531 (2011).

73. Li, Y., Liu, S. & Shu, L. Wind turbine fault diagnosis based on Gaussian process classifiers applied to operational data. *Renewable Energy* **134,** 357–366. doi:10.1016/j.renene.2018.10.088 (2019).

74. Ke, K.-C. & Huang, M.-S. Quality Prediction for Injection Molding by Using a Multilayer Perceptron Neural Network. *Polymers* **12**.8. doi:10.3390/polym12081812 (2020).

75. Al-kharaz, M., Ananou, B., Ouladsine, M., Combal, M. & Pinaton, J. *Quality Prediction in Semiconductor Manufacturing processes Using Multilayer Perceptron Feedforward Artificial Neural Network* in *2019 8th International Conference on Systems and Control (ICSC)* (2019), 423–428. doi:10.1109/ICSC47195.2019.8950664.

76. Wang, J., Zhuang, J., Duan, L. & Cheng, W. *A multi-scale convolution neural network for featureless fault diagnosis* in *2016 International Symposium on Flexible Automation (ISFA)* (2016), 65–70. doi:10.1109/ISFA.2016.7790137.

77. Yuan, M., Wu, Y. & Lin, L. *Fault diagnosis and remaining useful life estimation of aero engine using LSTM neural network* in *2016 IEEE International Conference on Aircraft Utility Systems (AUS)* (2016), 135–140. doi:10.1109/AUS.2016.7748035.

78. O'Leary, J., Sawlani, K. & Mesbah, A. Deep Learning for Classification of the Chemical Composition of Particle Defects on Semiconductor Wafers. *IEEE Transactions on Semiconductor Manufacturing* **33**.1, 72–85. doi:10.1109/TSM.2019.2963656 (2020).

79. Reddy, A., Indragandhi, V., Ravi, L. & Subramaniyaswamy, V. Detection of Cracks and damage in wind turbine blades using artificial intelligence-based image analytics. *Measurement* **147,** 106823. doi:10.1016/j.measurement.2019.07.051 (2019).

80. Ansari, M. A., Crampton, A., Garrard, R., Cai, B. & Attallah, M. A Convolutional Neural Network (CNN) classification to identify the presence of pores in powder bed fusion images. *The International Journal of Advanced Manufacturing Technology* **120**.7-8, 5133–5150. doi:10.1007/s00170-022-08995-7 (2022).

81. Song, L., Huang, W., Han, X. & Mazumder, J. Real-Time Composition Monitoring Using Support Vector Regression of Laser-Induced Plasma for Laser Additive Manufacturing. *IEEE Transactions on Industrial Electronics* **64**.1, 633–642. doi:10.1109/TIE.2016.2608318 (2017).

82. Golkarnarenji, G. *et al.* Support vector regression modelling and optimization of energy consumption in carbon fiber production line. *Computers & Chemical Engineering* **109,** 276–288. doi:10.1016/j.compchemeng.2017.11.020 (2018).

83. Patel, A. K., Chatterjee, S. & Gorai, A. K. Development of a machine vision system using the support vector machine regression (SVR) algorithm for the online prediction of iron ore grades. *Earth Science Informatics* **12,** 197–210. doi:10.1007/s12145-018-0370-6 (2 2019).

84. Takalo-Mattila, J., Heiskanen, M., Kyllönen, V., Määttä, L. & Bogdanoff, A. Explainable Steel Quality Prediction System Based on Gradient Boosting Decision Trees. *IEEE Access* **10,** 68099–68110. doi:10.1109/ACCESS.2022.3185607 (2022).

85. Lundberg, S. M. & Lee, S.-I. *A unified approach to interpreting model predictions* in *Proceedings of the 31st International Conference on Neural Information Processing Systems* (Curran Associates Inc., Long Beach, California, USA, 2017), 4768–4777. ISBN: 9781510860964.

86. Echeverria-Rios, D. & Green, P. L. Predicting product quality in continuous manufacturing processes using a scalable robust Gaussian Process approach. *Engineering Applications of Artificial Intelligence* **127,** 107233. doi:10.1016/j.engappai.2023.107233 (2024).

87. Anglani, A. & Pacella, M. Binary Gaussian Process classification of quality in the production of aluminum alloys foams with regular open cells. *Procedia CIRP* **99.** 14th CIRP Conference on Intelligent Computation in Manufacturing Engineering, 15-17 July 2020, 307–312. doi:10.1016/j.procir.2021.03.046 (2021).

88. Khelif, R. *et al.* Direct Remaining Useful Life Estimation Based on Support Vector Regression. *IEEE Transactions on Industrial Electronics* **64**.3, 2276–2285. doi:10.1109/TIE.2016.2623260 (2017).

89. Sun, C., Zhang, Z. & He, Z. Research on bearing life prediction based on support vector machine and its application. *Journal of Physics: Conference Series* **305**.1, 012028. doi:10.1088/1742-6596/305/1/012028 (2011).

90. Zhang, S., Zhao, C., Wang, S. & Wang, F. Pseudo Time-Slice Construction Using a Variable Moving Window k Nearest Neighbor Rule for Sequential Uneven Phase Division and Batch Process Monitoring. *Industrial & Engineering Chemistry Research* **56**.3, 728–740. doi:10.1021/acs.iecr.6b03743 (2017).

91. Scrimieri, D. & Ratchev, S. M. A k-Nearest Neighbour Technique for Experience-Based Adaptation of Assembly Stations. *Journal of Control, Automation and Electrical Systems* **25**.6, 679–688. doi:10.1007/s40313-014-0142-6 (2014).

92. Diez-Olivan, A., Pagan, J. A., Khoa, N. L. D., Sanz, R. & Sierra, B. Kernel-based support vector machines for automated health status assessment in monitoring sensor data. *The International Journal of Advanced Manufacturing Technology* **95**.1-4, 327–340. doi:10.1007/s00170-017-1204-2 (2018).

93. Chen, T. & Guestrin, C. *XGBoost: A Scalable Tree Boosting System* in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (ACM, San Francisco, California, USA, 2016), 785–794. ISBN: 978-1-4503-4232-2. doi:10.1145/2939672.2939785. http://doi.acm.org/10.1145/2939672.2939785.

94. Nasiri, H., Dadashi, A. & Azadi, M. Machine learning for fatigue lifetime predictions in 3D-printed polylactic acid biomaterials based on interpretable extreme gradient boosting model. *Materials Today Communications* **39,** 109054. doi:10.1016/j.mtcomm.2024.109054 (2024).

95. Scalabrini Sampaio, G., Vallim Filho, A. R. d. A., Santos da Silva, L. & Augusto da Silva, L. Prediction of Motor Failure Time Using An Artificial Neural Network. *Sensors* **19**.19. doi:10.3390/s19194342 (2019).

96. Orrù, P. F. *et al.* Machine Learning Approach Using MLP and SVM Algorithms for the Fault Prediction of a Centrifugal Pump in the Oil and Gas Industry. *Sustainability* **12**.11. doi:10.3390/su12114776 (2020).

97. Amir Falamarzi Sara Moridpour, M. N. & Cheraghi, S. Prediction of tram track gauge deviation using artificial neural network and support vector regression. *Australian Journal of Civil Engineering* **17**.1, 63–71. doi:10.1080/14488353.2019.1616357 (2019).

98. Zhao, R., Wang, J., Yan, R. & Mao, K. *Machine health monitoring with LSTM networks* in *2016 10th International Conference on Sensing Technology (ICST)* (2016), 1–6. doi:10.1109/ICSensT.2016.7796266.

99. Aghazadeh, F., Tahan, A. S. & Thomas, M. *Tool condition monitoring method in milling process using wavelet transform and long short-term memory* in *Surveillance, Vishno and AVE conferences* (Lyon, France, 2019). https://hal.science/hal-02188764.

100. Feng, K. *et al.* A novel vibration-based prognostic scheme for gear health management in surface wear progression of the intelligent manufacturing system. *Wear* **522.** 24th International Conference on Wear of Materials, 204697. doi:10.1016/j.wear.2023.204697 (2023).

101. Nanduri, A. & Sherry, L. *Anomaly detection in aircraft data using Recurrent Neural Networks (RNN)* in *2016 Integrated Communications Navigation and Surveillance (ICNS)* (2016), 5C2-1-5C2–8. doi:10.1109/ICNSURV.2016.7486356.

102. Vashisht, R. K. & Peng, Q. Online Chatter Detection for Milling Operations Using LSTM Neural Networks Assisted by Motor Current Signals of Ball Screw Drives. *Journal of Manufacturing Science and Engineering* **143.** doi:10.1115/1.4048001 (1 2021).

103. Kizito, R. *et al. The application of random forest to predictive maintenance* in *IIE annual conference. Proceedings* (2018), 354–359.

104. Dos Santos, T., Ferreira, F. J. T. E., Pires, J. M. & Damásio, C. *Stator winding short-circuit fault diagnosis in induction motors using random forest* in *2017 IEEE International Electric Machines and Drives Conference (IEMDC)* (2017), 1–8. doi:10.1109/IEMDC.2017.8002350.

105. Abdi, H. & Williams, L. J. Principal component analysis. *WIREs Computational Statistics* **2**.4, 433–459. doi:10.1002/wics.101 (2010).

106. Zermane, H. & Drardja, A. Development of an efficient cement production monitoring system based on the improved random forest algorithm. *The International Journal of Advanced Manufacturing Technology* **120**.3-4, 1853–1866. doi:10.1007/s00170-022-08884-z (2022).

107. Alshraideh, H., Castillo, E. D. & Gil Del Val, A. Process control via random forest classification of profile signals: An application to a tapping process. *Journal of Manufacturing Processes* **58,** 736–748. doi:10.1016/j.jmapro.2020.08.043 (2020).

108. Zhao, R., Yan, R., Wang, J. & Mao, K. Learning to Monitor Machine Health with Convolutional Bi-Directional LSTM Networks. *Sensors* **17**.2. doi:10.3390/s17020273 (2017).

109. Niu, J., Liu, C., Zhang, L. & Liao, Y. *Remaining Useful Life Prediction of Machining Tools by 1D-CNN LSTM Network* in *2019 IEEE Symposium Series on Computational Intelligence (SSCI)* (2019), 1056–1063. doi:10.1109/SSCI44817.2019.9002993.

110. Ronowicz, J., Thommes, M., Kleinebudde, P. & Krysiński, J. A data mining approach to optimize pellets manufacturing process based on a decision tree algorithm. *European Journal of Pharmaceutical Sciences* **73,** 44–48. doi:10.1016/j.ejps.2015.03.013 (2015).

111. Prihatno, A. T., Nurcahyanto, H. & Jang, Y. M. *Predictive Maintenance of Relative Humidity Using Random Forest Method* in *2021 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)* (2021), 497–499. doi:10.1109/ICAIIC51459.2021.9415213.

112. Diaz, P., Salas, J. C., Cipriano, A. & Núñez, F. Random forest model predictive control for paste thickening. *Minerals Engineering* **163,** 106760. doi:10.1016/j.mineng.2020.106760 (2021).

113. Zhu, M., Yang, Y., Feng, X., Du, Z. & Yang, J. Robust modeling method for thermal error of CNC machine tools based on random forest algorithm. *Journal of Intelligent Manufacturing* **34**.4, 2013–2026. doi:10.1007/s10845-021-01894-w (2023).

114. Wu, D., Wei, Y. & Terpenny, J. *Surface Roughness Prediction in Additive Manufacturing Using Machine Learning* in *Proceedings 13th International Manufacturing Science and Engineering Conference* (American Society of Mechanical Engineers, 2018). ISBN: 978-0-7918-5137-1. doi:10.1115/MSEC2018-6501.

115. Qian, N. *et al.* Predicting heat transfer of oscillating heat pipes for machining processes based on extreme gradient boosting algorithm. *Applied Thermal Engineering* **164,** 114521. doi:10.1016/j.applthermaleng.2019.114521 (2020).

116. Heitz, T. *et al.* Investigation on eXtreme Gradient Boosting for cutting force prediction in milling. *Journal of Intelligent Manufacturing.* doi:10.1007/s10845-023-02243-9 (2023).

117. Tapia, G., Elwany, A. & Sang, H. Prediction of porosity in metal-based additive manufacturing using spatial Gaussian process models. *Additive Manufacturing* **12.** Special Issue on Modeling & Simulation for Additive Manufacturing, 282–290. doi:10.1016/j.addma.2016.05.009 (2016).

118. Lee, S. H. Optimization of Cold Metal Transfer-Based Wire Arc Additive Manufacturing Processes Using Gaussian Process Regression. *Metals* **10**.4. doi:10.3390/met10040461 (2020).

119. Ogunsanyaa, M. & Desai, S. Predictive Modeling of Additive Manufacturing Process using Deep Learning Algorithm. *Proceedings of the IISE Annual Conference & Expo 2022* (ed K. Ellis, W. F.) https://par.nsf.gov/biblio/10335844 (2022).

120. Serin, G., Gudelek, M. U., Ozbayoglu, A. M. & Unver, H. O. *Estimation of parameters for the free-form machining with deep neural network* in *2017 IEEE International Conference on Big Data (Big Data)* (2017), 2102–2111. doi:10.1109/BigData.2017.8258158.

121. Shahin, M., Chen, F. F., Bouzary, H., Hosseinzadeh, A. & Rashidifar, R. A novel fully convolutional neural network approach for detection and classification of attacks on industrial IoT devices in smart manufacturing systems. *The International Journal of Advanced Manufacturing Technology* **123,** 2017–2029. doi:10.1007/s00170-022-10259-3 (5-6 2022).

122. Nalajam, P. K. & Varadarajan, R. A Hybrid Deep Learning Model for Layer-Wise Melt Pool Temperature Forecasting in Wire-Arc Additive Manufacturing Process. *IEEE Access* **9,** 100652–100664. doi:10.1109/ACCESS.2021.3097177 (2021).

123. *Unsupervised Learning Algorithms* (eds Celebi, M. E. & Aydin, K.) ISBN: 978-3-319-24209-5. doi:10.1007/978-3-319-24211-8 (Springer International Publishing, 2016).

124. Amer, M., Goldstein, M. & Abdennadher, S. *Enhancing one-class support vector machines for unsupervised anomaly detection* in *Proceedings of the ACM SIGKDD Workshop on Outlier Detection and Description* (Association for Computing Machinery, Chicago, Illinois, 2013), 8–15. ISBN: 9781450323352. doi:10.1145/2500853.2500857.

125. Ma, H., Hu, Y. & Shi, H. Fault Detection and Identification Based on the Neighborhood Standardized Local Outlier Factor Method. *Industrial & Engineering Chemistry Research* **52**.6, 2389–2402. doi:10.1021/ie302042c (2013).

126. Susto, G. A., Beghi, A. & McLoone, S. *Anomaly detection through on-line isolation Forest: An application to plasma etching* in *2017 28th Annual SEMI Advanced Semiconductor Manufacturing Conference (ASMC)* (2017), 89–94. doi:10.1109/ASMC.2017.7969205.

127. Bock, H.-H. in *Selected Contributions in Data Analysis and Classification* (eds Brito, P., Cucumel, G., Bertrand, P. & de Carvalho, F.) 161–172 (Springer Berlin Heidelberg, Berlin, Heidelberg, 2007). ISBN: 978-3-540-73560-1. doi:10.1007/978-3-540-73560-1_15.

128. Khan, K., Rehman, S. U., Aziz, K., Fong, S. & Sarasvady, S. *DBSCAN: Past, present and future* in *The Fifth International Conference on the Applications of Digital Information and Web Technologies (ICADIWT 2014)* (2014), 232–238. doi:10.1109/ICADIWT.2014.6814687.

129. Suman, S. & Das, A. Monitoring strategy for a multistage manufacturing facility using K-means clustering technique. *International Journal of Industrial and Systems Engineering* **36**.3, 301. doi:10.1504/IJISE.2020.110933 (2020).

130. Jin, C. H., Na, H. J., Piao, M., Pok, G. & Ryu, K. H. A Novel DBSCAN-Based Defect Pattern Detection and Classification Framework for Wafer Bin Map. *IEEE Transactions on Semiconductor Manufacturing* **32**.3, 286–292. doi:10.1109/TSM.2019.2916835 (2019).

131. Shi, Y., Zhang, Y. & Harik, R. Manufacturing feature recognition with a 2D convolutional neural network. *CIRP Journal of Manufacturing Science and Technology* **30,** 36–57. doi:10.1016/j.cirpj.2020.04.001 (2020).

132. Wang, Y., Yao, H. & Zhao, S. Auto-encoder based dimensionality reduction. *Neurocomputing* **184.** RoLoD: Robust Local Descriptors for Computer Vision 2014, 232–242. doi:10.1016/j.neucom.2015.08.104 (2016).

133. Lu, H., Du, M., Qian, K., He, X. & Wang, K. GAN-Based Data Augmentation Strategy for Sensor Anomaly Detection in Industrial Robots. *IEEE Sensors Journal* **22**.18, 17464–17474. doi:10.1109/JSEN.2021.3069452 (2022).

134. Jain, S., Seth, G., Paruthi, A., Soni, U. & Kumar, G. Synthetic data augmentation for surface defect detection and classification using deep learning. *Journal of Intelligent Manufacturing* **33**.4, 1007–1020. doi:10.1007/s10845-020-01710-x (2022).

135. Zhang, X., Zou, Y. & Li, S. Semi-supervised generative adversarial network with guaranteed safeness for industrial quality prediction. *Computers & Chemical Engineering* **153,** 107418. doi:10.1016/j.compchemeng.2021.107418 (2021).

136. Kim, G., Choi, J. G., Ku, M. & Lim, S. Developing a semi-supervised learning and ordinal classification framework for quality level prediction in manufacturing. *Computers & Industrial Engineering* **181,** 109286. doi:10.1016/j.cie.2023.109286 (2023).

137. Kang, P., Kim, D. & Cho, S. Semi-supervised support vector regression based on self-training with label uncertainty: An application to virtual metrology in semiconductor manufacturing. *Expert Systems with Applications* **51,** 85–106. doi:10.1016/j.eswa.2015.12.027 (2016).

138. Wang, Y., Gao, L., Gao, Y. & Li, X. A new graph-based semi-supervised method for surface defect classification. *Robotics and Computer-Integrated Manufacturing* **68,** 102083. doi:10.1016/j.rcim.2020.102083 (2021).

139. Kaelbling, L. P., Littman, M. L. & Moore, A. W. Reinforcement Learning: A Survey. *Journal of Artificial Intelligence Research* **4,** 237–285. doi:10.1613/jair.301 (1996).

140. Puterman, M. L. in *Stochastic Models* 331–434 (Elsevier, 1990). doi:10.1016/S0927-0507(05)80172-0.

141. Nievas, N., Pagès-Bernaus, A., Bonada, F., Echeverria, L. & Domingo, X. Reinforcement Learning for Autonomous Process Control in Industry 4.0: Advantages and Challenges. *Applied Artificial Intelligence* **38**.1. doi:10.1080/08839514.2024.2383101 (2024).

142. Meyes, R. *et al.* Motion Planning for Industrial Robots using Reinforcement Learning. *Procedia CIRP* **63.** Manufacturing Systems 4.0 – Proceedings of the 50th CIRP Conference on Manufacturing Systems, 107–112. doi:10.1016/j.procir.2017.03.095 (2017).

143. Liu, C., Gao, J., Bi, Y., Shi, X. & Tian, D. A Multitasking-Oriented Robot Arm Motion Planning Scheme Based on Deep Reinforcement Learning and Twin Synchro-Control. *Sensors* **20**.12. doi:10.3390/s20123515 (2020).

144. Beltran-Hernandez, C. C., Petit, D., Ramirez-Alpizar, I. G. & Harada, K. Variable Compliance Control for Robotic Peg-in-Hole Assembly: A Deep-Reinforcement-Learning Approach. *Applied Sciences* **10**.19. doi:10.3390/app10196923 (2020).

145. Wen, X. *et al.* Novel data-driven two-dimensional Q-learning for optimal tracking control of batch process with unknown dynamics. *ISA Transactions* **125,** 10–21. doi:10.1016/j.isatra.2021.06.007 (2022).

146. Masinelli, G., Le-Quang, T., Zanoli, S., Wasmer, K. & Shevchik, S. A. Adaptive Laser Welding Control: A Reinforcement Learning Approach. *IEEE Access* **8,** 103803–103814. doi:10.1109/ACCESS.2020.2998052 (2020).

147. Reinisch, N., Rudolph, F., Günther, S., Bailly, D. & Hirt, G. Successful Pass Schedule Design in Open-Die Forging Using Double Deep Q-Learning. *Processes* **9**.7. doi:10.3390/pr9071084 (2021).

148. Waschneck, B. *et al.* Optimization of global production scheduling with deep reinforcement learning. *Procedia CIRP* **72.** 51st CIRP Conference on Manufacturing Systems, 1264–1269. doi:10.1016/j.procir.2018.03.212 (2018).

149. Kuhnle, A., Schäfer, L., Stricker, N. & Lanza, G. Design, Implementation and Evaluation of Reinforcement Learning for an Adaptive Order Dispatching in Job Shop Manufacturing Systems. *Procedia CIRP* **81.** 52nd CIRP Conference on Manufacturing Systems (CMS), Ljubljana, Slovenia, June 12-14, 2019, 234–239. doi:10.1016/j.procir.2019.03.041 (2019).

150. Wang, D.-L., Sun, Q.-Y., Li, Y.-Y. & Liu, X.-R. Optimal Energy Routing Design in Energy Internet with Multiple Energy Routing Centers Using Artificial Neural Network-Based Reinforcement Learning Method. *Applied Sciences* **9**.3. doi:10.3390/app9030520 (2019).

151. Li, X., Luo, Q., Wang, L., Zhang, R. & Gao, F. Off-policy reinforcement learning-based novel model-free minmax fault-tolerant tracking control for industrial processes. *Journal of Process Control* **115,** 145–156. doi:10.1016/j.jprocont.2022.05.006 (2022).

152. Zhong, X., Zhang, L. & Ban, H. Deep reinforcement learning for class imbalance fault diagnosis of equipment in nuclear power plants. *Annals of Nuclear Energy* **184,** 109685. doi:10.1016/j.anucene.2023.109685 (2023).

153. Grieves, M. & Vickers, J. in *Transdisciplinary Perspectives on Complex Systems: New Findings and Approaches* (eds Kahlen, F.-J., Flumerfelt, S. & Alves, A.) 85–113 (Springer International Publishing, Cham, 2017). ISBN: 978-3-319-38756-7. doi:10.1007/978-3-319-38756-7_4.

154. Glaessgen, E. & Stargel, D. in *53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference* (). doi:10.2514/6.2012-1818.

155. Liu, M., Fang, S., Dong, H. & Xu, C. Review of digital twin about concepts, technologies, and industrial applications. *Journal of Manufacturing Systems* **58.** Digital Twin towards Smart Manufacturing and Industry 4.0, 346–361. doi:10.1016/j.jmsy.2020.06.017 (2021).

156. Lo, C., Chen, C. & Zhong, R. Y. A review of digital twin in product design and development. *Advanced Engineering Informatics* **48,** 101297. doi:10.1016/j.aei.2021.101297 (2021).

157. Yi, Y. *et al.* Digital twin-based smart assembly process design and application framework for complex products and its case study. *Journal of Manufacturing Systems* **58,** 94–107. doi:10.1016/j.jmsy.2020.04.013 (2021).

158. Agostino, Í. R. S., Broda, E., Frazzon, E. M. & Freitag, M. in *Scheduling in Industry 4.0 and Cloud Manufacturing* (eds Sokolov, B., Ivanov, D. & Dolgui, A.) 39–60 (Springer International Publishing, Cham, 2020). ISBN: 978-3-030-43177-8. doi:10.1007/978-3-030-43177-8_3.

159. Touhid, M. T. B. *et al.* Building a cloud-based digital twin for remote monitoring and control of a robotic assembly system. *The International Journal of Advanced Manufacturing Technology* **129**.9-10, 4045–4057. doi:10.1007/s00170-023-12611-7 (2023).

160. Fahim, M., Sharma, V., Cao, T.-V., Canberk, B. & Duong, T. Q. Machine Learning-Based Digital Twin for Predictive Modeling in Wind Turbines. *IEEE Access* **10,** 14184–14194. doi:10.1109/ACCESS.2022.3147602 (2022).

161. Xue, R. *et al.* Digital twin-driven CNC spindle performance assessment. *The International Journal of Advanced Manufacturing Technology* **119**.3-4, 1821–1833. doi:10.1007/s00170-021-08403-6 (2022).

162. Luo, D., Thevenin, S. & Dolgui, A. *A Digital Twin-Driven Methodology for Material Resource Planning Under Uncertainties* in *Advances in Production Management Systems. Artificial Intelligence for Sustainable and Resilient Production Systems* (eds Dolgui, A., Bernard, A., Lemoine, D., von Cieminski, G. & Romero, D.) (Springer International Publishing, Cham, 2021), 321–329. ISBN: 978-3-030-85874-2.

163. Semeraro, C., Lezoche, M., Panetto, H. & Dassisti, M. Data-driven invariant modelling patterns for digital twin design. *Journal of Industrial Information Integration* **31,** 100424. doi:10.1016/j.jii.2022.100424 (2023).

164. Min, Q., Lu, Y., Liu, Z., Su, C. & Wang, B. Machine Learning based Digital Twin Framework for Production Optimization in Petrochemical Industry. *International Journal of Information Management* **49,** 502–519. doi:10.1016/j.ijinfomgt.2019.05.020 (2019).

165. Jarosz, K. & Özel, T. Machine learning approaches towards digital twin development for machining systems. *International Journal of Mechatronics and Manufacturing Systems* **15,** 127. doi:10.1504/IJMMS.2022.124922 (2/3 2022).

166. Tancredi, G. P., Vignali, G. & Bottani, E. Integration of Digital Twin, Machine-Learning and Industry 4.0 Tools for Anomaly Detection: An Application to a Food Plant. *Sensors* **22**.11. doi:10.3390/s22114143 (2022).

167. Fowler, J. W. & Rose, O. Grand Challenges in Modeling and Simulation of Complex Manufacturing Systems. *SIMULATION* **80**.9, 469–476. doi:10.1177/0037549704044324 (2004).

168. Chang, J., Fabien, M. S., Knepley, M. G. & Mills, R. T. Comparative Study of Finite Element Methods Using the Time-Accuracy-Size(TAS) Spectrum Analysis. *SIAM Journal on Scientific Computing* **40**.6, C779–C802. doi:10.1137/18M1172260 (2018).

169. Marechal, Y., Ramdane, B. & Botelho, D. P. Computational Performances of Natural Element and Finite Element Methods. *IEEE Transactions on Magnetics* **50**.2, 405–408. doi:10.1109/TMAG.2013.2285259 (2014).

170. Alkan, B., Vera, D. A., Ahmad, M., Ahmad, B. & Harrison, R. Complexity in manufacturing systems and its measures: a literature review. *European Journal of Industrial Engineering* **12**.1, 116–150. doi:10.1504/EJIE.2018.089883 (2018).

171. Jankovic, A., Chaudhary, G. & Goia, F. Designing the design of experiments (DOE) – An investigation on the influence of different factorial designs on the characterization of complex systems. *Energy and Buildings* **250,** 111298. doi:10.1016/j.enbuild.2021.111298 (2021).

172. Fisher, O. J. *et al.* Considerations, challenges and opportunities when developing data-driven models for process manufacturing systems. *Computers & Chemical Engineering* **140,** 106881. doi:10.1016/j.compchemeng.2020.106881 (2020).

173. Yang, C., Shen, W. & Wang, X. The Internet of Things in Manufacturing: Key Issues and Potential Applications. *IEEE Systems, Man, and Cybernetics Magazine* **4**.1, 6–15. doi:10.1109/MSMC.2017.2702391 (2018).

174. Holst, C.-A. & Lohweg, V. Scarce Data in Intelligent Technical Systems: Causes, Characteristics, and Implications. *Sci* **4**.4. doi:10.3390/sci4040049 (2022).

175. Simpson, T., Toropov, V., Balabanov, V. & Viana, F. in *12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference* (2008). doi:10.2514/6.2008-5802.

176. Alizadeh, R., Allen, J. K. & Mistree, F. Managing computational complexity using surrogate models: a critical review. *Research in Engineering Design* **31,** 275–298. doi:10.1007/s00163-020-00336-7 (2020).

177. Pfrommer, J. *et al.* Optimisation of manufacturing process parameters using deep neural networks as surrogate models. *Procedia CIRP* **72.** 51st CIRP Conference on Manufacturing Systems, 426–431. doi:10.1016/j.procir.2018.03.046 (2018).

178. Razavi, S., Tolson, B. A. & Burn, D. H. Review of surrogate modeling in water resources. *Water Resources Research* **48**.7. doi:10.1029/2011WR011527 (2012).

179. Box, G. E. P. & Wilson, K. B. On the Experimental Attainment of Optimum Conditions. *Journal of the Royal Statistical Society Series B: Statistical Methodology* **13**.1, 1–38. doi:10.1111/j.2517-6161.1951.tb00067.x (1951).

180. Myers, R. H., Montgomery, D. C. & Anderson-Cook, C. M. *Response surface methodology: process and product optimization using designed experiments* (John Wiley & Sons, 2016).

181. Bezerra, M. A., Santelli, R. E., Oliveira, E. P., Villar, L. S. & Escaleira, L. A. Response surface methodology (RSM) as a tool for optimization in analytical chemistry. *Talanta* **76**.5, 965–977. doi:10.1016/j.talanta.2008.05.019 (2008).

182. Jun, S., Jeon, Y.-H., Rho, J. & Lee, D.-H. *Application of Collaborative Optimization Using Response Surface Methodology to an Aircraft Wing Design* in *10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference* (2004). doi:10.2514/6.2004-4442.

183. Stewart, P., Fleming, P. & MacKenzie, S. *On the response surface methodology and designed experiments for computationally intensive distributed aerospace simulations* in *Proceedings of the Winter Simulation Conference* **1** (2002), 476–482 vol.1. doi:10.1109/WSC.2002.1172919.

184. Gunaraj, V. & Murugan, N. Application of response surface methodology for predicting weld bead quality in submerged arc welding of pipes. *Journal of Materials Processing Technology* **88**.1, 266–275. doi:10.1016/S0924-0136(98)00405-1 (1999).

185. Onwubolu, G. C. & Kumar, S. Response surface methodology-based approach to CNC drilling operations. *Journal of Materials Processing Technology* **171**.1, 41–47. doi:10.1016/j.jmatprotec.2005.06.064 (2006).

186. Cressie, N. The origins of kriging. *Mathematical Geology* **22**.3, 239–252. doi:10.1007/BF00889887 (1990).

187. Xiaobo, Z. *Comparison of response surface method and Kriging method for approximation modeling* in *2017 2nd International Conference on Power and Renewable Energy (ICPRE)* (2017), 66–70. doi:10.1109/ICPRE.2017.8390502.

188. Raid, I., Kusnezowa, T. & Seewig, J. Application of ordinary kriging for interpolation of micro-structured technical surfaces. *Measurement Science and Technology* **24**.9, 095201. doi:10.1088/0957-0233/24/9/095201 (2013).

189. Gaspar, B., Teixeira, A. & Soares, C. G. Assessment of the efficiency of Kriging surrogate models for structural reliability analysis. *Probabilistic Engineering Mechanics* **37**, 24–34. doi:10.1016/j.probengmech.2014.03.011 (2014).

190. Jiang, P. *et al.* Optimization of welding process parameters by combining Kriging surrogate with particle swarm optimization algorithm. *The International Journal of Advanced Manufacturing Technology* **86**.9-12, 2473–2483. doi:10.1007/s00170-016-8382-1 (2016).

191. Vicario, G., Giraudo, M. T. & Calì, V. Kriging modelization in predicting metal sheet elongation. *Quality and Reliability Engineering International* **34**.7, 1390–1399. doi:10.1002/qre.2347 (2018).

192. Amari, S.-i. Backpropagation and stochastic gradient descent method. *Neurocomputing* **5**.4, 185–196. doi:10.1016/0925-2312(93)90006-O (1993).

193. Kingma, D. P. & Ba, J. *Adam: A Method for Stochastic Optimization* 2017. arXiv: 1412.6980 [cs.LG]. https://arxiv.org/abs/1412.6980.

194. Borra, S. & Di Ciaccio, A. Measuring the prediction error. A comparison of cross-validation, bootstrap and covariance penalty methods. *Computational Statistics & Data Analysis* **54**.12, 2976–2989. doi:10.1016/j.csda.2010.03.004 (2010).

195. Arlot, S. & Celisse, A. A survey of cross-validation procedures for model selection. *Statistics Surveys* **4.** doi:10.1214/09-SS054 (2010).

196. Le, V. T., Bui, M. C., Pham, T. Q. D., Tran, H. S. & Tran, X. V. Efficient prediction of thermal history in wire and arc additive manufacturing combining machine learning and numerical simulation. *The International Journal of Advanced Manufacturing Technology* **126**.9-10, 4651–4663. doi:10.1007/s00170-023-11473-3 (2023).

197. Viisainen, J. *et al.* Rapidly predicting the effect of tool geometry on the wrinkling of biaxial NCFs during composites manufacturing using a deep learning surrogate model. *Composites Part B: Engineering* **253,** 110536. doi:10.1016/j.compositesb.2023.110536 (2023).

198. Tu, Y. *et al.* Towards an instant structure-property prediction quality control tool for additive manufactured steel using a crystal plasticity trained deep learning surrogate. *Materials & Design* **213,** 110345. doi:10.1016/j.matdes.2021.110345 (2022).

199. Morand, L., Helm, D., Iza-Teran, R. & Garcke, J. A knowledge-based surrogate modeling approach for cup drawing with limited data. *IOP Conference Series: Materials Science and Engineering* **651**.1, 012047. doi:10.1088/1757-899X/651/1/012047 (2019).

200. Hürkamp, A. *et al.* Combining Simulation and Machine Learning as Digital Twin for the Manufacturing of Overmolded Thermoplastic Composites. *Journal of Manufacturing and Materials Processing* **4**.3. doi:10.3390/jmmp4030092 (2020).

201. Liu, Z., Zhu, D., Raju, L. & Cai, W. Tackling Photonic Inverse Design with Machine Learning. *Advanced Science* **8**.5, 2002923. doi:10.1002/advs.202002923 (2021).

202. Sikirica, A., Grbčić, L. & Kranjčević, L. Machine learning based surrogate models for microchannel heat sink optimization. *Applied Thermal Engineering* **222,** 119917. doi:10.1016/j.applthermaleng.2022.119917 (2023).

203. Liow, C. H. *et al.* Machine learning assisted synthesis of lithium-ion batteries cathode materials. *Nano Energy* **98,** 107214. doi:10.1016/j.nanoen.2022.107214 (2022).

204. Houchins, G. & Viswanathan, V. An accurate machine-learning calculator for optimization of Li-ion battery cathodes. *The Journal of Chemical Physics* **153**.5. doi:10.1063/5.0015872 (2020).

205. Asher, M. J., Croke, B. F. W., Jakeman, A. J. & Peeters, L. J. M. A review of surrogate models and their application to groundwater modeling. *Water Resources Research* **51**.8, 5957–5973. doi:10.1002/2015WR016967 (2015).

206. Westermann, P. & Evins, R. Surrogate modelling for sustainable building design – A review. *Energy and Buildings* **198,** 170–186. doi:10.1016/j.enbuild.2019.05.057 (2019).

207. Queipo, N. V. *et al.* Surrogate-based analysis and optimization. *Progress in Aerospace Sciences* **41**.1, 1–28. doi:10.1016/j.paerosci.2005.02.001 (2005).

208. McBride, K. & Sundmacher, K. Overview of Surrogate Modeling in Chemical Process Engineering. *Chemie Ingenieur Technik* **91**.3, 228–239. doi:10.1002/cite.201800091 (2019).

209. Viana, F. A., Gogu, C. & Goel, T. Surrogate modeling: tricks that endured the test of time and some recent developments. *Structural and Multidisciplinary Optimization*, 1–28. doi:10.1007/s00158-021-03001-2 (2021).

210. Kudela, J. & Matousek, R. Recent advances and applications of surrogate models for finite element method computations: A review. *Soft Computing* **26**.24, 13709–13733. doi:10.1007/s00500-022-07362-8 (2022).

211. Rosato, D. V. & Rosato, M. G. *Injection molding handbook* (Springer Science & Business Media, 2012).

212. Frizelle, W. G. in *Applied Plastics Engineering Handbook* (ed Kutz, M.) 205–214 (William Andrew Publishing, Oxford, 2011). ISBN: 978-1-4377-3514-7. doi:10.1016/B978-1-4377-3514-7.10013-3.

213. Kurt, M., Saban Kamber, O., Kaynak, Y., Atakok, G. & Girit, O. Experimental investigation of plastic injection molding: Assessment of the effects of cavity pressure and mold temperature on the quality of the final products. *Materials & Design* **30**.8, 3217–3224. doi:10.1016/j.matdes.2009.01.004 (2009).

214. Ciofu, C. & Mindru, D. T. Injection and micro injection of polymeric plastics materials: a review. *Int. J. Mod. Manufact. Technol* **1,** 49–68 (2013).

215. Bonollo, F., Gramegna, N. & Timelli, G. High-Pressure Die-Casting: Contradictions and Challenges. *JOM* **67,** 901–908. doi:10.1007/s11837-015-1333-8 (2015).

216. Casarotto, F., Franke, A. & Franke, R. in *Advanced Materials in Automotive Engineering* (ed Rowe, J.) 109–149 (Woodhead Publishing, 2012). ISBN: 978-1-84569-561-3. doi:10.1533/9780857095466.109.

217. Dumstorff, G., Pille, C., Tiedemann, R., Busse, M. & Lang, W. Smart aluminum components: Printed sensors for integration into aluminum during high-pressure casting. *Journal of Manufacturing Processes* **26,** 166–172. doi:10.1016/j.jmapro.2017.02.006 (2017).

218. Fan, S., Wang, X., Wang, G. G. & Weiler, J. P. in (IntechOpen, 2023). doi:10.5772/intechopen.110494.

219. Wang, Q.-g., Wang, A. & Coryell, J. Ultra-large aluminum shape casting: Opportunities and challenges. *China Foundry* **21**.5, 397–408. doi:10.1007/s41230-024-4111-9 (2024).

220. Merklein, M. & Lechler, J. Investigation of the thermo-mechanical properties of hot stamping steels. *Journal of Materials Processing Technology* **177**.1. Proceedings of the 11th International Conference on Metal Forming 2006, 452–455. doi:10.1016/j.jmatprotec.2006.03.233 (2006).

221. Karbasian, H. & Tekkaya, A. A review on hot stamping. *Journal of Materials Processing Technology* **210**.15, 2103–2118. doi:10.1016/j.jmatprotec.2010.07.019 (2010).

222. Neugebauer, R., Altan, T., Geiger, M., Kleiner, M. & Sterzing, A. Sheet metal forming at elevated temperatures. *CIRP Annals* **55**.2, 793–816. doi:10.1016/j.cirp.2006.10.008 (2006).

223. Nagathan, A. & Penter, L. *Chapter 7: Hot Stamping* in *Sheet Metal Forming—Processes and Applications* (eds Altan, T. & Tekkaya, A.) (2012), 153–163.

224. Hardell, J. *Tribology of hot forming tool and high strength steels* PhD thesis (Luleå University of Technology, Machine Elements, 2009), 133. ISBN: 978-91-7439-029-2.

225. CoreTech System Co., Ltd. *Moldex3D* version Moldex3D 2020 R2. 2020. https://moldex3d.com.

226. Cassisi, C., Montalto, P., Aliotta, M., Cannata, A. & Pulvirenti, A. in *Advances in Data Mining Knowledge Discovery and Applications* (ed Karahoca, A.) chap. 3 (IntechOpen, Rijeka, 2012). doi:10.5772/49941.

227. Sorjamaa, A., Hao, J., Reyhani, N., Ji, Y. & Lendasse, A. Methodology for long-term prediction of time series. *Neurocomputing* **70**.16. Neural Network Applications in Electrical Engineering Selected papers from the 3rd International Work-Conference on Artificial Neural Networks (IWANN 2005), 2861–2869. doi:10.1016/j.neucom.2006.06.015 (2007).

228. Ho, T. K. *Random decision forests* in *Proceedings of 3rd International Conference on Document Analysis and Recognition* **1** (1995), 278–282 vol.1. doi:10.1109/ICDAR.1995.598994.

229. Altair Engineering, Inc. *Altair Inspire Cast* version Altair Inspire Cast 2021.1. 2021. https://altair.com/inspire-cast/.

230. Smith, M. *ABAQUS/Standard User's Manual, Version 6.9* English (Dassault Systèmes Simulia Corp, United States, 2009).

231. Bergman, G. & Oldenburg, M. A finite element model for thermomechanical analysis of sheet metal forming. *International Journal for Numerical Methods in Engineering* **59,** 1167–1186. doi:https://onlinelibrary.wiley.com/doi/10.1002/nme.911 (2004).

232. Åkerström, P. *Modelling and Simulation of Hot Stamping* PhD thesis (Luleå Tekniska Universitet, 2006).

233. Palmieri, M. E., Lorusso, V. D. & Tricarico, L. Investigation of material properties of tailored press hardening parts using numerical and physical simulation. *Procedia Manufacturing* **50.** 18th International Conference on Metal Forming 2020, 104–109. doi:10.1016/j.promfg.2020.08.019 (2020).

234. Schönbach, T., Messner, M. & Gmainer, C. Press-Hardening Simulation - the next Level of Maturity. *Journal of Physics: Conference Series* **896,** 012062. doi:10.1088/1742-6596/896/1/012062 (2017).

235. Lejon, E., Kyösti, P. & Lindström, J. Machine learning for detection of anomalies in press-hardening: Selection of efficient methods. *Procedia CIRP* **72.** 51st CIRP Conference on Manufacturing Systems, 1079–1083. doi:10.1016/j.procir.2018.03.221 (2018).

236. Vollmer, R. & Palm, C. Process Monitoring And Real Time Algorithmic For Hot Stamping Lines. *Procedia Manufacturing* **29.** "18th International Conference on Sheet Metal, SHEMET 2019""New Trends and Developments in Sheet Metal Processing", 256–263. ISSN: 2351-9789. doi:10.1016/j.promfg.2019.02.135. https://www.sciencedirect.com/science/article/pii/S2351978919301684 (2019).

237. Penter, L., Link, P., Stoll, A., Albert, A. & Ihlenfeldt, S. Predictive Analysis from numerical and experimental data in press hardening. *IOP Conference Series: Materials Science and Engineering* **651,** 012060. doi:10.1088/1757-899X/651/1/012060 (2019).

238. Garcia-Llamas, E., Pujante, J., Torres, P. & Bonada, F. A Thermography-based Online Control Method for Press Hardening. *IOP Conference Series: Materials Science and Engineering* **1157**.1, 012010. doi:10.1088/1757-899x/1157/1/012010 (2021).

239. Valls, I., Hamasaiid, A. & Padré, A. High Thermal Conductivity and High Wear Resistance Tool Steels for cost-effective Hot Stamping Tools. *Journal of Physics: Conference Series* **896,** 012046. doi:10.1088/1742-6596/896/1/012046 (2017).

240. Pedregosa, F. *et al.* Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* **12,** 2825–2830 (2011).

241. Hart-Rawung, T., Buhl, J. & Bambach, M. A Fast Approach for Optimization of Hot Stamping Based on Machine Learning of Phase Transformation Kinetics. *Procedia Manufacturing* **47.** 23rd International Conference on Material Forming, 707–712. doi:10.1016/j.promfg.2020.04.218 (2020).

242. M. D. Mckay, R. J. B. & Conover, W. J. A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output From a Computer Code. *Technometrics* **42**.1, 55–61. doi:10.1080/00401706.2000.10485979 (2000).

243. Blanchet, F. G., Legendre, P. & Borcard, D. Forward Selection of Explanatory Variables. *Ecology* **89**.9, 2623–2632. doi:10.1890/07-0986.1 (2008).

244. Santamargarita, D., Salinas, G., Molinero, D., Bueno, E. J. & Vasić, M. Tradeoff Between Accuracy and Computational Time for Magnetics Thermal Model Based on Artificial Neural Networks. *IEEE Journal of Emerging and Selected Topics in Power Electronics* **11**.6, 5658–5674. doi:10.1109/JESTPE.2022.3203934 (2023).

245. Dogan, A. & Birant, D. Machine learning and data mining in manufacturing. *Expert Systems with Applications* **166,** 114060. doi:10.1016/j.eswa.2020.114060 (2021).

246. Lee, G., Kim, W., Oh, H., Youn, B. D. & Kim, N. H. Review of statistical model calibration and validation—from the perspective of uncertainty structures. *Structural and Multidisciplinary Optimization* **60**.4, 1619–1644. doi:10.1007/s00158-019-02270-2 (2019).

247. Alañón, A., Cerro-Prada, E., Vázquez-Gallo, M. J. & Santos, A. P. Mesh size effect on finite-element modeling of blast-loaded reinforced concrete slab. *Engineering with Computers* **34**.4, 649–658. doi:10.1007/s00366-017-0564-4 (2018).

248. Kaliakin, V. N. *Introduction to Approximate Solution Techniques, Numerical Modeling, and Finite Element Methods* ISBN: 9781315274461. doi:10.1201/9781315274461 (CRC Press, 2018).

249. Schnaubelt, E., Wozniak, M. & Schöps, S. Thermal thin shell approximation towards finite element quench simulation. *Superconductor Science and Technology* **36**.4, 044004. doi:10.1088/1361-6668/acbeea (2023).

250. Huang, C.-T., Hsu, Y.-H. & Chen, B.-S. Investigation on the internal mechanism of the deviation between numerical simulation and experiments in injection molding product development. *Polymer Testing* **75,** 327–336. doi:10.1016/j.polymertesting.2019.02.028 (2019).

251. Tercan, H. *et al.* Transfer-Learning: Bridging the Gap between Real and Simulation Data for Machine Learning in Injection Molding. *Procedia CIRP* **72.** 51st CIRP Conference on Manufacturing Systems, 185–190. doi:10.1016/j.procir.2018.03.087 (2018).

252. Müller, M. S., Jazdi, N. & Weyrich, M. Self-improving Models for the Intelligent Digital Twin: Towards Closing the Reality-to-Simulation Gap. *IFAC-PapersOnLine* **55**.2. 14th IFAC Workshop on Intelligent Manufacturing Systems IMS 2022, 126–131. doi:10.1016/j.ifacol.2022.04.181 (2022).

253. Trentsios, P., Wolf, M. & Gerhard, D. Overcoming the Sim-to-Real Gap in Autonomous Robots. *Procedia CIRP* **109.** 32nd CIRP Design Conference (CIRP Design 2022) - Design in a changing world, 287–292. doi:10.1016/j.procir.2022.05.251 (2022).

254. Forrester, A. I., Sóbester, A. & Keane, A. J. Multi-fidelity optimization via surrogate modelling. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* **463**.2088, 3251–3269. doi:10.1098/rspa.2007.1900 (2007).

255. Liu, H., Ong, Y.-S., Cai, J. & Wang, Y. Cope with diverse data structures in multi-fidelity modeling: A Gaussian process method. *Engineering Applications of Artificial Intelligence* **67,** 211–225. doi:10.1016/j.engappai.2017.10.008 (2018).

256. Giselle Fernández-Godino, M., Park, C., Kim, N. H. & Haftka, R. T. Issues in Deciding Whether to Use Multifidelity Surrogates. *AIAA Journal* **57**.5, 2039–2054. doi:10.2514/1.J057750 (2019).

257. Zhang, X., Xie, F., Ji, T., Zhu, Z. & Zheng, Y. Multi-fidelity deep neural network surrogate model for aerodynamic shape optimization. *Computer Methods in Applied Mechanics and Engineering* **373,** 113485. doi:10.1016/j.cma.2020.113485 (2021).

258. Nath, P., Sato, M., Karve, P. & Mahadevan, S. Multi-fidelity Modeling for Uncertainty Quantification in Laser Powder Bed Fusion Additive Manufacturing. *Integrating Materials and Manufacturing Innovation* **11**.2, 256–275. doi:10.1007/s40192-022-00260-9 (2022).

259. Menon, N., Mondal, S. & Basak, A. Multi-Fidelity Surrogate-Based Process Mapping with Uncertainty Quantification in Laser Directed Energy Deposition. *Materials* **15**.8. doi:10.3390/ma15082902 (2022).

260. Torrey, L. & Shavlik, J. in *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques* 242–264 (IGI global, 2010). doi:10.4018/978-1-60566-766-9.ch011.

261. Pan, S. J. & Yang, Q. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering* **22**.10, 1345–1359. doi:10.1109/TKDE.2009.191 (2010).

262. Hutchinson, M. L. *et al.* Overcoming data scarcity with transfer learning. *arXiv preprint arXiv:1711.05099.* doi:https://doi.org/10.48550/arXiv.1711.05099 (2017).

263. Olsson, D. T. *An LS-DYNA material model for simulations of hot stamping processes of ultra high strength steels* in *Proceedings of the 7th European LS-DYNA Conference, Sweden* (2009).

264. Tian, K. *et al.* Enhanced variable-fidelity surrogate-based optimization framework by Gaussian process regression and fuzzy clustering. *Computer Methods in Applied Mechanics and Engineering* **366,** 113045. doi:10.1016/j.cma.2020.113045 (2020).

265. Li, Z. & Montomoli, F. Aleatory uncertainty quantification based on multi-fidelity deep neural networks. *Reliability Engineering & System Safety* **245,** 109975. doi:10.1016/j.ress.2024.109975 (2024).

266. Xiao, M., Zhang, G., Breitkopf, P., Villon, P. & Zhang, W. Extended Co-Kriging interpolation method based on multi-fidelity data. *Applied Mathematics and Computation* **323,** 120–131. doi:10.1016/j.amc.2017.10.055 (2018).

267. Koziel, S., Tesfahunegn, Y. & Leifsson, L. Variable-fidelity CFD models and co-Kriging for expedited multi-objective aerodynamic design optimization. *Engineering Computations* **323**.8, 2320–2338. doi:10.1108/EC-09-2015-0277 (2016).

268. Pietrenko-Dabrowska, A. & Koziel, S. Antenna Modeling Using Variable-Fidelity EM Simulations and Constrained Co-Kriging. *IEEE Access* **8,** 91048–91056. doi:10.1109/ACCESS.2020.2993951 (2020).

269. Tan, C. *et al. A Survey on Deep Transfer Learning* in *Artificial Neural Networks and Machine Learning – ICANN 2018* (eds Kůrková, V., Manolopoulos, Y., Hammer, B., Iliadis, L. & Maglogiannis, I.) (Springer International Publishing, Cham, 2018), 270–279. ISBN: 978-3-030-01424-7.

270. Wang, M. & Deng, W. Deep visual domain adaptation: A survey. *Neurocomputing* **312,** 135–153. doi:10.1016/j.neucom.2018.05.083 (2018).

271. Ruder, S., Peters, M. E., Swayamdipta, S. & Wolf, T. *Transfer Learning in Natural Language Processing* in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Tutorials* (Association for Computational Linguistics, Minneapolis, Minnesota, 2019), 15–18. doi:10.18653/v1/N19-5004.

272. Deng, J. *et al. ImageNet: A large-scale hierarchical image database* in *2009 IEEE Conference on Computer Vision and Pattern Recognition* (2009), 248–255. doi:10.1109/CVPR.2009.5206848.

273. *WordNet: an electronic lexical database* (ed Fellbaum, C.) (MIT Press, 1998).

274. Reis, J. & Gonçalves, G. *Laser Seam Welding optimization using Inductive Transfer Learning with Artificial Neural Networks* in *2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA)* **1** (2018), 646–653. doi:10.1109/ETFA.2018.8502599.

275. Maschler, B. *et al.* Insights and Example Use Cases on Industrial Transfer Learning. *Procedia CIRP* **107.** Leading manufacturing systems transformation – Proceedings of the 55th CIRP Conference on Manufacturing Systems 2022, 511–516. doi:`10.1016/j.procir.2022.05.017` (2022).

276. Tian, K., Li, Z., Zhang, J., Huang, L. & Wang, B. Transfer learning based variable-fidelity surrogate model for shell buckling prediction. *Composite Structures* **273,** 114285. doi:`10.1016/j.compstruct.2021.114285` (2021).

277. Liao, P., Song, W., Du, P. & Zhao, H. Multi-fidelity convolutional neural network surrogate model for aerodynamic optimization based on transfer learning. *Physics of Fluids* **33**.12, 127121. doi:`10.1063/5.0076538` (2021).

278. Pujante, J., Garcıa-Llamas, E. & Casellas, D. *Study of wear in press hardening using a pilot facility* in *Proceedings of the 7th International Conference Hot Sheet Metal Forming of High-Performance Steel, Lulea, Sweden* (2019), 2–5.

279. Martín Abadi *et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems* Software available from tensorflow.org. 2015. `https://www.tensorflow.org/`.

280. O'Malley, T. *et al. Keras Tuner* 2019. `https://github.com/keras-team/keras-tuner`.

281. Cho, H. *et al.* Basic Enhancement Strategies When Using Bayesian Optimization for Hyperparameter Tuning of Deep Neural Networks. *IEEE Access* **8,** 52588–52608. doi:`10.1109/ACCESS.2020.2981072` (2020).

282. Chollet, F. *et al. Keras* `https://keras.io`. 2015.

283. Klambauer, G., Unterthiner, T., Mayr, A. & Hochreiter, S. *Self-Normalizing Neural Networks* in *Advances in Neural Information Processing Systems* (eds Guyon, I. *et al.*) **30** (Curran Associates, Inc., 2017).

284. Cybenko, G. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems* **2**.4, 303–314. doi:`10.1007/BF02551274` (1989).

285. Chen, F.-C. Back-propagation neural networks for nonlinear self-tuning adaptive control. *IEEE Control Systems Magazine* **10**.3, 44–48. doi:`10.1109/37.55123` (1990).

286. Le Gratiet, L. *Multi-fidelity Gaussian process regression for computer experiments* Theses (Université Paris-Diderot - Paris VII, 2013). `https://theses.hal.science/tel-00866770`.

287. Jondral, F. K. White Gaussian Noise – Models for Engineers. *Frequenz* **72**.5-6, 293–299. doi:`doi:10.1515/freq-2017-0064` (2018).