DOCTORAL (PHD) THESIS

Balázs Palotai

Veszprém

2026

DOI:10.18136/PE.2026.1025

University of Pannonia

Department of Process Engineering

Doctoral School of Chemical Engineering and Material Sciences

Reliable Digital Twin Operation via Surrogate-Based and Online Learning-Driven Flowsheet Model Maintenance

Doctoral (PhD) Thesis

Balázs Palotai

Supervisors:

Ágnes Takács-Bárkányi, PhD

Gábor Kis, PhD

Consultant:

Tibor Chován, PhD

Veszprém

2026

Reliable Digital Twin Operation via Surrogate-Based and Online Learning-Driven Flowsheet Model Maintenance

The thesis was prepared for the award of a doctoral degree (PhD) within the framework of the Doctoral School of Chemical Engineering and Material Sciences at University of Pannonia

in the discipline of Materials Science and Technology

written by: Balázs Palotai

Supervisors: Ágnes Takács-Bárkányi, Gábor Kis

I recommend the dissertation for acceptance: yes / no.

…….
supervisor

I recommend the dissertation for acceptance: yes / no.

…….
supervisor

I recommend the dissertation for peer review.

…….
chair of the DDHC

The PhD-candidate has achieved ……. % at the public debate.

The composition of the Final Examination Committee:

chair:…….

reviewers:……

members:…….

Veszprém,

…….
chair of the committee

Qualification of degree: ….....

Veszprém,

…….
chair of the UDHC

Abstract

Reliable Digital Twin Operation via Surrogate-Based and Online Learning-Driven Flowsheet Model Maintenance

Digital twins offer significant potential for monitoring, optimization, and decision support in the process industry, yet their industrial adoption remains limited due to the difficulty of maintaining model accuracy under changing operating conditions, equipment aging, fouling, and measurement uncertainty. Flowsheet-based process models provide a physically consistent basis for digital twins; however, their practical application requires automated and computationally efficient calibration methods.

This work presents a surrogate-based framework for automated flowsheet model calibration within digital twin systems. A global surrogate model is combined with a flexible, Taguchi-inspired loss function that enables the calibration objective to incorporate modeling priorities and measurement uncertainty, and to be adapted to specific operational scenarios. This formulation significantly reduces computational effort compared to direct flowsheet optimization.

Since surrogate model accuracy degrades as the real process evolves, an incremental learning strategy is developed to continuously update the surrogate while preserving previously learned global behavior. The main contribution of this work is an integrated workflow in which surrogate-based calibration and incremental surrogate retraining are performed jointly.

Validation on an industrially relevant case study shows that the proposed approach reduces flowsheet model evaluations by 92% during calibration while maintaining predictive accuracy across evolving operating regimes.

Kivonat

A digitális ikrek megbízható működésének biztosítása surrogate-alapú és online tanulással támogatott flowsheet modellkarbantartással

A digitális ikrek jelentős potenciált kínálnak a folyamatiparban az állapotfelügyelet, az optimalizálás és a döntéstámogatás területén, ipari alkalmazásukat azonban korlátozza a modellpontosság fenntartásának nehézsége a változó üzemállapotok, a berendezések öregedése, a fouling jelenségek és a mérési bizonytalanságok miatt. A flowsheet-alapú folyamatmodellek fizikai szempontból konzisztens alapot biztosítanak a digitális ikrek számára, gyakorlati alkalmazásuk azonban automatizált és számításigény szempontjából hatékony kalibrációs módszereket igényel.

A dolgozat egy surrogate modelleken alapuló keretrendszert mutat be a digitális ikrekben alkalmazott flowsheet modellek automatizált kalibrációjára. A megközelítés egy globális surrogate modellt kombinál egy rugalmas, Taguchi-inspirált veszteségfüggvénnyel, amely lehetővé teszi a kalibrációs célfüggvénynek a modellezési prioritások és a mérési bizonytalanságok figyelembevételét, valamint az adott üzemeltetési állapotokhoz való igazítását. Ez a megoldás csökkenti a számítási igényt a közvetlen flowsheet-optimalizációhoz képest.

Mivel a surrogate modellek pontossága a valós folyamat időbeli változásával romlik, egy inkrementális tanulási stratégiát dolgoztam ki a surrogate modellek folyamatos frissítésére, a korábban elsajátított globális viselkedés megőrzése mellett. A kutatás fő hozzájárulása egy integrált munkafolyamat, amelyben a surrogate-alapú kalibráció és az inkrementális surrogate modellfrissítés együttesen valósul meg.

Ipari esettanulmány igazolja, hogy az eljárás a kalibráció során 92%-kal csökkenti a flowsheet modellfuttatások számát, miközben megőrzi a predikciós pontosságot.

Zusammenfassung

Zuverlässiger Betrieb digitaler Zwillinge durch surrogatebasierte und online lerngestützte Instandhaltung von Flowsheet-Modellen

Digitale Zwillinge bieten großes Potenzial für die Überwachung, Optimierung und Entscheidungsunterstützung in der Prozessindustrie. Ihre industrielle Einführung wird jedoch durch den hohen Aufwand zur Aufrechterhaltung der Modellgenauigkeit unter veränderlichen Betriebsbedingungen erschwert. Flowsheet-basierte Prozessmodelle bilden hierfür eine physikalisch konsistente Grundlage, erfordern jedoch automatisierte und recheneffiziente Kalibrierungsverfahren.

Diese Arbeit stellt einen surrogatebasierten Ansatz zur automatisierten Kalibrierung von Flowsheet-Modellen vor. Ein globales Surrogatmodell wird mit einer flexiblen, Taguchi-inspirierten Verlustfunktion kombiniert, die Modellierungsprioritäten und Messunsicherheiten berücksichtigt. Dadurch wird der Rechenaufwand gegenüber der direkten Flowsheet-Optimierung deutlich reduziert.

Zur Anpassung an veränderte Prozessbedingungen wird eine Strategie des inkrementellen Lernens entwickelt, welche das Surrogatmodell kontinuierlich aktualisiert und gleichzeitig dessen globale Vorhersagefähigkeit erhält. Die zentrale wissenschaftliche Leistung dieser Arbeit ist die Integration der surrogatebasierten Kalibrierung und des inkrementellen Lernens in einem gemeinsamen Workflow.

Die Validierung anhand einer industriellen Fallstudie zeigt, dass der entwickelte Ansatz die Anzahl der Flowsheet-Modellauswertungen während der Kalibrierung um 92 % reduziert und gleichzeitig eine hohe Vorhersagegenauigkeit gewährleistet.

Acknowledgment

I would like to express my deepest gratitude to my supervisors, **Ágnes Takács-Bárkányi, PhD**, and **Gábor Kis, PhD**, for their continuous guidance, encouragement, and expertise throughout my doctoral work. Their scientific insight, patience, and commitment have played a fundamental role in shaping this research.

I would also like to extend my sincere appreciation to **Tibor Chován, PhD**, whose valuable discussions, practical insights, and continuous support greatly contributed to the development of this work. Although he was not an official supervisor, his help and guidance were essential throughout the research.

I am sincerely thankful to the **University of Pannonia** for providing an inspiring academic environment and for supporting my studies and professional development.

My special appreciation goes to **MOL Group**, whose collaboration made it possible to test the developed methods in real industrial environments. Their openness, trust, and professional support significantly strengthened the practical relevance and impact of this work.

Finally, I wish to acknowledge the support, patience, and encouragement I received outside the academic environment, which provided strength and motivation during the most demanding stages of this journey.

Contents

Abstract i

Kivonat ii

Zusammenfassung iii

Acknowledgment iv

List of Abbreviations ix

1 Introduction 1

1.1 Research background and motivation 1

1.2 Scope and structure of the thesis 2

2 Moving towards digital twins in the process industry 5

2.1 Chapter motivation and scope 5

2.2 Challenges of digital twin adoption in the process industry 7

2.2.1 Business value, trust, and organizational readiness 10

2.2.2 Integration, industrialization, and standardization aspects . . 14

2.2.3 Digital twin life cycle and model maintenance 17

2.3 Industrial case study: A phased pathway toward digital twins 29

2.3.1 Identifying a suitable use case 30

2.3.2 Phase I – Digital model 32

2.3.3 Phase II – Digital shadow 35

2.3.4 Phase III – Digital twin 37
2.3.5 Key learning points and future development areas 40
2.4 Conclusion 41

3 Surrogate-based calibration of flowsheet models 42
3.1 Chapter motivation and scope 42
3.2 Calibration and maintenance of flowsheet models in DT applications . 44
3.2.1 Challenges of flowsheet model calibration 44
3.2.2 Surrogate modeling as an enabler for scalable calibration . . . 46
3.3 Methodology for adaptive flowsheet model calibration 48
3.3.1 Surrogate models 49
3.3.2 Objective function 52
3.3.3 Optimizer 55
3.3.4 Methodology overview 57
3.4 Case study for surrogate-based flowsheet model calibration 58
3.4.1 Synthetic test data generation 61
3.4.2 Surrogate and partial inverse model training 62
3.4.3 Flowsheet model calibration 69
3.4.4 Surrogate-based flowsheet model calibration results 71
3.5 Summary and outlook 75

4 Online learning for incremental surrogate model maintenance 77
4.1 Chapter motivation and scope 77
4.2 Surrogate model sampling strategies for evolving processes 78
4.2.1 Offline surrogate sampling strategies 79
4.2.2 Limitations under evolving operating conditions 80
4.2.3 Motivation for online surrogate model updates 81
4.3 Online learning methodologies 83
4.3.1 Benefits and challenges of online learning 84

4.3.2 Types of online learning strategies 85
4.3.3 Performance Metrics 89
4.4 Case study for evaluating online learning 90
4.4.1 Analyzed retraining options 92
4.4.2 Results and discussion 94
4.5 Conclusion 103

5 Integrated framework for adaptive flowsheet model maintenance 104
5.1 Chapter motivation and scope 104
5.2 Online learning-supported methodology 106
5.2.1 Offline surrogate initialization 108
5.2.2 Flowsheet calibration 108
5.2.3 Incremental surrogate refinement 109
5.3 Case study: Adaptive maintenance of a heat exchanger network model 110
5.3.1 Initial surrogate training 111
5.3.2 Flowsheet calibration with online learning 114
5.3.3 Results 116
5.4 Industrial case studies: Continuous calibration of refinery models . . 122
5.4.1 Continuous calibration of a refinery sour water stripper unit . 122
5.4.2 Continuous calibration of a refinery naphtha redistillation unit 127
5.5 Discussion 134

6 Prototype software implementation and demonstrator platform 136
6.1 Objective and design principles 136
6.2 Flowsheet model connection 137
6.3 Dataset generation and sampling strategy 138
6.4 Surrogate model training and validation 139
6.5 Surrogate-based flowsheet calibration 140
6.6 Discussion and practical implications 140

7 Conclusions **143**

8 Thesis points and contributions **146**

Thesis points 146

Industrial Contributions 149

Scientific contributions 150

Appendices **153**

A Appendix to Chapter 2 **154**

A.1 General reference architecture for DT 154

A.2 Standards for model integration 155

B Appendix to Chapter 3 **156**

B.1 Case study: surrogate and partial inverse model parameters 156

B.2 Case study: calibration methodology comparison 157

B.3 Case study: calibration parameter estimation accuracy 157

B.4 Case study: calibration methodology comparison under noisy conditions 158

C Appendix to Chapter 4 **159**

C.1 Threshold selection for conditional retraining 159

D Appendix to Chapter 5 **160**

D.1 Performance improvement with online learning for FNN-low model . . 160

D.2 Pseudocode of surrogate-based flowsheet calibration with online learning 161

D.3 SWS industrial case – input–output correlation matrix 162

D.4 NRED industrial case – input–output correlation matrix 163

List of Abbreviations

AI	Artificial Intelligence
AIC	Akaike Information Criterion
ALAMO	Automatic Learning of Algebraic MOdels
ANN	Artificial Neural Network
APC	Advanced Process Control
API	Application Programming Interface
B2MML	Business to Manufacturing Markup Language
BIC	Bayesian Information Criterion
BNN	Bayesian Neural Network
CAPE-OPEN	Computer Aided Process Engineering – Open Interface
CI/CD	Continuous Integration and Continuous Deployment
CMSD	Core Manufacturing Simulation Data standard
CRISP-DM	Cross-Industry Standard Process for Data Mining
DCS	Distributed Control System
DataOps	Data Operations
DevOps	Development and Operations
DT	Digital Twin
DTE	Digital Twin Environment
ERP	Enterprise Resource Planning
EWC	Elastic Weight Consolidation
FDS	Fire Dynamics Simulator
FNN	Feedforward Neural Network
GCR	Gradient Coreset Replay
HLA	High-Level Architecture
IIoT	Industrial Internet of Things
LHS	Latin Hypercube Sampling
LNI	Light Naphtha Isomerization
LP	Linear Programming
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
MASE	Mean Absolute Scaled Error
MC	Monte Carlo
MES	Manufacturing Execution System
ML	Machine Learning
MLOps	Machine Learning Operations

MSE	Mean Squared Error
N2A	Naphthenes + 2×Aromatics (feed quality index)
NRED	Naphtha REDistillation unit
NRMSE	Normalized Root Mean Squared Error
OAGIS	Open Applications Group Integration Specification
OntoCAPE	Ontology for Computer Aided Process Engineering
OTS	Operator Training Simulator
PFA	Portable Format for Analytics
PMML	Predictive Model Markup Language
PoC	Proof of Concept
PSO	Particle Swarm Optimization
QIF	Quality Information Framework
RAMI 4.0	Reference Architectural Model Industrie 4.0
RBF	Radial Basis Function
ReLU	Rectified Linear Unit
RMSE	Root Mean Squared Error
RTO	Real-Time Optimization
SCR	Supervised Contrastive Replay
SimOps	Simulation Operations
SSA	Smart Sampling Algorithm
SWS	Sour Water Stripper
TOPSIS	Technique for Order Preference by Similarity to Ideal Solution
UoM	Unit of Measure
VV&A	Verification, Validation and Accreditation
XAI	Explainable Artificial Intelligence

Chapter 1

Introduction

1.1 Research background and motivation

The process industry is under increasing pressure to improve efficiency, flexibility, and sustainability due to volatile economic conditions, stricter environmental constraints, and safety regulations. Traditional model-based optimization and decision-support approaches, which are typically static and offline, struggle to cope with continuously changing operating conditions and long-term system degradation. As a result, digital transformation has become a strategic necessity, aiming to enable faster, data-driven, and more reliable decision-making across the full life cycle of industrial assets.

Within this transformation, Digital Twins (DTs) have gained prominence as a means to couple physical systems with their digital representations and to support operational decision-making. While such concepts have been rapidly adopted in manufacturing environments, their penetration into large-scale continuous process industries remains limited. This discrepancy is not rooted in a lack of modeling capability; on the contrary, the process industry has long relied on high-fidelity flowsheet simulation models for design, analysis, and optimization. Instead, the challenge lies in operationalizing these models as continuously reliable digital counterparts of evolving industrial systems. Flowsheet models offer a strong physical foundation,

capturing thermodynamic, kinetic, and transport phenomena in a transparent and interpretable manner. In principle, they provide an ideal backbone for DTs in safety-critical and economically sensitive applications. In practice, however, their industrial deployment is hindered by high computational cost, limited scalability, and the need for frequent recalibration. As real processes change due to fouling, equipment aging, or shifting feedstock characteristics, model parameters gradually drift, leading to a deterioration in predictive accuracy. Manual recalibration is often infeasible in operational settings, particularly when models are embedded in real-time or near-real-time decision workflows.

These limitations become especially critical when model predictions are directly linked to business processes such as production planning, scheduling, or operational optimization. In such contexts, declining model fidelity not only reduces the value of digitalization efforts but also erodes trust in model-based decision support. Consequently, despite their theoretical suitability, flowsheet models are rarely utilized as active, continuously maintained components of industrial DT systems.

This gap between the potential of flowsheet-based DTs and their practical applicability constitutes the central motivation of this research. The thesis addresses the need for methodologies that enable flowsheet models to remain accurate and computationally tractable over extended operational periods. By integrating surrogate modeling and online learning concepts with physics-based process models, the research aims to support adaptive, robust, and industrially viable DT implementations for the process industry.

1.2 Scope and structure of the thesis

The primary objective of this thesis is to develop methodological foundations and practical solutions that ensure the long-term reliability and industrial applicability of flowsheet-based DTs supported by machine learning. The research focuses on bridging

high-fidelity physics-based process models and adaptive data-driven techniques, with particular emphasis on surrogate modeling, automated calibration, and online learning.

The thesis begins by identifying and systematizing the key technological, organizational, and model maintenance challenges that limit the deployment and sustainable operation of DTs in the process industry. These challenges are analyzed from the perspectives of model integrability, life cycle management, and transparency, highlighting how principles from DevOps, MLOps, and model engineering can support more robust and reusable industrial modeling workflows.

Following this conceptual foundation, the thesis introduces a surrogate-based, multi-objective calibration methodology for flowsheet models. By replacing repeated high-cost process simulations with global surrogate models trained on flowsheet data, the proposed approach significantly reduces computational effort while maintaining calibration robustness under noisy and uncertain measurement conditions.

Subsequently, an online learning framework is developed to enable the continuous, incremental adaptation of surrogate models as new operating points become available. The proposed online learning strategy balances fast local adaptation with the preservation of previously acquired global knowledge, ensuring stable long-term predictive performance in non-stationary industrial environments.

Building on these methodological contributions, an additional chapter presents a Python-based framework that integrates surrogate model construction, automated flowsheet calibration, and online learning into a transparent and reproducible workflow. The framework serves as a reference implementation of the developed concepts, emphasizing modularity, traceability, and interpretability. By exposing model assumptions, data flows, and learning steps explicitly, the framework demonstrates how the proposed methods can be operationalized and systematically evaluated in practice, thereby supporting reproducibility and industrial adoption.

The thesis concludes with a summary of the main scientific contributions and

practical implications, followed by a discussion of limitations and future research directions. The concluding chapter outlines how the presented methods and the supporting framework can be extended toward fully operational, continuously learning DT systems for real-time decision support in the process industry.

Chapter 2

Moving towards digital twins in the process industry

This chapter synthesizes and critically evaluates the scientific literature related to flowsheet model-based DTs in the process industry. The review focuses on the key limitations that hinder industrial adoption, including challenges in system integration, model transparency, and long-term model life cycle management. By structuring the existing work along these dimensions, the chapter establishes the conceptual foundation for the next chapters contributions, particularly the development of surrogate-based calibration and online learning methodologies that address model degradation and adaptability in operational DT environments.

2.1 Chapter motivation and scope

The process industry is currently experiencing an accelerated digital transformation driven by the increasing availability of data, advances in computational capabilities, and the convergence of physical process modeling with data-driven methods. Digitalization has become a foundational element of Industry 4.0, where technologies such as the Industrial Internet of Things (IIoT), artificial intelligence, advanced process simulation, cognitive computing, and cloud-based infrastructures enable tighter

integration between physical assets and digital systems [1, 2]. These technologies facilitate improved data accessibility, enhanced system-level visibility, and faster, more informed decision-making across the entire production life cycle.

Within this digital transformation, the concept of the DT has emerged as a key paradigm for representing, analyzing, and optimizing complex industrial systems. Although the formal definition of the DT concept is commonly attributed to Grieves [3], who introduced it in the context of product life cycle management, the underlying principles of synchronized physical–digital representations have been applied in industrial practice for decades. Early examples include high-fidelity simulation environments used for mission-critical decision support, such as those employed during the Apollo 13 mission [4]. Current DT implementations build upon these foundations by emphasizing continuous data exchange, automated model updates, and bidirectional interaction between physical systems and their digital counterparts.

According to the ISO 23247 standard [5], a DT is defined as a digital representation of an observable manufacturing element that acquires data through device communication and uses this information to update its internal models. In industrial practice, DT implementations typically emphasize continuous data exchange and the alignment of physical and digital states. In the process industry, this is most often realized through flowsheet-based simulation models that represent the underlying physical and chemical phenomena of the process.

Despite their conceptual maturity and demonstrated potential, DTs are significantly less prevalent in the process industry than in discrete manufacturing. This gap is particularly notable given the economic scale of the process industry and the potential value of DT-supported optimization and decision-making. One contributing factor is the conservative nature of process industry operations, where safety, reliability, and regulatory compliance impose strict constraints on the adoption of novel digital technologies, despite the long-standing and successful use of model-based solutions such as advanced process control (APC) and inferential calculations. In

addition, the complexity and scale of process systems pose substantial challenges for model development, integration, and long-term maintenance.

As a result, the practical deployment of DTs in the process industry raises a number of unresolved questions related to system integration, model consistency, transparency, and sustainability over time. These challenges extend beyond individual modeling approaches and instead reflect broader issues in how digital representations are developed, validated, operated, and maintained throughout their life cycle.

This chapter examines the scientific and industrial literature that addresses the transition toward DT solutions in the process industry, with particular emphasis on the technical, organizational, and methodological challenges that accompany their adoption. It reviews existing definitions, architectures, standards, and implementation strategies, and critically discusses the limitations that currently hinder widespread adoption. By synthesizing these perspectives, the chapter establishes a structured understanding of the state of the art and identifies the key challenges that motivate the methodological developments presented in the subsequent chapters of this thesis.

2.2 Challenges of digital twin adoption in the process industry

To establish a systematic foundation for the literature review, a structured search was conducted using the Scopus database, focusing on peer-reviewed journal and conference publications published up to 2024. The objective of the search was to identify how DT concepts intersect with process industry applications, with particular attention to aspects related to model life cycle management and the broader digitalization of chemical engineering systems.

Multiple search strings and their combinations were evaluated to appropriately delimit the scope of the review, with queries restricted to titles, abstracts, and

keywords to ensure topical relevance. Table 2.1 summarizes the applied search queries and the corresponding number of retrieved publications, providing a transparent overview of the literature coverage and serving as the basis for identifying dominant research directions and gaps.

Table 2.1: Overview of Scopus search results.

Search String	Results
"digital twin"	29,105
"mlops"	476
"digital twin" AND ("mlops" OR "devops")	45
"digital twin" AND (LIMIT-TO (SUBJAREA, "CENG"))	1,346
"mlops" AND (LIMIT-TO (SUBJAREA, "CENG"))	8
"digital twin" AND ("mlops" OR "devops") AND (LIMIT-TO (SUBJAREA, "CENG"))	15
"digital twin" AND "model life cycle"	3
"surrogate model" AND ("mlops" OR "devops")	1
"digital twin" AND "model maintenance"	11

The analysis highlights several thematic clusters, including the integration of DT with DevOps practices, machine learning, and Industry 4.0 frameworks. These patterns reflect the interdisciplinary nature of DT research and its growing relevance in industrial contexts.

Despite the broad interest, most studies focus on developing digital models, however their integration into processes is often out of scope [6]. Within the domain of process control, DTs play a pivotal role, showcasing their versatility in the development of soft sensors and the seamless maintenance of process control systems, as demonstrated by Rui He and colleagues [7]. Additionally, DTs emerge as potent tools for decision-making support, offering predictive models that fortify informed choices. Their dynamic capabilities extend to risk mitigation and management, as highlighted by West et al. [8]. In the world of process and production optimization, DTs excel in orchestrating processes to maximize revenue while minimizing costs, with Duchene et al. [9] illustrating this profit optimization potential. In predictive maintenance, DTs possess a unique ability to detect irregular patterns, as highlighted by Desai et al. [10]. Virtual evaluation and verification represent another key

application domain for DTs, serving as robust verification mechanisms to assess technology development options [11]. This role extends further to actively mitigating failures during the development of new products or processes, as detailed by Sihan Wang et al. [12]. Furthermore, DTs contribute to improved collaboration by promoting transparency and understanding of models, fostering synergy between cross-functional teams and collaborators, as investigated by Zhang and colleagues [13]. The burgeoning adoption of various DTs underpins the pressing need for a robust methodology to guide their development and management throughout their entire life cycle [14].

Based on the definition of the ISO 23247 standard [5], DTs serve as synchronized digital representations of physical systems, bridging the gap between production systems and processes. This linkage allows dynamically collected data to be utilized for real-time monitoring, analysis of system behavior, asset health, and decision-making. By synchronizing digital and physical systems, either online or offline, production systems can be continuously optimized and monitored.

In the refining sector of the process industry, flowsheet models play a pivotal role in preliminary analysis, conducting "what if" scenario analyses, optimizing processes, and monitoring equipment health. These software tools enable the creation of process simulation models based on mathematical representations of physico-chemical processes. Widely used across various engineering fields, such as chemical engineering and supply chain management, these models offer cost-effective solutions to support process optimization, control, and design tasks. However, while process simulation models lack real-time connections to physical counterparts, they have been instrumental in technology development and Operator Training Systems (OTS) for decades. While simulation models focus more on design and analysis before implementation, DTs provide real-time monitoring and analysis capabilities. Therefore, leveraging the existing flowsheet process model can serve as a solid foundation for developing DTs in the refining sector. However, the complexity of these models and their potential for wider automated usage pose challenges for their adoption.

Adoption of DTs similar to other complex decision support systems has generally been hindered by a significant challenge, which is the lack of trust among stakeholders [15]. On one side, it does have crucial importance to identify appropriate pilot areas or projects for the validation of DT's benefit; on the other hand, the possibility of being able to "play" with the proposed solution even in the development phase is equally important to build understanding and trust. Once the benefits are validated, integrating DTs into existing process information systems and aligning them with relevant business processes is the next level of challenge. Proper life cycle management is the key to attaining transparency and credibility in the developed system to realize the vital added value.

In the subsequent sections, the aforementioned challenges are summarized and an overview of potential methodologies originating from software and ML development and model engineering areas is provided, aiming to offer a high-level understanding without delving into excessive detail.

2.2.1 Business value, trust, and organizational readiness

The process industry is in an early stage of adoption of DTs compared to the manufacturing sector [16]. Digitalization developments often follow a bottom-up approach, leading to limited implementation of isolated and problem-focused applications. For instance, there is a notable emphasis on applications like equipment or process monitoring. However, this approach tends to prioritize addressing burning problems rather than considering broader, industry-wide solutions [17]. Consequently, this mindset can inadvertently constrain the broader potential of DTs. To overcome these challenges and fully exploit the potential of DTs, a comprehensive understanding of DT solutions with the associated opportunities and challenges is crucial.

The list of challenges is opened by the quantification and validation of financial benefits. Implementing DTs requires notable investments in infrastructure and commitment toward a digitalization platform. To justify these costs, organizations must

identify sound, understandable, and nevertheless measurable business benefits [18]. On top of this, the lack of trust prevails in any digitalization solution regarding concerns about information leakage due to the high-level system integration. Therefore, building trust and ensuring that data security measures are in place can address these concerns and increase confidence in DT solutions. Another challenge lies in the current use of flowsheet simulation and modeling tools. Although they are common in industry, they typically require special expertise to operate and maintain, thereby limiting access to a narrow group of skilled experts. Generally, missing transparency and trialability further reduce the confidence and understanding of the end users, since they only see the results without fully understanding the underlying model structures. Transitioning from isolated, unit-level or application-specific models to larger-scale digital twins amplifies this challenge, as employees may not fully recognize the additional benefits and business value of DTs.

Introducing DTs also involves the transformation of traditional working methods and organizational structures, which can trigger resistance from employees who do not realize or understand tangible and measurable benefits. Designing a DT to meet all personal requirements is impractical; there will always be individuals who either do not directly benefit from it or perceive it as a threat and resist its implementation. Such a phenomenon was observed in other technological implementations, such as the negative effects of generative AI in business [19]. To overcome this challenge, it is essential to bridge the gap between digitalization tools and the highly skilled workforce. Embossing the intellectual aspects of human intervention, as envisioned in Industry 5.0 [20], can also foster acceptance and cooperation.

Successful deployment and adaptation of DT solutions require a carefully designed strategy to integrate them effectively into business processes. Similar to the adoption of machine learning (ML) solutions, the deployment process typically progresses through several stages of organizational readiness for AI technologies [21]. Drawing on the work of Parrott et al. [22], the adoption methodology presented in Figure 2.1

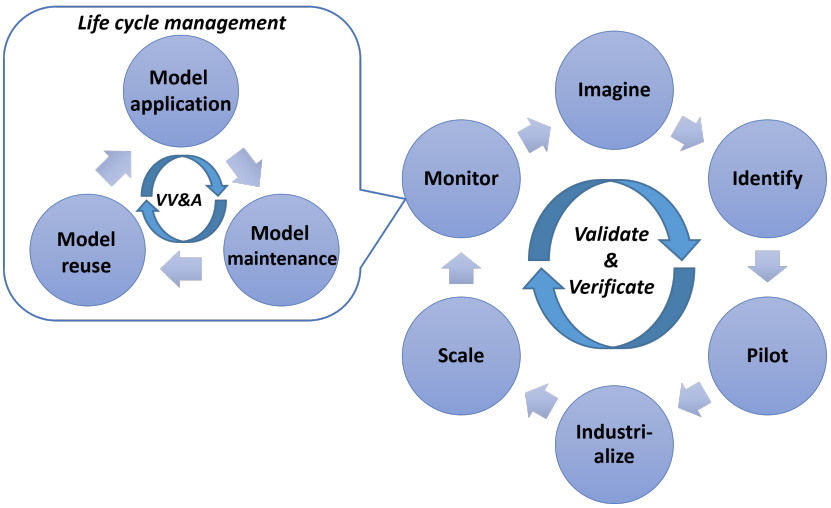


Figure 2.1: DT development steps and AI readiness combined with model life cycle management (VV&A: Verification, Validation, and Accreditation).

outlines the key steps for adapting DT solutions. This process emphasizes that adaptation should not cease after scaling up, and proper life cycle management is crucial to maintaining transparency and trust in the system.

The initial step is the tactical phase, aligned with the Machine Learning Operations (MLOps) methodology, which involves mapping potential use cases and identifying key processes [23]. Pilot programs are conducted to test and validate concepts, allowing for a better understanding of the challenges and benefits of the DT platform and the identification of optimal use cases. This phase focuses primarily on learning and gaining insight into how DTs can generate value. The exhibits in Section 2.3 also highlight the importance of this learning phase.

Once pilot programs confirm the benefits of a DT, the transition can be made from silo systems to integrated systems. This strategic phase involves integrating and extending the solutions developed into industrial systems, driven by business objectives. Continuous monitoring of the DT is crucial during this phase, as it provides information on the benefits of the DT application, validates assumptions, and helps identify the causes of process inefficiencies. These insights, in turn, contribute to the transformation of business processes.

During the development of a DT, it is important to define the appropriate balance

between model complexity and application scope. An oversimplified representation of the physical asset may fail to deliver the expected value of the DT, while an excessively complex model with extensive data and integration requirements can negatively affect usability, maintainability, and transparency. Determining the proper level of complexity is therefore essential not only during the initial implementation phase but throughout the entire DT life cycle, including deployment, operational monitoring, and continuous maintenance (Section 2.2.3).

Continuous evaluation of project deliverables and the application of project management methodologies originating from data science and software engineering, such as CRISP-DM, Scrum, or Kanban, can help ensure that DT development and operation remain aligned with business objectives [24]. However, the life cycle of a DT does not end with deployment. During the monitoring and operational phase, the delivered solution must be continuously evaluated to ensure that it is properly used, maintains its expected performance, and continues to address the original industrial problem that motivated its development. Since industrial processes, operational practices, and business objectives evolve over time, the intended purpose and application domain of the DT may also change. Operational usage frequently reveals new opportunities for model reuse, extension, or integration into additional decision-support workflows that were not foreseeable during the initial design phase.

This operational phase is closely connected to the Verification, Validation, and Accreditation (VV&A) cycle illustrated in Figure 2.1. Verification ensures that the implemented DT model is correctly constructed according to its specifications and mathematical formulation. Validation evaluates whether the DT adequately represents the real physical process within the intended operating conditions and accuracy requirements. Accreditation determines whether the DT is sufficiently reliable and trustworthy for its intended industrial application and decision-support role. In practice, these activities are not performed only once during development but must be continuously revisited during operation as new process data, operating conditions,

and system modifications emerge. Monitoring model performance, prediction errors, data quality, and user interaction therefore becomes an essential component of DT life cycle management.

Early identification and involvement of potential users through pilot programs and iterative deployment phases are also important. Gathering feedback from users enables trade-offs in system integration that improve usability and operator acceptance, even if this requires sacrificing some portion of theoretical benefits. If the system is not user-friendly, the full potential of a DT cannot be realized. To address these barriers, it is important to educate workers and managers about DT capabilities, limitations, and implementation strategies. Furthermore, defining access levels, responsibilities, and governance procedures during both implementation and operational phases is critical for ensuring transparency, traceability, and long-term maintainability of the DT system.

2.2.2 Integration, industrialization, and standardization aspects

An essential aspect of DT deployment is the seamless integration of DTs with existing business processes and enterprise systems, as illustrated in Figure 2.2. Proof of Concepts (PoCs) play a vital role in identifying potential use cases and processes that influence the integration requirements and expectations for business processes and information systems. However, it is not only the connection between physical and digital systems that is critical; equally important are the relationships between the various components within the DT system, which form an interdisciplinary field characterized by enormous complexity [25].

These relationships encompass connections between model information databases and relevant processes, as well as the interconnection between various models. DT models typically comprise diverse types of models, including mechanistic, flowsheet, or ML models. Each modeling tool has its strengths and weaknesses, depending

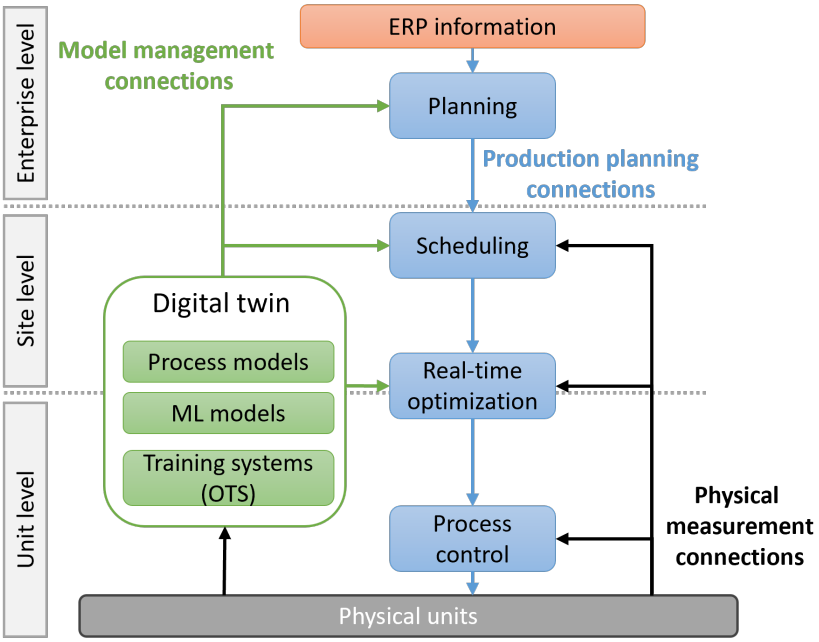


Figure 2.2: Example of DT integration into a standard planning and scheduling process (ERP — Enterprise Resource Planning).

on its original operational focus. Weaknesses can be addressed by replacing the corresponding subsystems in the selected modeling tool with improved models. These enhanced components frequently originate from the operating company itself and contain valuable intellectual property related to specific processes, which may not be accessible to commercial modelers. A notable research direction explores the potential of ML models as substitutes for computationally intensive, rigorous models [26]. These ML models, referred to as surrogate models, offer the advantage of requiring less computational power while enabling real-time analysis, all while maintaining prediction accuracy. However, ML models also have limitations, particularly their increasing uncertainty beyond the boundary of the training dataset. Consequently, the integration of mechanistic and ML knowledge, known as "hybrid science-guided machine learning" solutions, has gained attention as a promising research direction [27]. Such hybrid models could serve as a key driver for DT model development, as they can attain a balance between interpretability and fitting to real data [17].

Another aspect of integration is the management and connection of sensor or laboratory data. This is essential to implement feedback to the control system

and also a challenge in the case of having bidirectional closed-loop connections [28]. Data sources can vary in structure, ranging from unstructured to semi-structured and structured. Therefore, a data integration platform most probably, in any case, necessitates bringing the data from different sources into a format compatible with the DT. This practically comprises prefiltering, standardizing, and making it available for the DT coherently and consistently [29]. Furthermore, identifying data hosts and dependencies is crucial for model life cycle management and overall DT performance [30].

For addressing the challenge of integration requires adopting standardized methods, the CAPE-OPEN project paved the way for improving the integration of simulation tools throughout the DT development, initiated in 1997 with European Union funding. The result of this endeavor is the CAPE-OPEN standard, now commonly adopted by most simulation tools in the field of process industry [31]. CAPE-OPEN encompasses a set of specifications that extend the capabilities of process simulation technologies. These specifications define a software interface, enabling seamless inter-operation between a specific process modeling environment and third-party process modeling components. Although proprietary thermodynamics and individual equipment models can be incorporated into simulation models using CAPE-OPEN, it does not fully address the challenge of integrating heterogeneous model structures, where models are typically developed for individual equipment, plants, or processes. To tackle this issue, a comprehensive framework is required to establish connections between different disciplines, which encompasses various tools, simulation models, design systems, and data modeling solutions.

For manually linking different types of models can be a laborious and error-prone process, making it time-consuming and costly to manage; standardization plays an essential role in extending, reusing, and ensuring interoperability among DT models [32]. Over the past few decades, several model-related standards have been developed. High-Level Architecture (HLA) standards, developed by the Simulation

Interoperability Standards Organization, define model templates, interface specifications, and data standards that facilitate the interconnection and interoperability of models [33]. The extended modeling and simulation framework establishes standards for web-based modeling and simulation. Another widely adopted solution for standard information exchange between models is AutomationML [34]. Major cloud providers such as Microsoft's AzureDT and Amazon's AWS IoT TwinMaker already offer standard solutions for model integration. Open-source solutions for model integration, such as Salome, MyGrid, and Taverna, are also available.

In the manufacturing industry, the established standards offer a reference architecture (Appendix A.1) for DT integration [35]. Moreover, standardized solutions have been developed to facilitate data exchange between models and the physical model system, organized along ISO 95 levels as summarized in Appendix A.2. These standards support seamless integration and interoperability within DT systems.

2.2.3 Digital twin life cycle and model maintenance

DT solutions rely on digital representations of physical systems in the form of models. These models may be mechanistic, data-driven, or hybrid in nature; however, regardless of their formulation, effective life cycle management remains a critical challenge during both the development and operational phases [36]. The construction of a virtual representation that faithfully reflects the physical system often leads to DT implementations that are overly complex or contain redundant information, increasing the effort and cost associated with long-term maintenance. Addressing these challenges requires the early identification of critical modeling assumptions and data dependencies, continuous validation and verification, and systematic performance monitoring throughout the operational phase, as conceptually illustrated in Figure 2.1.

To further investigate how life cycle management is addressed in the existing literature, the refined body of publications identified during the structured review

was analyzed. Approximately 45 publications from engineering-focused domains were selected for detailed qualitative analysis, with particular relevance to DT life cycle considerations and their integration with operational frameworks. To uncover dominant research themes and conceptual linkages, a keyword co-occurrence analysis was performed using VOSviewer [37]. The resulting network, shown in Figure 2.3, provides an aggregated view of how DT-related concepts are positioned within the broader research landscape.

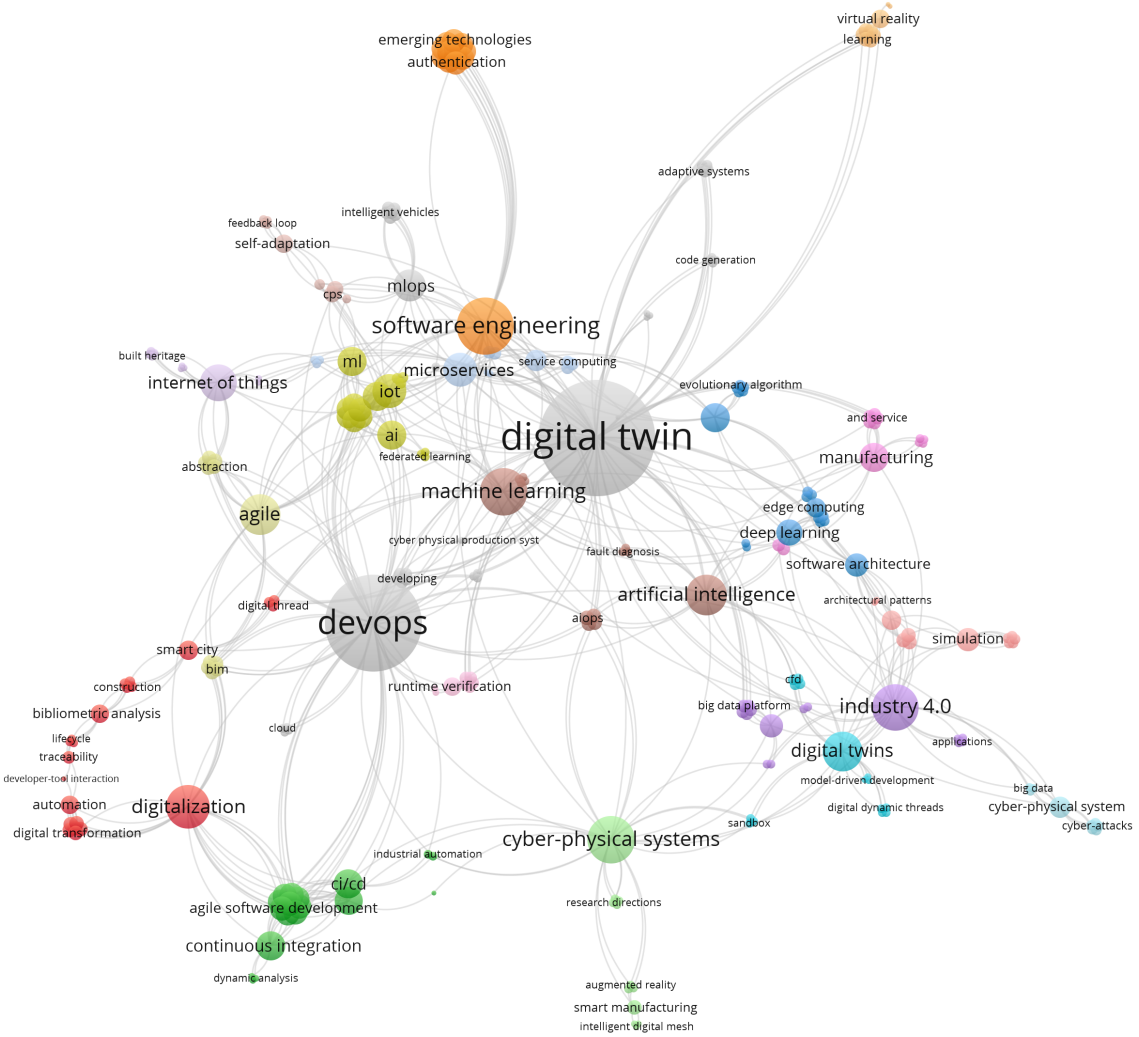


Figure 2.3: Keyword co-occurrence network derived from the literature review on DTs and related domains using VOSviewer. Node size represents the relative frequency of each keyword, while colors indicate the identified thematic clusters. Connections between nodes represent keyword co-occurrence relationships.

The visualization (Figure 2.3) reveals strong thematic associations between DTs and domains such as predictive maintenance, real-time analytics, model deployment, and Industry 4.0 integration. These connections indicate that DT life cycle management is rarely treated as an isolated modeling problem; instead, it is embedded within a broader ecosystem of data management, automation, and operational integration. In particular, the frequent co-occurrence of DT-related keywords with machine learning, cyber-physical systems, and industrial digitalization frameworks highlights the interdisciplinary nature of sustainable DT operation.

Given this complexity, several studies emphasize the importance of adopting structured life cycle methodologies originating from software engineering and data-driven system development. Approaches such as DevOps, MLOps, DataOps, and SimOps provide complementary perspectives for managing different aspects of DT systems across their life cycle. DevOps supports the continuous development, deployment, and quality assurance of DT frameworks, while MLOps focuses on the development and performance management of machine learning components embedded within DTs [30]. DataOps introduces process-oriented practices to ensure data quality and traceability, and SimOps [38] extends similar principles to the maintenance and evolution of simulation models, as illustrated in Figure 2.4.

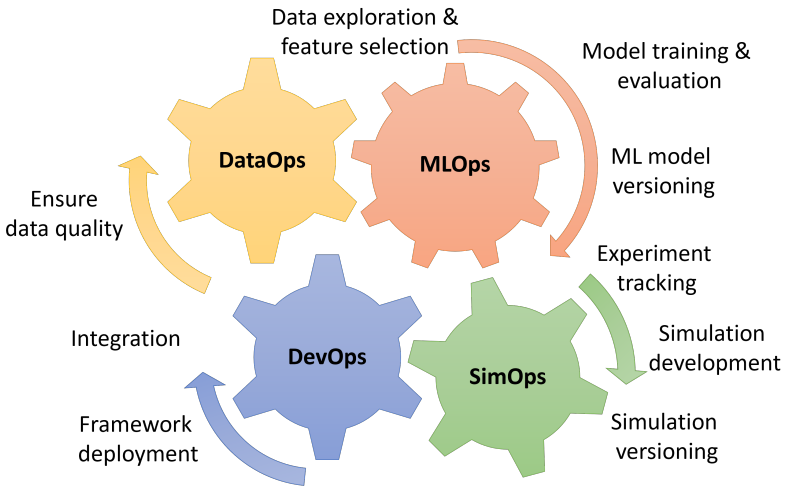


Figure 2.4: Overview of DT-related DevOps, DataOps, MLOps, and SimOps areas, including their associated responsibilities.

The heart of any DT system lies in its underlying model, underpinning the paramount importance of its life cycle management. As highlighted by Zhang et al. [39], various significant challenges emerge in this domain. These include identifying the optimal complexity, managing dependencies, implementing version control, monitoring and maintaining models, ensuring model reusability and composability, and fostering multidisciplinary collaboration and accessibility.

Complexity trade-off

The current approach to DT development often places excessive emphasis on achieving an identical representation of the physical system, driven by a misinterpretation of the term "twin". This false mental fixation on absolute accuracy leads to the development of highly complex DT models, where even minor improvements in accuracy result in a significant increase in model complexity. However, such high-complexity models introduce numerous challenges throughout their life cycle, including higher computational and resource costs, difficulties in model reuse and maintenance, and uncertainties arising from their transparency and dependencies. The model engineering study by Zhang et al. [13] proposes applying measurements to key phases of model development based on the associated costs. Those metrics can be utilized to optimize the short- and long-term impact of model development. The key factor in selecting the optimal model complexity is the accurate definition of requirements and their continuous validation and verification throughout model development and later in the model life cycle. In addition to using existing project management methodologies—as discussed in Section 2.2.1—to define modeling goals and translate business objectives into technical requirements, these methodologies and their metrics can also be applied to ensure proper requirement alignment. The evaluation metrics should consider various factors that influence the risks and costs associated with model development throughout its life cycle. These factors include the level of standardization, automation, complexity, reusability, model accuracy, and credibility.

The ultimate goal should be to optimize the complexity and minimize the costs of the model. To achieve this, techniques such as fuzzy-AHP [40] or TOPSIS [41], model verification, validation, and uncertainty quantification [42], Akaike (AIC) or Bayesian (BIC) information criterion [43], and multi-criteria framework [44] are available.

Dependency monitoring and version control

During the model development phase, it is typical for models to be developed under time constraints without adequate preprocessing, largely due to their complexity and limited development time [30]. This lack of thoroughness often leads to the omission of crucial steps such as separating development-test-production phases, conducting validation and verification tests between phases, and effectively managing model versions. Consequently, the model may contain redundancies and unused parts or functionalities, while the essential underlying dependencies remain hidden. These factors collectively contribute to the challenges faced in maintaining and operating the model, leading to increased costs and decreased business value derived from its utilization.

To address these issues, alongside leveraging "Ops" tool kits, incorporating ontologies presents a promising solution for detecting and managing model dependencies. In addition to well-known frameworks such as RAMI 4.0 [45], there are domain-specific ontologies such as OntoCAPE [46]. Romero et al. [47] used a hybrid deep learning and ontology-driven approach in their study, which offers several benefits in improving model documentation, transparency, and version management. By incorporating descriptive metadata of the equipment, model settings, and searchable structures, the ontology facilitates the linkage between model and equipment-level processes, maintenance data, and development-related information from ERP and MES systems. This, in turn, promotes proactive updating of the model during maintenance activities and serves as a valuable resource for decision support.

Effective management of dependencies between a model's input and output data

is needed to maintain its accuracy and reliability [30]. Changes in measurements, such as sensor failures, not only degrade the model's performance, but can also obscure dependencies, complicating the utilization of the model's results. Moreover, hidden dependencies present challenges in model maintenance over time. Thus, it is imperative to vigilantly monitor and address these dependencies to ensure the continued effectiveness of the model.

To mitigate the risks associated with dependencies, it is essential to incorporate appropriate metrics into the evaluation process. By accounting for dependencies, metrics can provide insight into the potential impact on the model's performance and guide decision-making. One approach to assess data quality and automate data processes throughout the model's life cycle is through the implementation of DataOps. DataOps applies DevOps principles to ensure data quality, offering a comprehensive set of metrics that assess various dimensions of data quality, including accuracy, completeness, consistency, timeliness, and skewness [48].

Model performance monitoring

Just like ML models, ensuring the performance and reliability of DT systems, which can rely on process simulation models, necessitates continuous testing and monitoring [49]. Given the analogous nature of simulation models and ML models, adapting elements of the MLOps methodology for simulation models can be beneficial to maintain the ongoing fitness of the DT model, as depicted in Figure 2.5. Automated tools and processes are already available for tasks such as model version management and detecting model drift during operation [50]. As maturity increases, the development, implementation, and maintenance of new models become faster and more cost-effective.

Maintenance of a model can vary depending on the level of intervention required. If maintenance is necessitated by natural performance degradation of the modeled equipment (e.g., heat-exchanger heat transfer degradation or catalyst deactivation)

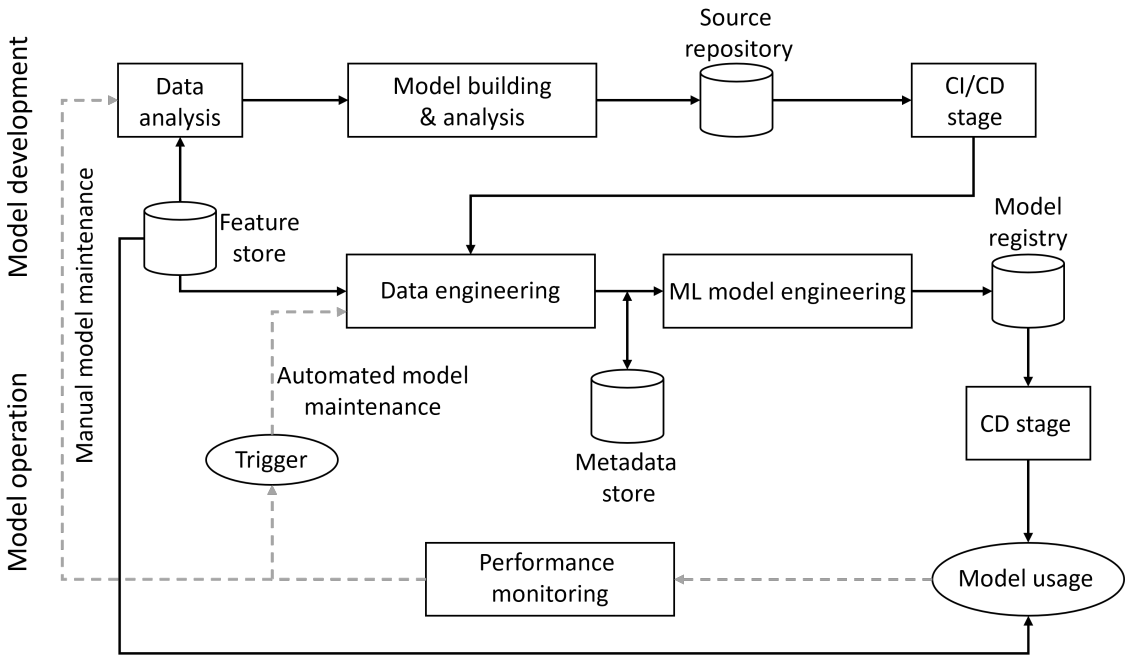


Figure 2.5: MLOps model development and operation pipeline, including CI/CD (continuous integration and continuous deployment) practices [21].

or by a gradual shift in the operating range, flowsheet simulation models can, in principle, be recalibrated without structural modification. In such cases, automated or semi-automated maintenance is theoretically feasible. Exploring the use of surrogate models to accelerate the model-fitting process has been identified as a promising direction in the literature [51] (see additional examples in Table 3.1); however, systematic and generally applicable methodologies for surrogate-supported, automated flowsheet model maintenance remain limited.

In contrast, when maintenance involves structural changes to the model—such as topology modifications or changes in governing equations—it continues to require manual intervention by domain experts, reflecting a fundamental boundary of current automation capabilities.

The quality and credibility of a DT model, in the context of its usage, are mainly reflected in the efficiency of the simulation performed with the model. Quantitative analysis, supported by measurable indicators, proves to be an effective tool for evaluating various model metrics. Bossel et al. [52] highlighted the importance

of verification, validation, and uncertainty quantification, which can be evaluated using different classes of metrics. These include quantitative methods (e.g., Bayesian estimation, Fourier analysis, correlation analysis, and wavelet analysis), qualitative approaches (e.g., the Turing test and entropy-based methods), as well as integrated techniques (e.g., analytic hierarchy process and fuzzy comprehensive evaluation). Furthermore, borrowing metrics from the broader field of modeling and simulation and data science and statistics, such as mean absolute error, mean squared error, root mean squared error, accuracy, precision, and confusion matrix, can further contribute to the assessment of model accuracy [53].

Model reusability and composability

The concept of reusing simulation models as a means to reduce time and costs in model development is an attractive proposition [54]. Although not a novel idea, the development of the high-level architecture [33] has sparked increased interest in model reuse. The possibility that modelers can save time and resources by reusing their own or others' models and model components holds significant appeal [55].

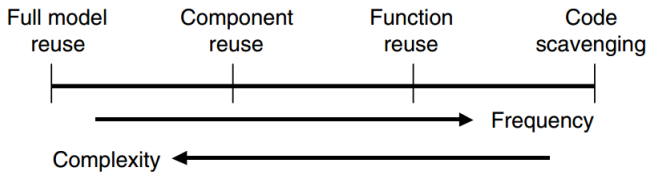


Figure 2.6: Spectrum of model reuse.

Figure 2.6 depicts the spectrum of model reuse, drawing an analogy to software development [56]. It illustrates four positions on a non-linear scale defined by two factors: frequency and complexity. On the frequency axis, reuse of code fragments is more prevalent towards the right side of the spectrum. Conversely, complexity runs in the opposite direction, suggesting that reusing code snippets is relatively straightforward, while effectively reusing entire simulation models can be highly

challenging. Achieving complete model reuse is often seen as the ultimate goal in the simulation realm, particularly when model development involves substantial costs and time investments. Full model reuse implies using a model in a different context than the one for which it had originally been developed. This raises important considerations about the validity and credibility of the model, necessitating rigorous validation.

The spectrum of reusability in the context of DTs is broad, encompassing various interpretations depending on the complexity of the models involved. Reusability can be understood in different ways. First, while it is unlikely that an entire DT model can be reused due to the inherent complexity and unique characteristics of different processes, the conceptual approach used to construct models for a particular asset type can be reused. This includes the reuse of modeling strategies, such as sharing input-output data requirements, metadata, and knowledge management related to the construction of complex models.

Second, DT models, like process simulation tools, can be decomposed into components or sub-models based on their specific functions and the processes they describe. Leveraging existing equipment models from a model database allows for the efficient construction of tailored complex models. Integration solutions such as CAPE-OPEN (Section 2.2.2) facilitate the utilization of heterogeneous models. Model libraries, supported by traditional database technology, service-oriented technology, or cloud computing, play a key role in storing and reusing model components, enabling efficient reuse of these building blocks across different models [57].

Finally, reusability can also extend to the reuse of framework components that manage the life cycle of DT models. By separating modeling functions based on complexity and process descriptions, modular construction becomes possible. This approach allows for the creation of complete process models using reusable building blocks [54], similar to the principles of object-oriented programming, which organizes design around data or objects rather than functions and logic [58]. While

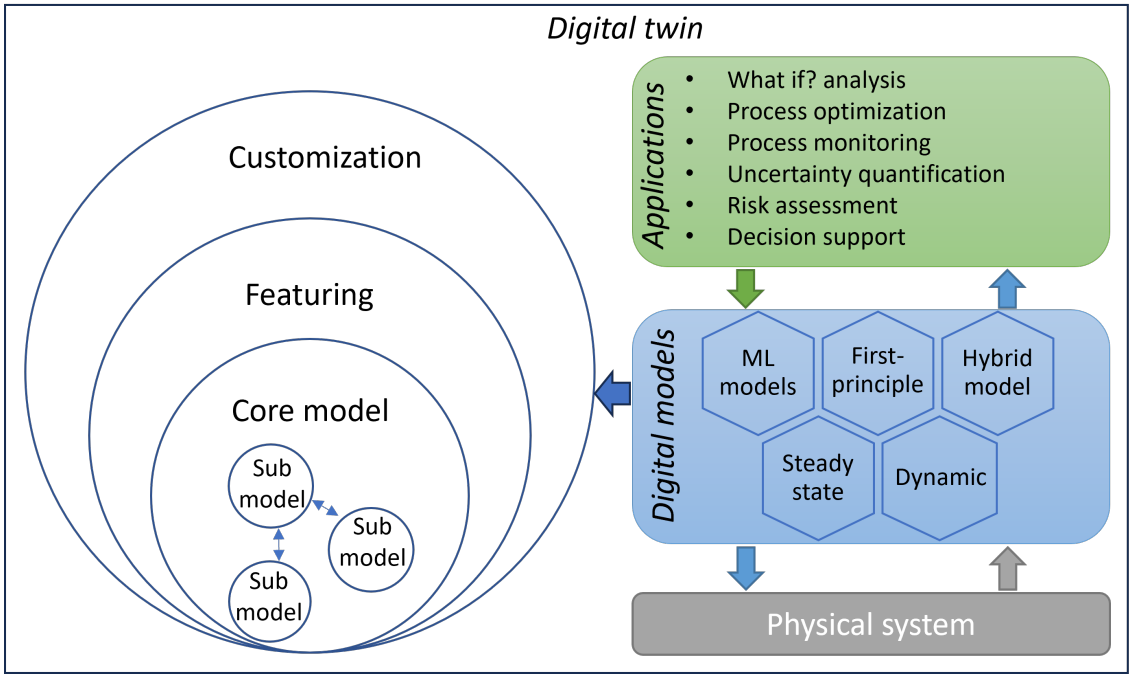


Figure 2.7: DT model modularization and framework component reuse supporting reduced complexity and multi-purpose model applications.

manual maintenance of models using expertise, customized knowledge, or individual machine-learning models can be costly, reusing framework components and adopting a modular approach helps reduce these costs and simplifies model complexity, since a single model can serve multiple purposes (Figure 2.7). However, integrating models and different layers can introduce additional complexity to the system. This modularization approach is aligned with ISO 23247 [5] and HLA standards [33], which emphasize the separation of model and application functions.

The core of the DT model is a general model or a set of sub-models, as shown in Figure 2.7. Sub-model composability has higher importance where disparate systems are composed with each other [59]. The integration of submodels is highlighted in Section 2.2.2. The base model is a raw description of the physical assets. A feature layer is built on top of the core model to compensate for the missing part of the base model (estimate missing parameters, increase the model accuracy). The customization layer transforms the model for the application task (optimization, monitoring, etc.).

Traditional ML and deep learning algorithms have generally been operated in isolation and trained to solve specific problems. As the distribution of the feature space changes, models often need to be rebuilt from scratch. Transfer learning overcomes this isolated learning paradigm by leveraging knowledge acquired from one task to solve other tasks. By adopting this approach, the model database can be simplified, and existing models can be reused to address a wider range of questions. This fosters increased reusability and efficiency in the utilization of available models [60].

Multidisciplinary collaboration and accessibility

The development of DT models similar to ML solutions presents unique challenges due to the scale and maturity of the system, necessitating a range of skills, including domain modeling, data analysis, data engineering, business analysis, and process automation [30]. However, managing fluctuations and complexities within these models is still a significant hurdle. The transition to data-driven design and ultimately the adoption of DTs requires a mindset change in simulation and modeling practices, highlighting the need for collaborative and cooperative thinking. The lack of willingness to collaborate not only jeopardizes understanding of business needs but also impedes the realization of collective benefits that could be attained in the long run [18].

Even the training of new colleagues is challenging. However, chaos can be mitigated by ensuring high-quality documentation and the use of standardized frameworks for model development. Ongoing model validation and verification processes further aid in training efforts. Similarly to MLOps solutions, a mixed team with a mutual understanding of various domain challenges can optimize the development of the model for the overall system. In some cases, breaking down the modeling tasks into specialized domains may be beneficial, particularly when research is required to develop the models (Figure 2.8).

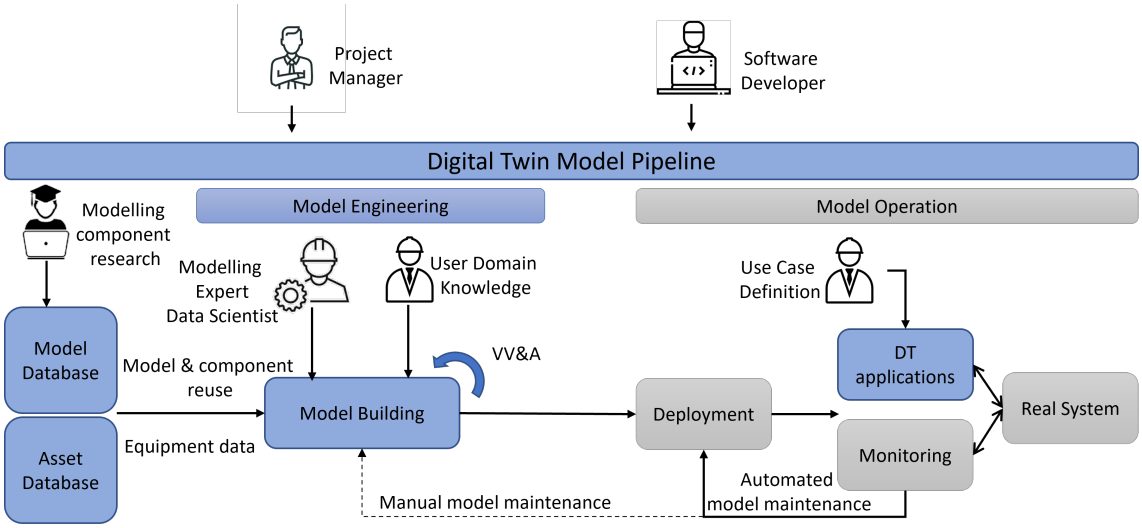


Figure 2.8: Example of DT development and operational roles, highlighting the separation of responsibilities and collaboration between specialized domains during model development and life cycle management.

Outsourcing certain aspects of model development, such as the creation of modeling strategies or the modeling of entirely new equipment, to universities can be a viable option [61]. Maintaining and developing domain-specific model databases can be separated from the modeling specialists, allowing them to leverage existing databases and create digital representations of physical systems that can be utilized by the users to obtain information [62]. On the other hand, industrial adoption necessitates user-friendly software implementations. In the realm of ML, the availability of deep learning frameworks like TensorFlow and Keras has democratized the technology, enabling novice users to adapt underlying code and apply it with ease, thereby promoting understanding and utilization of the technology [63]. Overcoming organizational barriers can be facilitated through the analogy of data models and the democratization of simulation, enabling data scientists, data engineers, and simulation specialists to share their experiences through a data-centric engineering layer.

Apart from technological advancements, empowering the existing industrial workforce to adeptly apply these methods in their daily operations is vital for augmenting productivity, efficiency, and the translation of digital solutions into tangible business

value. Long-term strategies should entail redefining university curricula to cultivate engineers who excel in data science and simulation [64]. Through comprehensive education initiatives and fostering collaborative environments, organizations can unlock the full potential of DT models and catalyze successful digital transformations.

2.3 Industrial case study: A phased pathway toward digital twins

The adoption of DT concepts in conservative industrial sectors, such as oil refining, is typically a gradual and resource-intensive process. This is primarily due to the complexity of implementing fully integrated and automated DT systems, which extends beyond technical development to encompass organizational, procedural, and cultural transformation. As discussed in the preceding sections, the lack of standardized, end-to-end DT solutions and the fragmented nature of existing modeling practices represent significant barriers to widespread industrial adoption.

Given these challenges, a phased approach to DT implementation is typically adopted in practice, as fully integrated systems rarely emerge from the outset. Incremental development allows organizations to align technical maturity with organizational readiness while managing risk and uncertainty. This staged evolution supports the gradual integration of models, data streams, and decision-support workflows, while enabling organizations to build internal expertise and confidence over time. Figure 2.9 illustrates a conceptual pathway for this transition, spanning from isolated, manually maintained digital models through intermediate digital shadow solutions toward fully integrated DT architectures. Throughout this progression, systematic verification, validation, and accreditation activities are essential to ensure technical correctness, operational relevance, and stakeholder trust, typically requiring collaboration among multidisciplinary expert roles, as outlined in Figure 2.8.

The refinery case study presented in this section serves to illustrate how the chal-

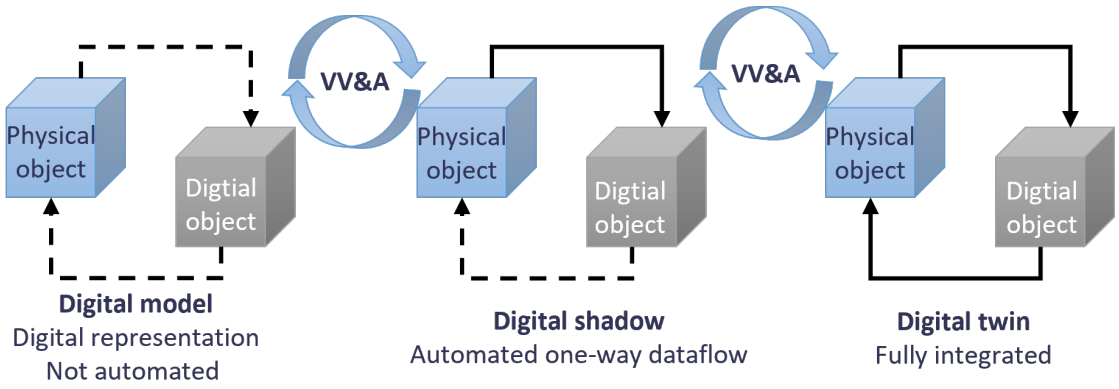


Figure 2.9: Phases of progression toward a fully integrated DT solution, from isolated digital models to integrated DT architectures supported by continuous VV&A activities.

lenges identified in the literature manifest in an industrial setting and how incremental DT adoption strategies are shaped by practical constraints. The initiative originated from limitations in automation, transparency, and maintainability within the existing modeling environment, which restricted the broader deployment of model-based decision-support tools. These limitations are particularly evident in applications such as Real-Time Optimization (RTO), where reliable and continuously updated digital representations are a prerequisite for effective operational optimization.

Due to the inherent complexity of DT systems, successful implementation requires a clearly articulated business case supported by a phased development strategy. Technical uncertainty, organizational inertia, and cultural resistance further reinforce the need for incremental deployment. Consequently, the case study emphasizes a step-by-step approach aimed at establishing a collaborative framework that connects technology, domain expertise, and operational knowledge, enabling transparent, trusted, and sustainable decision-making in DT-enabled industrial environments.

2.3.1 Identifying a suitable use case

Selecting an appropriate initial use case is critical for DT adoption in conservative, brownfield industrial environments. A suitable entry point must combine high business relevance, measurable economic impact, and sufficient technical maturity to

allow controlled experimentation without disrupting core operations. In the process industry, flowsheet-based simulation applications naturally meet these criteria and therefore represent a pragmatic starting point for DT-oriented developments.

Although flowsheet models are widely used for design, analysis, and operator training, they are typically developed and maintained in isolated contexts, with limited reuse, transparency, and life cycle management. Model updates are often manual and application-specific, which restricts scalability and long-term reliability as operational environments become more dynamic.

From a DT perspective, these limitations make flowsheet-based applications particularly suitable for early adoption: they provide established engineering credibility while exposing clear gaps in integration, automation, and governance. Evolving existing simulation models toward structured, continuously maintained digital representations offers a low-risk pathway for demonstrating DT value using familiar tools and workflows.

The PoC therefore targeted a brownfield refinery unit, where improvements could be evaluated against historical operation. The aromatic product line was selected due to its strong economic sensitivity and relevance for optimization-driven decision support. An economic objective function was formulated to maximize product-line profitability based on model outputs, utility costs, and market-priced products, subject to technological constraints.

The resulting optimization workflow is illustrated in Figure 2.10. Optimized operating conditions were returned to process engineering as actionable recommendations, enabling open-loop evaluation of technical feasibility and economic relevance.

While long-term profitability forecasts remain uncertain in complex process systems, historical operation and industrial benchmarks indicate realistic improvement ranges. Studies reported by Linde suggest efficiency and yield improvements of 1–4% for complex conversion units [65]. For example, for assets such as steam crackers, a pessimistic 1% yield improvement may still correspond to annual benefits on the order

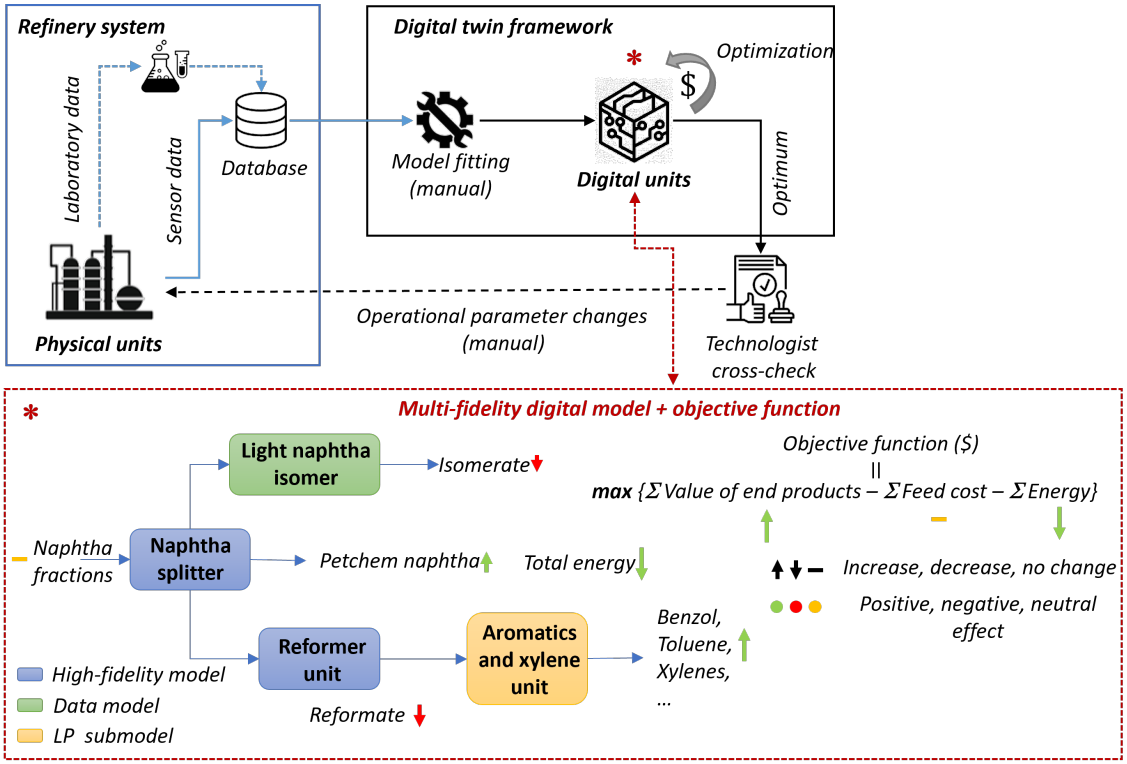


Figure 2.10: Optimization application workflow within the DT framework, including the interaction and complexity levels of the applied models. Colored arrows conceptually illustrate the relative influence of different model components on the objective function during optimization.

of 1–12 million USD, depending on plant capacity and product pricing [66, 67]. These benchmarks are reinforcing the relevance of optimization-driven DT applications as a credible entry point. This framing motivates the subsequent development of the underlying digital model.

2.3.2 Phase I – Digital model

The first implementation phase focused on establishing a robust *digital model* as the technical foundation for subsequent DT maturity stages. Given the complexity of the aromatic product line, a multi-fidelity modeling strategy (Figure 2.10) was adopted to balance development effort and decision-support value. At this stage, data handling, model execution, and evaluation were intentionally kept manual to ensure transparency and reduce technical risk.

High-fidelity steady-state flowsheet simulations were developed for the naphtha splitter and reformer units using Aspen HYSYS, where operational recommendations were expected to have the highest impact. Supporting units were represented using data-driven regression models or planning-level submodels. Model development was carried out collaboratively by modeling experts and unit technologists, ensuring both numerical robustness and technological realism.

To support integration, reuse, and version control, a dedicated Digital Twin Environment (DTE) was developed. The architecture was aligned with the ISO 23247 reference framework and provided standardized model interfaces based on HLA standard, validity bounds for inputs, and multiple model aspects to enable reuse across scenarios. As shown in Figure 2.11, the implementation focused on the core model management and simulation capabilities required for the presented PoC, while excluding full sensing and control integration.

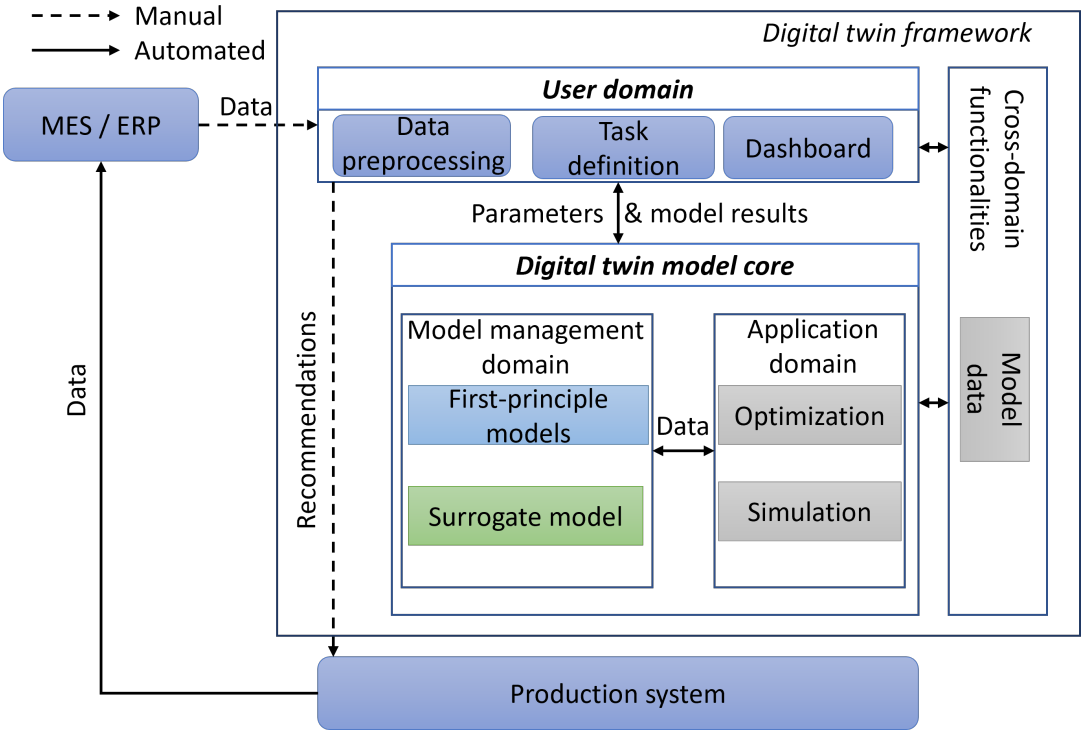


Figure 2.11: Developed DT framework for the PoC, illustrating the interaction between the application domain, model management domain, and supporting cross-domain functionalities adapted from the ISO 23247 reference architecture.

As shown in Figure 2.11, the “Model Management Domain” provides access to different modeling approaches, including first-principles process models and data-driven models such as surrogate models, enabling their reuse across multiple applications. The “Application Domain” represents the layer where these models are utilized for specific industrial use cases and decision-support tasks. The “User Domain” allows users to configure model-driven applications and analyze simulation or optimization results through dashboards and visualization tools. The “Cross-domain functionalities” represent supporting platform-level services that are not directly connected to the process model itself but are required for reliable DT operation, such as database management, communication infrastructure, cybersecurity, interoperability, and system integration services. The presented framework therefore does not represent a fully integrated DT implementation but rather a partial architecture that adopts and integrates the relevant components of the reference framework (ISO 23247) required for the given development stage and application scope.

The integrated digital model was validated in an open-loop industrial test run. A modified cut-point was recommended for the naphtha splitter to increase aromatic precursor content in the reformer feed. As shown in Figure 2.12, both the model-predicted and measured responses indicated the same directional trend for the N2A ratio, namely an increase following the proposed operational change. In addition, the measured plant data showed increased benzene and toluene production in the aromatic product streams, consistent with the expected process behavior. The magnitude of the predicted and measured N2A changes was comparable. While deviations between predicted and measured values were observed, these differences were considered acceptable for the industrial test conditions, since the digital model assumed steady-state operation whereas the unit measurements inherently included process fluctuations and measurement noise. The results therefore confirmed that the integrated multi-fidelity model was able to capture the relevant process sensitivities and provide technologically feasible operating recommendations.

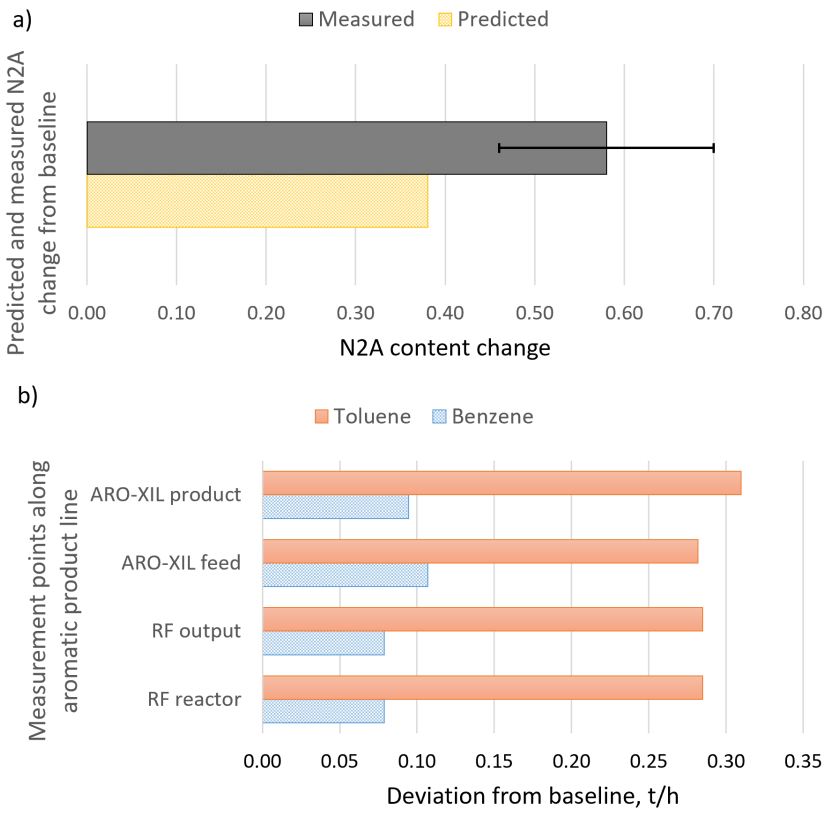


Figure 2.12: a) Model predicted and measured N2A changes (measurement reproducibility error based on the ASTM standard [68]), b) the measured benzene and toluene content changes through the aromatics product line (the deviation of the data comes from the inaccuracy of the measurements).

This phase established a credible digital representation while deliberately deferring automation, creating a stable foundation for subsequent more integrated operation.

2.3.3 Phase II – Digital shadow

The second phase advanced the system toward a *digital shadow* by automating data integration and model execution, while maintaining human-in-the-loop decision making. Real process data were continuously synchronized with the digital model, enabling systematic evaluation of current operating conditions and automated generation of optimization recommendations.

Heterogeneous data sources with different temporal resolutions were integrated through an automated preprocessing pipeline, including resampling, data quality

checks, and missing-value handling. The digital shadow focused explicitly on normal operating regimes; anomaly detection and rule-based filtering were applied to exclude invalid operating points and ensure consistency with model assumptions. The overall processing logic of the digital shadow is illustrated in Figure 2.13, which highlights the key stages of data ingestion, data validation, model validation, and task execution. At each stage, the validity of the data and the applicability of the model are explicitly assessed, while the optimization step quantifies the achievable performance improvement. Additional metadata associated with each step ensure transparency and enable users to trace why a given operating point was accepted, rejected, or forwarded for optimization.

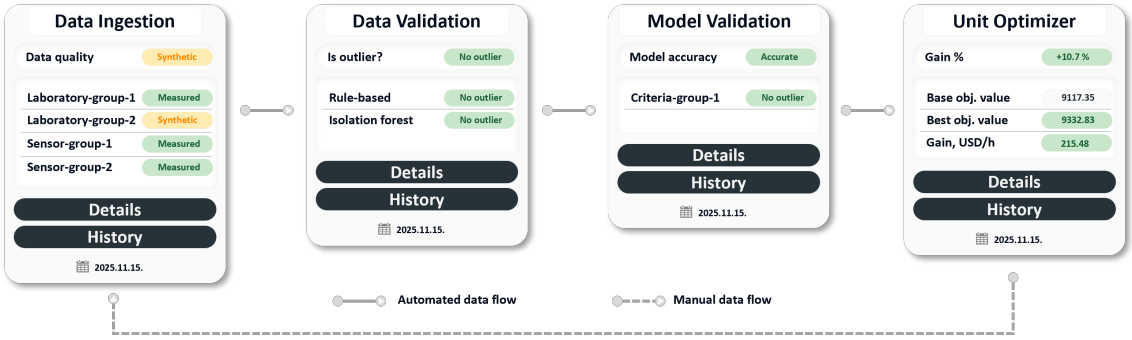


Figure 2.13: Key steps of the digital shadow workflow, including automated data ingestion, data validation, model validation, and execution optimization.

Model validation was embedded as a routine step by comparing selected model outputs against measured process data. Only validated operating points were forwarded to optimization, which was executed automatically but implemented manually after technologist review. Transparency was further reinforced through workflow tracking and summary visualizations, as illustrated in Figure 2.14. This dashboard-style representation exposes the potential benefits of model-based operational decision support over time, while clearly marking periods where recommendations were withheld due to missing data, detected anomalies, or insufficient model accuracy. In this way, the digital shadow communicates not only optimization opportunities, but also the limitations that currently prevent their realization.

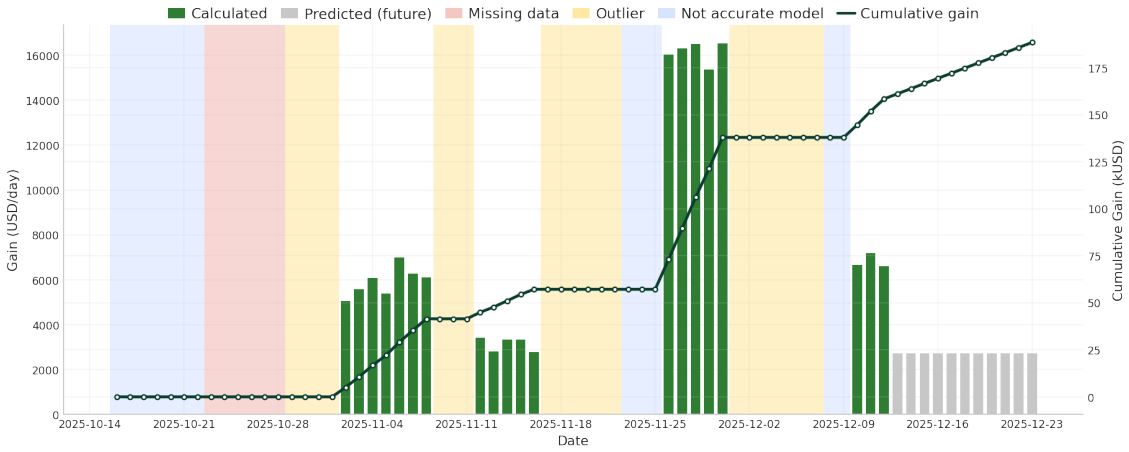


Figure 2.14: Summary visualization of digital shadow operation showing potential optimization benefits over time, while highlighting operating points excluded.

While data quality management is a necessary prerequisite for reliable digital shadow operation, this phase also revealed that model maintenance is equally critical. Even with validated and consistent data, a drifting or outdated model cannot provide reliable recommendations. Consequently, continuous monitoring of model accuracy and systematic model maintenance emerge as essential requirements to advance beyond the digital shadow stage.

Overall, this phase demonstrated continuous, data- and model-driven decision support with a high level of transparency, while preserving operational authority. The digital shadow represents a critical transition toward higher maturity of DT.

2.3.4 Phase III – Digital twin

The final stage of the proposed maturity pathway corresponds to a full *Digital Twin*, in which the digital representation is continuously synchronized with the real process and embedded into operational decision-support workflows. In contrast to the digital shadow phase—where recommendations are generated automatically but implemented manually—this phase extends the workflow with additional validation layers and tighter integration with dynamic process behavior, while explicitly preserving human authority over final operational decisions.

A key limitation of steady-state optimization-based decision support is that dynamically feasible transitions between operating points are not guaranteed, even when all steady-state constraints are satisfied. To address this limitation, the DT concept incorporates a dynamic feasibility validation step using an OTS. In the proposed workflow, optimized operating conditions generated by the digital shadow are first evaluated in the OTS environment before being forwarded to the real unit. The OTS provides a high-fidelity dynamic representation of the process and its control system, enabling assessment of transient behavior, controllability, and operational robustness, and allowing potential issues—such as actuator saturation, control-loop interaction, or unacceptable transient excursions—to be identified prior to plant implementation.

The applicability of OTS-based validation strongly depends on the availability of an existing and well-maintained simulator. In installations where an OTS is already deployed for training or control-system testing, this approach provides a natural extension toward dynamic validation of DT recommendations. In contrast, developing an OTS solely for this purpose may not be economically justified, in which case alternative validation strategies—such as simplified dynamic checks or conservative constraint tightening—may be required.

For dynamic validation to be meaningful, the OTS must accurately reflect the behavior of the real control system. Accordingly, the DT design assumes that the same control logic used in the distributed control system (DCS) — including regulatory control and, where applicable, APC strategies — is available in the OTS environment. This requirement highlights that full DT deployment extends beyond modeling and analytics and necessitates close integration with control system engineering practices.

In addition to dynamic validation, the DT workflow incorporates a state consistency check to verify that the process is still operating within the conditions under which the optimization was performed. Since real processes are subject to disturbances, drift, and operational changes, recommendations must be validated

against the current process state before further evaluation. If deviations exceed predefined thresholds, the workflow is halted or restarted to prevent the application of outdated or context-inconsistent recommendations.

The complete DT workflow, extending the digital shadow logic with dynamic validation and explicit approval steps, is illustrated in Figure 2.15. The figure highlights how data ingestion, data validation, model validation, steady-state optimization, dynamic feasibility checks, and operator approval are combined into a transparent, auditable decision-support chain.

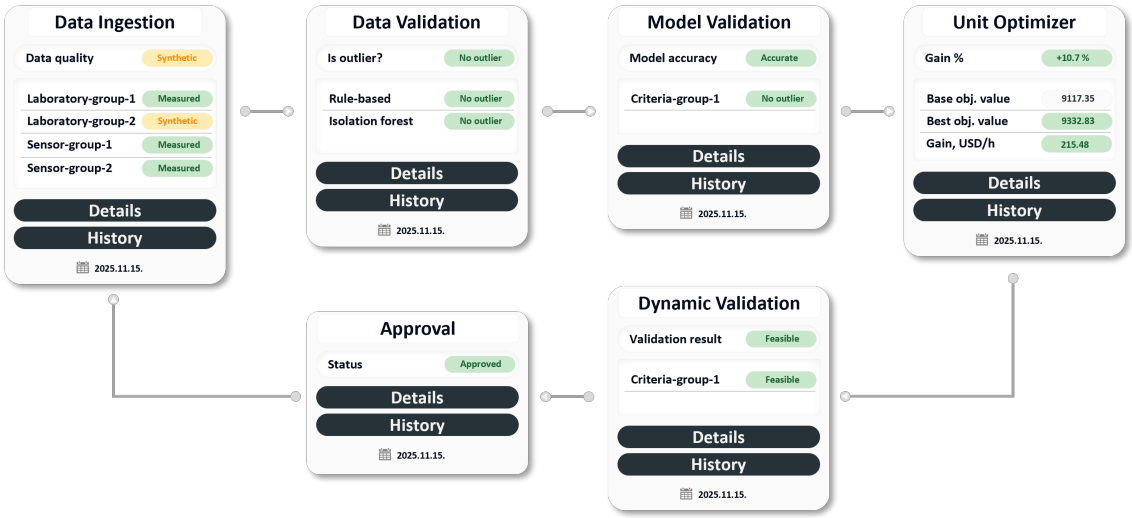


Figure 2.15: Extended DT workflow incorporating dynamic feasibility validation using an OTS and explicit operator approval before implementation.

Despite the increased level of automation and validation, the transition from recommendation to execution is not intended to be autonomous. Final implementation remains subject to explicit operator approval, ensuring safety, accountability, and auditability. All stages of the DT workflow—from data ingestion and optimization through OTS validation to operator decision—are logged and traceable, supporting regulatory compliance and fostering trust among operators, engineers, and management.

While this phase has not yet been fully implemented or validated in industrial operation, it defines a clear and realistic target architecture. Together with the preceding digital model and digital shadow phases, it illustrates how DT capabilities

can be introduced incrementally in conservative industrial environments, enabling risk-managed progression toward advanced, dynamically validated decision support.

2.3.5 Key learning points and future development areas

This case study demonstrated that DT concepts can be introduced incrementally in conservative industrial environments by aligning technical feasibility with organizational readiness and economic justification. The phased progression from a digital model through a digital shadow to a conceptually defined DT showed that measurable operational value can be achieved without requiring immediate closed-loop automation.

A central outcome of the study is the recognition that sustaining DT benefits depends critically on maintaining model credibility over time. While continuous monitoring, data quality management, and version control are well established for data-driven models through MLOps frameworks, comparable automated life cycle management solutions are largely absent for flowsheet-based process models. As operating conditions evolve, model drift and performance degradation therefore remain major risks in DT-enabled industrial applications.

The developed DTE highlighted the benefits of separating model development from model usage, enabling broader access to simulation and optimization capabilities while increasing the importance of governance, traceability, and validation. In practice, multi-fidelity modeling—combining high-fidelity flowsheet models with data-driven or surrogate models—is often unavoidable, particularly for large, interconnected systems [69]. However, such hybrid model structures further amplify the need for automated validation, monitoring [49], and robust data quality management [70].

Validation emerged as a key enabler of transparency and operational acceptance. Embedding validation steps directly into DT workflows reduces the risk of propagating inaccurate recommendations and helps maintain user trust. In this context, explainability and user engagement also play a critical role; integrating explainable

AI techniques [71] supports informed decision-making and long-term adoption.

In summary, the case study confirms that phased DT adoption can deliver tangible value while progressively increasing system sophistication. At the same time, it clearly identifies automated calibration and life cycle management of flowsheet-based models as a critical bottleneck and a primary focus for future research and development.

2.4 Conclusion

This chapter examined the industrial transition toward DT solutions based on a literature review and an industrial case study. The analysis confirms that DT adoption in the process industry is inherently incremental, limited not only by technical complexity but also by organizational readiness, integration effort, and long-term model sustainability. Key challenges repeatedly identified include the selection of high-impact use cases, the integration of heterogeneous modeling layers into operational workflows, and the maintenance of model validity under changing process conditions.

The case study demonstrated that partial DT implementations—such as digital models and digital shadows—can already deliver operational value when embedded in transparent workflows. At the same time, it exposed a central limitation of current practice: while flowsheet simulation models provide the physical foundation of DT systems, their predictive accuracy degrades over time due to process drift, equipment aging, and evolving operating regimes. The absence of automated performance monitoring and recalibration mechanisms makes long-term model maintenance a critical bottleneck.

These conclusions motivate the next chapter, which focuses on surrogate-based flowsheet calibration and online model maintenance strategies designed to ensure sustained model accuracy and reliability in dynamic industrial environments.

Chapter 3

Surrogate-based calibration of flowsheet models

The preceding chapter showed that while flowsheet-based DTs can deliver operational value, their long-term reliability is fundamentally limited by the lack of automated model maintenance mechanisms. In particular, gradual process drift and changing operating conditions lead to a progressive loss of model fidelity, which directly undermines trust in DT–supported decision-making. This chapter addresses this limitation by introducing a surrogate-based calibration framework designed to sustain flowsheet model accuracy under evolving industrial conditions.

3.1 Chapter motivation and scope

DTs are increasingly adopted in industrial environments to support data-driven monitoring, analysis, and decision-making across interconnected business processes [16]. In the process industry, high-fidelity flowsheet simulation models represent a well-established and physically grounded option for DT implementations, as they provide consistent representations of thermodynamics, mass and energy balances, and unit-level behavior. These models form the analytical backbone of many DT applications, including process analysis, scenario evaluation, performance optimization, and equip-

ment condition monitoring [72].

As DT systems evolve from offline analysis tools toward continuously operating decision-support systems, flowsheet models are expected to remain synchronized with the physical plant. In practice, however, maintaining model accuracy over extended time horizons remains a major challenge. Real industrial processes are subject to gradual but persistent changes arising from equipment aging, fouling, catalyst deactivation, operational adjustments, and feedstock variability [38]. If such effects are not systematically addressed, discrepancies between model predictions and plant measurements accumulate, ultimately degrading the credibility of DT-based recommendations, especially when model outputs are embedded into production planning, scheduling, or control workflows.

Model calibration represents a primary mechanism for sustaining flowsheet model fidelity by adjusting uncertain parameters to reduce the mismatch between simulated and observed process behavior. In this dissertation, the term calibration is used to denote parameter estimation in first-principles flowsheet models with fixed model structure. However, in long-term DT operation, calibration is complicated by concept drift, whereby the input–output relationships of the system evolve over time due to underlying physical changes [73]. Although a wide range of drift detection techniques has been proposed [50], model adaptation in industrial practice is still largely manual, expert-driven, and computationally expensive.

In many industrial applications—such as heat exchanger fouling or catalyst activity loss—the dominant degradation mechanisms are gradual and systematic, making them well suited for automated or semi-automated calibration strategies. Building on this observation, this chapter formulates flowsheet model maintenance as an adaptive calibration problem and introduces a surrogate-based methodology that reduces computational cost while preserving physical consistency. The proposed approach aligns calibration objectives with modeling priorities and operational requirements, supporting continuous flowsheet model adaptation in DTs.

3.2 Calibration and maintenance of flowsheet models in DT applications

The long-term reliability of flowsheet-based DTs critically depends on the ability to maintain model fidelity under evolving industrial conditions. As discussed in the previous section, gradual process drift, equipment aging, and operational changes inevitably lead to discrepancies between simulated and observed behavior if models remain static. Addressing these discrepancies requires systematic calibration and maintenance strategies that can operate efficiently within a DT environment.

3.2.1 Challenges of flowsheet model calibration

In DT applications, flowsheet simulation models are coupled with real processes to support continuous analysis and decision-making. The reliability of these functions depends directly on the predictive accuracy of the underlying models. When model fidelity deteriorates, decision-support systems may fail to detect gradual equipment degradation, misrepresent process constraints, or overlook emerging optimization opportunities, potentially leading to unstable and less effective operation or unplanned shutdowns. Beyond immediate operational risks, persistent model inaccuracies gradually erode user confidence in both the model and the DT as a whole [74].

Unlike purely data-driven models, flowsheet models embed physical and thermodynamic knowledge and are traditionally developed for offline analysis or scenario evaluation. Their construction typically follows a structured, expert-driven workflow, as illustrated in Figure 3.1. The “Model application” step may represent either offline or online model utilization; however, in conventional industrial practice, the workflow usually terminates after this step, without systematic online performance monitoring or continuous revalidation. The dashed section in Figure 3.1 highlights this commonly missing life cycle management step, where model predictions are continuously compared against plant behavior to evaluate whether recalibration is

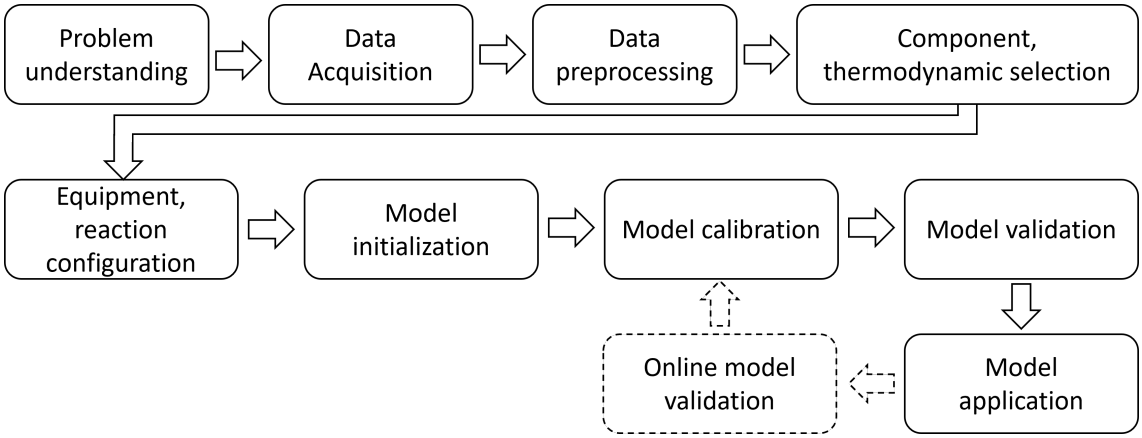


Figure 3.1: General workflow of flowsheet model development and application, including the typically missing online validation and recalibration feedback loop highlighted by the dashed section.

required. This feedback mechanism becomes particularly important in online DT applications, where model degradation caused by process drift, equipment aging, or changing operating conditions can significantly reduce prediction reliability over time. Without continuous validation, subsequent model applications may require restarting the calibration workflow due to uncertainty regarding the actual model accuracy under current operating conditions. Therefore, automated real-time validation and recalibration become essential capabilities for maintaining long-term model reliability within DT environments.

A central challenge in long-term model maintenance is concept drift, where the relationship between model inputs and outputs changes due to physical degradation, operational adjustments, or process reconfiguration [73]. While certain drift sources—such as equipment replacement or process redesign—require expert intervention, many degradation phenomena in the process industry (e.g., fouling or catalyst deactivation) evolve gradually and can, in principle, be addressed through automated or semi-automated calibration strategies.

From a computational perspective, flowsheet model calibration is commonly formulated as an optimization problem that minimizes the discrepancy between simulated and measured process variables. However, direct optimization of high-

fidelity flowsheet models is computationally expensive and often impractical in an online or near-real-time context. Gradient-based algorithms require accurate derivative information, which is difficult to obtain for complex flowsheets, while derivative-free methods suffer from slow convergence in high-dimensional parameter spaces [75]. In addition, flowsheet simulations frequently encounter convergence failures under extreme or poorly initialized operating conditions, further complicating their use in automated calibration workflows [17].

These limitations highlight the need for calibration approaches that reduce computational burden, improve robustness, and enable systematic model adaptation without continuous expert involvement.

3.2.2 Surrogate modeling as an enabler for scalable calibration

Surrogate models—also referred to as metamodels or reduced-order models—offer a promising pathway to address the computational and practical challenges associated with flowsheet model calibration [26]. By approximating the input–output behavior of high-fidelity simulation models using simplified mathematical representations, surrogate models enable efficient optimization while significantly reducing computational cost.

Previous studies have demonstrated the effectiveness of surrogate-based optimization in complex process systems, showing improvements in convergence behavior, robustness against noisy data, and overall computational efficiency. Applications range from explicit equation-based optimization of large-scale chemical processes [76] to Bayesian symbolic regression for constructing analytical surrogates of flowsheet models [77]. AI-based surrogate models have also been shown to accurately predict flowsheet behavior and provide reliable initial conditions for simulation, even in systems with recycle streams or strong nonlinearities [78]. Similar approaches have been successfully applied in pharmaceutical process design, where surrogate-based

Table 3.1: Examples of surrogate-based calibration of high-fidelity models.

Article	Description	Ref.
Machine learning-based surrogate model for calibrating fire source properties in FDS models of façade fire tests	Surrogate models used to efficiently calibrate fire dynamics simulation parameters without direct coupling to optimization.	[51]
Surrogate modeling-based calibration of hydrodynamic river model parameters	Calibration of hydrodynamic models using surrogate conceptual representations prior to full-scale simulation.	[84]
Surrogate-based model calibration for pressurized water reactor physics calculations	Calibration of neutronics and thermal-hydraulics parameters using surrogate-assisted optimization of reactor core models.	[85]
Development and surrogate-based calibration of a CO2 reservoir model	Surrogate-assisted calibration of high-fidelity subsurface reservoir simulation models.	[86]

optimization enables efficient handling of constrained, noisy, and regulation-driven design problems [79].

Flowsheet model calibration inherently involves multiple outputs and competing objectives, as different process variables vary in importance depending on modeling goals and operational priorities [80]. Measurement uncertainty and process noise further complicate this task. While multi-objective optimization frameworks are available [81], industrial calibration practice typically aims to identify a single, practically acceptable parameter set rather than a full Pareto front. This is commonly achieved by aggregating objectives using weighted formulations informed by modeling expertise and operational considerations [82]. Expert knowledge therefore remains essential for defining calibration objectives that balance accuracy, robustness, and physical interpretability [83].

Despite the extensive use of surrogate models in process optimization, their application to flowsheet model calibration remains comparatively limited. Existing studies primarily focus on calibrating computationally intensive models at fixed operating points, often as a preprocessing step for high-fidelity model development, as summarized in Table 3.1. In contrast, the calibration of flowsheet models across a range of operating conditions, as required in DT applications, has received considerably less attention.

Building on these observations, this chapter introduces a surrogate-driven cal-

ibration framework specifically designed for flowsheet process simulation models operating within DT systems. The framework addresses two central challenges: reducing the computational burden of calibrating high-dimensional flowsheet models while avoiding infeasible operating points, and systematically incorporating modeling priorities, physical constraints, and measurement uncertainties into the calibration objective without requiring frequent retraining of surrogate models.

The proposed methodology is detailed in Section 3.3. Section 3.4 presents an industrial refinery case study that illustrates surrogate model training and the formulation of calibration objectives informed by modeling considerations. Finally, Section 3.4.4 evaluates the performance of the framework and discusses its implications for long-term DT operation.

3.3 Methodology for adaptive flowsheet model calibration

Surrogate models are extensively used in optimization to address challenges associated with high computational costs, noisy data, and convergence difficulties [26]. In the approach proposed in this paper, surrogate models are employed to replace computationally intensive flowsheet models during the calibration process. This strategy aims to estimate calibration factors that minimize the residual error between simulated and observed process data, thereby reducing the number of infeasible operating points and significantly decreasing the overall calibration time.

The success of this approach hinges on the accuracy of both the surrogate and the original flowsheet models, as these directly impact the precision of the adaptive calibration. A less accurate surrogate model results in less precise calibration factors for the flowsheet model. However, achieving higher accuracy for the surrogate model requires a larger number of simulation runs, which can be computationally demanding.

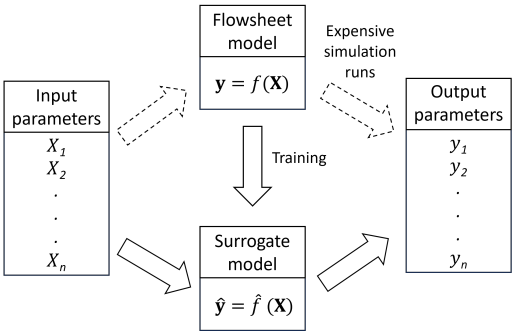


Figure 3.2: General surrogate model methodology.

Another crucial aspect of the calibration process is the formulation of the objective function, which defines the calibration goals. The calibration of flowsheet models is influenced by various sources of uncertainty, including sensor measurement errors such as high variability, noise, and sensor failures, as well as limitations inherent to the flowsheet model itself. These limitations may result from model simplifications or constraints, which can reduce the accuracy of certain parameters. Therefore, a flexible objective function is necessary to effectively integrate these uncertainties and modeling considerations into the calibration process.

3.3.1 Surrogate models

Surrogate models, also referred to as metamodels, provide simplified mathematical representations of complex systems and are commonly employed to reduce the computational burden associated with optimization and calibration of high-fidelity flowsheet simulation models. By approximating the input–output behavior of the original simulator, surrogate models enable efficient exploration of the parameter space while preserving the dominant system characteristics (Figure 3.2).

Within surrogate-based modeling frameworks, a key distinction can be made between *global* and *local* surrogate models. Global surrogate models aim to approximate the behavior of the original system over a wide operating domain and are therefore suitable for multi-point analysis, optimization, and calibration tasks across varying operating conditions [87]. In contrast, local surrogate models focus

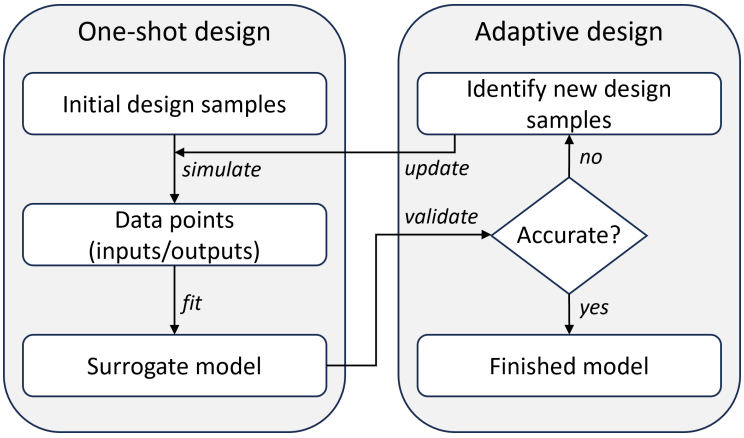


Figure 3.3: One-shot and adaptive methods for global surrogate model design.

on accurately representing system behavior in the vicinity of a specific operating point, often providing higher local accuracy at the expense of generalizability. In the context of DT applications—where operating conditions may shift over time and calibration must remain valid across multiple regimes—global surrogate models offer a more appropriate foundation for sustained model maintenance.

The construction of surrogate models relies critically on the quality and distribution of the training data. This data is typically obtained by sampling the input space of the underlying flowsheet model and evaluating the corresponding outputs. However, there is no universally optimal sampling strategy, as the effectiveness of a given design depends strongly on model nonlinearity, dimensionality, and the intended use of the surrogate [88]. Consequently, surrogate sampling strategies are generally categorized into one-shot (stationary) and sequential (adaptive) approaches [17], as illustrated in Figure 3.3.

One-shot sampling methods determine all sampling points prior to surrogate training, typically using space-filling designs to achieve uniform coverage of the input domain. Common techniques include Latin Hypercube Sampling (LHS) [89, 90], Sobol sequences [91], Halton sequences [92], full factorial designs [93], and Box–Behnken designs [94]. These approaches are straightforward to implement and well suited for constructing global surrogate models in stationary settings. However, their non-adaptive nature may lead to inefficient allocation of samples, oversampling

regions with limited sensitivity while underrepresenting highly nonlinear or critical areas of the input space [95, 87, 96].

Sequential or adaptive sampling strategies iteratively refine the surrogate model by incorporating information gained during training, often guided by prediction error or uncertainty estimates [97]. These methods balance exploration of poorly sampled regions with exploitation of areas exhibiting high model error [98]. Representative techniques include LOLO-Voronoi [99], the Smart Sampling Algorithm (SSA) [100], and ALAMO [101]. Adaptive sampling can significantly reduce the number of required simulation runs while maintaining accuracy, particularly for complex and highly nonlinear systems. Nevertheless, these strategies generally assume stationary system behavior during surrogate construction and may incur additional computational overhead due to repeated simulator evaluations [102, 103].

A wide range of surrogate modeling techniques has been reported in the chemical process engineering literature, including polynomial regression, kriging, artificial neural networks (ANNs), and radial basis functions (RBFs) [87]. While the selection of an appropriate surrogate type depends on the complexity of the modeled system, ANNs are particularly well suited for constructing global surrogate models of complex, nonlinear flowsheet simulations. For simpler systems, polynomial models or Gaussian processes may provide sufficient accuracy, whereas increasing system complexity typically necessitates deeper neural network architectures with higher representational capacity. In the context of surrogate-based flowsheet calibration, particular attention must be paid to ensuring that the training dataset adequately spans the relevant calibration parameter space, as insufficient coverage directly limits the surrogate's ability to support robust and reliable model maintenance. Since some input parameters of the surrogate model are used for calibration during the optimization process, the general form of the surrogate model can be expressed as shown in Eq. 3.1.

$$\hat{\mathbf{y}} = f(\mathbf{X}, \boldsymbol{\theta}) \tag{3.1}$$

Where $\mathbf{X}$ represents the input parameters, and $\boldsymbol{\theta}$ denotes those input parameters considered as calibration parameters for the original flowsheet model. While these parameters are treated as input variables in the surrogate model, they can be adjusted during the flowsheet calibration process to compensate model drift.

Beyond the surrogate-based approach, it is also possible to train a partial inverse model that directly predicts the calibration factors. This method bypasses the need for additional optimization, resulting in a quicker calibration process; however, its main disadvantage is that it considers fewer influencing factors than the surrogate-based method, providing fewer options to incorporate modeling and operational considerations. To evaluate the effectiveness of the surrogate-based adaptive calibration, in the Section 3.4 the results from the proposed approach (Eq.3.2) are compared with the results from a partial inverse model, which serves as a baseline (Eq.3.3).

$$\hat{\mathbf{y}} = f(\mathbf{X_b}, \mathbf{X_o}, \boldsymbol{\theta}_{surr}) \tag{3.2}$$

$$\hat{\boldsymbol{\theta}}_{p.inv} = f(\mathbf{X_b}, \mathbf{X_o}, \mathbf{y}) \tag{3.3}$$

In Eq.3.2 and Eq.3.3, $\hat{\mathbf{y}}$ represents the estimated values, $\mathbf{y}$ denotes the measured values, $\mathbf{X_b}$ includes boundary input conditions (such as inlet flow, temperature, quality), $\mathbf{X_o}$ encompasses unit operation conditions, and $\boldsymbol{\theta}$ and $\hat{\boldsymbol{\theta}}$ represent the original and estimated calibration factors. A subscript specifies whether these factors were estimated using a partial inverse model or a surrogate model.

3.3.2 Objective function

In multi-objective optimization, transforming multiple objectives into a single one is a common approach [82]. This involves utilizing an error function, such as Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error

(RMSE), or Huber loss [104], to calculate the residual error between measured and estimated parameters. Each error term corresponding to a fitted output parameter is then adjusted by a weight, reflecting its relative importance in the overall objective function [105]. The general form of the objective function is expressed in Eq. 3.4.

$$F(\boldsymbol{\theta}^*) = \sum_{k=1}^{M} w_k \cdot \left(\frac{1}{N} \sum_{i=1}^{N} \phi\Big(y_{k,i}, \hat{y}_{k,i}\Big) \right) \tag{3.4}$$

Where:

- $F(\boldsymbol{\theta}^*)$: The objective function that we seek to minimize.
- **M**: The number of model output parameters (objectives) used to calculate model residual error.
- **N**: The number of observations (data points).
- w_k: The weight assigned to the k-th objective, which adjusts its contribution to the overall objective function. The weights should satisfy $\sum_{k=1}^{M} w_k = 1$, and each weight $w_k \in [0, 1]$.
- $\phi(y_{k,i}, \hat{y}_{k,i})$: The error function used to compute the distance between the true value $y_{k,i}$ and the predicted value $\hat{y}_{k,i}$.
- $\boldsymbol{\theta}^*$: Those input parameters which are used as calibration parameters of the model that are adjusted by the optimizer to minimize the objective function. These parameters are updated during the optimization process.
- $y_{k,i}$: The true value (target) for the k-th objective and i-th observation.
- $\hat{y}_{k,i}$: The predicted output for the k-th objective and i-th observation, computed by the model.

In this study, the squared error is utilized as the error function (ϕ), and the weighted sum of errors becomes the single objective function to be minimized, formulated as Eq. 3.5.

$$F(\boldsymbol{\theta}^*) = \sum_{k=1}^{M} \frac{1}{N} \sum_{i=1}^{N} (y_{k,i} - \hat{y}_{k,i})^2 w_k \tag{3.5}$$

If the calibration process involves fitting the model to one operating point at a time, the formula can be simplified as demonstrated by Eq. 3.6.

$$F(\boldsymbol{\theta}^*) = \sum_{k=1}^{M} (y_k - \hat{y}_k)^2 w_k \tag{3.6}$$

While weighting functions offer flexibility in assigning importance to specific output parameters during optimization, uncertainties in weight selection can significantly influence the optimization outcome. An alternative approach focuses on measurement variability and uncertainty, introducing the Taguchi loss function. This concept aligns with Taguchi methods, developed by Genichi Taguchi for quality control in manufacturing [106]. Taguchi's philosophy centers on minimizing customer dissatisfaction with product performance deviations. This function penalizes deviations from target values based on the standard deviation of measured parameters.

The objective function integrating weighting functions with the Taguchi loss formulation is defined in Eq. 3.7:

$$F(\boldsymbol{\theta}^*) = \sum_{k=1}^{M} \tau \left(y_k - \hat{y}_k\right)^2 w_k \tag{3.7}$$

where τ denotes a penalty function that reflects acceptable and unacceptable deviations between model predictions and observed values. In general, the choice of penalty structure and thresholds is use-case specific and should be aligned with process sensitivity, measurement reliability, and operational priorities. In this study, a simple two-level penalty scheme is adopted, with $\tau = 0.001$ if $|y_k - \hat{y}_k| \leq \sigma_k$ and $\tau = 1$ otherwise. Here, σ_k denotes the measured standard deviation of the observed variable k, and $w_k \in \{0, 1\}$ is a weighting factor indicating whether the corresponding output variable is included in the objective function.

This formulation reflects the assumption that deviations within the measurement uncertainty are acceptable and should not dominate the calibration process, while larger discrepancies indicate model mismatch and should be penalized more strongly. Although alternative penalty functions could be employed, the selected structure provides a transparent and robust objective formulation suitable for the investigated case study.

3.3.3 Optimizer

The calibration task is formulated as a parameter optimization problem, where the objective is to identify the set of calibration parameters that minimizes a problem-specific objective function:

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}^* \in \Theta} F(\boldsymbol{\theta}^*) \tag{3.8}$$

where $\hat{\boldsymbol{\theta}}$ denotes the optimized calibration parameters and Θ represents the feasible parameter space.

Among the wide range of available optimization techniques [107], Particle Swarm Optimization (PSO), originally proposed by Kennedy and Eberhart [108], was selected for this study. PSO belongs to the class of population-based, derivative-free optimization algorithms and has been widely applied to complex, nonlinear optimization problems. In PSO, each particle represents a candidate solution, and particles iteratively update their positions by combining individual experience with information shared across the swarm, allowing efficient exploration and exploitation of the search space.

The choice of PSO is motivated by several characteristics of the flowsheet calibration problem. First, the objective function is typically nonlinear, multimodal, and may be non-smooth due to surrogate approximations and physical constraints, which limits the applicability of gradient-based methods. Second, calibration parameters often differ in scale and sensitivity, and PSO is relatively insensitive to variable

scaling. Third, PSO does not require gradient information, making it robust to noisy objective functions arising from measurement uncertainty and surrogate prediction errors.

The PSO update equations for particle velocity and position are given by:

$$\mathbf{v_{id}^{t+1}} = w\mathbf{v_{id}^{t}} + c_1\text{rand}(0,1)(\mathbf{p_{id}^{t}} - \mathbf{x_{id}^{t}}) + c_2\text{rand}(0,1)(\mathbf{p_{gd}^{t}} - \mathbf{x_{id}^{t}}) \tag{3.9}$$

$$\mathbf{x_{id}^{t+1}} = \mathbf{x_{id}^{t}} + \mathbf{v_{id}^{t+1}} \tag{3.10}$$

where $\mathbf{v_{id}^{t}}$ and $\mathbf{x_{id}^{t}}$ denote the velocity and position of particle i at iteration t, w is the inertia weight, c_1 and c_2 are the cognitive and social coefficients, and $\mathbf{p_{id}^{t}}$ and $\mathbf{p_{gd}^{t}}$ represent the personal and global best positions, respectively. The personal best position corresponds to the best solution previously found by particle i, while the global best position represents the best solution identified by the entire swarm up to iteration t.

During calibration, particle positions are initialized within parameter bounds derived from flowsheet sensitivity analysis and expert knowledge. The required swarm size depends on the dimensionality and nonlinearity of the calibration problem, with more complex systems requiring larger populations for adequate exploration.

Although direct application of PSO to high-fidelity flowsheet models can be computationally expensive due to the parallel evaluation of particles, the use of surrogate models substantially reduces this burden. By replacing costly flowsheet evaluations with fast surrogate predictions, PSO becomes a practical and efficient optimizer for calibration tasks in DT environments. In addition, PSO is inherently parallelizable, allowing further performance gains on multi-core computing architectures.

While alternative optimization algorithms could be considered, PSO was selected due to its robustness, scalability, and compatibility with surrogate-based calibration. A systematic comparison of optimization strategies is considered outside the scope of this work but represents a relevant direction for future research.

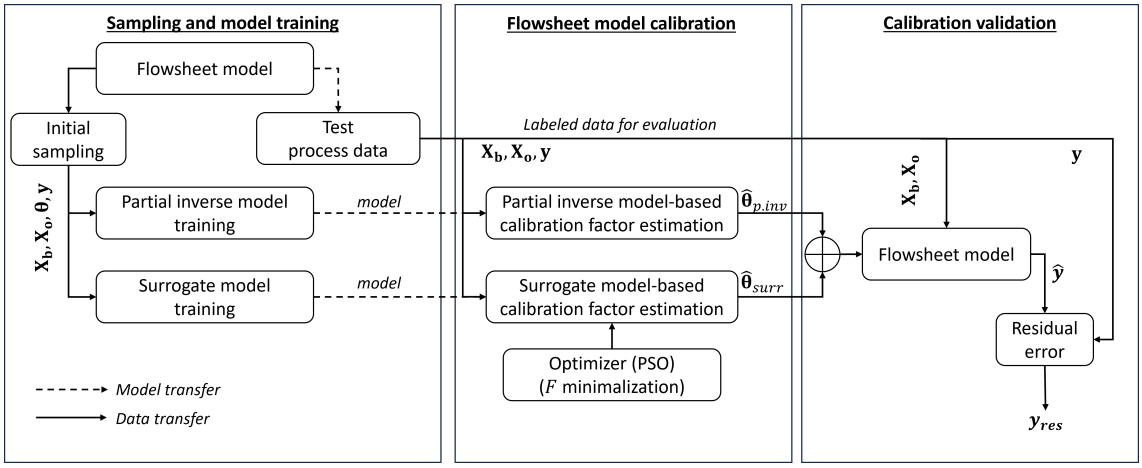


Figure 3.4: Main steps of the investigated surrogate-based adaptive flowsheet model calibration workflow, including sampling, model training, calibration, and evaluation.

3.3.4 Methodology overview

This section presents a high-level overview of the proposed surrogate-based calibration framework, summarized in Figure 3.4. The methodology is designed to support computationally efficient and repeatable calibration of flowsheet simulation models within DT environments, while maintaining physical consistency and seamless integration with industrial simulation tools.

The workflow begins with the generation of training data using a high-fidelity flowsheet simulation model. Input–output samples are collected over a predefined calibration parameter domain that represents the normal operating range of the physical unit, ensuring that the resulting surrogate models remain relevant for industrial operation. Depending on the application, the sampling strategy may follow either a stationary (one-shot) or a sequential (adaptive) design, enabling flexible trade-offs between computational cost and surrogate accuracy.

Based on the generated dataset, a global surrogate model is trained to approximate the forward behavior of the flowsheet model. In parallel, a partial inverse model is constructed to estimate calibration parameters from measured process data. During DT operation, these surrogate models replace direct flowsheet evaluations within the calibration loop, enabling rapid parameter estimation.

Calibration is formulated as an optimization problem in which the surrogate models are used to evaluate candidate parameter sets. In this case a population-based optimization algorithm is employed to minimize a problem-specific objective function defined over selected process variables. To ensure physical validity, the estimated calibration parameters are subsequently validated using the original flowsheet simulation model, and the resulting residual errors are evaluated to confirm consistency between surrogate-based predictions and high-fidelity model behavior.

The complete workflow is implemented and automated in Python (v3.10) and integrated with Aspen HYSYS V14 as the high-fidelity simulation environment. The following section presents an industrial refinery case study that applies this methodology in practice, detailing the model setup, data generation strategy, and calibration results under realistic operating conditions.

3.4 Case study for surrogate-based flowsheet model calibration

To demonstrate and evaluate the proposed surrogate-based flowsheet calibration methodology, a simplified heat exchanger network was selected as an industrially representative case study (Figure 3.5). Heat exchangers are among the most common unit operations in process units and are well known for exhibiting gradual performance degradation due to fouling, making them a suitable and practically relevant benchmark for model maintenance and calibration studies. At the same time, the reduced structural complexity of the selected network enables systematic analysis of the calibration workflow while minimizing disturbing effects related to numerical instability or non-convergent simulation states and due the simplicity it provide a good interpretability and explainability.

The selected use case allows the methodological aspects of surrogate-based calibration to be investigated in isolation, without the additional challenges introduced

by large-scale or highly interconnected flowsheet models. Both the flowsheet and surrogate models represent steady-state behavior, and the intended application corresponds to online engineering support rather than closed-loop control. Consequently, model updates are assumed to follow the relatively slow time scale of process drift – e.g., gradual changes caused by equipment fouling – rather than fast dynamic variations occurring on the scale of seconds or minutes. Under this assumption, daily calibration updates are sufficient to capture the relevant degradation effects.

The primary objective of the case study is to assess the ability of the proposed framework to maintain flowsheet model accuracy under evolving operating conditions. Specifically, the study evaluates surrogate-based calibration performance in comparison with a partial-inverse model approach and investigates the sensitivity of the calibration outcome to objective function formulation and parameter weighting.

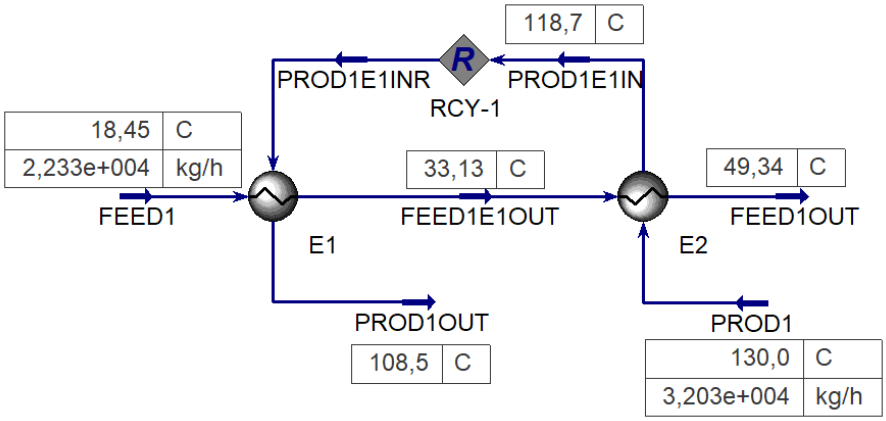


Figure 3.5: Flowsheet representation of the simplified heat exchanger network modeled in Aspen HYSYS for the case study.

A steady-state flowsheet model of the heat exchanger network was developed in Aspen HYSYS V14, consisting of two heat exchangers and two interconnected process streams. The model inputs, outputs, and corresponding operating ranges are summarized in Table 3.2. The operating ranges were defined based on engineering experience, ensuring that they reflect realistic industrial conditions while adequately covering the expected variability of the process. Since the use case is synthetic, these ranges were selected to be representative rather than derived from plant data.

Physically acceptable parameter ranges were therefore defined based on engineering knowledge to support surrogate model training and subsequent calibration.

The fouling factors of heat exchangers `E1` and `E2` were selected as calibration parameters, as they cannot be measured directly and vary gradually over time. This factor accounts for the additional resistance to heat transfer and enables the calculation of the overall heat transfer coefficient. Fouling affects the overall heat transfer performance by increasing the thermal resistance of the exchanger [109], which can be described through the overall heat transfer coefficient in Eq. 3.11:

$$\frac{1}{U} = \frac{1}{U_{\text{clean}}} + R_f \tag{3.11}$$

where U is the effective heat transfer coefficient, U_{clean} corresponds to the clean condition, and R_f represents the fouling resistance. As fouling increases, the heat transfer efficiency decreases, leading to deviations in temperature profiles and process variables. Therefore, accurate estimation of fouling factors is essential for maintaining model fidelity.

Table 3.2: Simplified heat exchanger network model input-output parameters with the used parameter ranges.

Type	Description	Unit	Operating range
Input	FEED1 mass flow rate	kg/h	10 000–30 000
Input	FEED1 temperature	°C	10–35
Input	PROD1 mass flow rate	kg/h	20 000–60 000
Input	PROD1 temperature	°C	90–135
Input	E1 Fouling factor	$\text{h°C} \cdot \text{m}^2/\text{kJ}$	0.00025–0.0030
Input	E2 Fouling factor	$\text{h°C} \cdot \text{m}^2/\text{kJ}$	0.00025–0.0030
Output	PROD1E1IN temperature	°C	80-132
Output	FEED1E1OUT temperature	°C	17-66
Output	FEED1OUT temperature	°C	26-88
Output	PROD1OUT temperature	°C	72-124

The following subsection describes the generation of training and test datasets used for surrogate model construction and calibration, including the selection of sampling strategy.

3.4.1 Synthetic test data generation

To evaluate the proposed surrogate-based calibration framework under controlled yet realistic conditions, a synthetic dataset was generated to represent approximately two years of daily operation. Periods of long steady-state operation were intentionally excluded, as they provide limited information for assessing calibration performance. The resulting dataset consists of 530 samples and reflects a time-varying operating environment characterized by gradual shifts in operating conditions, process noise, and the progressive accumulation of heat exchanger fouling.

The temporal evolution of the generated input parameters is illustrated in Figure 3.6. The imposed trends and fluctuations are designed to emulate realistic industrial behavior, thereby providing a suitable basis for evaluating the ability of the surrogate-based calibration approach to track slow process drift and maintain model accuracy over time. The trends were generated by defining parameter-wise gradual variations within the specified operating ranges for input variables, to which small-amplitude random noise (on the order of a few percent) was superimposed to mimic measurement variability and process fluctuations.

To establish a reference scenario, the flowsheet model was first evaluated on the synthetic dataset without any calibration updates. As shown in Figure 3.7, the model predictions progressively deviate from the synthetic "true" data as fouling effects accumulate. In particular, the absence of continuous updates to the fouling factors leads to pronounced prediction errors in specific operating regions, demonstrating the inability of a static flowsheet model to remain accurate under evolving conditions.

These results motivate applying the surrogate-based calibration method (Section 3.3.4). The following sections apply the workflow (Figure 3.4) to a synthetic dataset to assess its ability to maintain flowsheet model accuracy under varying conditions.

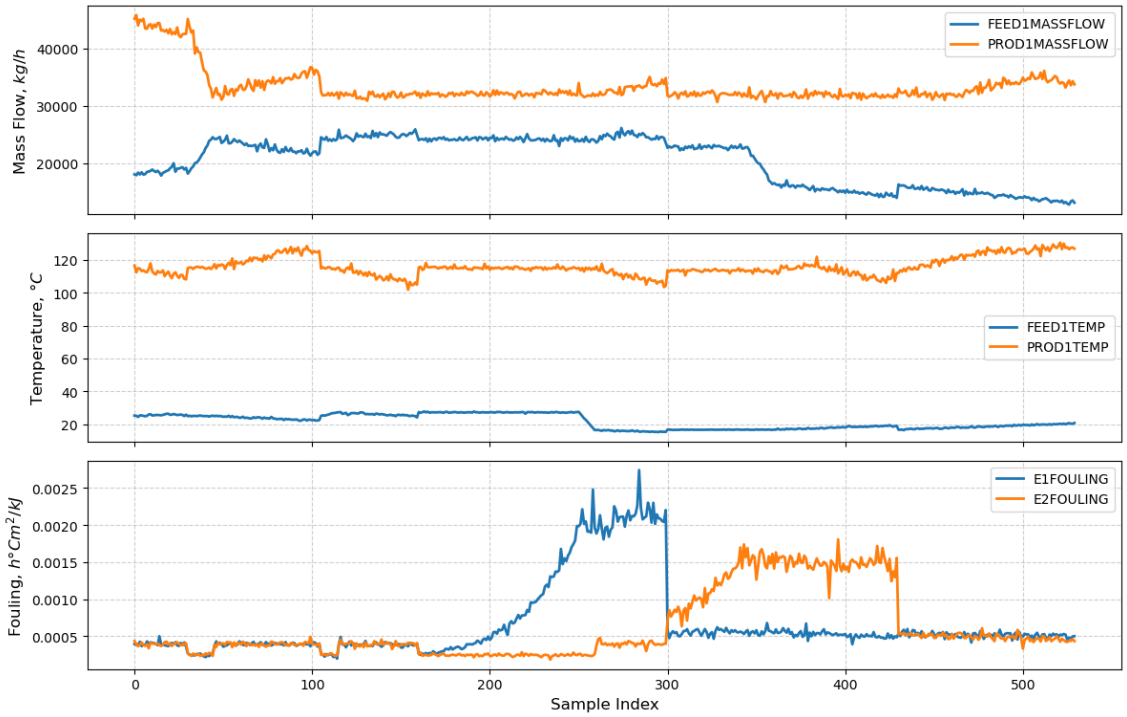


Figure 3.6: Temporal variation of synthetic input parameters used in the evaluation.

3.4.2 Surrogate and partial inverse model training

The objective of the surrogate modeling stage is to obtain a globally accurate and computationally efficient approximation of the underlying simulator for subsequent calibration tasks, rather than to develop novel training or model selection methodologies. As there is no universally optimal strategy for surrogate sampling or model selection, and performance depends strongly on problem characteristics, this work uses a reproducible training protocol instead of problem-specific tuning.

Two related models were considered. The *surrogate model* approximates the simulator response using six input parameters and four output variables. In addition, a *partial inverse model* was trained to directly predict fouling-related parameters; this model uses eight input variables (including selected simulator outputs) and produces two fouling factor outputs. The partial inverse model was introduced to assess whether direct estimation of fouling parameters can complement or simplify the calibration workflow.

For both models, the training datasets were generated exclusively using LHS,

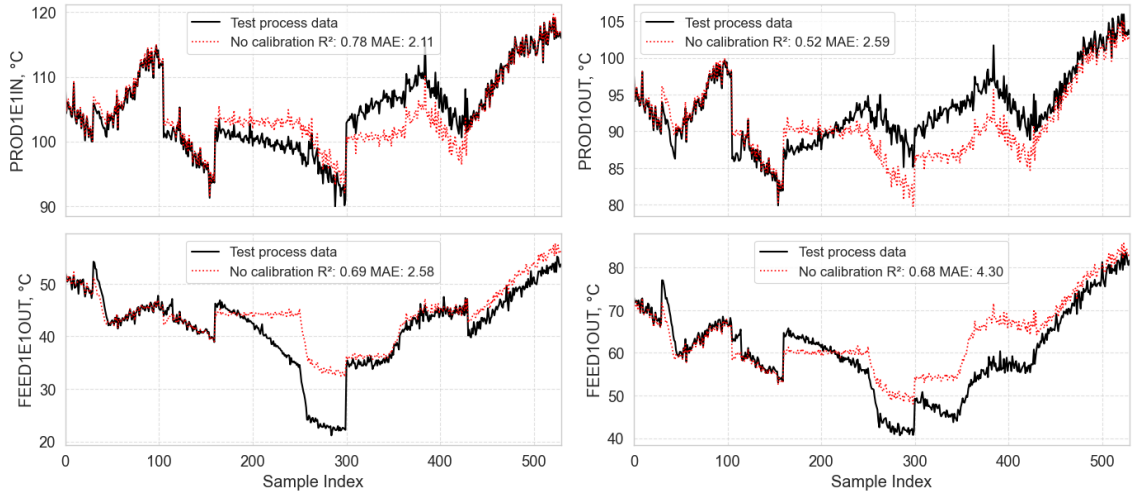


Figure 3.7: Flowsheet model prediction accuracy on the synthetic test dataset (black line) without calibration (red dotted line).

a space-filling design technique commonly employed in computer experiments and surrogate modeling. LHS ensures uniform marginal coverage of each input dimension and provides good global exploration of the parameter space without requiring large sample sizes [110]. Alternative sampling strategies were not considered in order to isolate the effects of training set size and model capacity within a consistent experimental design framework. The evaluation and selection of alternative sampling approaches remain an important direction for future research.

To analyze the influence of training data availability, a predefined ladder of sample sizes was considered. A single maximin LHS design of size $N_{\max}$ was generated, and smaller datasets were obtained as nested subsets of this design,

$$\mathcal{D}_{N_1} \subset \mathcal{D}_{N_2} \subset \cdots \subset \mathcal{D}_{N_{\max}}, \qquad N_1 < N_2 < \cdots < N_{\max}, \tag{3.12}$$

where $\mathcal{D}_N$ denotes a dataset containing N input–output pairs. This nested sampling strategy ensures consistent information growth across experiments and allows performance improvements to be attributed solely to the addition of new data rather than to differences in random sampling realizations [111].

The maximum training set size was intentionally kept moderate. While larger

datasets may further improve surrogate accuracy in this simplified case, such sample sizes are often impractical for the calibration of complex industrial or physics-based models, where individual simulations are computationally expensive. The selected range, therefore, reflects a realistic compromise between achievable accuracy and computational cost.

Neural network surrogates were implemented as multi-layer perceptrons with multiple outputs using the `MLPRegressor` implementation provided by the `scikit-learn` library. Rather than fixing a single architecture a priori, a small set of candidate architectures with increasing representational capacity was defined. Model architecture was treated as a hyperparameter and selected jointly with the training set size using cross-validation. This approach follows standard statistical learning practice, in which model complexity is chosen based on estimated generalization performance rather than only accuracy [112].

For each dataset size, all candidate architectures were evaluated using 5-fold cross-validation. The analysis investigated multiple neural network architectures with different hidden-layer depths and neuron configurations, while the activation function was fixed to the rectified linear unit (ReLU) for all experiments because of its stable gradient propagation. Although some investigated architectures contained relatively high neuron counts compared to the available sample size, the effective model complexity was controlled through several complementary regularization and validation strategies. Input and output variables were standardized prior to training and transformed back for evaluation. Model training employed gradient-based optimization with ℓ_2 regularization, adaptive learning rate, and early stopping based on validation performance, where training terminated automatically if the validation loss did not improve for a predefined number of iterations. In addition, repeated cross-validation and multiple random initializations were applied to reduce sensitivity to initialization effects and minimize the risk of selecting overfitted architectures. Furthermore, the datasets originated from deterministic flowsheet simulations rather

than noisy experimental measurements, resulting in substantially lower stochastic variability than typical industrial datasets. Consequently, somewhat larger neural network architectures could generalize reliably even with moderate sample sizes. The final network configurations were therefore selected empirically based on cross-validated predictive performance and stability rather than training accuracy alone.

Surrogate performance was assessed using the normalized root-mean-square error (NRMSE) which provides a scale-independent measure of prediction accuracy while retaining sensitivity to larger deviations. Normalization by the output standard deviation enables consistent comparison across outputs and dataset sizes and prevents high-magnitude variables from dominating the error metric [113]. For each output variable k, the NRMSE is defined as

$$\mathrm{NRMSE}_k = \frac{\sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(y_{ik} - \hat{y}_{ik}\right)^2}}{\sigma(y_{\cdot k})}, \tag{3.13}$$

where y_{ik} and $\hat{y}_{ik}$ denote the true and predicted values of output k at sample i, respectively, and $\sigma(y_{\cdot k})$ is the standard deviation of the true output values. The overall model error is computed as the mean NRMSE across all output variables, ensuring that outputs with different numerical scales contribute equally to the performance metric.

For the surrogate model, a clear improvement in predictive accuracy is observed when the training dataset size increases beyond $N = 80$. However, further increases in sample size from $N = 120$ to $N = 240$ yield only marginal reductions in NRMSE, indicating diminishing returns relative to the additional simulation effort. Based on this trade-off, an LHS training set of size $N = 160$ was selected as a balanced compromise between surrogate accuracy and computational cost. Within this dataset size, a neural network architecture with a single hidden layer of 128 neurons achieved consistently low cross-validated NRMSE and stable training behavior, and was therefore selected as the preferred surrogate configuration.

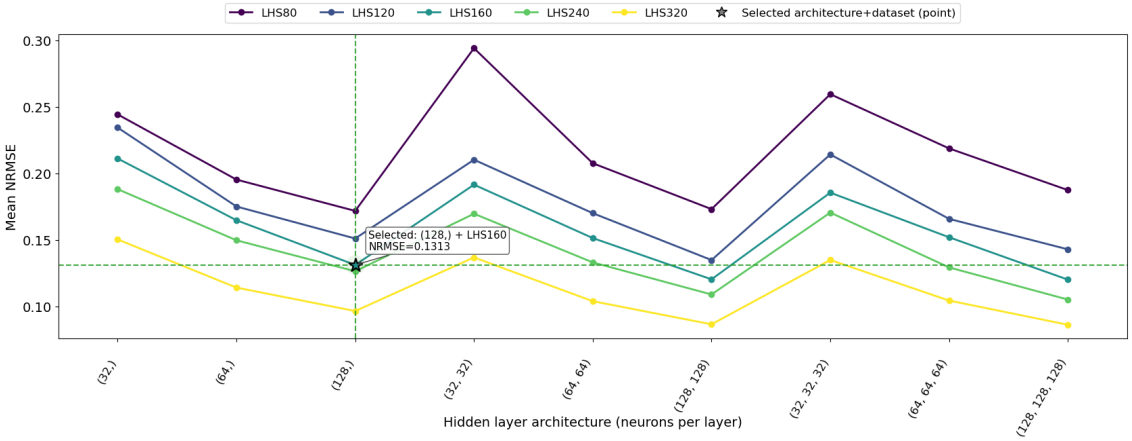


Figure 3.8: Cross-validated surrogate model performance for different LHS sample sizes (line colors) and neural network architectures (x-axis), reported in terms of NRMSE.

Although larger network architectures and training datasets achieved slightly lower minimum NRMSE values, the observed performance gains were minor compared to the increased training cost and model complexity. Given the intended use of the surrogate model within a calibration workflow, priority was given to robustness, generalization, and computational efficiency rather than absolute minimization of prediction error.

An analogous selection procedure was applied to the partial inverse model. Due to the increased input dimensionality and the more challenging inverse mapping from process variables to fouling factors, higher training data requirements were observed. As shown in Figure 3.9, predictive accuracy improves with increasing dataset size, while smaller datasets exhibit higher error and variability. For a dataset size of $N = 240$, a neural network with a single hidden layer of 128 neurons achieved stable and consistently low NRMSE. Further increases in network complexity yielded only marginal improvements, indicating that model capacity was no longer the limiting factor. Accordingly, an LHS dataset of size $N = 240$ and a single-layer 128-neuron architecture were selected as a balanced trade-off between accuracy, stability, and training efficiency for the partial inverse model.

Overall, the learning-curve analysis demonstrates that surrogate accuracy is

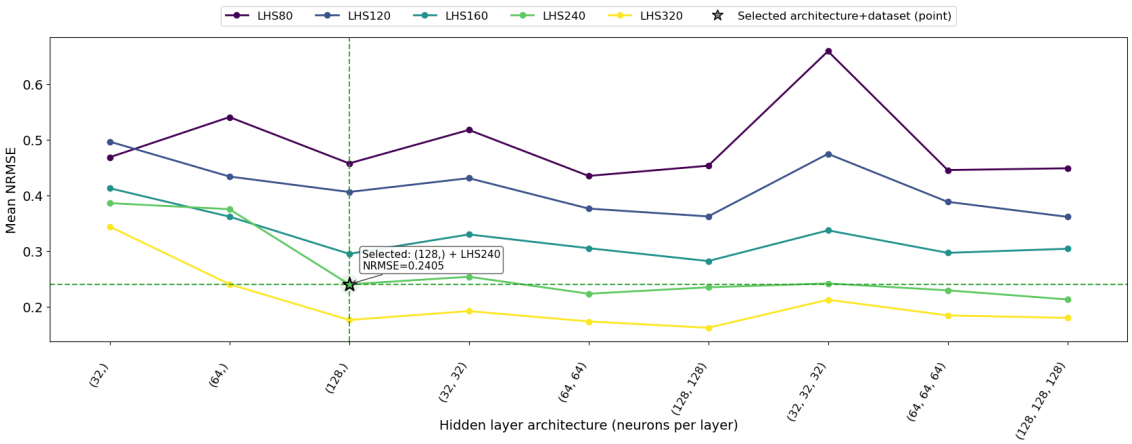


Figure 3.9: Cross-validated performance of partial inverse models for different LHS sample sizes (line colors) and neural network architectures (x-axis), reported in terms of NRMSE.

strongly influenced by both sample size and model capacity, but that beyond a problem-dependent threshold additional data yields only marginal improvements. The selected configurations therefore represent pragmatic choices that ensure sufficient accuracy for calibration tasks while remaining computationally feasible for more complex models.

After model selection, the chosen architectures were retrained on their respective datasets using a fixed 70–30 train–validation split. The splitting was performed randomly using the `train_test_split` function from the `scikit-learn` library to ensure an unbiased partitioning of the data. This final training step was performed only after the sample size and network architecture had been determined, ensuring a clear separation between model selection and final model fitting. A summary of the selected neural network configurations for both the surrogate and partial inverse models is provided in Appendix B.1.

Figures 3.10 and 3.11 summarize the predictive performance of the selected surrogate and partial inverse models on the training and validation datasets. Both models were also evaluated on an independent synthetic test dataset. This test dataset was not used for training or model selection and serves only to illustrate expected predictive behavior in the flowsheet calibration workflow.

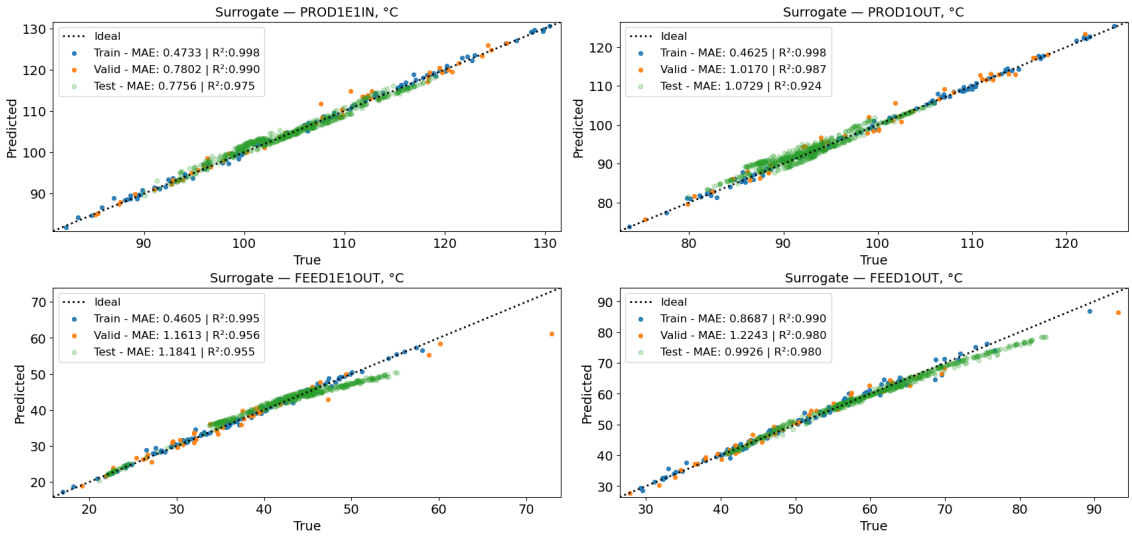


Figure 3.10: Predictive performance of the selected surrogate model on training (blue), validation (orange), and independent synthetic test (green) datasets.

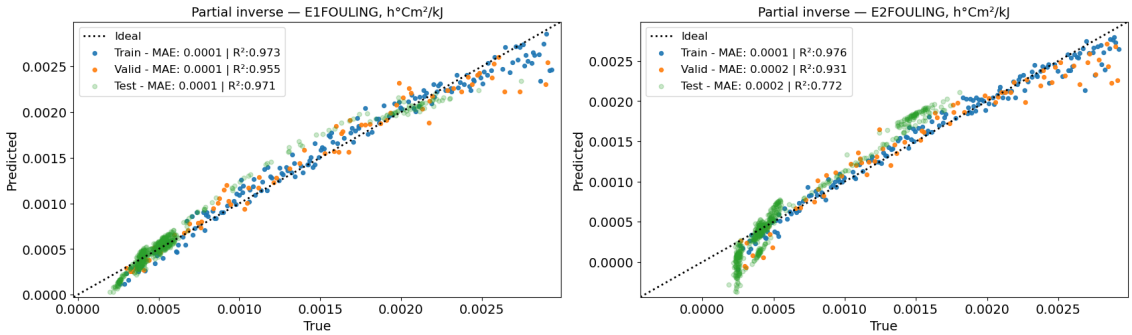


Figure 3.11: Predictive performance of the selected partial inverse model on training (blue), validation (orange), and independent synthetic test (green) datasets.

The results show that both models exhibit good agreement between predictions and reference values on the training and validation data, indicating stable learning behavior and effective regularization. On the independent test dataset, the surrogate model maintains high predictive accuracy, demonstrating robust generalization across the input space. In contrast, the partial inverse model exhibits a noticeable reduction in accuracy on the test data, reflecting the increased difficulty of the inverse mapping and the higher sensitivity of fouling factor estimation to input uncertainty.

3.4.3 Flowsheet model calibration

Effective surrogate-based calibration of flowsheet models requires careful formulation of the objective function and considered selection of the output variables included in the calibration process. In complex industrial models with many correlated outputs, it is often neither necessary nor desirable to include all variables in the objective function. Instead, calibration is typically focused on those outputs that exhibit strong sensitivity to the calibration parameters, in order to avoid redundancy and weak optimization problems.

In the present simplified case study, the relationship between the calibration parameters—namely the heat exchanger fouling factors—and the outlet temperatures is well understood from first-principles heat transfer considerations. Although strong correlations exist among some of the outlet temperature variables, all outputs were intentionally included in the objective function. This design choice enables a transparent demonstration of the influence of output weighting and the Taguchi loss formulation on calibration behavior, without the additional complexity of output selection strategies.

These modeling considerations were further refined through the selection of weighting factors and acceptance ranges within the objective function. The acceptance range, which defines the penalty threshold in the Taguchi loss formulation (Eq. 3.7), represents the combined effect of measurement and modeling uncertainty used to characterize the expected accuracy of the flowsheet calibration process.

In this case study, the test data were generated directly from the flowsheet model and therefore do not contain measurement noise or unmodeled disturbances typically present in industrial data. Under these controlled conditions, surrogate validation performance provides a practical substitute for the available accuracy of surrogate-assisted calibration. Accordingly, the acceptance range for each output variable k was defined using the mean absolute error of the surrogate model evaluated on the

validation dataset,

$$\sigma_k = \mathrm{MAE}_{k,\mathrm{val}}. \tag{3.14}$$

This formulation reflects the assumption that deviations within the validated surrogate error are acceptable, while larger discrepancies indicate meaningful calibration mismatch.

For the weighted objective function, the mean absolute percentage error ($\mathrm{MAPE}_{k,\mathrm{val}}$) computed on the validation dataset was used as a relative error measure for each output variable. The corresponding weights were calculated according to

$$w_k = \frac{\frac{1}{\mathrm{MAPE}_{k,\mathrm{val}}}}{\sum_{k=1}^{M} \frac{1}{\mathrm{MAPE}_{k,\mathrm{val}}}}, \tag{3.15}$$

assigning higher weight to outputs that can be predicted more reliably by the surrogate model. This weighting strategy prioritizes calibration consistency for variables with lower relative uncertainty while reducing the influence of noisier or less predictable outputs. The resulting weights are summarized in Table 3.3.

The following section presents the calibration results obtained using this objective function formulation and evaluates the effectiveness of the proposed surrogate-based calibration approach under evolving operating conditions.

Table 3.3: Objective function weights and Taguchi acceptance limits for the heat exchanger network case study.

Objective func. parameter	Unit	Surrogate MAE / MAPE	Weight	Taguchi acceptance limit
PROD1E1IN	°C	0.780 / 0.725	0.443	0.780
PROD1OUT	°C	1.017 / 1.042	0.308	1.017
FEED1E1OUT	°C	1.161 / 2.871	0.112	1.161
FEED1OUT	°C	1.224 / 2.351	0.137	1.224

3.4.4 Surrogate-based flowsheet model calibration results

The performance of the proposed surrogate-based calibration framework was evaluated using two alternative objective function formulations: a weighted formulation and a Taguchi loss-based formulation. In addition, calibration results were compared against two reference approaches: (i) a partial inverse model that directly predicts calibration parameters from process measurements, and (ii) direct optimization performed on the high-fidelity flowsheet model using the Taguchi-based objective function. The last one serves as a computational baseline, verifying that the objective function formulation itself enables accurate calibration and providing a reference against which surrogate-induced approximation errors can be assessed.

For all optimization-based calibration runs, PSO was employed with 20 particles. Given the limited complexity of the selected case study, this configuration provided a reasonable trade-off between solution robustness and computational effort. While increasing the number of particles can improve the likelihood of identifying a global optimum, it also proportionally increases computational cost; therefore, no universal rule exists, and the choice must be tailored to the specific calibration problem.

Figure 3.12 presents the calibrated flowsheet outputs obtained by applying the estimated calibration parameters $\hat{\boldsymbol{\theta}}$ to the flowsheet model. The black line represents the synthetic test data, while the orange and green lines correspond to surrogate-based calibration using the weighted and Taguchi loss functions, respectively. Results obtained from the partial inverse model are shown with a purple dotted line, and the cyan line denotes direct calibration using the flowsheet model. All methods demonstrate high predictive accuracy across the evaluated output variables. Because the test data were generated by simulation, process noise and unmodeled disturbances are absent, and no significant accuracy differences are observed among the calibration strategies at this level of model complexity.

Localized reductions in calibration accuracy are observed in regions where the surrogate or partial inverse models exhibit reduced predictive performance. These

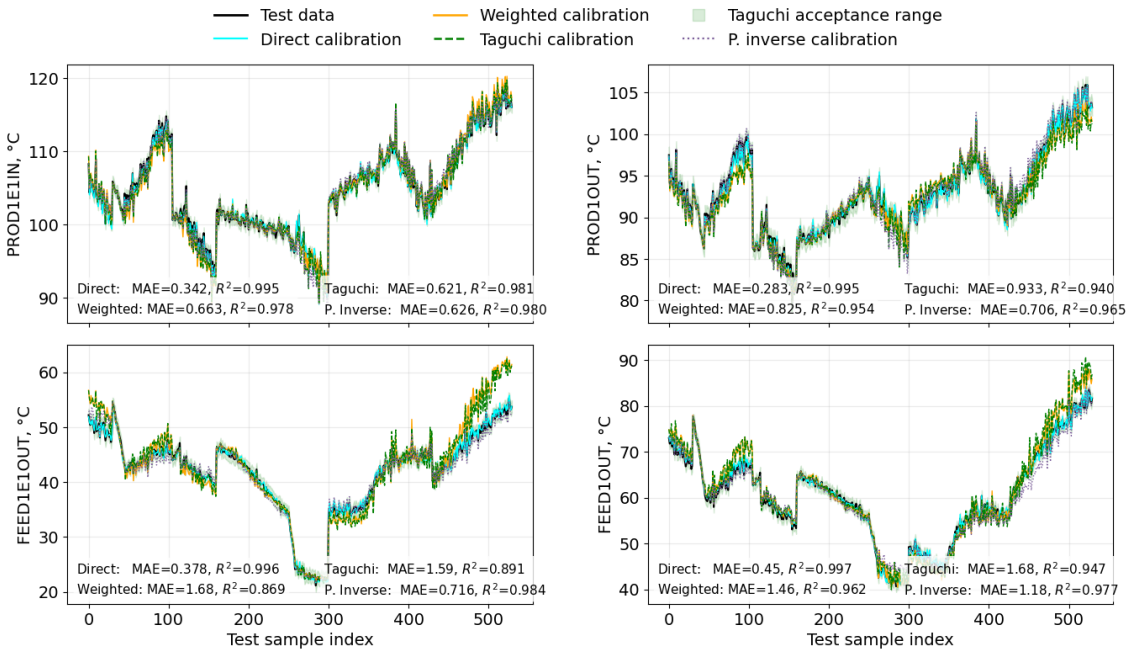


Figure 3.12: Comparison of accuracy of different calibration methodologies on simulation validation. Calibration MAE and R^2 metrics are reported in Appendix Table B.2.

discrepancies indicate insufficient representation of such regions in the training data and suggest that adaptive sampling strategies could further improve surrogate fidelity in localized operating regimes.

Figure 3.13 compares the true fouling factors used to generate the test data with the estimated calibration parameters obtained using the weighted surrogate approach, the Taguchi-based surrogate approach, the partial inverse model, and direct calibration. All approaches achieve accurate parameter estimation. However,

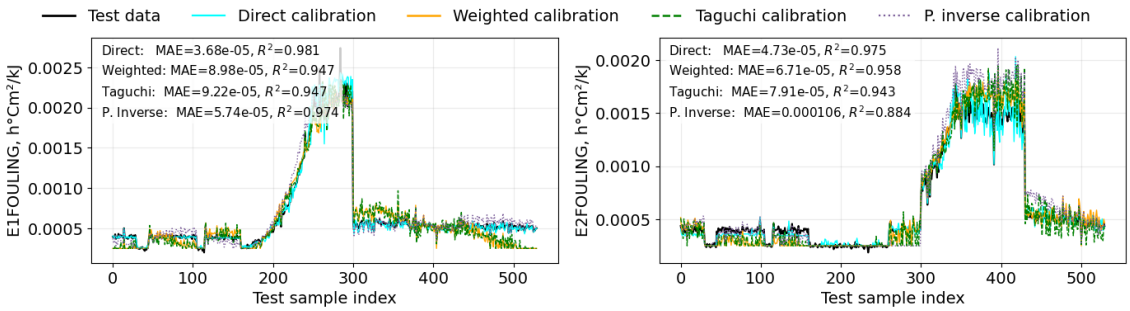


Figure 3.13: The original and estimated calibration parameters (fouling factors). Calibration MAE and R^2 metrics are reported in Appendix Table B.3.

reduced parameter estimation accuracy overlap with regions of lower surrogate or inverse model accuracy, confirming that surrogate fidelity is a key factor influencing calibration quality in addition to objective function formulation. Compared to weighted objectives, the Taguchi loss formulation offers improved interpretability by explicitly defining acceptable deviation ranges, making it more transparent and user-friendly for domain experts.

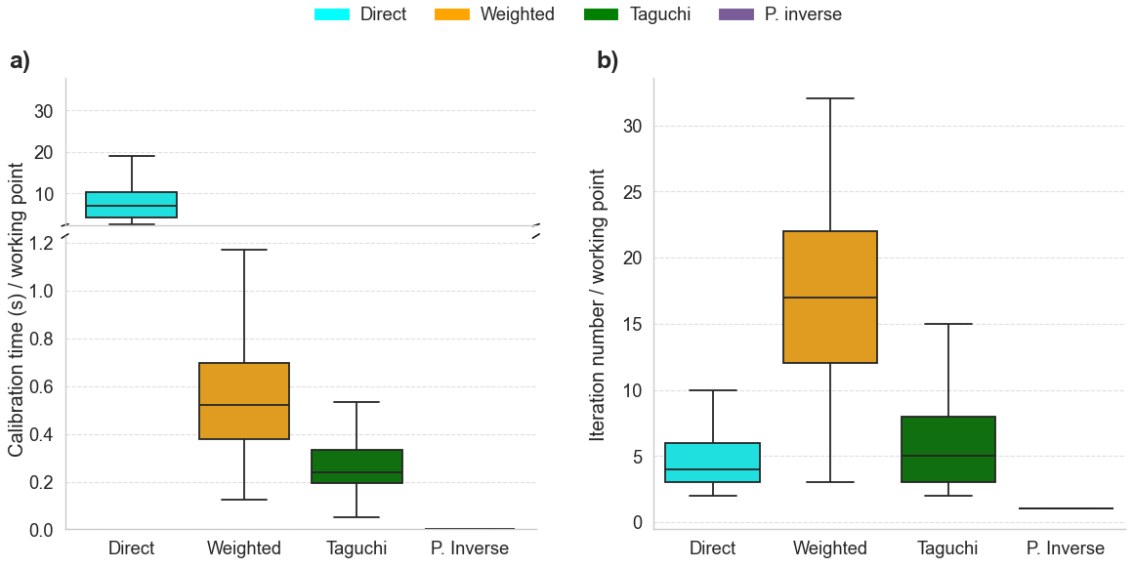


Figure 3.14: Comparison of computational performance of direct and surrogate-based calibration approaches in terms of (a) calibration time and (b) number of optimization iterations.

Computational performance is summarized in Figure 3.14. Direct calibration using the flowsheet model is by far the most computationally expensive approach due to the cost of repeated high-fidelity simulations and the risk of encountering non-convergent operating points during optimization. In contrast, surrogate-based calibration reduces computation time by orders of magnitude. While the weighted and Taguchi surrogate formulations exhibit comparable solution times, the Taguchi-based approach consistently requires fewer optimizer iterations due to its implicit early-stopping behavior once predictions fall within acceptable tolerance ranges.

The partial inverse model achieves the highest computational performance, as calibration parameters are obtained through direct prediction rather than optimiza-

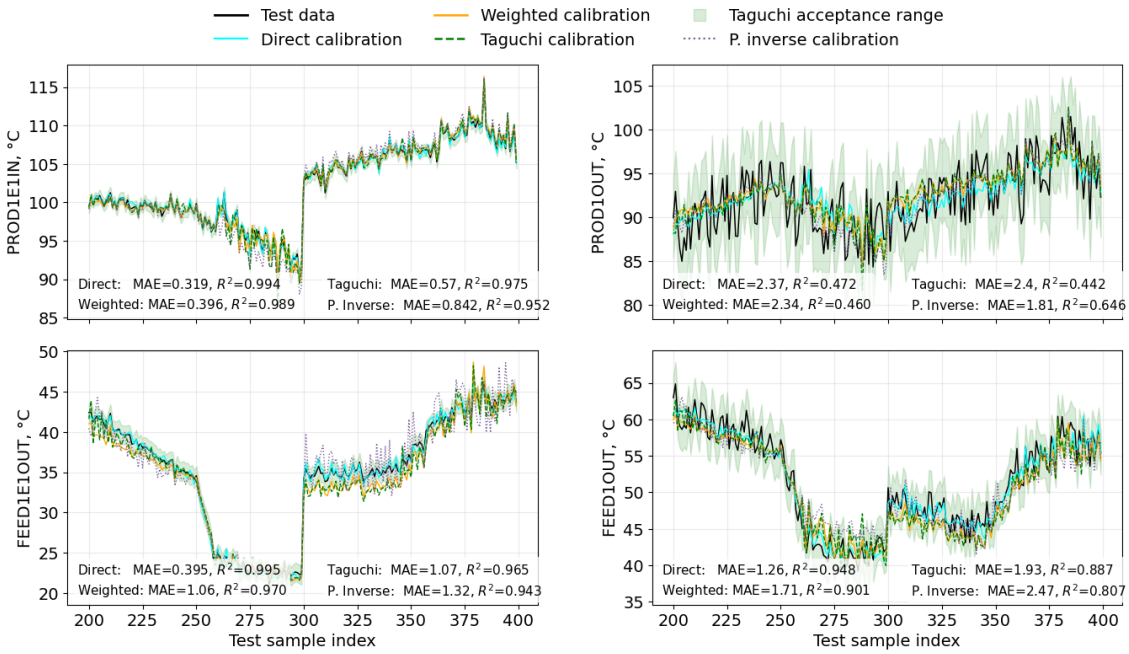


Figure 3.15: Comparison of accuracy of different calibration methodologies on simulation validation in noisy scenario. Calibration MAE and R^2 metrics are reported in Appendix Table B.4.

tion. However, this efficiency comes at the cost of reduced flexibility. The inverse model is trained specifically for flowsheet calibration and cannot easily incorporate modeling considerations, uncertainty handling, or changes in calibration priorities without retraining. In contrast, surrogate-based calibration remains adaptable by modifying the objective function while preserving the same surrogate model, making it more suitable for long-term DT deployment.

To further illustrate this flexibility, a noisy scenario was introduced by adding up to 5% random noise to selected output variables (`PROD1OUT` and `FEED1OUT`). In response, the surrogate-based calibration was adapted by reducing the corresponding weights in the weighted objective formulation and increasing the acceptance limits in the Taguchi loss formulation. Figure 3.15 demonstrates that surrogate-based methods remain stable under these conditions, whereas the partial inverse model exhibits increased variability (`FEED1E1OUT`). This behavior highlights a key limitation of inverse models: uncertainty handling and modeling considerations cannot be incorporated without retraining the estimator.

Overall, these results demonstrate that surrogate-based calibration provides a favorable balance between accuracy, computational efficiency, and flexibility. Compared to direct optimization, surrogate models dramatically reduce calibration time. Compared to partial inverse models, they offer substantially greater adaptability to modeling considerations, measurement uncertainty, and evolving calibration objectives. Among the tested objective function formulations, the Taguchi loss function emerges as particularly attractive from a practical perspective due to its transparency, interpretability, and reduced iteration count, aligning well with the principles of explainable and user-centric AI.

3.5 Summary and outlook

This chapter demonstrated that surrogate-based calibration provides an effective and flexible solution for automated flowsheet model maintenance in digital twin applications. By decoupling calibration from expensive flowsheet simulations, the proposed framework significantly reduces computational cost while preserving calibration accuracy across a range of operating conditions. Compared to direct calibration, surrogate-based approaches offer substantial performance gains, while avoiding the rigidity inherent to partial inverse models.

The results further show that objective function design plays a critical role in practical deployment. While both weighted and Taguchi-based formulations achieve high accuracy, the Taguchi loss function offers improved interpretability and ease of adjustment, making it better suited for user-guided calibration in industrial environments. Importantly, surrogate-based calibration enables uncertainty handling and modeling considerations to be incorporated directly into the optimization problem—capabilities that are not readily available in partial inverse-model-based approaches.

At the same time, the case study highlights a fundamental dependency of the proposed framework: calibration accuracy ultimately relies on the predictive quality of the surrogate model. Regions of reduced surrogate accuracy directly translate into degraded calibration performance. Moreover, as real industrial processes evolve over time, surrogate accuracy may deteriorate due to process drift and changing operating regimes.

These observations motivate the focus of the next chapter. To enable robust long-term deployment of surrogate-based calibration in industrial digital twins, surrogate models themselves must be maintained and updated incrementally. The following chapter therefore investigates online learning strategies designed to preserve surrogate accuracy under continuously evolving process conditions, enabling fully autonomous and sustainable DT operation.

Chapter 4

Online learning for incremental surrogate model maintenance

Surrogate models significantly reduce the computational cost of high-fidelity process simulations, but their predictive accuracy depends on the representativeness of the offline training data. In industrial operation, evolving process conditions often violate this assumption, leading to performance degradation over time. This chapter investigates online learning strategies for incremental surrogate model maintenance to ensure robust and reliable deployment under changing operating regimes.

4.1 Chapter motivation and scope

The increasing digitalization of the process industry, driven by Industry 4.0 technologies, has intensified the demand for continuously updated and reliable process models to support monitoring, maintenance, and optimization tasks [114]. Within this context, flowsheet-based process models remain a cornerstone of industrial decision support due to their strong physical interpretability and predictive capabilities [72].

As demonstrated in the preceding chapters, maintaining flowsheet model accuracy over extended periods is challenging when systems are subject to operational drift, equipment aging, and evolving operating conditions. While automated calibration

strategies can mitigate these effects, direct optimization of complex flowsheet models remains computationally demanding and often suffers from convergence issues [75, 17]. Surrogate models provide an effective mechanism to mitigate this computational burden by approximating the flowsheet response with significantly reduced evaluation cost.

However, the effectiveness of surrogate-based calibration depends strongly on the representativeness of the surrogate training data. Classical surrogate construction implicitly assumes that future operating conditions will remain within the bounds of the offline training domain. In real industrial operation, this assumption is frequently violated. Process conditions, including both process variables and model parameters, may gradually drift or unexpectedly shift, and previously unobserved operating regimes may emerge during long-term operation. As a consequence, surrogates trained exclusively offline may exhibit degraded predictive accuracy when deployed in dynamic environments.

This chapter addresses this limitation by focusing on online learning strategies for incremental surrogate model maintenance. The scope of the chapter is to analyze how different online update mechanisms can sustain surrogate accuracy under evolving conditions while balancing computational efficiency, adaptability, and long-term stability. Particular emphasis is placed on surrogate models used for flowsheet calibration within digital twin frameworks.

4.2 Surrogate model sampling strategies for evolving processes

The design of an appropriate sampling strategy is a central element of surrogate model development, as it defines where reliable predictions can be expected. In industrial applications, it must support accurate initial models and account for long-term process evolution.

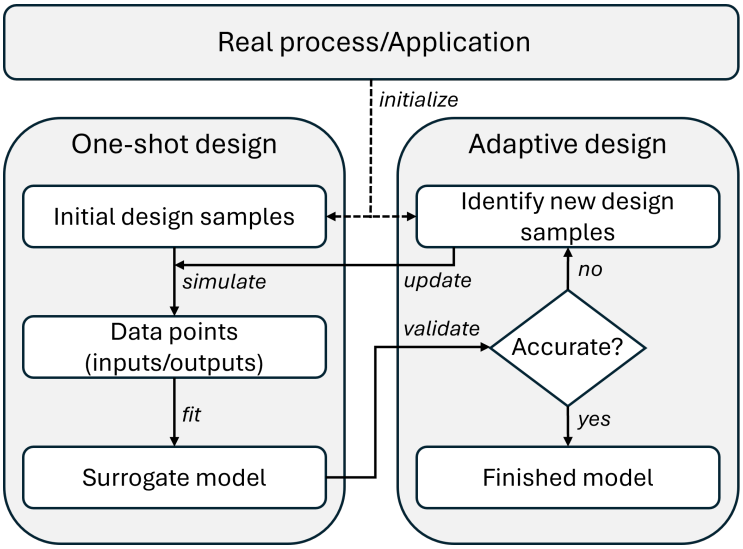


Figure 4.1: One-shot and adaptive methods for global surrogate model design.

4.2.1 Offline surrogate sampling strategies

Surrogate models are widely used to approximate computationally expensive simulations or experimental systems [26]. Their predictive performance depends strongly on the sampling strategy used to generate the training data. Traditionally, surrogate models are constructed using offline sampling of the underlying high-fidelity model, where all training data are generated prior to deployment. Common approaches include one-shot sampling methods, such as LHS [115], as well as adaptive sampling strategies that iteratively refine the training dataset [116, 117].

Offline surrogate construction typically requires predefined minimum and maximum bounds for each input parameter, often derived from nominal operating windows. While this is feasible for directly measurable process variables, it becomes problematic for derived or latent parameters—such as fouling factors or degradation coefficients—where prior knowledge is limited. If the assumed parameter ranges differ substantially from the values encountered during real operation, prediction performance of surrogate can be reduced significantly.

Figure 4.1 illustrates the two dominant classes of offline surrogate design strategies: one-shot and adaptive approaches.

One-shot offline surrogate construction provides a simple and robust starting point for model development, as all training data are generated in advance and the resulting surrogate exhibits predictable global coverage of the predefined design space. This makes one-shot approaches attractive in early design phases and for systems with well-characterized and stable operating ranges. However, their static nature limits their ability to capture localized nonlinearities efficiently and prevents any adaptation once the surrogate is deployed. Consequently, prediction accuracy may degrade rapidly when the process operates in regions that were insufficiently informative during offline sampling.

Adaptive offline strategies partially mitigate these shortcomings by concentrating sampling effort in regions where surrogate uncertainty or approximation error is high. By iteratively refining the training dataset, they typically achieve higher accuracy with fewer samples than one-shot approaches. Nevertheless, adaptive methods remain fundamentally offline in nature: they assume that the system behavior is stationary during surrogate construction and that all relevant operating regimes can be explored through repeated high-fidelity simulations. In industrial contexts, where operating conditions evolve continuously and simulation budgets are limited, this assumption becomes increasingly difficult to satisfy, and the computational overhead of repeated model evaluations can restrict practical applicability.

4.2.2 Limitations under evolving operating conditions

The assumptions underlying offline surrogate construction are frequently violated in industrial environments, where process conditions evolve over time due to operational drift, equipment aging, fouling, or changes in feedstock and operating objectives. Under such conditions, the statistical distribution of operational data gradually diverges from the offline training distribution.

These limitations are particularly critical in surrogate-based flowsheet model calibration, where both operating conditions and calibration parameters may drift

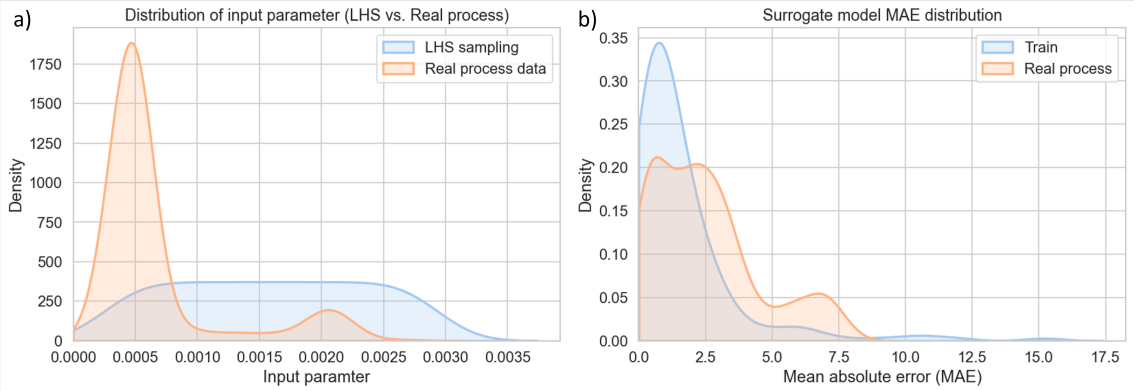


Figure 4.2: (a) Sampling range mismatch between offline model and real process. (b) Offline model prediction error on training data and real process data.

over time [118]. When the real process behavior deviates from the original training domain, the surrogate may no longer provide reliable predictions, leading to biased calibration results and degraded decision support.

Figure 4.2 illustrates the consequences of such distribution mismatch. Even if offline training error is low, prediction accuracy can deteriorate substantially when the surrogate is evaluated on real process data drawn from previously unobserved regions.

These observations highlight a fundamental limitation of purely offline surrogate modeling approaches: high offline accuracy does not guarantee reliable performance during long-term industrial deployment.

4.2.3 Motivation for online surrogate model updates

To sustain surrogate accuracy under evolving operating conditions, static offline training must be complemented by online update mechanisms. This requirement aligns closely with the objectives of MLOps, which emphasize maintaining reliable, monitored, and continuously updated models in production environments [119].

Online learning – also referred to as incremental learning—provides a natural framework for such adaptation [120, 121]. In an online learning setting, the surrogate model is updated sequentially as new high-fidelity data become available, avoiding

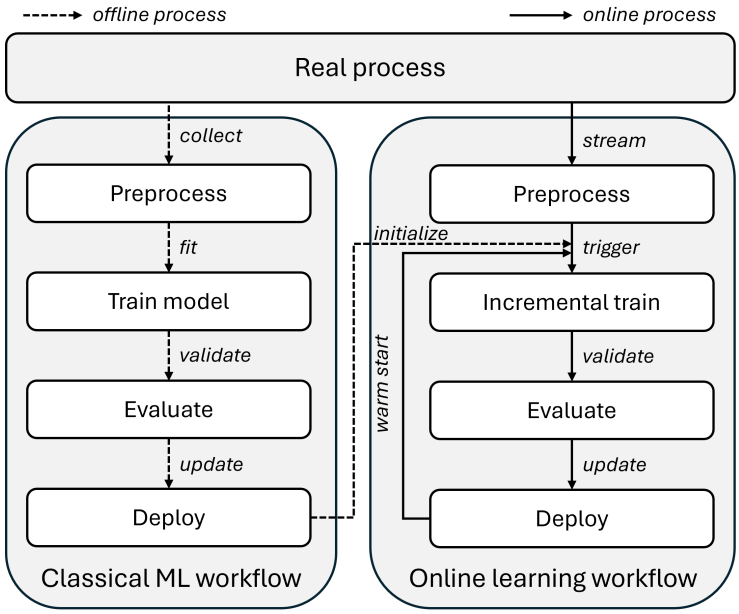


Figure 4.3: Illustration of classical offline training versus online learning methodologies.

complete retraining from scratch. This enables rapid adaptation with limited computational overhead and makes online learning particularly suitable for industrial monitoring, robotics, and interactive decision support systems [122, 123]. In the context of this work, the term "online data" refers specifically to continuously arriving process measurements or newly generated high-fidelity data, depending on the application, collected after the initial offline model training and deployment phase, while "offline data" refers to the dataset collected prior to application deployment.

As illustrated in Figure 4.3, in the case of online learning workflow, additional retraining are not necessarily initiated immediately after deployment, but are instead activated through predefined trigger conditions. These triggers may represent periodic update intervals, the availability of sufficient new operational data, or performance-related criteria such as increasing prediction error or model drift identified during operation. Data stream learning techniques can further regulate update frequency to balance predictive accuracy and computational cost [124].

In summary, while one-shot and adaptive offline sampling strategies provide a solid foundation for initial surrogate development, they are insufficient for long-term

deployment in evolving industrial environments. Online learning complements these approaches by enabling continuous surrogate refinement during operation, thereby improving robustness, adaptability, and long-term reliability within digital twin frameworks.

4.3 Online learning methodologies

Machine learning methods are widely used to extract information from data and to construct predictive models that support decision-making in complex systems. Classical batch learning approaches assume that all training data are available prior to model construction and can be processed simultaneously. While effective in static settings, this assumption becomes increasingly restrictive in practical applications, where data arrive sequentially and the underlying system behavior may evolve over time. In such cases, incorporating new information typically requires retraining the model from scratch, which is computationally expensive and may result in models that lag behind the current state of the system [120].

To address these limitations, a paradigm shift from static batch learning toward sequential data processing is required. By treating incoming data as a stream, learning algorithms can update models as soon as new information becomes available, enabling continuously up-to-date predictions while reducing storage and computational requirements [124]. This capability is particularly important in non-stationary environments, where timely adaptation is essential to maintain model reliability.

Incremental and online learning algorithms naturally support this paradigm, as they are designed to update model parameters continuously with minimal processing time and memory usage. Due to their ability to handle evolving data distributions and real-time learning scenarios, online learning methods have gained increasing attention in recent years, especially in applications characterized by large-scale, dynamic, or streaming data [125].

In the context of surrogate-assisted flowsheet calibration, this paradigm enables predictive models to be updated continuously as new calibration-relevant data become available, allowing the model to remain aligned with the current process behavior without repeated offline retraining.

Formally, the online learning process can be expressed as a sequence of incremental model updates:

$$s^{(1)}, s^{(2)}, \ldots, s^{(t)} \quad \Rightarrow \quad h^{(1)}, h^{(2)}, \ldots, h^{(t)} \tag{4.1}$$

where each training sample $s^{(t)} = (x^{(t)}, y^{(t)})$ consists of an input–output pair, and $h^{(t)}$ denotes the surrogate model after incorporating the t-th sample.

Each update depends on the previous model state and the newly observed sample:

$$h^{(t)} = f(h^{(t-1)}, s^{(t)}) \tag{4.2}$$

This formulation captures the core principle of online learning: the surrogate model evolves sequentially as new calibration-relevant information becomes available, optionally retaining a limited memory of past observations.

4.3.1 Benefits and challenges of online learning

In online learning, training proceeds sequentially, typically using individual samples or small batches, allowing models to respond rapidly to newly available information. This learning paradigm is well established in domains such as robotics, autonomous systems, and interactive learning environments, where continuous adaptation is a core requirement [122, 123].

One of the primary advantages of online learning is efficient resource utilization. By processing data incrementally, memory and computational requirements are significantly reduced compared to repeated batch retraining [126]. In addition, continuous adaptation enables models to remain relevant in non-stationary environments where data distributions and system dynamics change over time, such as industrial

processes, financial systems, or environmental forecasting [125].

Despite these advantages, online learning also introduces several challenges. A major concern is catastrophic forgetting, where models gradually lose previously acquired knowledge while adapting to new data [127]. Furthermore, handling concept drift—whether gradual or sudden changes occur in the underlying data distribution—often requires dedicated detection and adaptation strategies. Frequent updates based on limited recent data may also increase the risk of overfitting, highlighting the need to balance adaptability with robustness and generalization [128].

4.3.2 Types of online learning strategies

To ensure reliable long-term surrogate performance under evolving operating conditions, different online learning strategies can be employed, each offering a distinct trade-off between adaptability, memory retention, and computational cost. Five representative update mechanisms are considered in this work and conceptually illustrated in Figure 4.4. The presented strategies address two closely related aspects of online learning: (i) when model retraining or updating is triggered, and (ii) what data are used during the update process. The most commonly applied online learning approaches are represented by the incremental, mini-batch, and replay-based strategies shown in Figure 4.4-a–c. Incremental and mini-batch learning primarily differ in the amount of newly arriving online data used during each retraining step, while replay-based learning extends these approaches by partially reusing offline training samples during updates to mitigate catastrophic forgetting. In contrast, cumulative learning (Figure 4.4-d) corresponds to the traditional retraining approach, where the model is repeatedly retrained from scratch using both historical offline data and all accumulated online data. Additional aspects can also be incorporated into online learning strategies, such as conditional update triggering based on predictive performance, as illustrated in Figure 4.4-e, where retraining is activated only for operating regions associated with elevated prediction errors. In practice, these mechanisms are

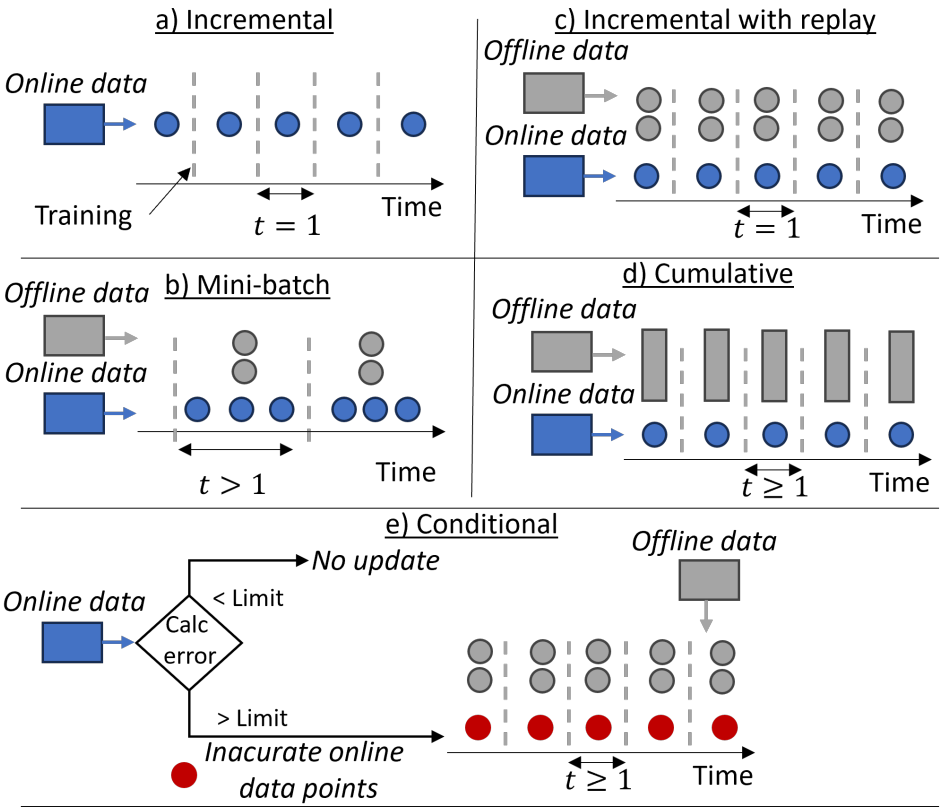


Figure 4.4: Conceptual comparison of online learning update strategies.

not mutually exclusive and can be combined depending on application requirements.

All considered strategies can be expressed within a common online learning formulation, where the model is updated sequentially as new data become available. Let $h^{(t)}$ denote the model after the t-th update, and $\mathcal{D}_{\text{train}}^{(t)}$ the dataset used at that update step. The generic online update can then be written as

$$h^{(t)} = \mathcal{U}\left(h^{(t-1)}, \mathcal{D}_{\text{train}}^{(t)}\right), \tag{4.3}$$

where $\mathcal{U}(\cdot)$ denotes a generic learning operator. The different online learning strategies considered in this work primarily differ in the composition of $\mathcal{D}_{\text{train}}^{(t)}$, as detailed below.

- **Incremental**: The model is updated with each newly arriving data point ($\mathcal{D}_{\text{new}}$), using only the most recent observation for training. This strategy enables immediate adaptation to changing conditions but is particularly sensitive

to catastrophic forgetting due to the absence of explicit memory retention mechanisms [124].

$$\mathcal{D}_{\text{train}}^{(t)} = \mathcal{D}_{\text{new}}^{(t)} \tag{4.4}$$

- **Incremental replay**: Each update incorporates the newly observed data $\mathcal{D}_{\text{new}}^{(t)}$ together with a small fraction r of the original offline training dataset $\mathcal{D}_{\text{offline}}$. This approach mitigates catastrophic forgetting by explicitly reinforcing previously learned patterns while still allowing adaptation to new operating regions [129].

$$\mathcal{D}_{\text{train}}^{(t)} = \mathcal{D}_{\text{new}}^{(t)} \cup \mathcal{D}_{\text{offline}}^{(r)} \tag{4.5}$$

- **Mini-batch**: Updates are performed after accumulating a small batch of size B of new samples ($\mathcal{D}_{\text{new}}$). This buffering reduces sensitivity to individual data points and smooths stochastic fluctuations, providing improved numerical stability compared to purely incremental updates [120].

$$\mathcal{D}_{\text{train}}^{(t)} = \bigcup_{i=t-B+1}^{t} \mathcal{D}_{\text{new}}^{(i)} \tag{4.6}$$

- **Cumulative**: The model is retrained using the complete offline dataset ($\mathcal{D}_{\text{offline}}$) augmented with all newly acquired samples ($\mathcal{D}_{\text{new}}$). This strategy provides strong memory retention and stable global performance, but its computational cost increases monotonically over time, limiting scalability in long-term operation.

$$\mathcal{D}_{\text{train}}^{(t)} = \mathcal{D}_{\text{offline}} \cup \bigcup_{i=1}^{t} \mathcal{D}_{\text{new}}^{(i)} \tag{4.7}$$

- **Conditional**: Model updates are governed by predictive performance at each incoming working point. At update step t, the calibration objective function $F\left(h^{(t-1)}, \mathbf{x}^{(t)}\right)$ is evaluated. Since $F(\cdot)$ is formulated as a minimization objective, larger values indicate insufficient predictive accuracy.

If the objective value exceeds a predefined practical threshold F_{thr}, the model is updated incrementally using the newly arriving data point in order to improve local predictive performance. If the objective value remains below the threshold, no update is performed at that step, thereby avoiding unnecessary adaptation in regions that are already well represented.

$$\mathcal{D}_{\text{train}}^{(t)} = \begin{cases} \mathcal{D}_{\text{new}}^{(t)}, & \text{if } F\Big(h^{(t-1)}, \mathbf{x}^{(t)}\Big) > F_{\text{thr}}, \\ \varnothing, & \text{if } F\Big(h^{(t-1)}, \mathbf{x}^{(t)}\Big) \leq F_{\text{thr}}. \end{cases} \tag{4.8}$$

The practical accuracy threshold is defined by application-specific requirements rather than absolute numerical optimality. Pursuing maximal accuracy at all times does not necessarily improve calibration outcomes and may increase computational cost or the risk of overfitting, which in turn can contribute to catastrophic forgetting in the online learning process.

From a methodological perspective, the presented strategies can be interpreted within a unified mini-batch framework, which also encompasses replay-based mechanisms by augmenting the mini-batch with offline stored samples. Incremental learning corresponds to a special case of mini-batch learning with a batch size of $B = 1$, providing immediate adaptation at the cost of increased sensitivity to noise and forgetting. Similarly, both cumulative and conditional retraining can be naturally integrated with mini-batch learning, where the mini-batch mechanism controls the number of newly collected samples required before a model update is triggered. In this sense, cumulative and conditional strategies primarily differ in how and when model updates are activated, while mini-batching governs the temporal aggregation of incoming data. This unified view highlights that the choice among online learning strategies is largely a design decision balancing adaptation speed, computational efficiency, and long-term model stability.

4.3.3 Performance Metrics

To evaluate the effectiveness of online surrogate updates, two complementary performance metrics are defined using the calibration objective function F introduced in Eq. 3.7.

The first metric is the *local calibration improvement* (4.9), inspired by the Mean Absolute Scaled Error (MASE) [130]. After each update, the surrogate $h^{(t)}$ is evaluated at the subsequent operating point $\mathbf{x}^{(t+1)}$. Improvement relative to the previous model $h^{(t-1)}$ is quantified as:

$$\rho_F^{(t)} = \frac{F(h^{(t-1)}, \mathbf{x}^{(t+1)})}{F(h^{(t)}, \mathbf{x}^{(t+1)})} \tag{4.9}$$

Values $\rho_F^{(t)} > 1$ indicate improvement. Averaging $\rho_F^{(t)}$ across update steps reveals the overall improvement trend.

The second metric is the *global accuracy retention* (Eq. 4.10). To assess whether the updated surrogate retains its original predictive capability, each fine-tuned model $h^{(t)}$ is evaluated on the original offline dataset $\mathcal{D}_{\text{offline}}$ using the same calibration cost function F as in Eq. 3.7. The ratio of the baseline model's ($h^{(0)}$) error (obtained after offline training) to the current model's error is computed as:

$$\gamma_F^{(t)} = \frac{F\left(h^{(0)}, \mathcal{D}_{\text{offline}}\right)}{F\left(h^{(t)}, \mathcal{D}_{\text{offline}}\right)} \tag{4.10}$$

A value $\gamma_F^{(t)} \approx 1$ indicates that the updated model has preserved its original offline accuracy. If $\gamma_F^{(t)} < 1$, the model is beginning to forget, whereas $\gamma_F^{(t)} > 1$ suggests that the updates have further improved global accuracy. Averaging $\gamma_F^{(t)}$ across updates provides a long-term view of retention trends.

4.4 Case study for evaluating online learning

To evaluate the effectiveness of different online learning strategies, this case study focuses exclusively on surrogate model maintenance rather than on the full surrogate-assisted flowsheet calibration workflow. The objective is to isolate and analyze the impact of surrogate updates under evolving data conditions. To this end, it is assumed that the underlying flowsheet model accurately represents the true process behavior across all operating points. This assumption allows the influence of online surrogate updates to be examined independently of flowsheet calibration errors.

The analysis is based on the same simplified heat exchanger network model introduced in Chapter 3.4. Synthetic test data generated from this model are treated as representative of ideal operational measurements. Under the simplifying assumption that the flowsheet model can be calibrated perfectly, the generated test data—including the corresponding calibration factor values—are considered suitable for retraining and updating the surrogate model. This abstraction enables a focused investigation of surrogate retraining behavior and facilitates a controlled comparison of different online learning strategies.

The surrogate model architecture and offline training setup are identical to those used in the surrogate-based flowsheet calibration study presented in Chapter 3.4.3. Specifically, a global ANN with a single hidden layer consisting of 128 neurons was trained on 160 offline LHS data points. This model serves as the baseline offline surrogate, denoted as $h^{(0)}$.

To establish an upper-bound reference for achievable model accuracy, an *ideal surrogate model* was also trained using only the synthetic test dataset. This reference model employs the same ANN architecture, training procedure, and a 70–30 train–validation split as the baseline surrogate. As such, it represents the best possible predictive performance that could be achieved with the given model structure if the training data were perfectly representative of the operational regime. The comparison between the baseline offline surrogate and the ideal surrogate highlights

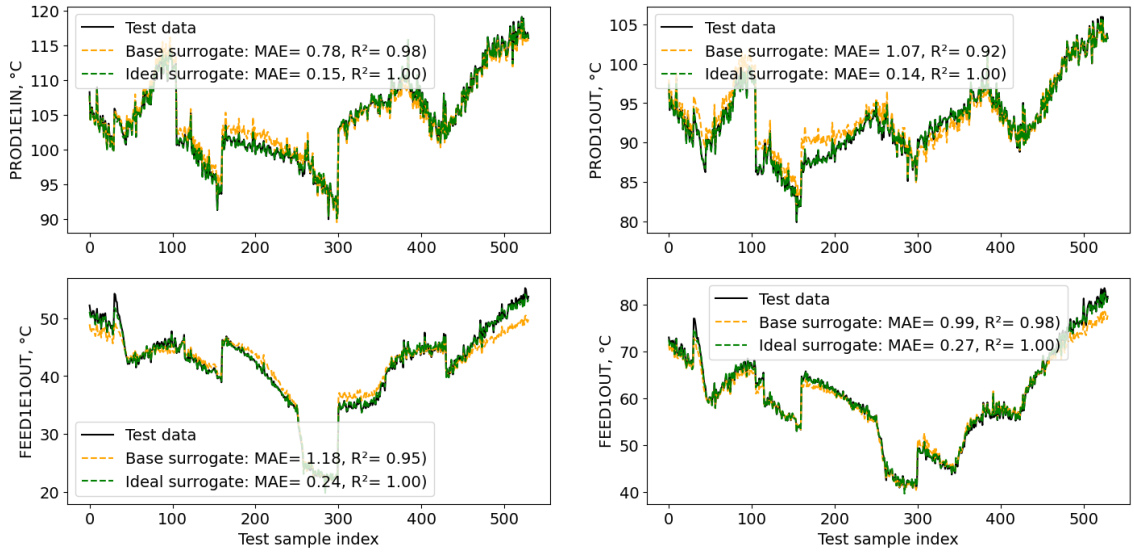


Figure 4.5: Comparison of baseline offline surrogate performance and ideal surrogate performance on the test dataset.

the performance gap that online learning aims to progressively reduce through incremental data incorporation.

As shown in Figure 4.5, the ideal surrogate model achieves near-perfect predictive accuracy on the test dataset, with an average R^2 of 1.00 and a MAE of 0.2. While the baseline offline surrogate exhibits reasonable overall performance (MAE of 1.01), it displays significantly higher errors in certain local regions of the input space. In the surrogate-based calibration workflow, such localized inaccuracies lead to suboptimal calibration results and increased reliance on high-fidelity flowsheet evaluations.

The baseline surrogate model was implemented using the `MLPRegressor` class from the `scikit-learn` library, which supports incremental parameter updates via the `partial_fit()` method. This enables progressive model refinement without reinitializing the network weights and is therefore well suited for online learning. Accordingly, all online learning strategies except the cumulative approach use `partial_fit()`. In contrast, the cumulative strategy retrains a new model from scratch using the aggregated dataset at each update step. This distinction enables a direct comparison between online updates and full retraining in terms of adaptability and computational efficiency.

4.4.1 Analyzed retraining options

To evaluate the behavior of different online learning strategies under realistic operating conditions, the performance of the baseline surrogate model was assessed using the test dataset in a sequential manner, emulating a streaming data scenario. New data points were introduced one by one, corresponding to successive working points of the process. This setup reflects practical deployment conditions, where new observations become available incrementally during operation.

At each working point, surrogate performance was evaluated in two complementary ways. First, predictive accuracy at the newly observed data point was assessed to quantify the model's local adaptation capability using the local calibration improvement metric. Second, the model was evaluated on the original offline dataset to assess its ability to retain previously learned knowledge, measured through the global accuracy retention metric. Together, these metrics provide insight into the trade-off between adaptability and robustness for each retraining strategy.

Based on the online learning methodologies introduced in Section 4.2, the following retraining configurations were analyzed:

- **Incremental learning**: Model updates were performed at every newly arriving data point. To investigate the effect of memory retention, replay buffers containing 0, 1, 2, 3, or 4 samples drawn from the original offline dataset were incorporated during each update step.

- **Mini-batch learning**: Model updates were carried out after accumulating batches of $B = 2, 5$, and 10 new data points. For each batch configuration, an equal number of replay samples from the original offline dataset was included in the update (e.g., 5 new samples combined with 5 replay samples), ensuring a consistent balance between adaptation to new data and retention of prior knowledge. The choice of a 1:1 ratio was initially heuristic; however, its effectiveness is supported by the results presented later (Figure 4.6), where

this balance was found to provide stable performance.

- **Cumulative learning**: The surrogate model was retrained from scratch at each update step using the complete offline dataset augmented with all newly acquired data points. This configuration serves as a reference representing the classical offline retraining approach. Different batch sizes were evaluated to reduce the frequency of full model retraining.

- **Conditional learning**: Model updates were performed selectively based on predictive performance at each working point. Incremental retraining was triggered only when the calibration objective indicated insufficient accuracy; otherwise, the model state was left unchanged. This conditional strategy was evaluated using different batch sizes B and replay buffer, allowing the effect of selective retraining intensity and memory reinforcement to be systematically assessed. By restricting updates to poorly predicted operating regions, this approach aims to improve local accuracy while avoiding unnecessary model adaptation.

This systematic exploration of retraining strategies enables a comprehensive comparison of different update mechanisms with respect to adaptation speed, robustness against catastrophic forgetting, and computational efficiency.

In this work, the replay fraction r was sampled uniformly at random from the offline dataset. This simple approach was sufficient for the current regression-based calibration problem, where the process does not exhibit discrete operational modes. In scenarios involving multimodal distributions or sharply separated regimes, more informed replay-buffer construction may be advantageous. Continual learning research provides several examples of such strategies. For instance, Gradient Coreset Replay (GCR) selects replay samples that best approximate the full gradient of historical data, maintaining knowledge of earlier tasks with a minimal subset that preserves the essential gradient information [131]. Likewise, methods such as Supervised

Contrastive Replay (SCR) prioritize samples that preserve embedding structure [132]. Although these methods were developed primarily for classification settings, the underlying principles may be transferable to surrogate model retraining tasks with more complex operational landscapes. The investigation of such approaches may form the basis of future research, particularly in more complex systems where advanced replay strategies may be required to ensure robust model performance.

4.4.2 Results and discussion

This section evaluates the performance of different different update mechanisms for surrogate model maintenance, with particular emphasis on the trade-off between fast local adaptation to new working points and retention of global predictive accuracy learned from the original offline dataset. The analysis is structured around four learning strategies: incremental, mini-batch, cumulative, and conditional retraining. Performance is assessed using (i) a local calibration improvement metric and (ii) global accuracy retention evaluated on the offline test dataset.

Incremental learning

In incremental learning, the surrogate model is updated after each newly arriving data point. The expected behavior of this strategy is rapid local adaptation, accompanied by a risk of catastrophic forgetting if the model is exposed exclusively to new operating conditions. Figure 4.6 illustrates this behavior.

The results indicate that the local improvement metric remains consistently greater than one, indicating successful adaptation to each new working point. However, without replay, global accuracy retention greatly degrades, confirming that incremental updates alone are insufficient to preserve previously learned information. Introducing different replay buffers effectively mitigates this effect, however slightly decreasing the adaptability as well. With replay, the model maintains global accuracy comparable to the baseline while preserving strong local adaptivity.

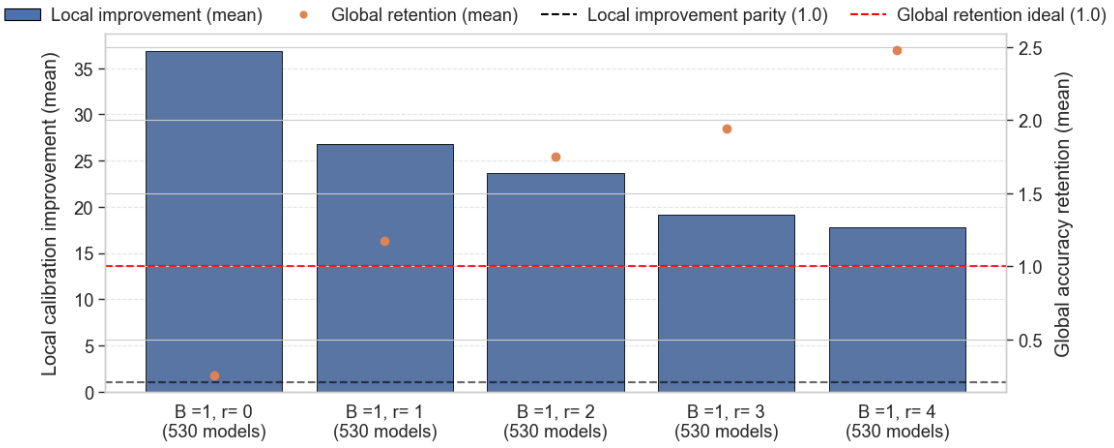


Figure 4.6: Effect of incremental learning on local calibration improvement and global accuracy retention (B: batch size, r: replay buffer size). The x-axis labels also indicate the number of model updates performed for each configuration (e.g., 530 updates).

This result is particularly relevant from an industrial perspective, as it demonstrates that catastrophic forgetting can be effectively mitigated through the use of a replay buffer. The incremental strategy with minimal replay therefore represents a highly efficient solution for continuously operating digital twin environments.

Mini-batch learning

Mini-batch learning reduces the frequency of model updates by aggregating multiple new data points before retraining. While this decreases computational effort, it also delays adaptation to changes in operating conditions. The corresponding results are shown in Figure 4.7.

Local improvement values remain above one, confirming that the surrogate adapts once retraining occurs. However, compared to incremental learning, adaptation is less responsive. The larger update steps lead to bigger jumps in the local improvement metric. This highlights an important limitation of the local improvement metric: it is best suited for comparing strategies with similar update frequencies and should be interpreted cautiously when batch sizes differ.

Without replay, mini-batch learning also exhibits clear forgetting of the offline

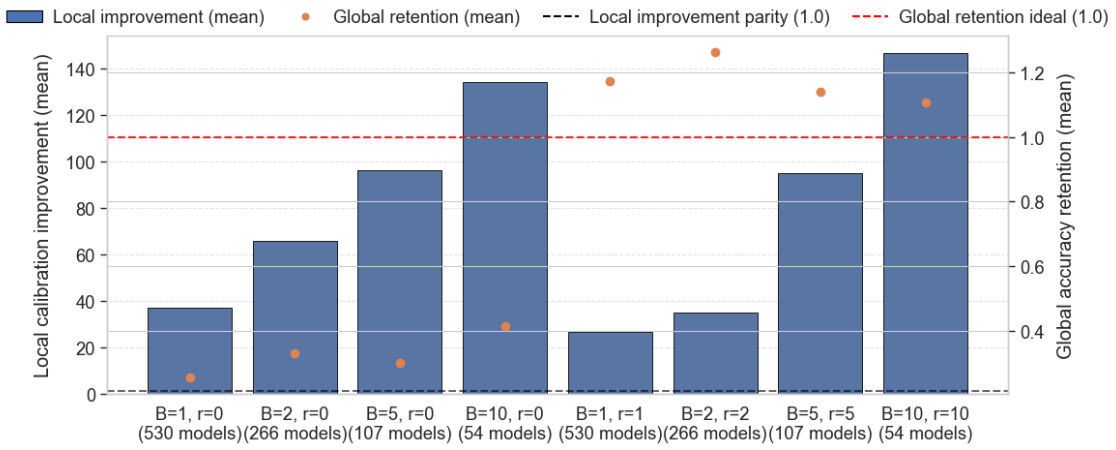


Figure 4.7: Effect of mini-batch learning on local calibration improvement and global accuracy retention (B: batch size, r: replay buffer size). The x-axis labels also indicate the number of model updates performed for each configuration (e.g., 530 updates).

dataset. When a replay buffer of equal size to the incoming batch is introduced, global accuracy retention improves significantly and approaches that of the incremental strategy. These results indicate that replay is equally essential for mini-batch learning, but larger replay buffers are required to compensate for the reduced update frequency. In the present study, a 1:1 ratio between new and replay samples was applied for higher batch sizes, and the results demonstrate that this configuration achieves near-ideal global retention metrics while maintaining effective adaptation.

Cumulative learning

In cumulative learning, the surrogate model is retrained from scratch using the union of the original offline dataset and all newly collected data. The network architecture was kept fixed throughout the analysis. The results are shown in Figure 4.8.

Local improvement is consistently observed; however, its magnitude is strongly influenced by batch size, similarly to the mini-batch case. As a result, direct comparison of local improvement values between cumulative and incremental learning is not meaningful. More importantly, cumulative learning exhibits systematically lower global accuracy retention across all configurations.

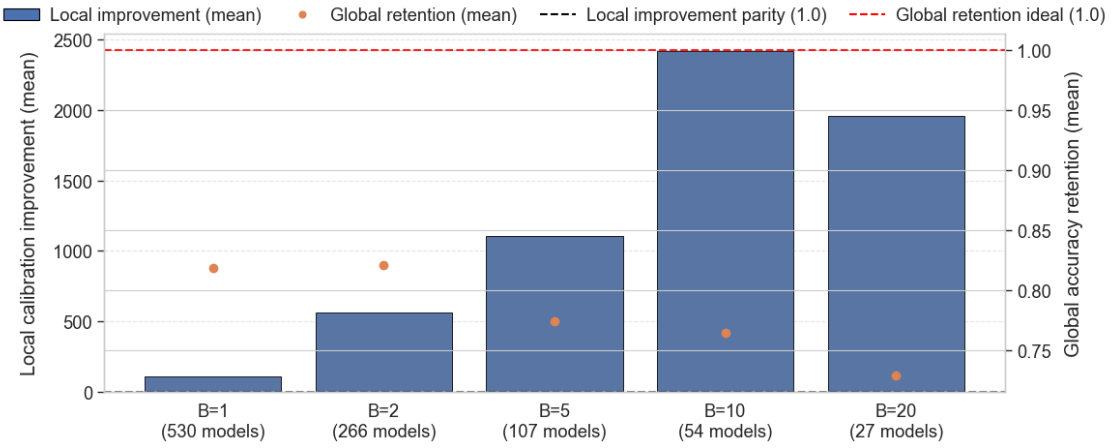


Figure 4.8: Effect of cumulative learning on local calibration improvement and global accuracy retention (B: batch size, r: replay buffer size). The x-axis labels also indicate the number of model updates performed for each configuration (e.g., 530 updates).

This behavior can be explained by the imbalance between the offline dataset and the growing set of locally clustered new samples. Since the offline test set contains substantially more points than the original training set, the retrained model increasingly overfits the recently observed operating region, effectively underrepresenting the original global domain. These results highlight a fundamental limitation of cumulative retraining in long-term operation: while conceptually simple, it does not provide a robust mechanism for balancing historical and newly acquired knowledge.

A potential direction for further improvement is the introduction of weighted loss formulations, in which newly acquired and historical samples contribute unequally to the training objective, allowing the influence of local adaptation to be controlled explicitly [133]. Alternatively, controlled replay strategies or regularization-based constraints that penalize deviation from previously learned global behavior could be employed to mitigate imbalance effects and improve long-term stability under non-uniform data arrival [134].

Conditional retraining

Conditional retraining introduces a decision layer that triggers model updates only when predictive performance degrades beyond a predefined threshold. In this study, the threshold value $F_{\text{thr}} = 0.05$ was selected based on the density distribution of the incremental learning objective (Appendix C.1), ensuring that retraining is initiated only for high-error working points.

As shown in Figure 4.9, this strategy significantly reduces the number of retraining events. Local improvement is achieved in all configurations; however, its magnitude is generally higher than that observed for mini-batch learning with comparable batch sizes, since not every new data point triggers a model update. The higher local improvement values may occur because retraining is activated exclusively at high-error working points, which can inflate the local improvement metric. When the batch size is one, such high-error points are detected immediately, enabling rapid local adaptation. In contrast, for larger batch sizes, retraining is delayed until multiple samples are accumulated, which reduces local responsiveness.

Global accuracy retention remains below ideal in most cases. This indicates that the replay buffer sizes adopted from the mini-batch experiments are insufficient to counteract forgetting when updates are infrequent and focused on high-error working points. Conditional retraining represents a computationally efficient but highly sensitive to thresholds and replay design, and increasing the replay ratio ($r > B$) may improve retention.

Overall comparison and discussion

All evaluated strategies achieved local improvement values greater than one, confirming that each method is capable of learning from newly arriving data. However, without replay mechanisms, all online strategies suffer from catastrophic forgetting, leading to a gradual loss of global predictive accuracy. Cumulative learning additionally exhibits a structural imbalance between the original offline dataset

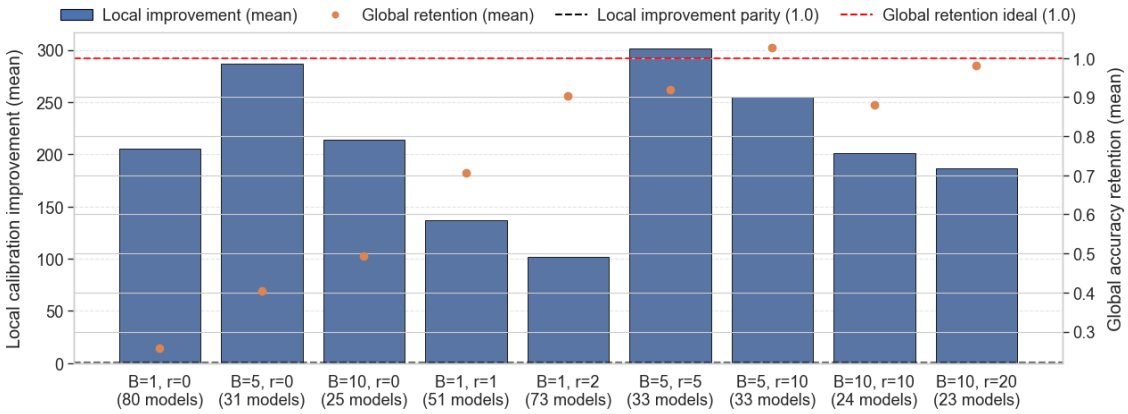


Figure 4.9: Effect of conditional learning on local calibration improvement and global accuracy retention (B: batch size, r: replay buffer size). The x-axis labels also indicate the number of model updates performed for each configuration (e.g., 80 updates).

and the continuously growing set of newly collected samples, resulting in persistent degradation of global accuracy over long-term operation.

Replay buffers consistently improved global accuracy retention across all applicable strategies, even when their size was minimal. Since the local improvement metric is most meaningful when update frequencies are comparable, additional global performance indicators—including MAE and R^2—were evaluated for all configurations to enable a fair comparison across strategies. A summary of the results is provided in Table 4.1.

Table 4.1: Summary of surrogate retraining results (B: batch size, r: replay size).

Strategy	Configuration	Models	Local impr.	Global ret.	MAE	R^2
Ideal	Ideal Model	1	-	-	0.20	1.00
Base	Base Model	1	-	-	1.01	0.96
Incremental	B =1, r= 0	530	36.87	0.26	0.37	0.99
Incremental	B =1, r= 1	530	26.75	1.17	0.39	0.99
Incremental	**B =1, r= 2**	**530**	**23.66**	**1.75**	**0.34**	**0.99**
Incremental	B =1, r= 3	530	19.13	1.94	0.36	0.99

Continued on next page

Strategy	Configuration	Models	Local impr.	Global ret.	MAE	R^2
Incremental	B =1, r= 4	530	17.74	2.48	0.36	0.99
Mini-batch	B =2, r= 0	266	65.92	0.33	0.46	0.99
Mini-batch	B =5, r= 0	107	96.15	0.30	0.57	0.98
Mini-batch	B =10, r= 0	54	134.09	0.41	0.66	0.98
Mini-batch	B =2, r= 2	266	34.74	1.26	0.41	0.99
Mini-batch	B =5, r= 5	107	94.79	1.14	0.53	0.99
Mini-batch	B =10, r= 10	54	146.69	1.11	0.61	0.98
Cumulative	B =1	530	105.04	0.82	0.40	0.99
Cumulative	B =2	266	564.04	0.82	0.41	0.99
Cumulative	B =5	107	1105.19	0.77	0.45	0.99
Cumulative	B =10	54	2420.52	0.76	0.50	0.99
Cumulative	B =20	27	1958.39	0.73	0.59	0.99
Conditional	B =1, r= 0	80	205.65	0.26	0.55	0.99
Conditional	B =5, r= 0	31	286.47	0.41	0.65	0.98
Conditional	B =10, r= 0	25	213.75	0.49	0.75	0.98
Conditional	B =1, r= 1	51	136.58	0.71	0.48	0.99
Conditional	B =1, r= 2	73	101.99	0.90	0.53	0.99
Conditional	B =5, r= 5	33	301.46	0.92	0.64	0.99
Conditional	B =5, r= 10	33	254.73	1.03	0.65	0.98
Conditional	B =10, r= 10	24	201.48	0.88	0.72	0.98
Conditional	B =10, r= 20	23	186.22	0.98	0.71	0.98

Figure 4.10 shows that all evaluated configurations achieve lower MAE and higher R^2 values than the base surrogate on test data, confirming the trends indicated by the local improvement metric. While none of the online strategies fully reach the performance of the ideal reference model, their results approach this upper bound closely. It should be noted that the ideal model was trained directly on the test dataset and therefore represents a theoretical maximum achievable accuracy rather

than a practically deployable solution.

Among the evaluated approaches, incremental learning with replay buffers demonstrates the most favorable balance between rapid adaptation and global stability, achieving the lowest MAE values across configurations. Mini-batch learning trades local responsiveness for reduced retraining frequency, resulting in intermediate performance. Cumulative learning yields results comparable to mini-batch strategies in terms of local improvement; however, its global accuracy retention deteriorates due to the increasing imbalance between historical and newly acquired data.

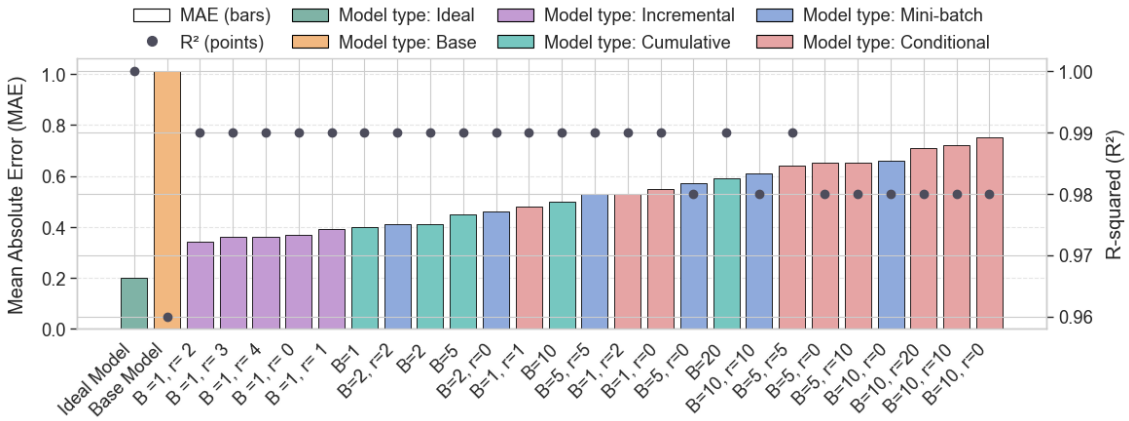


Figure 4.10: MAE and R^2 comparison of analyzed surrogate retraining methods (B: batch size, r: replay buffer size).

Conditional learning yields the smallest accuracy gains among online strategies but remains consistently better than the base model. It is therefore a computationally efficient, accuracy-sensitive alternative, strongly dependent on threshold selection and replay buffer design. Across all strategies, replay mechanisms substantially mitigate catastrophic forgetting.

Figure 4.11 compares the best-performing configuration – incremental learning with a replay buffer ($B = 1$, $r = 2$) – against both the original offline surrogate and an idealized offline-retrained reference. The results indicate that appropriately designed online learning strategies can closely approach offline retraining performance while operating under realistic computational constraints.

The results demonstrate that incremental learning represents a particularly

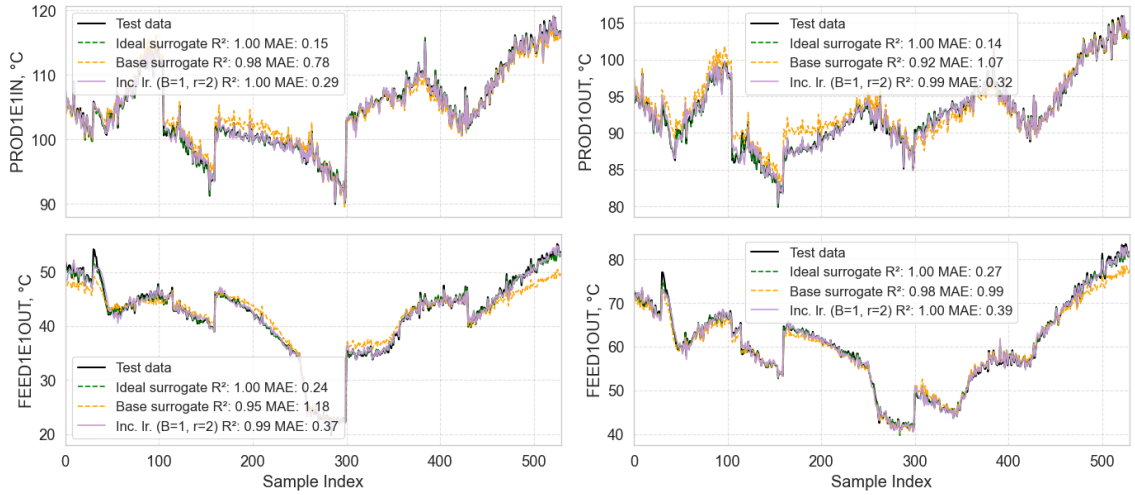


Figure 4.11: Comparison of selected online learning strategies against the baseline surrogate and an ideal offline-trained reference.

promising strategy for online surrogate model maintenance; however, no universal rule exists for selecting optimal replay buffer sizes, batch sizes, or retraining frequencies, as the appropriate configuration is inherently application-dependent. Future developments may therefore focus on hybrid maintenance strategies that combine incremental or mini-batch updates with conditional retraining mechanisms and adaptive replay sizing. In such approaches, minor model updates could continuously adapt the surrogate to newly observed operating conditions, while periodic or condition-triggered cumulative retraining could be used to restore global predictive performance when significant model drift or degradation is detected. Continuous monitoring of both local and global prediction metrics may therefore play a key role in determining when more comprehensive model retraining becomes necessary.

Beyond replay-based solutions, more advanced continual learning techniques – such as Elastic Weight Consolidation (EWC) – represent promising directions for future investigation, particularly in scenarios where long-term memory preservation is critical [135]. Another promising avenue is the development of task-aware sampling strategies that exploit the gradual evolution of calibration parameters in real processes, enabling focused local exploration while maintaining sufficient global coverage [136].

Finally, these results naturally motivate the integration of online surrogate learning

into the surrogate-based flowsheet calibration workflow. Since each calibration step produces validated simulation data, these results can be directly reused to incrementally update the surrogate model, forming a closed-loop, self-maintaining digital twin architecture. This integration is discussed in detail in the following section.

4.5 Conclusion

This chapter showed that purely offline-trained surrogate models are insufficient for long-term industrial deployment, as predictive accuracy degrades under changing operating conditions. Online learning strategies help maintain surrogate accuracy by enabling continuous adaptation to newly observed operating regimes.

Multiple online retraining approaches were evaluated, demonstrating that no single strategy is universally optimal; the appropriate choice depends on application-specific requirements such as computational cost, adaptation speed, and robustness against forgetting. For the studied case, incremental learning combined with a replay buffer provided the best balance between local accuracy improvement and global accuracy retention, while alternative strategies also achieved acceptable performance. The conditional retraining approach, in particular, can be advantageous when computational cost is a primary consideration.

These results motivate the integration of online surrogate learning into surrogate-based flowsheet calibration workflows. As calibration step generates validated simulation data, these data can be directly reused to incrementally update the surrogate model, forming a closed-loop, self-maintaining framework. The design of retraining strategies must balance performance gains with long-term stability, supported by continuous performance monitoring to ensure reliable industrial operation.

Chapter 5

Integrated framework for adaptive flowsheet model maintenance

The industrial realization of Digital Twins requires not only accurate process models, but also robust mechanisms for their long-term maintenance under continuously evolving operating conditions. This chapter consolidates the surrogate-based flowsheet calibration and online learning strategies introduced previously into a unified, adaptive framework, enabling co-evolution of physical and data-driven model layers within a single operational workflow.

5.1 Chapter motivation and scope

Digital Twin systems deployed in the process industry are inherently exposed to non-stationary operating environments. Equipment aging, fouling phenomena, feedstock variability, sensor drift, and operational reconfiguration progressively invalidate the assumptions under which process models are initially developed. As a consequence, maintaining model accuracy over time becomes a central requirement for safe and reliable monitoring, optimization, and decision support. Without systematic model maintenance, even high-fidelity flowsheet models deteriorate into static representations that fail to reflect the continuously evolving process state, undermining both

predictive performance and engineering trust in the model.

Surrogate-based calibration, as developed in the previous chapters, provides an effective solution to reduce the computational burden of frequent flowsheet model recalibration, while maintaining accuracy and enabling practical digital twin applications. By approximating the behavior of computationally expensive simulations, surrogate models enable rapid optimization and significantly shorten calibration cycles. However, this advantage introduces a new dependency: the accuracy of the calibration process becomes contingent on the validity of the surrogate itself. As demonstrated in the preceding chapters, surrogate models trained offline exhibit performance degradation when the process drifts beyond the originally sampled operating domain. Consequently, maintaining surrogate fidelity is as critical as maintaining the underlying flowsheet model.

Online learning provides a complementary mechanism to address this limitation by enabling incremental adaptation of surrogate models as new operational data becomes available. Through controlled updates, surrogate models can track local process changes while preserving previously learned global behavior. When embedded within a calibration workflow, online learning allows information generated during flowsheet recalibration to be immediately reintegrated into the surrogate model, closing the loop between physical simulation and data-driven approximation. This interaction mitigates the sensitivity to initial surrogate accuracy and reduces reliance on extensive offline design of experiments, which is often costly or infeasible in industrial settings.

The central contribution of this chapter is the integration of surrogate-based flowsheet calibration and online surrogate learning into a coherent, synergistic framework. Rather than treating calibration and learning as isolated tasks, the proposed approach organizes them into a shared maintenance cycle, where each component continuously supports and corrects the other. Within this structure, the surrogate and flowsheet models operate as a paired representation of the process: the

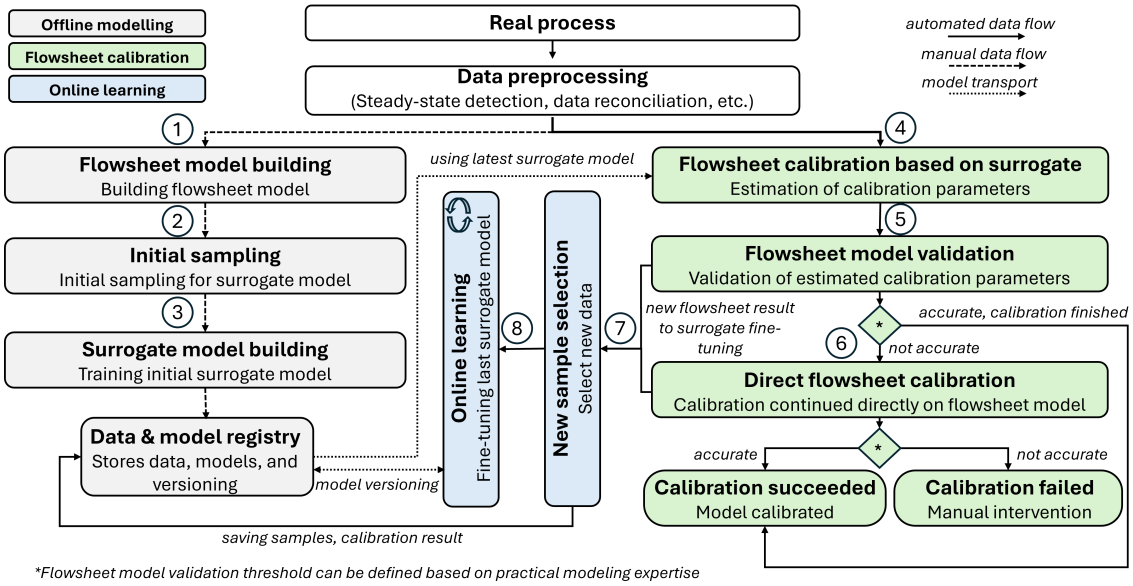


Figure 5.1: Workflow for surrogate-based flowsheet model calibration, highlighting the integration of online surrogate adaptation steps.

surrogate ensures computational efficiency and responsiveness, while the flowsheet model preserves physical consistency and interpretability.

This integrated framework enables flexible model utilization depending on task requirements. Time-critical applications such as what-if analysis, operational screening, or local optimization can rely primarily on the surrogate, while high-impact decisions and validation steps invoke the flowsheet model as a physical reference. By dynamically balancing accuracy, computational cost, and adaptability, the proposed methodology establishes a foundation for responsive, scalable, and sustainable DT deployments in industrial environments.

5.2 Online learning-supported methodology

This section introduces a structured workflow for the continuous calibration of flowsheet models using surrogate models enhanced with online learning techniques (Figure 5.1). The proposed methodology integrates classical offline surrogate model development with ongoing flowsheet calibration and incremental surrogate refinement, enabling sustained predictive performance under evolving process conditions.

Surrogate models are employed to accelerate the calibration of computationally intensive flowsheet simulations by approximating their input–output behavior. Common surrogate modeling approaches include polynomial response surfaces, Gaussian process regression, support vector machines, and neural networks [87]. These methods exhibit different trade-offs in terms of interpretability, scalability, and sample efficiency.

In this work, neural network surrogates are adopted due to their suitability for high-dimensional, nonlinear process models and their ability to support incremental parameter updates without retraining from scratch. Although less interpretable than simpler surrogate forms, neural networks provide the flexibility required for online learning scenarios, where the surrogate must adapt to changing operating conditions while retaining previously acquired knowledge [120].

While surrogate-based flowsheet calibration has been demonstrated previously [118], the focus here is on extending this methodology by embedding the surrogate within an online learning loop. In the proposed framework, the surrogate model is not treated as a static approximation, but as an evolving component that is continuously refined using information generated during calibration and validation.

The key innovation lies in the integration of online learning steps within the calibration workflow (Figure 5.1, Steps 7–8). Incremental updates are applied selectively to preserve surrogate accuracy while maintaining computational efficiency and avoiding catastrophic forgetting. Importantly, the objective of the workflow is not to track all deviations in plant behavior, but to maintain reliable model performance within the normal operating window. Abnormal situations, equipment faults, or transient disturbances mitigated by the control system are intentionally excluded from surrogate updates. When neither surrogate-based nor direct flowsheet calibration satisfies predefined acceptance criteria, the operating point is flagged as an outlier and escalated for expert assessment rather than incorporated into the learning process.

The following subsections describe main stages of the workflow, with emphasis on the role of online learning in enabling scalable and robust digital twin model maintenance.

5.2.1 Offline surrogate initialization

The workflow begins with conventional offline model development. A detailed flowsheet model is constructed based on historical or design data (Step 1), followed by surrogate model training using sampling strategies one-shot or adaptive sampling guided by uncertainty metrics (Steps 2–3; see Chapter 3, Section 3.3.1). The surrogate is trained to approximate the flowsheet model's input–output mapping over an expected operating envelope.

As discussed in Chapter 3, one-shot and adaptive sampling strategies define how the *initial* training dataset is generated prior to deployment. While adaptive sampling improves efficiency relative to purely space-filling designs, both approaches remain offline and inherently assume stationary system behavior during surrogate construction.

In contrast, the online learning strategy introduced in Chapter 4 addresses model maintenance *after deployment*. Rather than refining the initial sampling design, online learning focuses on preserving surrogate accuracy as new operating conditions emerge that were not represented in the offline dataset. This distinction is particularly important in calibration workflows, where surrogate inaccuracies directly influence calibration stability and reliability.

5.2.2 Flowsheet calibration

The central observation exploited in this chapter is that the calibration process itself generates valuable information for surrogate maintenance. During calibration (Step 4), the surrogate model is used to estimate calibration parameters by approximating the flowsheet response. These parameters are subsequently validated using high-fidelity

flowsheet simulations (Step 5), producing trusted input–output pairs that reflect the current operating regime.

If surrogate-based calibration fails to meet the acceptance criteria, two situations are distinguished: (i) the surrogate lacks sufficient local accuracy in the relevant operating region, or (ii) the process has undergone unmodeled structural changes. In the first case, direct flowsheet calibration is performed (Step 6) to recover accurate parameter estimates. In the second case, where direct calibration also fails to achieve acceptable accuracy, the operating point is excluded from automated updating and flagged for manual investigation. Successful validation results obtained in Steps 5 and 6 are stored for potential surrogate refinement.

5.2.3 Incremental surrogate refinement

To sustain surrogate accuracy under process drift, incremental refinement is performed using online learning (Steps 7–8). Newly validated data obtained from surrogate-based or direct flowsheet calibration are incorporated to improve local surrogate accuracy, thereby reducing the frequency of future computationally expensive flowsheet optimizations.

This interaction establishes a bidirectional coupling between the surrogate and the flowsheet model. At calibration step t, the surrogate model $h^{(t)}$ is used to estimate calibration parameters $\boldsymbol{\theta}^{(t)}$ by minimizing the calibration objective function $F(\cdot)$:

$$\boldsymbol{\theta}^{(t)} = \arg\min_{\boldsymbol{\theta}} F\Big(h^{(t)}(\mathbf{X}, \boldsymbol{\theta})\Big) \tag{5.1}$$

The estimated parameters are then validated using the flowsheet model S through a high-fidelity validation function f_{val}:

$$\mathbf{y}^{(t)} = f_{\text{val}}\Big(S, \mathbf{X}, \boldsymbol{\theta}^{(t)}\Big) \tag{5.2}$$

The validated data point is incorporated into the surrogate training set, and the

surrogate is updated incrementally:

$$h^{(t+1)} \leftarrow \mathcal{U}\Big(h^{(t)}, \big(\mathbf{X}, \mathbf{y}^{(t)}\big)\Big) \tag{5.3}$$

where $\mathcal{U}(\cdot)$ denotes the online learning update operator.

Through this mechanism, the surrogate model adapts continuously to the operating regions encountered during calibration, while the flowsheet model provides a physically consistent reference that constrains surrogate evolution. The proposed approach complements offline sampling strategies by relaxing the requirement for highly accurate initial surrogate models and treating surrogate accuracy as an evolving property maintained through systematic interaction with the flowsheet model.

This incremental refinement enables responsiveness to process changes while limiting overfitting and computational overhead. Embedded within a DT architecture, the framework supports continuous, resilient model maintenance and forms the basis for the comparative case study presented in the following section.

5.3 Case study: Adaptive maintenance of a heat exchanger network model

To ensure a consistent and unbiased evaluation of the proposed integrated calibration and online learning framework, the same heat exchanger network case study introduced in Chapter 3 (Section 3.4) and further employed in Chapter 4 is reused in this chapter. The flowsheet model, system configuration, and synthetic dataset are identical across all chapters, enabling a direct comparison between offline surrogate-based calibration, standalone online learning, and the integrated methodology proposed here.

The case study represents a heat exchanger network subject to gradual performance degradation, where fouling-induced changes in heat transfer behavior lead to

progressive model mismatch over time. A synthetically generated dataset is used to emulate realistic operating conditions, including controlled drift in key process parameters, while preserving full access to ground-truth model responses. This setup allows systematic assessment of calibration accuracy, surrogate robustness, and adaptation behavior under reproducible and well-defined conditions.

By reusing the same case study and dataset, the analysis isolates the impact of methodological differences rather than confounding effects arising from changes in system complexity or data characteristics. This continuity provides a coherent experimental basis for evaluating how offline surrogate initialization, surrogate-based calibration, and online learning interact within a unified workflow, and highlights the benefits of incremental surrogate refinement driven by flowsheet validation.

5.3.1 Initial surrogate training

To investigate how the choice of initial surrogate model (Figure 5.1, Steps 2–3) influences calibration efficiency and subsequent online learning performance, several surrogate models were initialized using different sampling strategies and training set sizes. Two classes of offline surrogate development approaches were considered: *one-shot sampling* and *adaptive sampling*.

One-shot sampling For the one-shot approach, the same datasets and surrogate training configurations introduced in Chapter 3 (Section 3.4.2) were reused to ensure consistency and comparability. Based on the surrogate performance analysis presented in Chapter 3 (Figure 3.8), a baseline surrogate trained on 160 LHS samples (LHS-160) was selected, along with an additional surrogate trained on an extended dataset of 500 LHS samples (LHS-500) to represent the effect of increased offline data availability.

Two feedforward neural network (FNN) surrogate models were trained on these datasets:

- **FNN-low**: a single hidden layer with 128 neurons and ReLU activation, trained on 160 LHS samples.
- **FNN-high**: a single hidden layer with 128 neurons and ReLU activation, trained on 500 LHS samples.

Both models were implemented using the `MLPRegressor` class from `scikit-learn`, which supports incremental updates via the `partial_fit()` method, making the models suitable for subsequent online learning experiments. While a fixed network architecture is assumed in this study, recent work has explored dynamic architectures that allow structural adaptation during incremental retraining [137].

Adaptive sampling The adaptive sampling strategy employed a Bayesian Neural Network (BNN) with Monte Carlo (MC) dropout to estimate predictive uncertainty, following the approach proposed by Gal and Ghahramani [138]. The **BNN** surrogate was implemented in `TensorFlow` using a single hidden layer with 128 neurons and ReLU activation, matching the network capacity used in the corresponding `scikit-learn` surrogate models to ensure comparability.

The adaptive sampling procedure was carried out iteratively as follows:

1. An initial set of 20 samples was generated using LHS and used for model training with a 70–30 train–validation split.
2. At each iteration, 10 additional samples were selected from regions exhibiting high predictive uncertainty, as estimated from MC dropout on the validation set.
3. Sampling was continued until achieving a similar performance on validation data as FNN-low, resulting in a total of 110 training samples (Figure 5.2).

Uncertainty-driven acquisition improves sampling efficiency, particularly in high-dimensional and nonlinear systems, and has been shown to outperform random

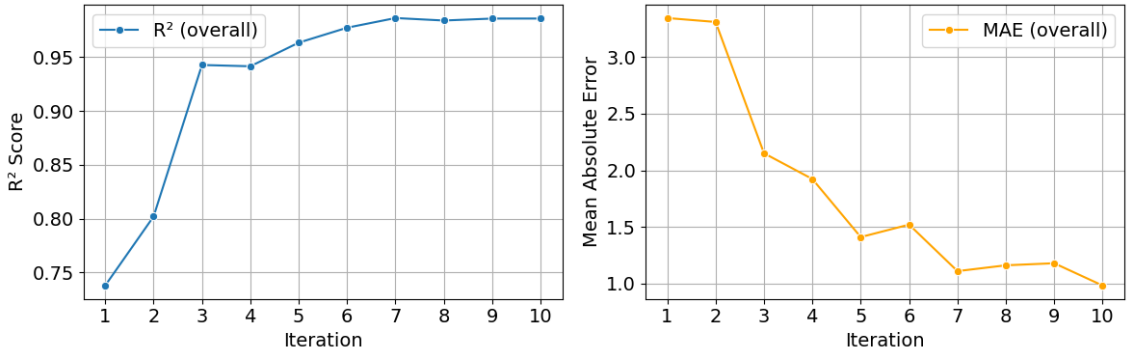


Figure 5.2: Prediction accuracy of the BNN surrogate model over successive adaptive sampling iterations.

or purely space-filling strategies in process modeling applications [139]. However, explicit uncertainty estimation increases prediction time and computational cost, which can be limiting in calibration and online learning contexts, as also reflected in the computational time comparison presented later in Section 5.3.3 (Figure 5.7).

To retain the benefits of adaptive sampling while reducing computational overhead, an additional surrogate model, denoted **FNN-BNN-low**, was trained on the BNN-selected dataset using the same architecture as **FNN-low**. This surrogate enables efficient calibration and online learning while preserving the informative sampling characteristics obtained through uncertainty-based acquisition.

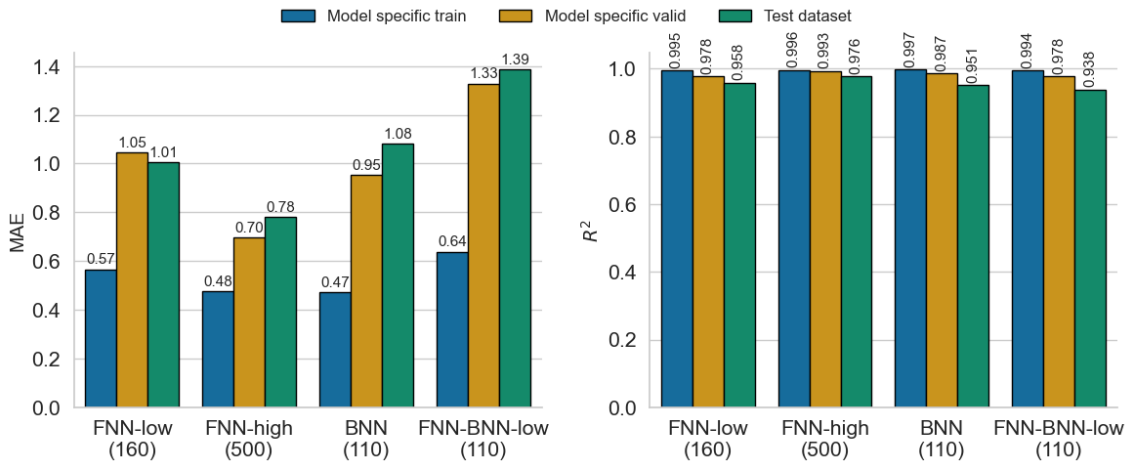


Figure 5.3: Comparison of baseline surrogate model performance on sampling-specific training datasets and a common synthetic test dataset.

Figure 5.3 illustrates that all baseline surrogate models achieve high predictive accuracy on their respective training and validation datasets, while also demonstrating good generalization performance on the common synthetic test set. As expected, the **FNN-high** model attains the highest overall accuracy, owing to the larger size of its training dataset. Notably, the **BNN** surrogate achieves performance comparable to that of the **FNN-low** model, despite being trained on a substantially smaller number of samples, indicating the effectiveness of uncertainty-driven adaptive sampling. In contrast, the **FNN-BNN-low** models exhibit slightly reduced accuracy relative to the other surrogates. Overall, these results confirm that adaptive sampling can achieve predictive performance comparable to—or even exceeding—that of one-shot sampling when training samples are selected strategically.

5.3.2 Flowsheet calibration with online learning

At each test-data working point, a surrogate-based calibration step was performed to estimate the fouling factors from the current process conditions, following the workflow shown in Figure 5.1 (Step 4) and the algorithm described in Appendix D.2. The calibrated values were validated through the flowsheet simulation in Aspen HYSYS (Step 5), and the corresponding outputs were stored. If the calibration accuracy did not meet the predefined threshold, the calibration continued using direct flowsheet simulation, with successful results also recorded.

The validation and direct-calibration results were then used to incrementally update the baseline surrogate models using selected online learning strategies introduced in Chapter 4 (Section 4.3.2) and summarized in Figure 5.4. In this analysis, the focus was on the integration of incremental learning with surrogate-based flowsheet calibration and on the selection of new operating points. Accordingly, at each time step the surrogate model update depended on whether the validation result $y_{\text{valid}}^{(t)}$ satisfied the accuracy criterion $F^{(t)}$, as well as on the chosen update strategy:

- **Always-one:** If validation accuracy is sufficient ($F^{(t)} \leq$ threshold), only the validation result $y^{(t)}_{\text{valid}}$ is used to incrementally update the surrogate $h^{(t)}$, together with a small replay buffer fraction r from the offline dataset $\mathcal{D}^{(r)}_{\text{offline}}$ based on Eq. 4.5. If surrogate-based calibration fails ($F^{(t)} >$ threshold), the surrogate is fine-tuned with the final direct calibration result $y^{(t)}_{\text{direct}}$.
- **Always-both:** Both validation ($y^{(t)}_{\text{valid}}$) and direct calibration ($y^{(t)}_{\text{direct}}$) results are included whenever available. If the surrogate calibration already meets the accuracy threshold, only $y^{(t)}_{\text{valid}}$ is applied.
- **Conditional retrain:** Updates are performed only when surrogate-based calibration fails ($F^{(t)} >$ threshold), using the most recent direct calibration result $y^{(t)}_{\text{direct}}$ together with the replay buffer $\mathcal{D}^{(r)}_{\text{offline}}$.
- **No-replay:** Same as *Always-one*, but without replay, to isolate the effect of replay buffering.

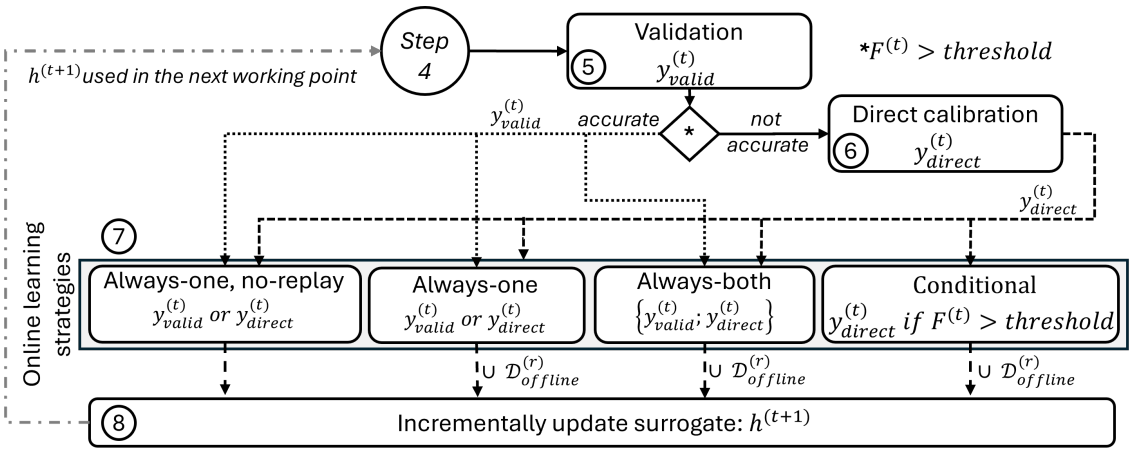


Figure 5.4: Detailed view of Step 7-8 of Figure 5.1, illustrating the investigated online update strategies for incrementally updating the surrogate model using validation and direct calibration results.

These strategies were applied to the initial surrogate models (FNN-low, FNN-high, and FNN-BNN-low). The BNN model itself was excluded from online learning experiments because of the high computational cost of repeated uncertainty evaluations (see later on Figure 5.7).

After each update, the retrained surrogate $h^{(t+1)}$ was applied to calibrate the next operating point. Performance was continuously monitored using the two metrics defined in Chapter 4 Section 4.3.3: *local calibration improvement* and *global accuracy retention*. Both were assessed at every timestep where updates occurred, enabling a direct comparison of the different online learning strategies in balancing update frequency, data usage, and long-term robustness.

As a reference, the calibration process was also performed using direct flowsheet simulations and the surrogate models (FNN-low, FNN-high, BNN, and FNN-BNN-low) without online learning updates (i.e., static surrogate models). These baseline cases provide a benchmark to evaluate the additional benefit of online adaptation.

The selection of the replay fraction r was investigated for the baseline model in Chapter 4 (Section 4.4). Based on these results, a replay fraction of $r = 2$ was adopted for all online learning experiments presented in this chapter. It should be noted, however, that the optimal replay fraction is application-dependent and may vary with process dynamics, data availability, and the desired trade-off between rapid adaptation and retention of previously learned behavior. In this study, the threshold value was set to $F_{\text{thr}} = 0.1$, which corresponds to a realistic mean accuracy target computed from the Taguchi acceptance limits in Chapter 3 (Table 3.3) using Eq. 3.7.

5.3.3 Results

The proposed framework aims to minimize computationally expensive direct calibration runs by improving the *local accuracy* of the surrogate at each operating point, while retaining its *global accuracy* across the operating range. Achieving this balance is nontrivial: frequent local updates improve short-term calibration accuracy but risk degrading generalization through overfitting or catastrophic forgetting, whereas infrequent updates preserve global accuracy but may require more direct simulations. The main calibration accuracy metric was the validation cost function (Eq. 3.7), evaluated after applying surrogate-optimized fouling factors to the flowsheet.

Comparison of validation accuracy of calibration. Figure 5.5 summarizes the validation cost function of calibration across all investigated cases, including no calibration, direct calibration, static surrogates without online learning (FNN-low, FNN-high, BNN, FNN-BNN-low), and the different online learning strategies.

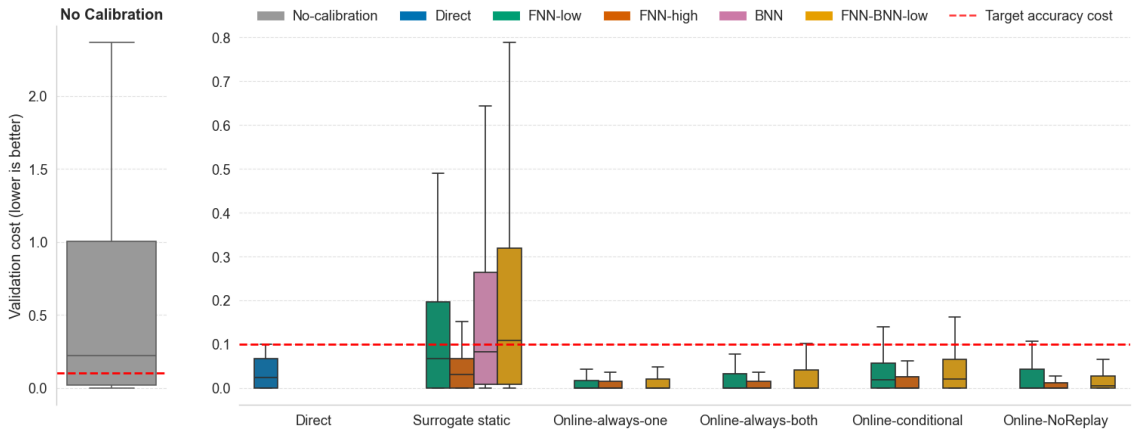


Figure 5.5: Boxplot of validation cost function of calibration for the analyzed cases. Online learning substantially improves validation accuracy compared to static surrogates, with the largest gains observed for always-update strategies.

The no-calibration case showed the highest validation cost function, demonstrating the necessity of parameter adjustment. Direct calibration consistently reached the target accuracy and served as a benchmark that the flowsheet model can be calibrated with the desired accuracy. Static surrogate-based calibration without online learning achieved higher validation cost function than direct calibration; among these, FNN-high (trained on more LHS samples) outperformed FNN-low and BNN, indicating the dependency of model quality.

Introducing online learning significantly improved validation accuracy compared with static surrogates. The always-update strategies (*Always-one* and *Always-both*) achieved the lowest validation cost functions and the most consistent performance. Conditional retraining also improved accuracy relative to static surrogates, though to a lesser extent, since updates were only triggered when calibration performance was insufficient. Replay buffers further reduced variability and prevented catastrophic forgetting, ensuring both local improvements and global retention.

Importantly, online learning narrowed the performance gap between initial models: while FNN-high still performed better, FNN-low with online updates achieved nearly comparable results. This shows that even relatively weak "good-enough" initial models can be refined effectively during operation.

Direct calibration steps and computational cost. The number of direct calibration steps (Step 6) strongly influences total computational time. Figure 5.6 shows that while all surrogate model solution reduced the number of direct simulations relative to a fully direct workflow, online learning further decreased this requirement. Always-update strategies required the fewest direct calibrations, while conditional retraining needed more, reflecting its more conservative update policy.

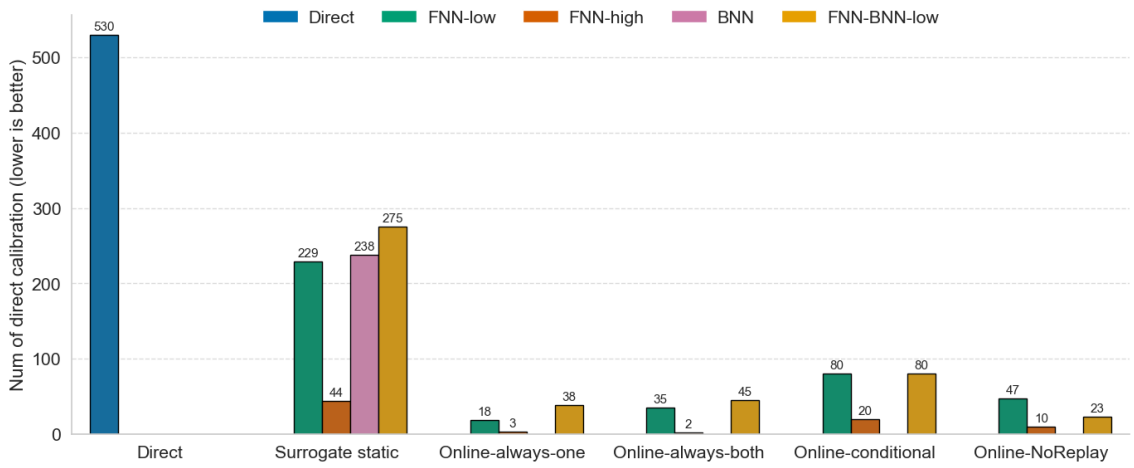


Figure 5.6: Number of direct calibration steps required. Online learning consistently reduces simulation calls, with always-update strategies achieving the largest reductions.

In terms of total calibration time (Figure 5.7), the static BNN surrogate resulted in the longest calibration time. This is because the number of direct calibration steps was almost the same as for the FNN-low surrogate, while each calibration step additionally required MC-dropout-based uncertainty evaluation. As a result, the total calibration time was even higher than that of direct calibration without using a surrogate. For this reason, the BNN surrogate was not evaluated in combination with online learning.

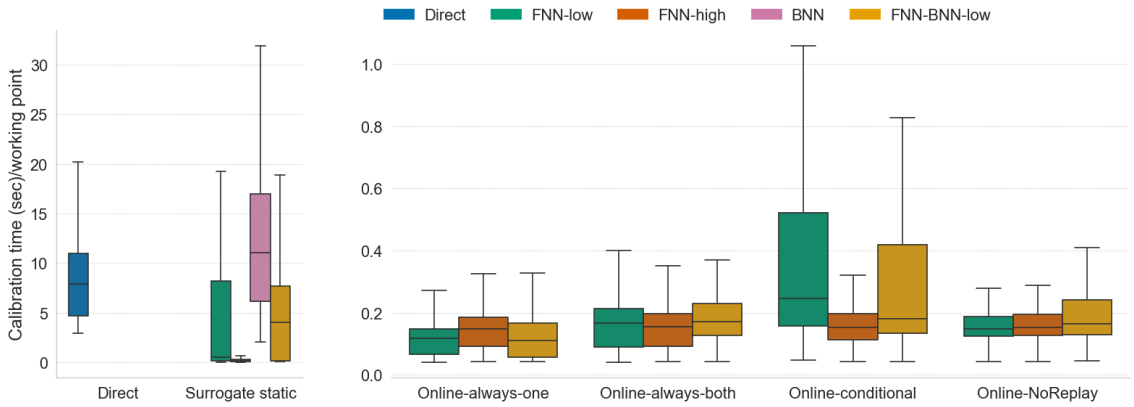


Figure 5.7: Boxplot of total calibration time for the analyzed cases. BNN uncertainty evaluation introduces significant overhead, while FNN-based models benefit more from online learning.

The FNN–BNN-low approach reduced this overhead by retaining the benefits of adaptive sampling during surrogate construction, while avoiding uncertainty calculations during calibration. All surrogate models showed significant improvement when combined with online learning, leading to a clear reduction in both the number of direct calibration steps and the total calibration time. Since the prediction accuracy of the FNN-low and FNN–BNN-low models was similar, the performance improvement achieved through online learning was also comparable for these two models.

In this simplified case study, the FNN-high surrogate achieved sufficiently high accuracy to be used for flowsheet calibration without online learning. As a result, only a very small number of direct calibration steps was required, and the total calibration time was comparable to that of the online learning–based solutions. However, combining the FNN-high surrogate with online learning further reduced the need for direct calibration steps.

Local improvement and global accuracy retention. All online learning strategies improved surrogate accuracy on incoming operational data (Figure 5.8), though omitting the replay buffer caused higher variability and occasional degradation due to catastrophic forgetting. While more accurate base models also benefited, the

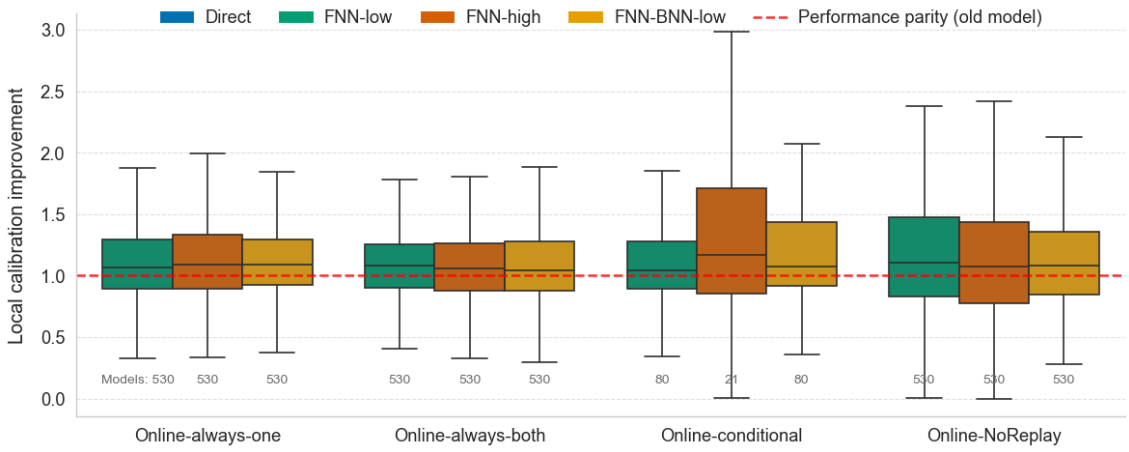


Figure 5.8: Boxplot of local calibration improvement for online learning strategies. Replay buffers reduce variability and mitigate catastrophic forgetting.

relative gains from online learning were similar across model types, reinforcing that extremely high initial accuracy is not a prerequisite for effective deployment.

Global accuracy retention (Figure 5.9) confirms this pattern. Without replay, accuracy on the offline dataset deteriorated, indicating knowledge loss. Conditional retraining limited overfitting risk by updating less often, while always-update strategies with replay not only preserved but improved global accuracy by incorporating both new and historical samples.

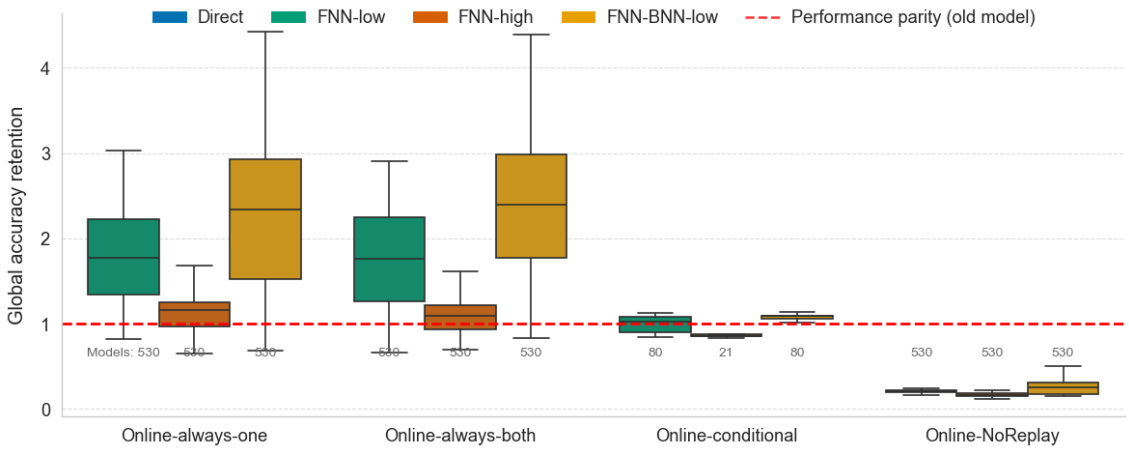


Figure 5.9: Boxplot of global accuracy retention for online learning strategies. Replay buffers preserve or improve global performance by reinforcing historical knowledge.

Overall performance. Online learning consistently outperformed static surrogates across all metrics and improved prediction accuracy for all output parameters (Figure 5.10). It improved calibration accuracy, closing the gap with direct calibration, while simultaneously reducing reliance on expensive direct simulations. By incorporating replay buffers, online learning maintained or even improved global accuracy, ensuring robust long-term performance.

While better initial models (e.g., FNN-high) still provided a slight advantage, the gap narrowed substantially with online updates, showing that highly optimized initial surrogates are not a prerequisite for effective deployment (Appendix Table D.1).

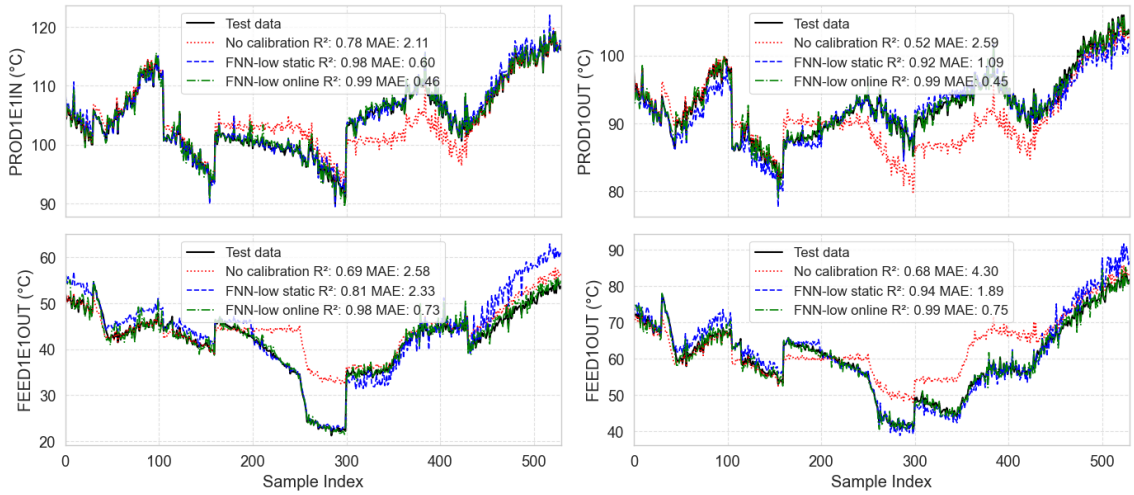


Figure 5.10: Effect of online learning (always-one) on calibration accuracy with FNN-low surrogate model.

The proposed online learning workflow effectively maintains surrogate performance when integrated with a flowsheet model, reducing dependence on the initial surrogate's accuracy and minimizing the computational cost of calibration in dynamic process environments. Replay buffers mitigate catastrophic forgetting, while conditional retraining provides a practical trade-off between computational cost and accuracy improvement. Even though higher-quality initial models still deliver slightly better results, the gap narrows significantly with online learning, making it unnecessary to invest in building an extensively optimized initial model before deployment in this flowsheet model scenario.

5.4 Industrial case studies: Continuous calibration of refinery models

The methodological developments presented in the previous chapters were validated under controlled and semi-industrial conditions. However, the ultimate objective of this research is to demonstrate applicability in real process industry environments, where measurement noise, structural model simplifications, equipment degradation, and operational variability continuously influence model accuracy.

To evaluate the robustness and scalability of the developed online learning–supported surrogate-based calibration workflow, two refinery case studies are presented. The selected units represent distinct degradation mechanisms and different levels of structural complexity, enabling comprehensive assessment of the proposed methodology under realistic operating conditions using real plant measurement data. Together, the cases examine the scalability of the framework to higher-dimensional industrial calibration problems and its robustness in structurally complex refinery environments.

5.4.1 Continuous calibration of a refinery sour water stripper unit

The first industrial case study focuses on a Sour Water Stripper (SWS) unit (Figure 5.11) operating under varying load and feed composition conditions. The SWS unit plays a critical role in refinery wastewater treatment by removing hydrogen sulfide (H_2S) and ammonia (NH_3) generated during crude oil processing. The separation is achieved in a distillation column, where the sour water feed is preheated and stripping is driven by vapor generated in a steam-heated reboiler. The overhead acidic gases are routed to a downstream Claus unit for further treatment. Efficient heat exchange is essential to maintain the required column inlet temperature and reboiler duty, thereby ensuring stable stripping performance.

A dominant operational challenge in this unit is heat exchanger fouling. Fouling

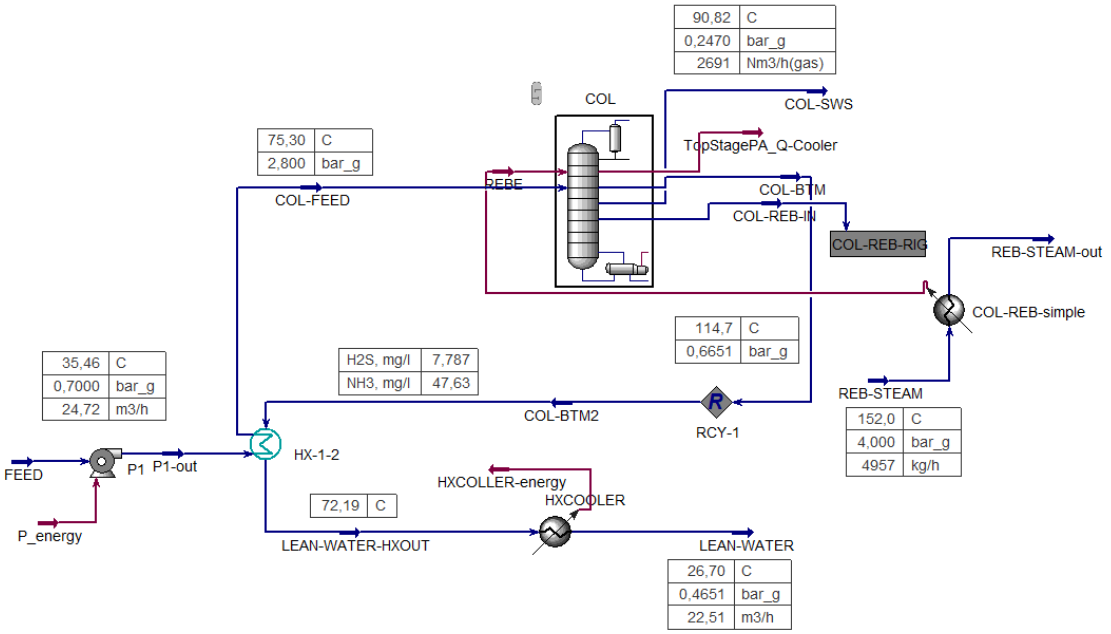


Figure 5.11: SWS unit representation in Aspen Hysys.

reduces thermal efficiency, lowers the column feed temperature, increases steam demand, and progressively deteriorates stripping performance. As a result, effective heat transfer coefficients and separation efficiencies deviate from their nominal values, leading to systematic model–plant mismatch. Traditional flowsheet models calibrated at fixed operating points are unable to follow this gradual degradation, causing prediction errors to accumulate over time.

Table 5.1 summarizes the input and output variables considered in this study together with their operating ranges, based on plant data collected between 01.01.2021 and 31.12.2022.

The surrogate model was developed using the methodology in Chapter 3 (Section 3.4.2). Several LHS sample sizes and neural network configurations were evaluated. Cross-validated NRMSE (Figure 5.12) showed no significant improvement beyond 320 LHS points or higher model complexity. Thus, a single-hidden-layer network with 512 neurons trained on 320 LHS samples was selected as the baseline. The final model used a 70–30 train–validation split.

Prior to the application of collected process data, the dataset required systematic

Table 5.1: Input–output parameters and operating ranges for the SWS case study.

Type	Parameter	Unit	Operating range
Input	FEED-VOLF	m^3/h	16.0–29.0
Input	FEED-TEMP	°C	19.0–39.0
Input	FEED-H2S	wt%	0.1–4.2
Input	FEED-NH3	mg/l	2000–40000
Input	COL-TOP-TEMP	°C	77.0–97.5
Input	COL-REFLUX-VOLF	m^3/h	14.0–24.0
Input	COL-REFLUX-TEMP	°C	31.0–58.0
Input	HX-FOULING	$h°C \cdot m^2/kJ$	0.00025–0.00294
Input	COL-EFF	%	70–95
Output	COL-FEED-TEMP	°C	73.5-92.7
Output	REB-STEAM-MASSF	kg/h	3509-6399
Output	COL-SWS-VOLF	Nm^3/h	105-2740
Output	LEAN-WATER-H2S	mg/l	0.01-44.00
Output	LEAN-WATER-NH3	mg/l	1.30-300.38
Output	LEAN-WATER-TEMP	°C	52.0-103.5

preprocessing to ensure consistency with the steady-state assumption of the flowsheet model. Periods corresponding to unit shutdowns, start-up and transition phases were removed, as these conditions violate steady-state modeling assumptions. In addition, data points with missing or delayed laboratory measurements were excluded to avoid inconsistencies between process variables and quality indicators. Obvious sensor anomalies and outliers—identified based on physical feasibility limits and rapid, non-physical fluctuations—were also filtered out.

After preprocessing, approximately 130 representative steady-state operating points were retained for calibration analysis. These points cover a broad range of load levels and feed compositions, providing a realistic and industrially relevant test basis for evaluating the robustness and efficiency of the online learning–supported surrogate-based calibration framework.

The calibration targeted two lumped performance parameters representing dominant degradation effects:

- heat exchanger fouling factor (HX-FOULING),
- column efficiency (COL-EFF).

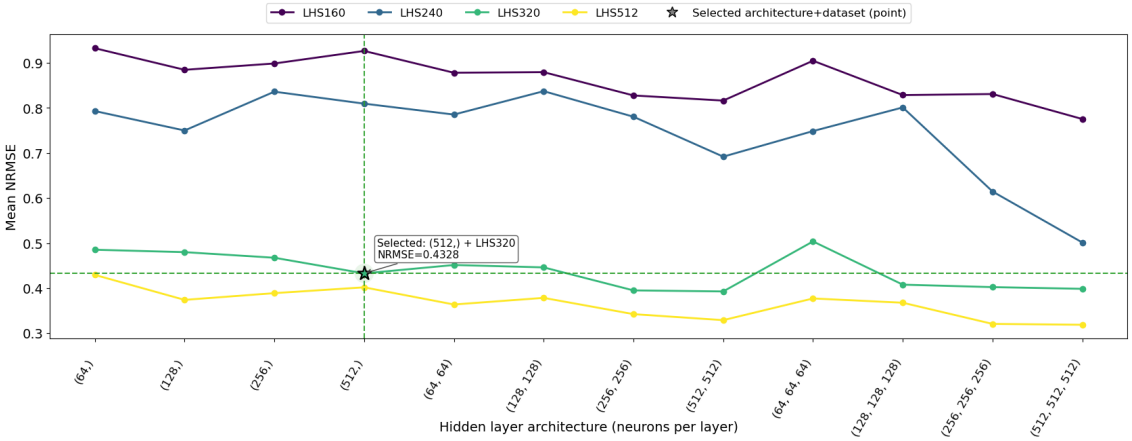


Figure 5.12: Cross-validated performance of surrogate models for different LHS sample sizes (line colors) and neural network architectures (x-axis), reported in terms of NRMSE.

The Taguchi-based objective function (Eq. 3.7) was formulated using selected output variables listed in Table 5.2. Due to correlations between certain outputs, not all measured variables were included.

COL-FEED-TEMP was selected as it directly reflects heat exchanger performance (Appendix D.3). REB-STEAM-MASSF and COL-SWS-VOLF were included because they are influenced by column efficiency. Lean water temperature was excluded due to sensor uncertainty during the studied period, and lean water quality variables were omitted because their behavior was sufficiently represented by the selected process indicators.

The Taguchi acceptance limits were defined based on engineering considerations representing desired modeling accuracy rather than strict measurement tolerances, and equal weights were assigned to all included parameters.

Table 5.2: Objective function Taguchi acceptance limits for the SWS unit.

Output parameter	Unit	Included in objective	Taguchi acceptance limit
COL-FEED-TEMP	°C	+	1
REB-STEAM-MASSF	kg/h	+	500
COL-SWS-VOLF	Nm^3/h	+	200
LEAN-WATER-H2S	mg/l	-	20
LEAN-WATER-NH3	mg/l	-	40
LEAN-WATER-TEMP	°C	-	1

The online learning–supported surrogate-based calibration workflow was applied to the industrial dataset. A single direct flowsheet-based calibration step requires approximately 10 minutes per operating point. The surrogate model was incrementally updated using replay-enhanced retraining based on validated surrogate-assisted calibration samples, as well as on samples obtained from direct calibrations when the surrogate-based solution did not satisfy the predefined accuracy criteria. This strategy is sufficient to demonstrate the practical applicability and computational efficiency of the proposed online learning–supported framework in an industrial environment.

Figure 5.13 presents the calibration results. The workflow significantly improved predictive accuracy for column feed temperature, reboiler steam consumption, and stripped gas flow. Other variables remained close to predefined accuracy limits, confirming stable global model performance.

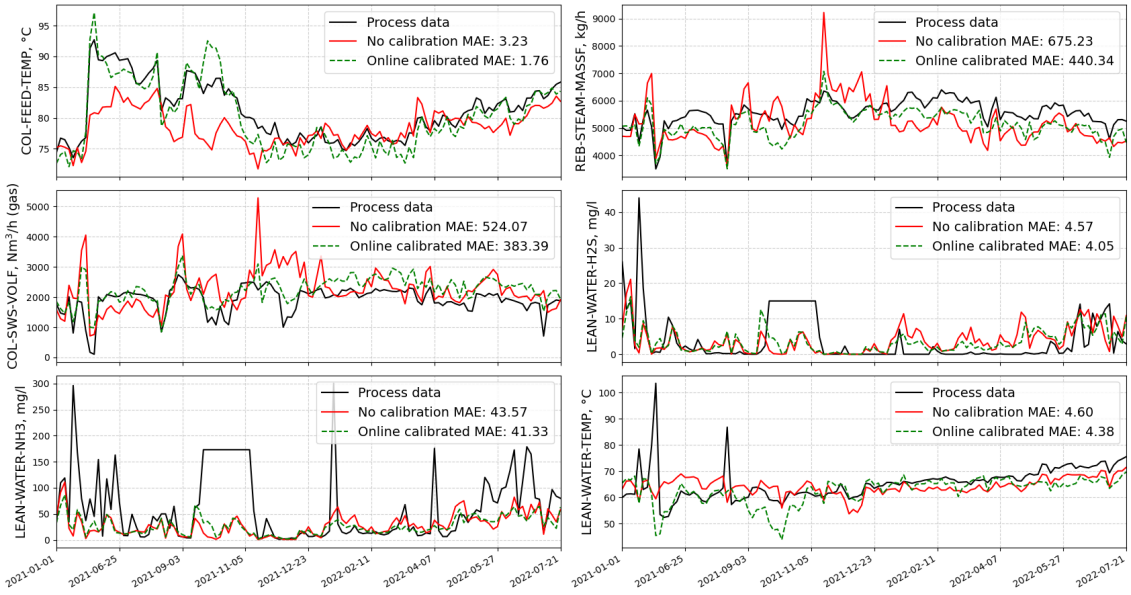


Figure 5.13: Calibration results for selected output variables.

Figure 5.14 shows the evolution of the calibrated fouling factor and column efficiency. Although short-term fluctuations are observed due to measurement noise and model simplifications, a clear long-term fouling trend is identified.

A planned shut-down in May 2021 included mechanical cleaning of the heat

exchanger. Following start-up, temporary plugging effects were observed in plant operation, which are reflected in the calibration results as increased fouling and reduced separation efficiency. After corrective intervention in September 2021, the calibrated fouling factor decreased and column efficiency improved. Thereafter, a gradual fouling increase is visible until the next operational limit is reached.

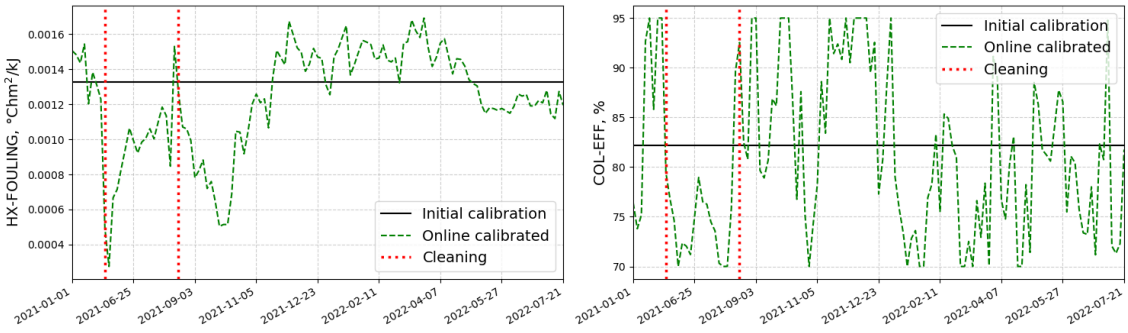


Figure 5.14: Evolution of calibrated performance parameters.

The results demonstrate that the developed online learning–supported surrogate-based calibration workflow is capable of tracking gradual performance degradation in a real refinery environment. The calibrated fouling trajectory is consistent with operational observations, validating the physical interpretability of the method.

Beyond maintaining model accuracy, the calibrated parameters enable quantification of fouling-related energy penalties, providing a basis for maintenance planning and economic evaluation. The case study therefore confirms both the technical robustness and the practical industrial value of the proposed continuous calibration framework.

5.4.2 Continuous calibration of a refinery naphtha redistillation unit

The second industrial case study investigates a refinery naphtha redistillation unit (NRED), representing a substantially more complex calibration problem than the previous SWS case. The unit consists of two thermally coupled distillation columns

with strongly interacting operating variables and two feed streams with detailed compositional information (Figure 5.15).

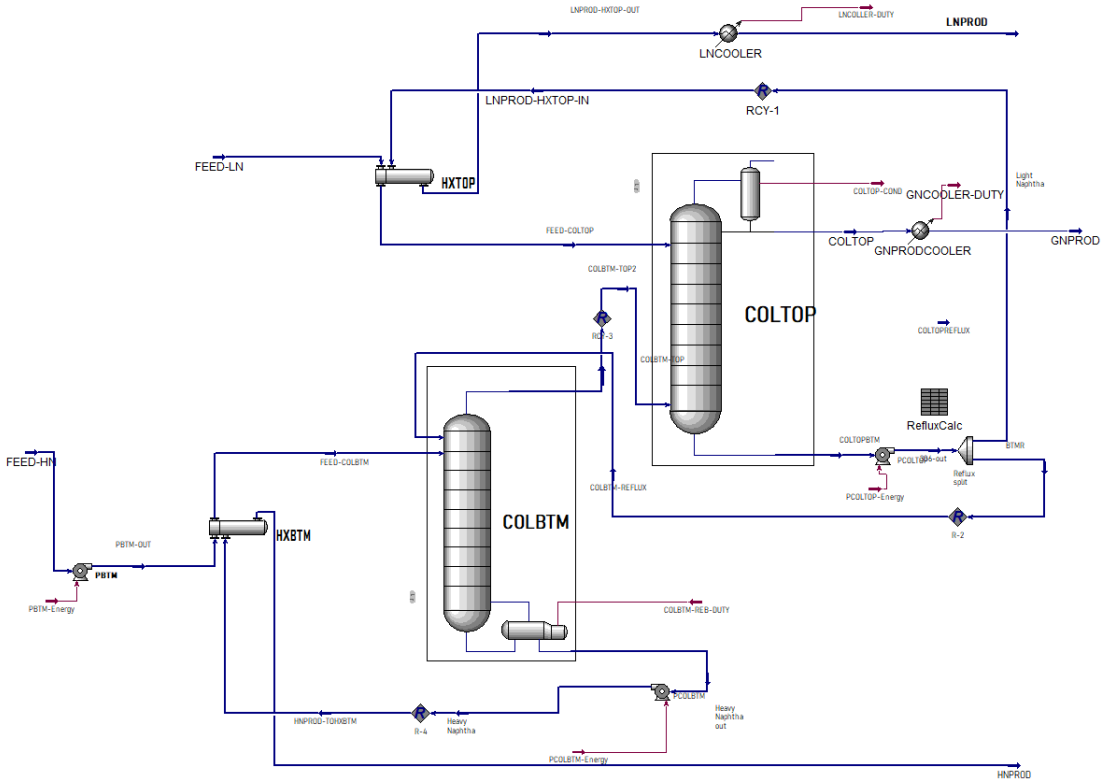


Figure 5.15: Naphtha redistillation unit representation in Aspen Hysys.

The purpose of the NRED unit is to redistill hydrotreated naphtha into fractions serving different downstream units. The top column product (gas naphtha) mainly contains C5/C6 components and provides feed for Light Naphtha Isomerization (LNI). The intermediate cut is used as petrochemical naphtha in steam crackers, while the bottom product (heavy naphtha) serves as feed to the naphtha reformer. The strong thermal and compositional coupling between the columns significantly increases calibration complexity.

Compared to the SWS case, the calibration space is higher dimensional. Two feed streams with 29-component compositions are considered, together with multiple pressure, temperature, and operational variables (74 input parameters and 12 outputs). The presence of detailed compositional inputs and interacting column specifications substantially increases optimization difficulty and surrogate modeling complexity.

Table 5.3 summarizes the input and output variables and their operating ranges, based on plant data collected between 01.01.2025 and 01.01.2026.

Table 5.3: Input–output parameters and operating ranges for the NRED case study (excluding the detailed feed compositions of the two streams, each described by 29 components).

Type	Parameter	Unit	Operating range
Input	FEED-LN-MASSF	t/h	22.8–40.7
Input	FEED-HN-MASSF	t/h	56.4–79.3
Input	FEED-LN-TEMP	°C	35.5–77.0
Input	FEED-HN-TEMP	°C	75.1–98.6
Input	FEED-LN-COMP (29 comp.)	wt%	48 lab samples
Input	FEED-HN-COMP (29 comp.)	wt%	48 lab samples
Input	COLTOP-TOP-PRES	barg	1.83–1.96
Input	COLTOP-BTM-PRES	barg	2.00–2.18
Input	COLBTM-TOP-PRES	barg	2.01–2.22
Input	COLBTM-BTM-PRES	barg	2.28–2.49
Input	COLTOP-TOP-TEMP	°C	72.8–83.5
Input	COLBTM-REFLUX-RATIO	ratio	0.702–0.900
Input	COLBTM-BTM-TEMP	°C	167.9–176.3
Input	COLBTM-BTMTEMP-BIAS	°C	-5–5
Input	HXTOP-FOULING	$h°C \cdot m^2/kJ$	0.00025–0.00294
Input	HXBTM-FOULING	$h°C \cdot m^2/kJ$	0.00025–0.00294
Input	COLTOP-EFF	%	75–95
Input	COLBTM-EFF	%	85–95
Output	FEED-COLTOP-TEMP	°C	59.4–97.1
Output	FEED-COLBTM-TEMP	°C	132.1–145.9
Output	COLBTM-TOP-TEMP	°C	111.2–120.3
Output	COLBTM-REFLUX-TEMP	°C	108.9–118.3

Continued on next page

Type	Parameter	Unit	Operating range
Output	GNPROD-MASSF	t/h	8.7–22.0
Output	HNPROD-MASSF	t/h	52.2–78.7
Output	GNPROD-N-HEXAN	wt%	0.40–15.80
Output	GNPROD-I-PENTAN	wt%	15.45–36.00
Output	GNPROD-N-PENTAN	wt%	24.03–51.34
Output	GNPROD-BENZENE	wt%	0.07–2.25
Output	HNPROD-IBP	°C	94.2–103.0
Output	HNPROD-D05	°C	104.4–110.2

The surrogate model was developed following the methodology presented in Chapter 3 (Section 3.4.2). For scalar process variables, LHS sampling was applied. For feed compositions, measured industrial compositions were used to generate realistic combination scenarios instead of artificial LHS sampling, ensuring physically meaningful mixture distributions.

Cross-validated NRMSE results (Figure 5.16) showed that increasing model complexity or sample size beyond 512 LHS points did not provide significant performance improvement. Therefore, a single-hidden-layer neural network with 1024 neurons trained on 512 LHS samples was selected. The final model was trained using a 70–30 train–validation split.

Industrial measurement data were preprocessed similarly to the SWS case. Shutdown periods, transition phases, missing laboratory measurements, and physically inconsistent data points were removed to ensure steady-state consistency. After preprocessing, approximately 48 representative steady-state operating points were retained for calibration analysis.

Five parameters were selected to represent performance deviations and modeling simplifications:

- column feed heat exchanger fouling factors (HXTOP-FOULING, HXBTM-

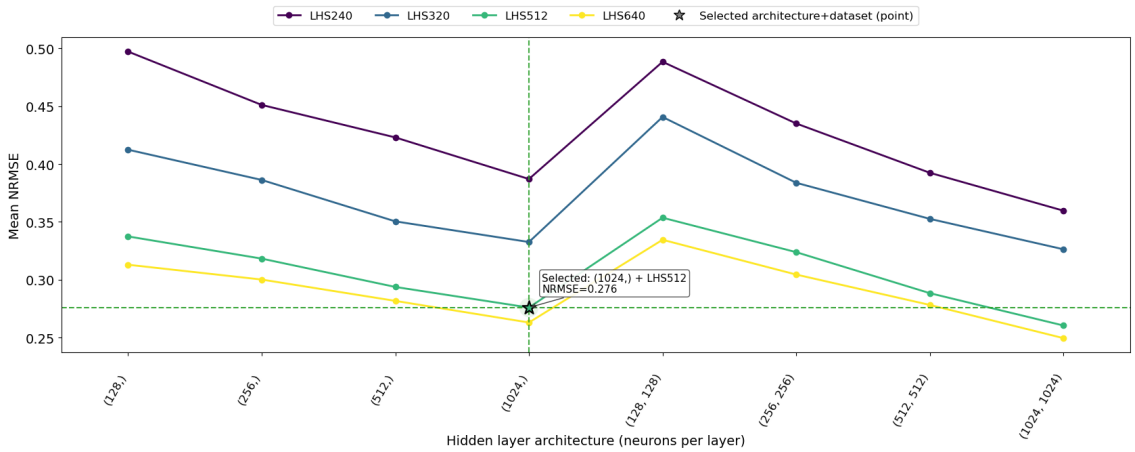


Figure 5.16: Cross-validated performance of surrogate models for different LHS sample sizes (line colors) and neural network architectures (x-axis), reported in terms of NRMSE.

FOULING),

- top and bottom column efficiencies (COLTOP-EFF, COLBTM-EFF),
- bottom column temperature bias (COLBTM-BTMTEMP-BIAS).

The bottom temperature bias parameter accounts for structural modeling simplifications, since the measured bottom temperature cannot be directly enforced without introducing systematic deviations.

The Taguchi-based objective function (Eq. 3.7) was constructed using selected output variables listed in Table 5.4. Due to correlations among outputs, only representative variables were included. Column inlet temperatures were selected to capture heat exchanger performance. For product quality representation, selected key components of the gas naphtha (n-hexane, i-pentane) and its mass flow were included, while heavy naphtha quality was represented by the Initial Boiling Point (IBP). Other variables were excluded because their behavior was sufficiently reflected by these indicators.

The Taguchi acceptance limits were defined based on engineering accuracy requirements rather than strict measurement tolerances, and equal weights were assigned to all included parameters.

Table 5.4: Objective function Taguchi acceptance limits for the NRED unit.

Output parameter	Unit	Included in objective	Taguchi acceptance limit
FEED-COLTOP-TEMP	°C	+	1
FEED-COLBTM-TEMP	°C	+	1
COLBTM-TOP-TEMP	°C	+	1
COLBTM-REFLUX-TEMP	°C	+	1
GNPROD-MASSF	t/h	+	1
HNPROD-MASSF	t/h	-	1
GNPROD-N-HEXAN	wt%	+	1
GNPROD-I-PENTAN	wt%	+	2
GNPROD-N-PENTAN	wt%	-	2
GNPROD-BENZENE	wt%	-	0.5
HNPROD-IBP	°C	+	2
HNPROD-D05	°C	-	2

The online learning–supported surrogate-based calibration workflow was applied in the same manner as in the previous case study. Due to the extremely high computational demand of direct flowsheet calibration in this complex model, only Step 5 of the workflow (Figure 5.1) was applied for surrogate maintenance. The surrogate model was incrementally updated using replay-enhanced retraining based solely on calibration validation samples. The direct calibration step was not incorporated into the calibration process.

Figure 5.17 shows the calibration results. The largest improvements can be seen in the column inlet temperatures and in the heavy naphtha IBP prediction. The other variables show similar performance to the baseline surrogate model, without significant change. The lack of improvement in the gas naphtha product (GNPROD) mass flow rate and composition is likely due to the limited influence of the selected calibration parameters on these variables. If GNPROD predictions had shown larger errors, additional calibration factors with stronger impact—such as column top temperature bias or reflux bias—could have been included in the calibration (Appendix D.4).

For the side product, no measurement data were available during the analyzed period. Therefore, these streams were not included in the calibration objective

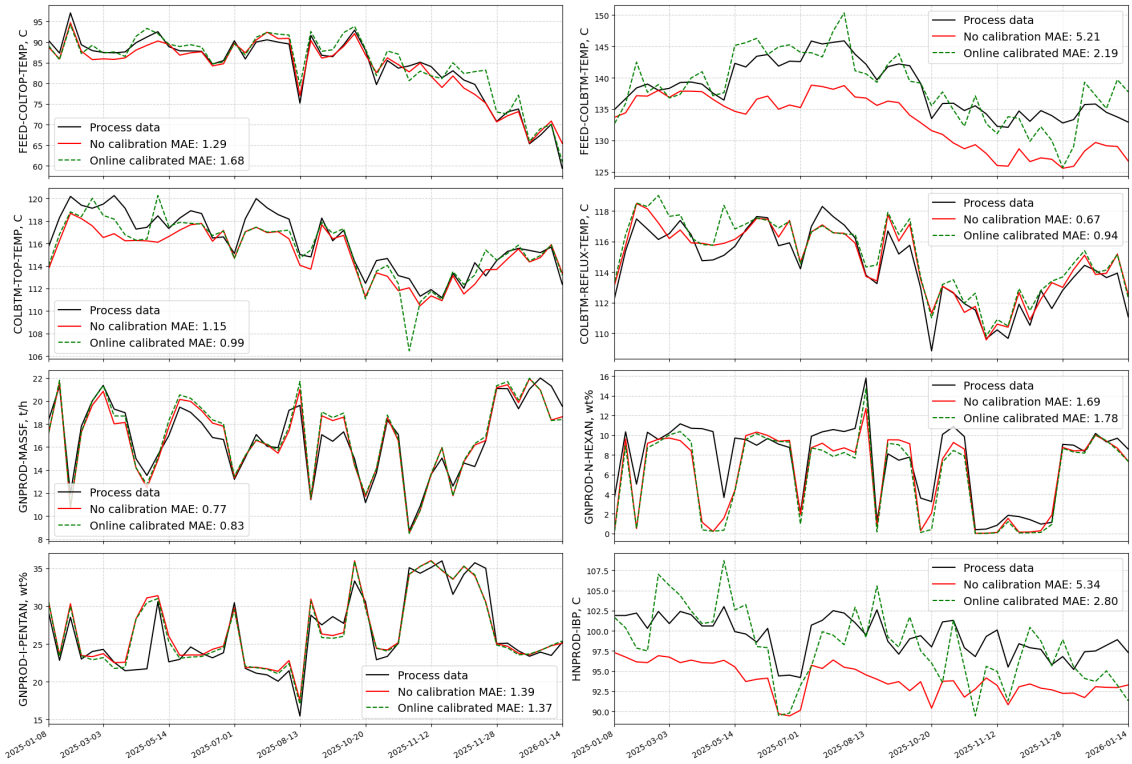


Figure 5.17: Calibration results for selected output variables.

function, since there was no reliable information about their flow rate or quality.

Figure 5.18 presents the evolution of the calibrated performance parameters. Unlike in the SWS case, no long-term degradation trend was expected during the studied period, and the results confirm this assumption. The calibrated parameters show only moderate variations, which mainly reflect normal operating changes rather than structural process deterioration.

A slight increase in top heat exchanger fouling and bottom column efficiency, together with a decrease in bottom heat exchanger fouling, is consistent with operational observations. The remaining fluctuations are mainly caused by measurement noise and interactions between multiple calibration variables. Better data quality and a clearer separation of calibration factors could further reduce these parameter fluctuations and improve stability.

This case study demonstrates that the proposed workflow remains robust in a higher-dimensional, strongly coupled industrial system. Even without incorporating

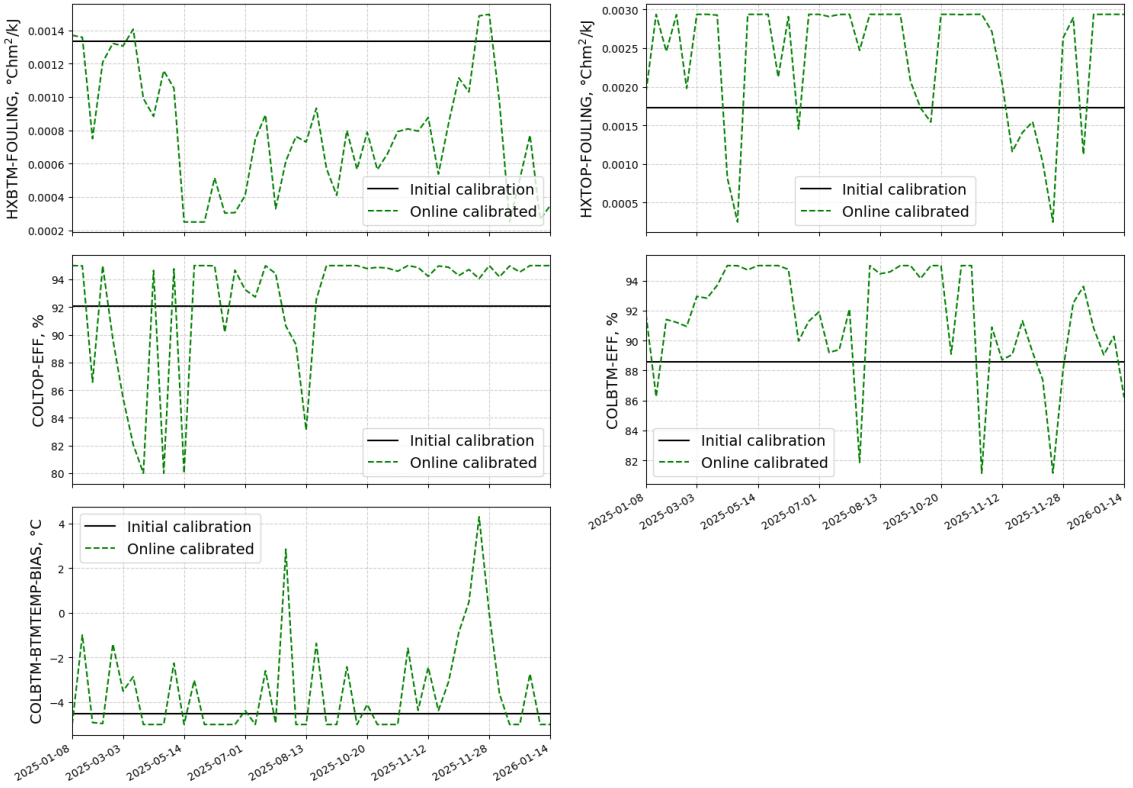


Figure 5.18: Evolution of calibrated performance parameters.

direct calibration outputs into surrogate retraining, the online learning–supported approach maintains predictive stability and improves key quality indicators.

The results confirm that the developed framework scales to complex refinery models and can support continuous calibration in multi-variable industrial environments where degradation trends may not be dominant but structural model deviations and operational variability must still be managed systematically.

5.5 Discussion

This chapter shows that offline-trained surrogate models gradually lose predictive reliability as operating conditions evolve, reducing calibration efficiency and increasing reliance on computationally expensive direct simulations. Incremental online learning provides an effective solution by maintaining local surrogate accuracy without requiring repeated full retraining.

The proposed workflow integrates surrogate-based calibration with systematic validation against the high-fidelity flowsheet model. Surrogate predictions are accepted only if a predefined accuracy threshold is satisfied; otherwise, the procedure reverts to direct calibration. This safeguard ensures physical consistency and prevents error accumulation, while replay-enhanced updates mitigate catastrophic forgetting and preserve global predictive performance.

The industrial case studies confirm both robustness and scalability. In the SWS unit, the workflow successfully tracked fouling-related degradation with physically interpretable calibration factors. In the more complex NRED unit, stable predictive performance was maintained in a higher-dimensional, strongly coupled system.

The study focuses on steady-state applications with moderate process drift. More dynamic environments may require shorter update intervals and enhanced data preprocessing.

Overall, the results demonstrate that online learning–supported surrogate calibration is a practical and robust strategy for sustaining model accuracy and computational efficiency in industrial digital twin applications.

Chapter 6

Prototype software implementation and demonstrator platform

This chapter presents a prototype software framework developed to operationalize the surrogate-based flowsheet calibration and online learning methodology introduced in the previous chapters. The platform serves as a demonstrator for integrating sampling, surrogate modeling, calibration, and incremental model maintenance into a unified and reproducible workflow suitable for digital twin applications.

6.1 Objective and design principles

The developed framework integrates the methodologies presented in Chapters 3–5 — including flowsheet model sampling, surrogate construction, automated calibration, and incremental online learning — into a unified and reproducible application. The objective was not to develop a standalone software product, rather to demonstrate how surrogate-supported flowsheet model maintenance strategies can be operationalized in a transparent and modular manner.

The architecture follows clear separation-of-responsibility principles, where sampling, model training, calibration, and surrogate update are implemented as distinct modules with explicit data and model dependencies. This modular design supports

traceability, reproducibility, and future extensibility.

The platform was implemented in Python using a Flask-based backend API, which orchestrates flowsheet execution, surrogate training, and calibration routines. A web-based interface enables interactive configuration of experiments and structured visualization of results. The framework implements the core elements of the integrated workflow introduced in Chapter 5 (Figure 5.1), excluding process data acquisition and initial model development. The main dashboard (Figure 6.1) provides access to model connection, dataset generation, surrogate training, and surrogate-assisted calibration within a unified environment.

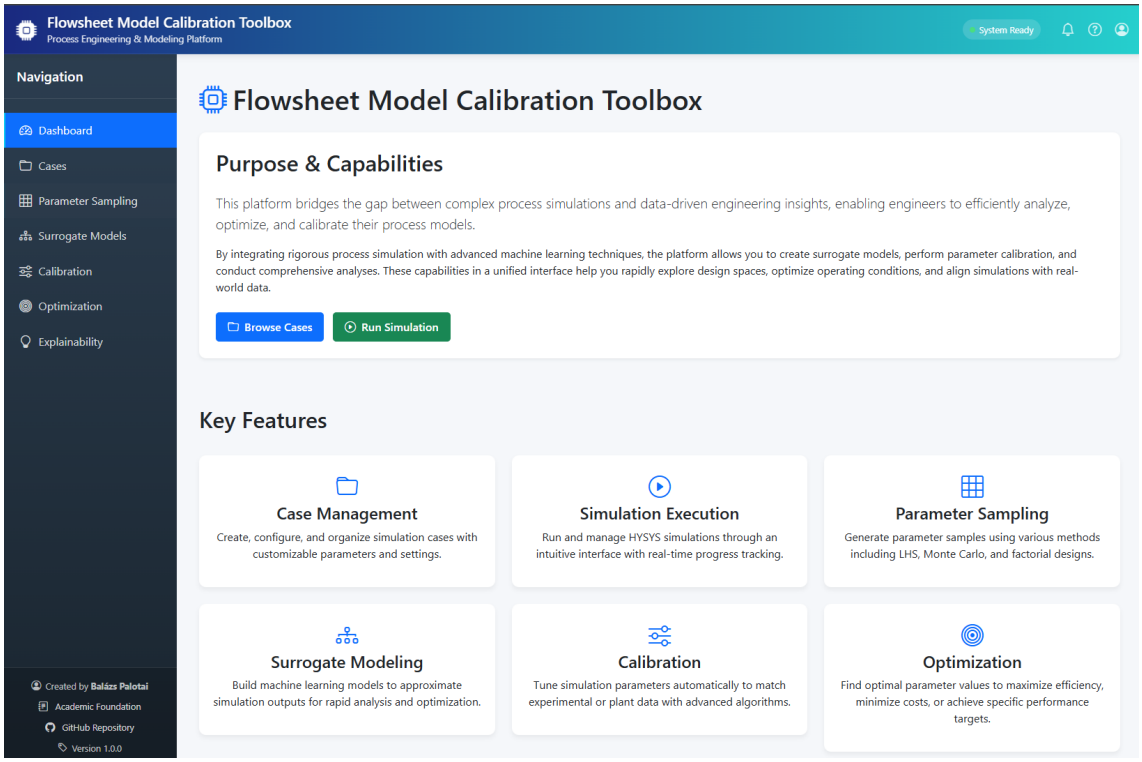


Figure 6.1: Main dashboard illustrating the application functionalities.

6.2 Flowsheet model connection

The first step in the workflow is setting up the flowsheet model within the platform (Figure 6.2). The user configures input and output parameter mappings, specifying which variables are written to and read from the simulation model. This explicit

parameter mapping ensures traceability and consistency between the surrogate and the flowsheet model.

At its current stage, the framework supports steady-state simulation models implemented in Aspen HYSYS. The architecture, however, was designed to be extensible toward other simulation environments supporting programmable interfaces.

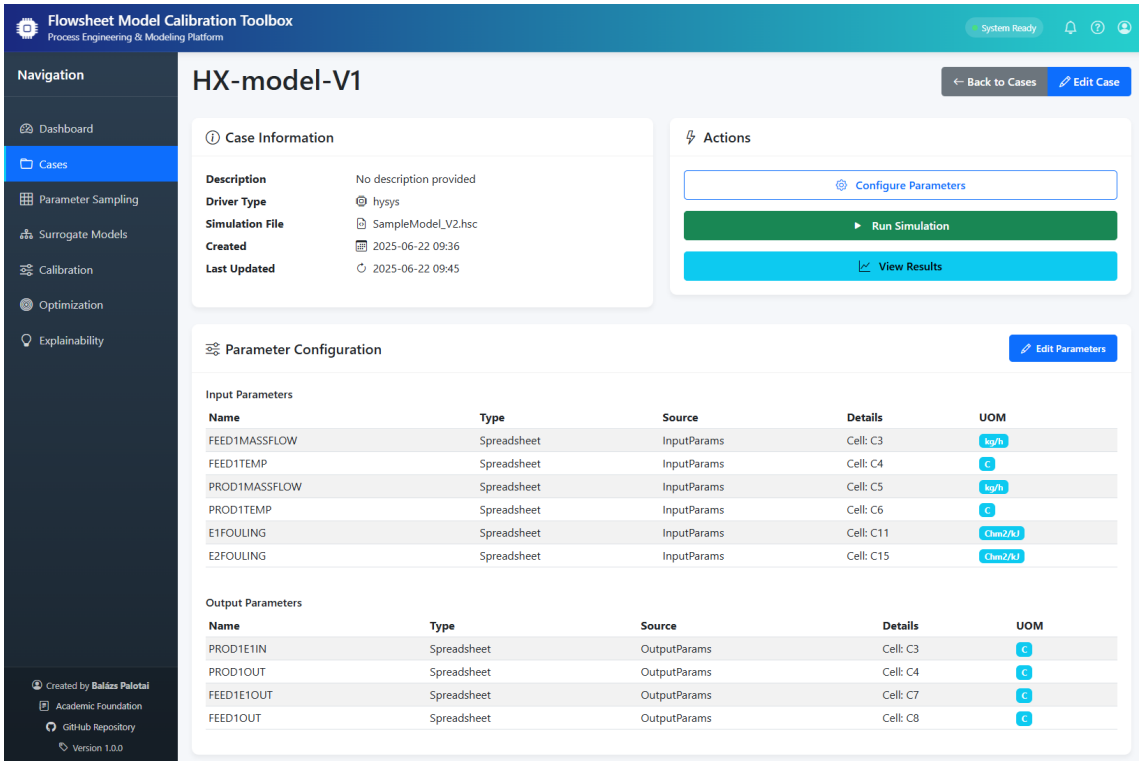


Figure 6.2: Configuration of simulation model connection and parameter mapping.

6.3 Dataset generation and sampling strategy

After establishing the model connection, the framework enables systematic dataset generation within user-defined parameter ranges (Figure 6.3). The selected sampling strategy determines the exploration of the input space, after which the flowsheet model is executed automatically and the resulting data are stored for surrogate training.

Currently implemented sampling methods include LHS, random sampling, full factorial sampling, and Monte Carlo sampling for scalar parameters. These options

allow the investigation of different trade-offs between coverage, computational cost, and model generalization performance.

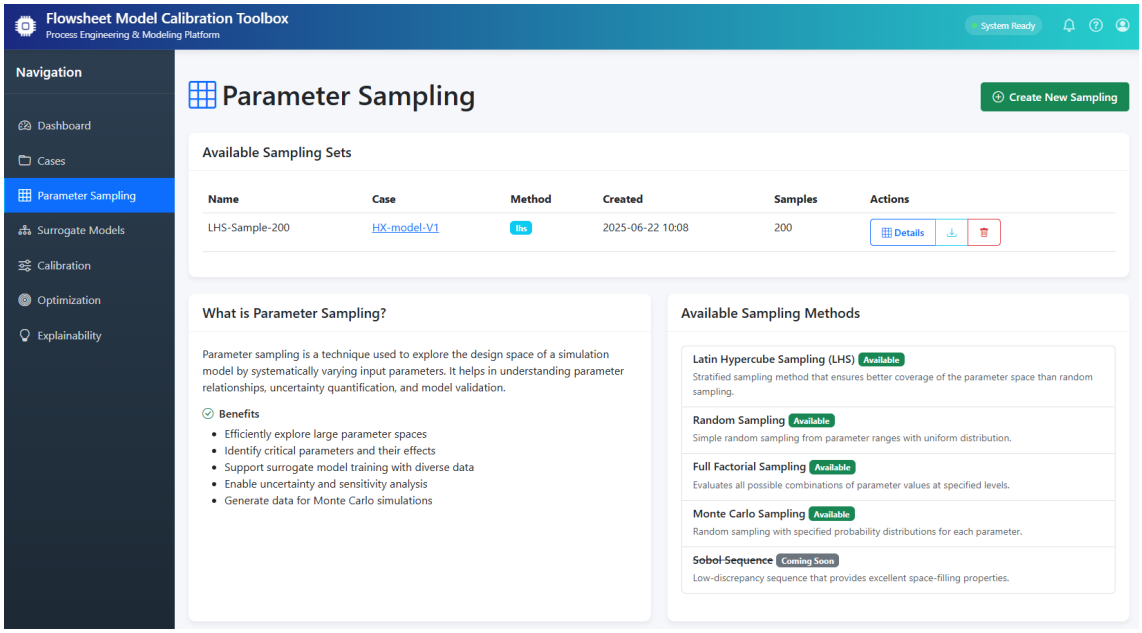


Figure 6.3: Sampling configuration and dataset generation interface.

6.4 Surrogate model training and validation

Based on the generated dataset, surrogate models can be trained and evaluated within the platform (Figure 6.4). The framework supports the configuration of model hyperparameters and provides performance comparison across alternative modeling approaches.

Currently implemented surrogate models include artificial neural networks and random forest regressors using the `scikit-learn` library. Performance metrics are calculated on validation datasets to ensure consistent and transparent model selection.

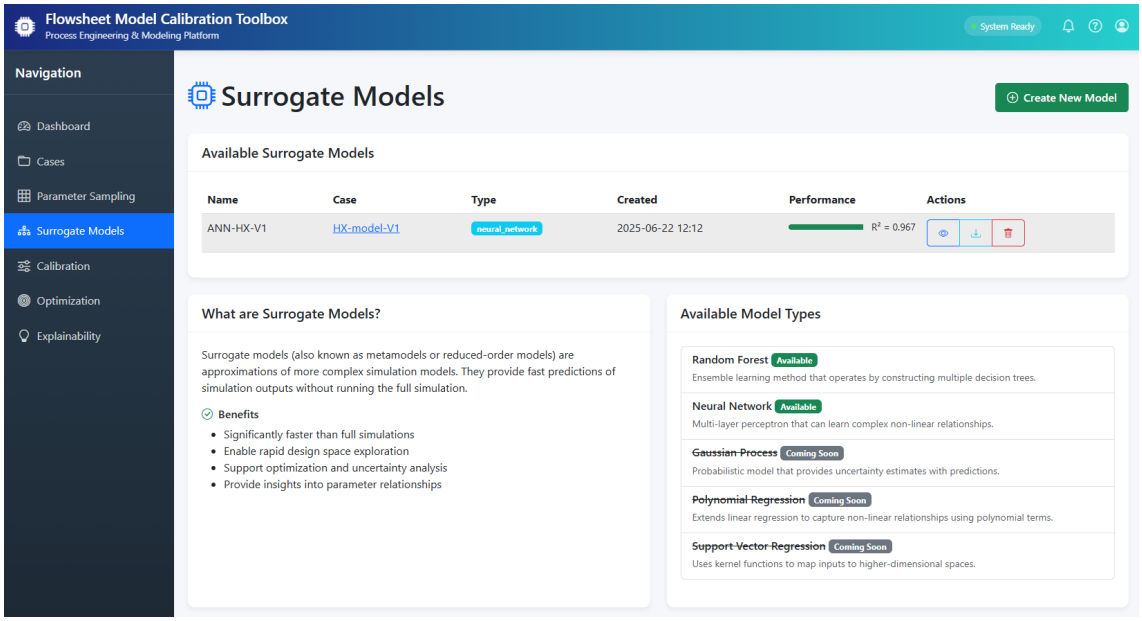


Figure 6.4: Surrogate model training and validation interface.

6.5 Surrogate-based flowsheet calibration

The calibration module enables the use of trained surrogate models for parameter estimation of the underlying flowsheet model (Figure 6.5). Users can configure objective functions and optimization strategies consistent with the methodology presented in Chapter 3.

Calibration can be performed for individual operating points or for batches. The application provides visualization of surrogate predictions and validated flowsheet outputs, supporting transparent assessment of calibration quality.

6.6 Discussion and practical implications

The developed framework demonstrates how the methodological contributions of this thesis can be translated into an operational flowsheet model maintenance workflow. By integrating surrogate modeling, automated calibration, and incremental learning within a unified software environment, the platform bridges the gap between conceptual methodology and practical deployment.

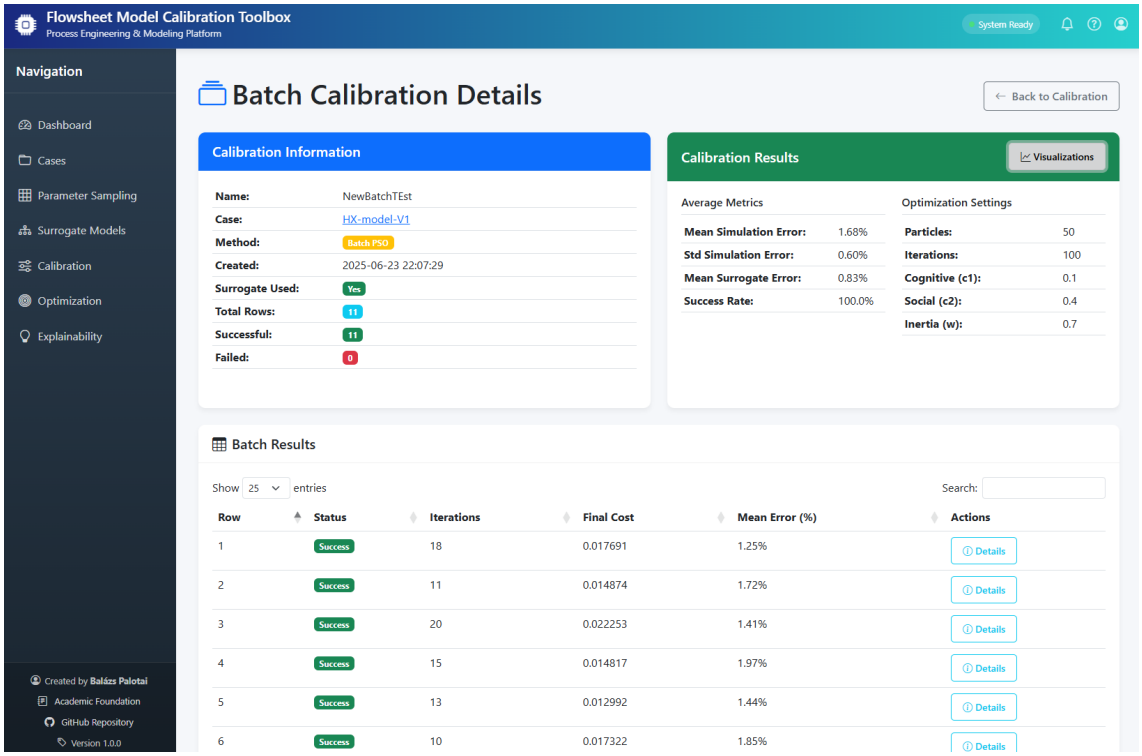


Figure 6.5: Surrogate-based flowsheet calibration interface.

The modular architecture supports systematic experimentation, reproducibility, and transparent evaluation of alternative model maintenance strategies. From an industrial perspective, such a framework enables controlled validation of surrogate-based calibration approaches before full-scale deployment in production environments, thereby reducing implementation risk and supporting life cycle management.

Future extensions may include the integration of explainable artificial intelligence (XAI) techniques [140] to enhance model transparency (Figure 6.6). Explainability methods can provide quantitative insights into parameter sensitivities, clarify the contribution of input variables to specific predictions, and support validation against engineering domain knowledge.

Overall, the prototype platform serves as a proof-of-concept implementation that confirms the practical feasibility of the proposed surrogate-supported digital twin maintenance strategy and illustrates its potential for scalable and transparent industrial adoption. [1]

[1]Repository available at: `https://github.com/bpalotai/Flowsheet-toolbox`

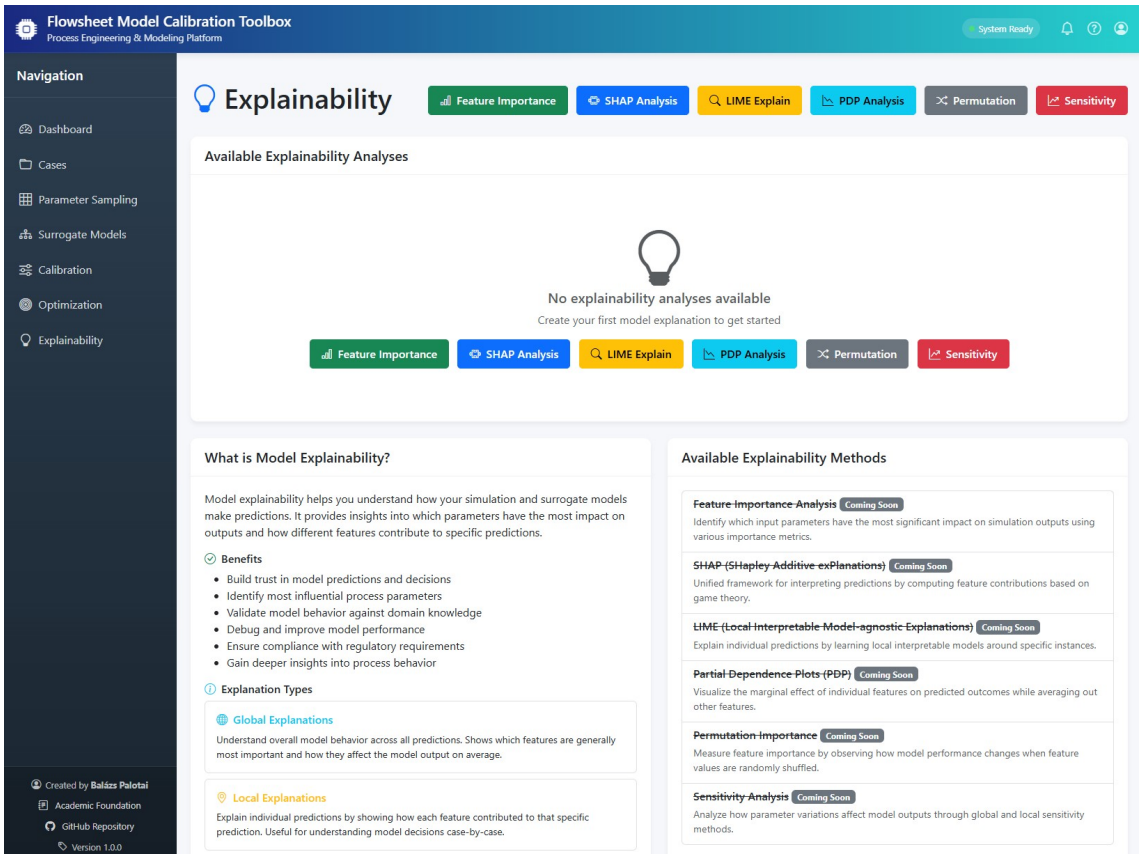


Figure 6.6: Conceptual integration of explainable artificial intelligence techniques to enhance transparency and explainability within the surrogate-based maintenance workflow.

Chapter 7

Conclusions

The process industry is undergoing a profound digital transformation driven by economic volatility, increasing energy costs, stricter environmental regulations, and growing demands for operational flexibility. In this context, digital twins have emerged as a promising paradigm for enabling data-driven, model-based decision support across the life cycle of industrial assets. While high-fidelity flowsheet models provide a physically rigorous foundation for such systems, their long-term industrial applicability is constrained by computational cost, limited scalability, and the progressive deterioration of predictive accuracy under changing operating conditions. This dissertation addressed the central challenge of maintaining reliable, computationally tractable, and industrially deployable flowsheet-based digital twins through the integration of surrogate modeling and online learning.

The research began by systematically analyzing the technological and organizational barriers that hinder the operationalization of Digital Twins in the process industry. The identified challenges were structured along three principal dimensions: integrability, life cycle management, and transparency. By mapping DevOps, MLOps, DataOps, and model engineering principles to flowsheet-based environments, the work established a structured methodological foundation for sustainable Digital Twin deployment. This contribution extends beyond algorithmic development by empha-

sizing governance, traceability, and interdisciplinary integration as prerequisites for long-term industrial success.

Building on this conceptual framework, a surrogate-based multi-objective calibration methodology was developed to reduce the computational burden associated with flowsheet model maintenance. Global neural-network surrogates were trained to approximate the behavior of high-fidelity simulations, enabling calibration to be performed with substantially lower computational cost. The introduction of a Taguchi-based objective function allowed measurement uncertainty and modeling priorities to be incorporated directly into the optimization problem, thereby enhancing robustness, interpretability, and practical usability. Industrial validation demonstrated significant reductions in calibration time while maintaining accuracy under noisy operating conditions.

Recognizing that offline-trained surrogate models lose predictive reliability under process drift, the dissertation further developed an online learning framework for incremental surrogate maintenance. Replay-enhanced incremental learning strategies were investigated to balance rapid local adaptation with global knowledge retention, thereby mitigating catastrophic forgetting. The results confirmed that adaptive updating significantly improves predictive stability in evolving operating regimes, ensuring that surrogate-assisted calibration remains reliable over extended operational periods.

The principal methodological contribution of the thesis lies in the integration of surrogate-based calibration and online learning into a unified, closed-loop workflow. In this framework, each validated calibration step generates new information that incrementally updates the surrogate model, forming a self-maintaining cycle. A flowsheet-level validation safeguard ensures physical consistency and prevents the accumulation of systematic model errors. Industrial case studies demonstrated that this integrated approach can drastically speed up the calibration runs while preserving long-term predictive performance.

From an industrial perspective, the proposed methodology enables automated, adaptive model maintenance for process units affected by fouling, aging, feedstock variability, and other non-stationary phenomena. The dual-model architecture – combining fast surrogate evaluation with physics-based validation – provides a balanced solution that meets industrial requirements for computational efficiency, transparency, and engineering credibility. Beyond calibration, the continuously maintained surrogate models offer a scalable platform for advanced applications such as operational optimization, what-if analysis, and predictive maintenance.

In a broader scientific context, the dissertation contributes to bridging physics-based process modeling and adaptive machine learning within safety-critical industrial environments. It demonstrates that sustainable digital twin deployment requires not only accurate models, but also structured life cycle management and adaptive maintenance mechanisms. The overarching message of this work is that the long-term value of digital twins in the process industry depends on their ability to evolve continuously with the physical system they represent.

Future research may extend the presented framework toward dynamic process models, uncertainty-driven active learning, plant-wide integration, and tighter coupling with economic optimization and enterprise decision systems. Nevertheless, the results presented herein establish a coherent methodological pathway toward adaptive, robust, and industrially viable digital twin systems that combine physical rigor with data-driven adaptability.

Chapter 8

Thesis points and contributions

Thesis points

1. I developed a framework supporting the industrial adoption of digital twins and demonstrated that one of the most critical prerequisites for their successful deployment and long-term operation is the automated maintenance of the underlying flowsheet models.

- I demonstrated that the limited industrial adoption of digital twins in the process industry is primarily caused by the combined challenges of model integration, life cycle management, and model maintenance rather than by a single technological limitation.
- Based on these findings, I developed a life cycle-oriented framework supporting the industrial deployment of digital twins. The framework defines the development stages from a digital model to a fully integrated digital twin and identifies the roles of DevOps, DataOps, MLOps, and model engineering practices in supporting model integration, reusability, and life cycle management.
- I further demonstrated that flowsheet models provide a suitable foundation for digital twin implementation in the process industry. At the same time, I

showed that the automated maintenance of flowsheet models is one of the key prerequisites for the long-term reliable operation of digital twins.

Supporting publications: [1, 7, 8, 9, 10, 11]

2. I demonstrated that the proposed surrogate-based calibration methodology substantially reduces the computational cost of flowsheet model calibration while improving its robustness without compromising the required accuracy, thereby supporting the efficient application of flowsheet models in digital twin systems.

- I demonstrated that the predictive accuracy of flowsheet models gradually deteriorates during operation, while the high computational cost of conventional direct calibration limits the automated maintenance of flowsheet-based digital twins.
- I developed a surrogate-based calibration methodology in which the computationally intensive flowsheet model is replaced by a neural network surrogate during optimization. The methodology was enhanced by a composite objective function that combines engineering priorities with measurement uncertainty, thereby improving calibration robustness.
- The case study demonstrated a 96% reduction in calibration computational cost while preserving the required accuracy and maintaining robust performance under noisy measurement conditions. These results confirm that surrogate-based calibration provides an effective solution for the automated maintenance of flowsheet models.

Supporting publications: [2, 5]

3. I demonstrated that online (incremental) learning enables the continuous adaptation of surrogate models to changing operating conditions while preserving the previously acquired global knowledge of the system. Furthermore, I demonstrated that the timing of model updates and the selection of training data are independent design decisions, allowing different update strategies to be freely combined.

- I demonstrated that the predictive accuracy of offline-trained surrogate models may deteriorate when the real process deviates from the operating region covered by the original training dataset, limiting their long-term applicability.
- I developed an online learning framework that enables the incremental updating of surrogate models using newly acquired operational data. I demonstrated that controlled replay of samples from the original offline training dataset enables adaptation to new operating conditions while preserving previously acquired global knowledge, thereby preventing catastrophic forgetting.
- Furthermore, I demonstrated that the timing of model updates and the selection of training data represent two independent design dimensions, allowing different online learning strategies to be flexibly combined. The case study confirmed continuous improvement of local prediction accuracy while preserving the global predictive performance of the surrogate model.

Supporting publications: [3, 12, 13]

4. I developed an integrated framework that combines surrogate-based flowsheet model calibration with online learning of surrogate models, thereby reducing the computational effort required for flowsheet model calibration and improving calibration efficiency, while effectively supporting the long-term adaptation of flowsheet models in digital twin systems.

- I demonstrated that surrogate-based calibration and online learning, when applied independently, are insufficient to ensure the long-term adaptation of flowsheet models.

- I developed an integrated closed-loop framework in which surrogate-based flowsheet calibration and online learning mutually reinforce each other. Each successful flowsheet model calibration provides new training data for the surrogate model, enabling continuous adaptation to changing operating conditions while preserving previously acquired global knowledge.

- The case study demonstrated that the proposed methodology reduced the number of direct flowsheet model calibrations by 92% while maintaining the long-term predictive accuracy of the surrogate model. Consequently, the surrogate model evolves from a static approximation into a continuously maintained model that supports the long-term adaptation of flowsheet models and the reliable operation of digital twins.

Supporting publications: [4, 6, 14, 15]

Industrial contributions

The results of this research can be directly applied to industrial digital twin systems that combine flowsheet-based process models with machine learning techniques for real-time decision support. The integration of surrogate-based calibration and online learning enables the rapid and reliable updating of first-principles models while significantly reducing computational cost. This is particularly beneficial for process units where equipment aging, heat exchanger fouling, catalyst deactivation, feedstock variability, or other nonlinear phenomena require frequent model recalibration. Owing to its flexibility, the proposed methodology can be readily extended to heat exchanger networks, distillation systems, reactors, and other process units where continuous

model maintenance is essential.

Within digital twin-based decision support systems, the proposed framework enables more reliable process optimization, reduced operating costs, and condition-based operation of industrial equipment. The automated, data-driven model maintenance process allows the digital twin to continuously adapt to changing process, economic, and operating conditions without requiring manual model intervention. Furthermore, continuously updated surrogate models provide a foundation for predictive functionalities such as early fault detection, maintenance scheduling, and rapid evaluation of alternative operating scenarios.

The proposed methodology also offers broader opportunities beyond model maintenance. A continuously updated surrogate model can serve as a computationally efficient approximation for optimization, operating window exploration, multi-objective economic optimization, and rapid what-if analysis. Within this framework, the flowsheet model acts as a first-principles reference for validating surrogate model predictions, thereby ensuring physical consistency and engineering credibility. This synergistic combination of first-principles and data-driven models provides a foundation for faster and more reliable decision support and offers potential for future applications in process intensification, investment planning, and predictive maintenance.

Scientific contributions

Peer-reviewed journal publications

1. B. Palotai, Á. Bárkányi, G. Kis, and J. Abonyi. “Overview of challenges in life cycle management of digital twins”. *Heliyon*, Available at SSRN 4831947 (preprint).

2. B. Palotai, G. Kis, J. Abonyi, and Á. Bárkányi. "Surrogate-based flowsheet model maintenance for digital twins". *Digital Chemical Engineering*, 15 (2025) 100228.

3. B. Palotai, G. Kis, T. Chován, and Á. Bárkányi. "Online learning for surrogate model maintenance". *Hungarian Journal of Industry and Chemistry*, 53(2) (2025) 37–43.

4. B. Palotai, G. Kis, T. Chován, and Á. Bárkányi. "Online learning supported surrogate-based flowsheet model maintenance". *Digital Chemical Engineering*, (2025) 100287.

International conference full papers

5. B. Palotai, Á. Bárkányi, G. Kis, and J. Abonyi. "Operationalization management: Enhancing life cycle management of digital twins". *Computer Aided Chemical Engineering*, 53 (2024) 3025–3030.

6. B. Palotai, G. Kis, J. Abonyi, and Á. Bárkányi. "Synergies of adaptive learning for surrogate-based flowsheet model maintenance". *ESCAPE 35 – European Symposium on Computer Aided Process Engineering*, Ghent, Belgium, 6–9 July 2025. ISBN: 978-9-492859-36-5.

International conference

7. B. Palotai, G. Kis, Z. Mathe. "Unlocking value through operationalization and advanced digital twin applications in downstream". *Emerson User Exchange EMEA 2024*, Düsseldorf, Germany, 2024.

8. B. Palotai, G. Kis, Z. Mathe. "Digital twins in action: Joint effort of Hysys and DeltaV to make profit real". *Emerson Exchange 2025*, San Antonio, Texas, USA, 19–22 May 2025.

9. B. Palotai, G. Kis, L. Grad-Gyenge, Z. Máthé, T. Chován, and Á. Bárkányi. "Toward transparent decision support: Building fully integrated digital twin for refinery operations". *Advances 2025*, Manchester, UK, 16–17 October 2025.

10. B. Palotai, G. Kis, L. Grad-Gyenge, Z. Máthé, T. Chován, and Á. Bárkányi. "Building a fully integrated digital twin for refinery operations". *IChemE Process Management and Control Special Interest Group Webinar*, Online, 27 March 2026.

National conference

11. B. Palotai. "Moving towards digital twins in the process industry". *30th Process Control Systems Meeting (PCS)*, Miskolc-Lillafüred, Hungary, 14–16 October 2024.

12. B. Palotai, G. Kis, T. Chován, and Á. Bárkányi. "Online learning for surrogate model maintenance". *Műszaki Kémiai Napok'25*, Veszprém, Hungary, 15–17 April 2025.

13. B. Palotai, G. Kis, T. Chován, and Á. Bárkányi. "Exploring AI and Machine Learning-Based Decision Support Systems for Real Chemical Processes". *University Research Scholarship Program (EKÖP) Institutional Conference*, University of Pannonia, Veszprém, Hungary, June 25, 2025.

14. B. Palotai, G. Kis, T. Chován, and Á. Bárkányi. "Maintenance of flowsheet models in digital twins through real-time data-driven learning in the chemical industry". *Műszaki Kémiai Napok'26*, Veszprém, Hungary, 21–23 April 2026.

15. B. Palotai, G. Kis, T. Chován, and Á. Bárkányi. "Exploring AI and Machine Learning-Based Decision Support Systems for Real Chemical Processes". *University Research Scholarship Program (EKÖP) Institutional Conference*, University of Pannonia, Veszprém, Hungary, June 25, 2026.

Appendices

Appendix A

Appendix to Chapter 2

A.1 General reference architecture for DT

ISO 23247 provides a general reference architecture for digital twins, defining guidelines for their development, integration, and interoperability. While originally developed for manufacturing, the framework is also applicable to the process industry (Figure A.1).

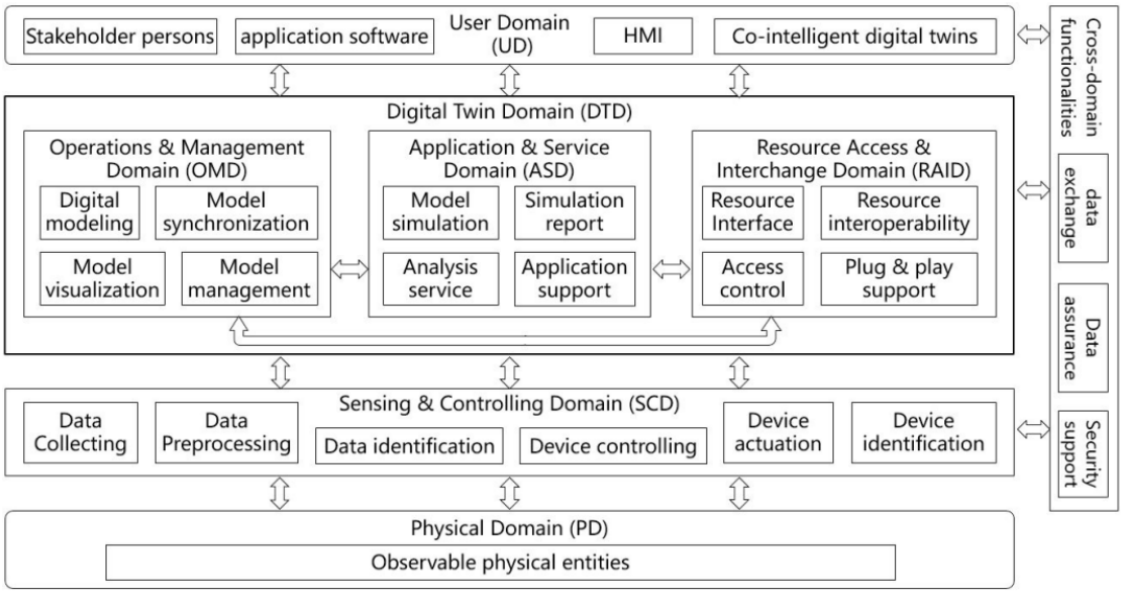


Figure A.1: General reference architecture (ISO 23247) of DT framework [5].

A.2 Standards for model integration

Standardized solutions for model and data exchange are largely defined within the ISO 95 framework. Table A.2 lists the standards relevant to ISO 95 levels 3–4.

Table A.2: ISO 95 level 3–4 model and data integration standards.

Level	Standard	Description
Level 3	CMSD	Enables standardized data exchange between simulation models and manufacturing applications [141].
	IEC 62264	Defines manufacturing operations management activities and interfaces at levels 3–4 [142].
	ISO 15531	Supports interoperability between planning, scheduling, simulation, and execution systems [143].
	QIF	Supports digital thread concepts from design through manufacturing and quality inspection [144].
	PMML	Standard format for representing predictive models and related data transformations [145].
	PFA	JSON-based specification focusing on portable and executable analytics models [146].
Level 4	OAGIS	Provides a canonical XML-based business language for application integration [147].
	IEEE 1516 (HLA)	Interoperability standard for distributed simulation systems [33].
	ISO/AWI 23247	Defines a reference framework for digital twin development and integration [5].
	STEP (ISO 10303)	Enables exchange and sharing of product data across engineering systems [148].
	B2MML	XML-based standard for linking ERP and manufacturing systems [149].

Appendix B

Appendix to Chapter 3

B.1 Case study: surrogate and partial inverse model parameters

Table B.1 summarizes the key neural network configuration parameters of the forward surrogate and partial inverse models used in the case study.

Table B.1: Neural network configuration summary for the surrogate and partial inverse models.

Parameter	Surrogate	Partial inverse
Purpose	Forward mapping	Inverse mapping
Inputs	6	8
Outputs	4	2
Training samples	160	240
Sampling design	Nested LHS	Nested LHS
Architecture	(128,)	(128,)
Activation	ReLU	ReLU
Optimizer	Adam	Adam
Regularization	ℓ_2	ℓ_2
Early stopping	Yes	Yes
Train/validation split	70/30	70/30

B.2 Case study: calibration methodology comparison

Table B.2 summarizes the calibration accuracy of the investigated calibration methodologies on the synthetic test dataset.

Table B.2: Comparison of calibration methodologies on the synthetic test dataset based on MAE and R^2 metrics.

	PROD1E1IN		PROD1OUT		FEED1E1OUT		FEED1OUT	
Calibration type	MAE	R^2	MAE	R^2	MAE	R^2	MAE	R^2
Direct	0.342	0.995	0.283	0.995	0.378	0.996	0.450	0.997
Weighted	0.663	0.978	0.825	0.954	1.680	0.869	1.460	0.962
Taguchi	0.621	0.981	0.933	0.940	1.590	0.891	1.680	0.947
Partial inverse	0.626	0.980	0.706	0.965	0.716	0.984	1.180	0.977

B.3 Case study: calibration parameter estimation accuracy

Table B.3 summarizes the accuracy of the estimated calibration parameters with respect to the original fouling factors used in the synthetic test dataset.

Table B.3: Comparison of calibration methodologies for fouling parameter estimation on the synthetic test dataset.

	E1FOULING		E2FOULING	
Calibration type	MAE	R^2	MAE	R^2
Direct	3.68E-05	0.981	4.73E-05	0.975
Weighted	8.98E-05	0.947	6.71E-05	0.958
Taguchi	9.22E-05	0.947	7.91E-05	0.943
Partial inverse	5.74E-05	0.974	1.06E-04	0.884

B.4 Case study: calibration methodology comparison under noisy conditions

Table B.4 summarizes the calibration accuracy of the investigated calibration methodologies on the synthetic test dataset under noisy conditions.

Table B.4: Comparison of calibration methodologies on the synthetic test dataset under noisy conditions based on MAE and R^2 metrics.

	PROD1E1IN		PROD1OUT		FEED1E1OUT		FEED1OUT	
Calibration type	MAE	R^2	MAE	R^2	MAE	R^2	MAE	R^2
Direct	0.319	0.994	2.370	0.472	0.395	0.995	1.260	0.948
Weighted	0.396	0.989	2.340	0.460	1.060	0.970	1.710	0.901
Taguchi	0.570	0.975	2.400	0.442	1.070	0.965	1.930	0.887
Partial inverse	0.842	0.952	1.810	0.646	1.320	0.943	2.470	0.807

Appendix C

Appendix to Chapter 4

C.1 Threshold selection for conditional retraining

The threshold value F_{thr} for conditional retraining was selected based on an empirical analysis of incremental learning results. Figure C.1 shows the density distribution of the objective values obtained without conditional filtering. The chosen threshold limits retraining to high-error working points while preserving performance close to fully incremental learning.

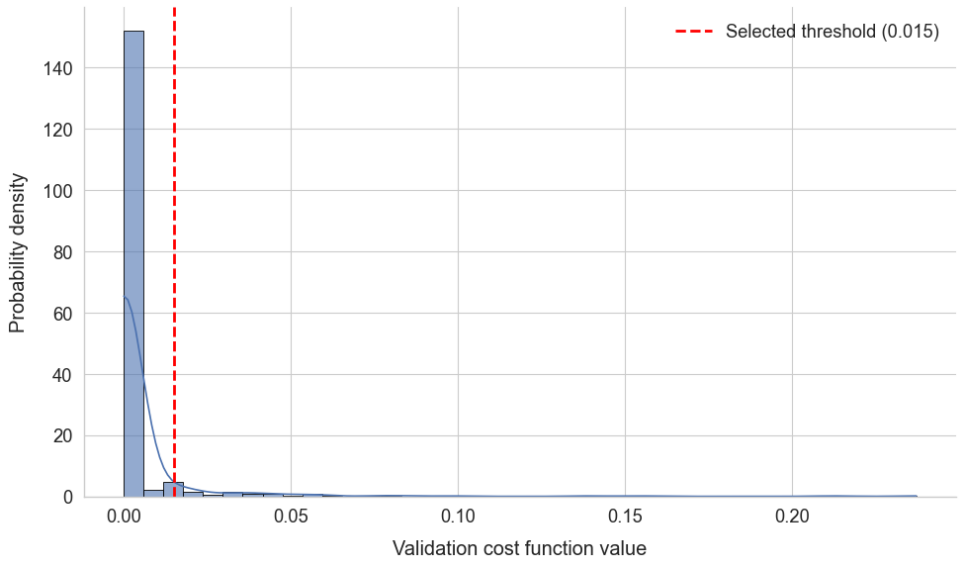


Figure C.1: Density distribution of the incremental learning objective used for selecting the conditional retraining threshold $F_{\text{thr}} = 0.015$.

Appendix D

Appendix to Chapter 5

D.1 Performance improvement with online learning for FNN-low model

Table D.1 compares the flowsheet calibration average performance of the FNN-low model with and without online learning for all output parameters, and includes as a baseline the accuracy of the uncalibrated flowsheet model. The results show that integrating online learning significantly improves calibration accuracy and reduces errors across all outputs compared to the initial static surrogate models.

Table D.1: Flowsheet calibration average performance with and without online learning (FNN-low model - always-one update).

Metric	P1E1IN	P1OUT	F1E1OUT	F1OUT
No Cal. R^2	0.781	0.518	0.691	0.682
FNN-low static R^2	0.983	0.921	0.806	0.936
FNN-low online R^2	0.990	0.986	0.982	0.991
No Cal. MAE	2.114	2.594	2.579	4.302
FNN-low static MAE	0.596	1.086	2.325	1.892
FNN-low online MAE	0.456	0.448	0.732	0.745

P1E1IN: PROD1E1IN temperature; P1OUT: PROD1OUT temperature;
F1E1OUT: FEED1E1OUT temperature; F1OUT: FEED1OUT temperature.

D.2 Pseudocode of surrogate-based flowsheet calibration with online learning

Algorithm 1 Surrogate-based flowsheet calibration with online learning

Require: Process dataset $D = \{\mathbf{x}_t, \mathbf{y}_t\}$, flowsheet model S, surrogate h, update strategy $\mathcal{U}$, replay buffer r

Ensure: Calibrated flowsheet sequence $\{S^{(t)}\}$, updated surrogates $\{h^{(t)}\}$, metrics $\rho_F^{(t)}, \gamma_F^{(t)}$

for each operating point $\mathbf{x}_t \in D$ **do**
  Estimate fouling factors with surrogate $h^{(t)}$ using PSO $\rightarrow g_{\text{best}}^{(t)}$
  Validate with flowsheet S: $y_{\text{valid}}^{(t)}$, compute validation cost func. $F^{(t)}$
  if $F^{(t)} > F_{thr}$ **then**
    Run direct calibration with $S \rightarrow g_{\text{direct}}^{(t)}$, $y_{\text{direct}}^{(t)}$
    if direct calibration fails (no convergence) **then**
      Terminate workflow
    else
      Set $y_{\text{cal}}^{(t)} \leftarrow y_{\text{direct}}^{(t)}$
    end if
  else
    Set $y_{\text{cal}}^{(t)} \leftarrow y_{\text{valid}}^{(t)}$
  end if
  Save calibrated output $y_{\text{cal}}^{(t)}$
  Select new data $\mathcal{D}_{\text{new}}$ based on $\mathcal{U}$:

$$\mathcal{D}_{\text{new}} = \begin{cases} y_{\text{valid}}^{(t)} \text{ or } y_{\text{direct}}^{(t)}, & \mathcal{U} = \text{Always-one} \\ \{y_{\text{valid}}^{(t)}, y_{\text{direct}}^{(t)}\}, & \mathcal{U} = \text{Always-both} \\ y_{\text{direct}}^{(t)} \text{ if } F^{(t)} > \text{threshold}, & \mathcal{U} = \text{Conditional} \\ y_{\text{valid}}^{(t)} \text{ or } y_{\text{direct}}^{(t)}, & \mathcal{U} = \text{No-replay (no } r\text{)} \end{cases}$$

  if replay buffer enabled **then**
    Sample fraction r from $\mathcal{D}_{\text{offline}} \rightarrow \mathcal{D}_{\text{offline}}^{(r)}$
    $\mathcal{D}_{\text{train}} \leftarrow \mathcal{D}_{\text{new}} \cup \mathcal{D}_{\text{offline}}^{(r)}$
  else
    $\mathcal{D}_{\text{train}} \leftarrow \mathcal{D}_{\text{new}}$
  end if
  Incrementally update surrogate $h^{(t+1)}$ with $\mathcal{D}_{\text{train}}$
  Compute metrics: local improvement $\rho_F^{(t)}$, global retention $\gamma_F^{(t)}$
end for

D.3 SWS industrial case – input–output correlation matrix

Figure D.3 presents the Pearson correlation coefficients calculated between the input and output variables of the SWS industrial case study LHS sampling data. The correlation matrix provides an overview of the linear dependencies among the considered process variables and supports the interpretation of surrogate model behavior and parameter sensitivity.

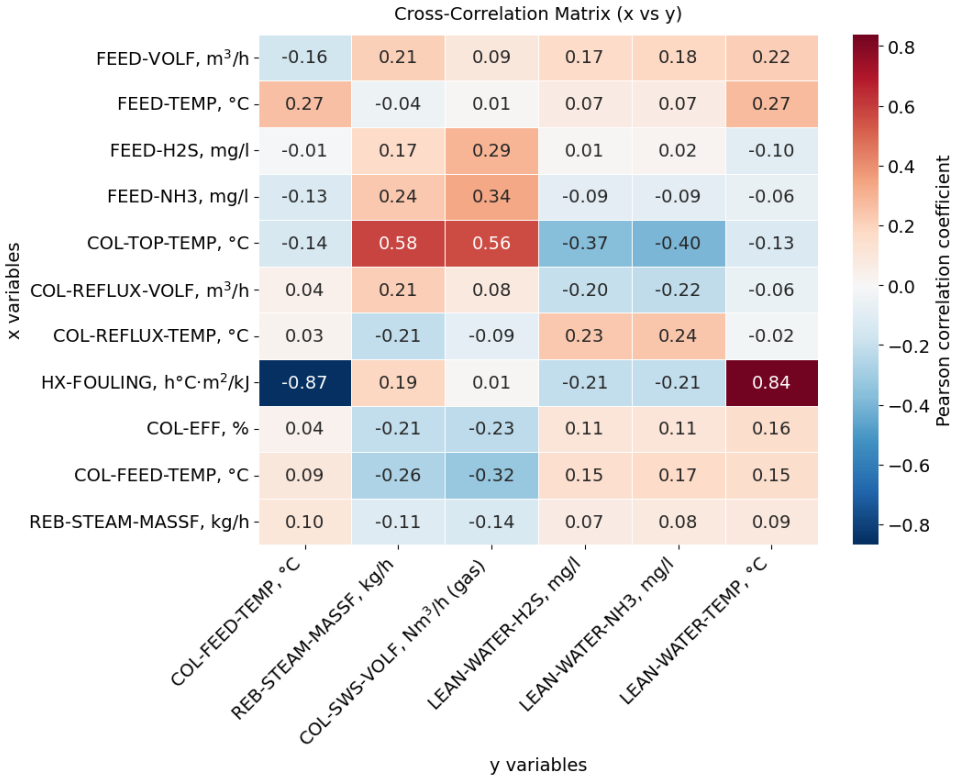


Figure D.3: Pearson correlation matrix of input and output variables for the SWS industrial case.

D.4 NRED industrial case – input–output correlation matrix

Figure D.4 shows the Pearson correlation coefficients between the selected input and output variables of the NRED industrial case study LHS sampling data. The light and heavy feed compositions were excluded from this analysis to improve clarity and focus on the operational variables.

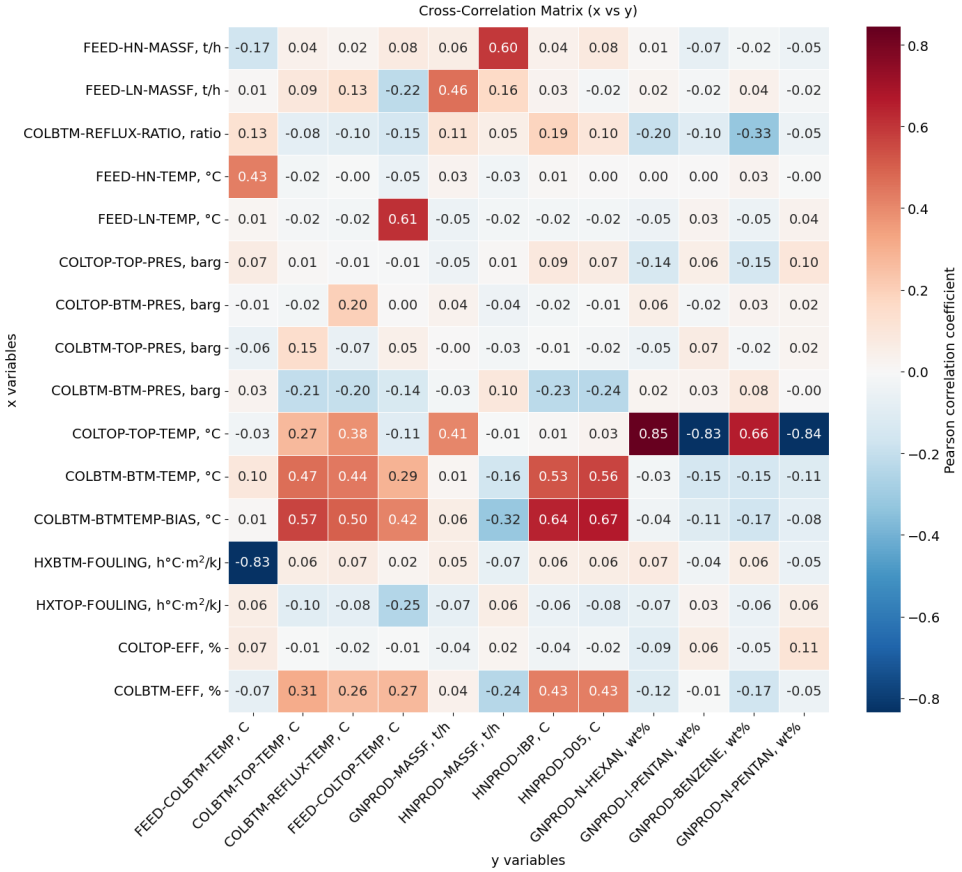


Figure D.4: Pearson correlation matrix of input and output variables for the NRED industrial case (excluding light and heavy feed compositions).

Bibliography

[1] S. Janík, P. Szabó, M. MÍkva, and M. Mareček-Kolibiskỳ. "Effective data utilization in the context of Industry 4.0 technology integration". In: *Applied Sciences* 12.20 (2022), p. 10517.

[2] S. Jaskó, A. Skrop, T. Holczinger, T. Chován, and J. Abonyi. "Development of manufacturing execution systems in accordance with Industry 4.0 requirements: A review of standard-and ontology-based methodologies and tools". In: *Computers in Industry* 123 (2020), p. 103300.

[3] M. Grieves and J. Vickers. "Origins of the digital twin concept". In: *Florida Institute of Technology* 8 (2016), pp. 3–20.

[4] B. D. Allen. "Digital twins and living models at NASA". In: *Digital Twin Summit*. 2021.

[5] *ISO 23247-1:2021 Automation systems and integration — Digital twin framework for manufacturing — Part 1: Overview and general principles*. Accessed: 2023-01-01.

[6] Y. Jiang, S. Yin, K. Li, H. Luo, and O. Kaynak. "Industrial applications of digital twins". In: *Philosophical Transactions of the Royal Society A* 379.2207 (2021), p. 20200360.

[7] R. He, G. Chen, C. Dong, S. Sun, and X. Shen. "Data-driven digital twin technology for optimized control in process systems". In: *ISA Transactions* 95 (2019), pp. 221–234.

[8] S. West, O. Stoll, J. Meierhofer, and S. Züst. “Digital twin providing new opportunities for value co-creation through supporting decision-making”. In: *Applied Sciences* 11.9 (2021), p. 3750.

[9] P. Duchêne, L. Mencarelli, and A. Pagot. “Optimization approaches to the integrated system of catalytic reforming and isomerization processes in petroleum refinery”. In: *Computers & Chemical Engineering* 141 (2020), p. 107009.

[10] P. S. Desai, V. Granja, and C. F. Higgs III. “Lifetime prediction using a tribology-aware, deep learning-based digital twin of ball bearing-like tribosystems in oil and gas”. In: *Processes* 9.6 (2021), p. 922.

[11] M. Liu, S. Fang, H. Dong, and C. Xu. “Review of digital twin about concepts, technologies, and industrial applications”. In: *Journal of Manufacturing Systems* 58 (2021), pp. 346–361.

[12] H. Sihan, G. Wang, D. Lei, D. Lei, and Y. Yan. “Toward digital validation for rapid product development based on digital twin: A framework”. In: *The International Journal of Advanced Manufacturing Technology* (Jan. 2022).

[13] L. Zhang, L. Zhou, and B. K. Horn. “Building a right digital twin with model engineering”. In: *Journal of Manufacturing Systems* 59 (2021), pp. 151–164.

[14] M. Perno, L. Hvam, and A. Haug. “Implementation of digital twins in the process industry: A systematic literature review of enablers and barriers”. In: *Computers in Industry* 134 (2022), p. 103558.

[15] J. M. Müller. “Antecedents to digital platform usage in Industry 4.0 by established manufacturers”. In: *Sustainability* 11.4 (2019), p. 1121.

[16] M. Singh, R. Srivastava, E. Fuenmayor, V. Kuts, Y. Qiao, N. Murray, and D. Devine. “Applications of digital twin across industries: A review”. In: *Applied Sciences* 12.11 (2022), p. 5727.

[17] Z. Hao, C. Zhang, and A. A. Lapkin. “Efficient surrogates construction of chemical processes: Case studies on pressure swing adsorption and gas-to-liquids”. In: *AIChE Journal* 68.6 (2022), e17616.

[18] C. Heavin and D. J. Power. “Challenges for digital transformation–towards a conceptual decision support guide for managers”. In: *Journal of Decision Systems* 27.sup1 (2018), pp. 38–45.

[19] K. Wach, C. D. Duong, J. Ejdys, R. Kazlauskaitė, P. Korzynski, G. Mazurek, J. Paliszkiewicz, and E. Ziemba. “The dark side of generative artificial intelligence: A critical analysis of controversies and risks of ChatGPT”. In: *Entrepreneurial Business and Economics Review* 11.2 (2023), pp. 7–30.

[20] T.-A. Tran, T. Ruppert, G. Eigner, and J. Abonyi. “Retrofitting-based development of brownfield Industry 4.0 and Industry 5.0 solutions”. In: *IEEE Access* 10 (2022), pp. 64348–64374.

[21] D. Kreuzberger, N. Kühl, and S. Hirschl. “Machine learning operations (MLOps): Overview, definition, and architecture”. In: *IEEE access* 11 (2023), pp. 31866–31879.

[22] A. Parrott and L. Warshaw. “Industry 4.0 and the digital twin”. In: *Deloitte University Press*. 2017, pp. 1–17.

[23] J. Holmström. “From AI to digital transformation: The AI readiness framework”. In: *Business Horizons* 65.3 (2022), pp. 329–339.

[24] J. S. Saltz, I. Shamshurin, and K. Crowston. “Comparing Data Science Project Management Methodologies via a Controlled Experiment”. In: *Hawaii International Conference on System Sciences*. 2017.

[25] N. Li, L. Zhao, C. Bao, G. Gong, X. Song, and C. Tian. “A real-time information integration framework for multidisciplinary coupling of complex aircrafts: An application of IIIE”. In: *Journal of Industrial Information Integration* 22 (2021), p. 100203.

[26] Á. Bárkányi, T. Chovan, S. Nemeth, and J. Abonyi. "Modelling for digital twins—Potential role of surrogate models". In: *Processes* 9.3 (2021), p. 476.

[27] N. Sharma and Y. Liu. "A hybrid science-guided machine learning approach for modeling chemical processes: A review". In: *AIChE Journal* 68.5 (2022), e17609.

[28] A. Garmendia, M. Wimmer, A. Mazak-Huemer, E. Guerra, and J. de Lara. "Modelling production system families with AutomationML". In: *2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*. Vol. 1. IEEE. 2020, pp. 1057–1060.

[29] V. Çetin and O. Yıldız. "A comprehensive review on data preprocessing techniques in data analysis". In: *Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi* 28.2 (2022), pp. 299–312.

[30] D. Sculley, G. Holt, D. Golovin, E. Davydov, T. Phillips, D. Ebner, V. Chaudhary, M. Young, J.-F. Crespo, and D. Dennison. "Hidden technical debt in machine learning systems". In: *Advances in neural information processing systems* 28 (2015).

[31] J. van Baten and M. Pons. "CAPE-OPEN: Interoperability in Industrial Flowsheet Simulation Software". In: *Chemie Ingenieur Technik* 86.7 (2014), pp. 1052–1064.

[32] H. Duan and F. Tian. "The development of standardized models of digital twin". In: *IFAC-PapersOnLine* 53.5 (2020), pp. 726–731.

[33] IEEE Standards Association and others. "IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA)—Framework and Rules". In: *Institute of Electrical and Electronics Engineers, New York. IEEE Standard* 1516-2010 (2010), pp. 10–1109.

[34] C. Seeber and A. Lämmle. “AutomationML as seamless data exchange format for integration of automated planning tools for assembly line design”. In: *Procedia CIRP* 107 (2022), pp. 1421–1426.

[35] G. Shao. “Use case scenarios for digital twin implementation based on ISO 23247”. In: *National Institute of Standards: Gaithersburg, MD, USA* (2021).

[36] S. Wazir, G. S. Kashyap, and P. Saxena. “Mlops: A review”. In: *arXiv preprint arXiv:2308.10908* (2023).

[37] N. Van Eck and L. Waltman. “Software survey: VOSviewer, a computer program for bibliometric mapping”. In: *scientometrics* 84.2 (2009), pp. 523–538.

[38] I. Pan, L. R. Mason, and O. K. Matar. “Data-centric Engineering: Integrating simulation, machine learning and statistics. Challenges and opportunities”. In: *Chemical Engineering Science* 249 (2022), p. 117271.

[39] L. Zhang, Y. Shen, X. Zhang, X. Song, F. Tao, and Y. Liu. “Model engineering for complex system simulation”. In: *The 26th European Modeling & Simulation Symposium (Simulation in Industry), Bordeaux*. 2014, pp. 10–12.

[40] Y. Liu, C. M. Eckert, and C. Earl. “A review of fuzzy AHP methods for decision-making with subjective judgements”. In: *Expert Systems with Applications* 161 (2020), p. 113738.

[41] Y.-J. Lai, T.-Y. Liu, and C.-L. Hwang. “Topsis for MODM”. In: *European Journal of Operational Research* 76.3 (1994), pp. 486–500.

[42] G. Shao, J. Hightower, and W. Schindel. “Credibility consideration for digital twins in manufacturing”. In: *Manufacturing Letters* 35 (2023), pp. 24–28.

[43] A. Chakrabarti and J. K. Ghosh. “AIC, BIC and recent advances in model selection”. In: *Philosophy of Statistics* (2011), pp. 583–605.

[44] K. Otto and K. Hölttä. "A multi-criteria framework for screening preliminary product platform concepts". In: *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. Vol. 46962. 2004, pp. 125–136.

[45] M. Hankel and B. Rexroth. "The reference architectural model Industrie 4.0 (RAMI 4.0)". In: *Zvei* 2.2 (2015), pp. 4–9.

[46] J. Morbach, A. Wiesner, and W. Marquardt. "OntoCAPE—A (re) usable ontology for computer-aided process engineering". In: *Computers & Chemical Engineering* 33.10 (2009), pp. 1546–1556.

[47] M. Romero, W. Guédria, H. Panetto, and B. Barafort. "A hybrid deep learning and ontology-driven approach to perform business process capability assessment". In: *Journal of Industrial Information Integration* 30 (2022), p. 100409.

[48] F. Bayram, B. S. Ahmed, E. Hallin, and A. Engman. "DQSOps: Data Quality Scoring Operations Framework for data-driven applications". In: *Proceedings of the 27th International Conference on Evaluation and Assessment in Software Engineering*. EASE '23. Oulu, Finland: Association for Computing Machinery, 2023, pp. 32–41. ISBN: 9798400700446.

[49] F. Tao, H. Zhang, and C. Zhang. "Advancements and challenges of digital twins in industry". In: *Nature Computational Science* 4.3 (2024), pp. 169–177.

[50] F. Bayram, B. S. Ahmed, and A. Kassler. "From concept drift to model degradation: An overview on performance-aware drift detectors". In: *Knowledge-Based Systems* 245 (2022), p. 108632.

[51] H. T. Nguyen, Y. Abu-Zidan, G. Zhang, and K. T. Nguyen. "Machine learning-based surrogate model for calibrating fire source properties in FDS models of façade fire tests". In: *Fire Safety Journal* 130 (2022), p. 103591.

[52] H. Bossel. *Modeling and simulation*. AK Peters/CRC Press, 2018.

[53] L. Zhang, B. P. Zeigler, and Y. Laili. “Introduction to model engineering for simulation”. In: *Model Engineering for Simulation*. Elsevier, 2019, pp. 1–23.

[54] J. Charpentier and T. McKenna. “Managing complex systems: Some trends for the future of chemical and process engineering”. In: *Chemical Engineering Science* 59.8-9 (2004), pp. 1617–1640.

[55] O. Balci, J. D. Arthur, and W. F. Ormsby. “Achieving reusability and composability with a simulation conceptual model”. In: *Journal of Simulation* 5 (2011), pp. 157–165.

[56] S. Robinson, R. E. Nance, R. J. Paul, M. Pidd, and S. J. Taylor. “Simulation model reuse: Definitions, benefits and obstacles”. In: *Simulation Modelling Practice and Theory* 12.7-8 (2004), pp. 479–494.

[57] F. A. Polack, T. Hoverd, A. T. Sampson, S. Stepney, and J. Timmis. “Complex systems models: Engineering simulations”. In: *Artificial Life XI: Proceedings of the Eleventh International Conference on the Simulation and Synthesis of Living Systems*. MIT press. 2008, pp. 482–489.

[58] J. Robinson and P. Otaduy. *An application of object-oriented programming to process simulation*. Tech. rep. Oak Ridge National Lab., TN (USA), 1988.

[59] M. P. Uysal. “Machine learning-enabled healthcare information systems in view of industrial information integration engineering”. In: *Journal of Industrial Information Integration* 30 (2022), p. 100382.

[60] D. Sarkar. “A comprehensive hands-on guide to transfer learning with real-world applications in deep learning”. In: *Towards Data Science* 20 (2018), p. 2020.

[61] A. Göppert, L. Grahn, J. Rachner, D. Grunert, S. Hort, and R. H. Schmitt. “Pipeline for ontology-based modeling and automated deployment of digital twins for planning and control of manufacturing systems”. In: *Journal of Intelligent Manufacturing* (2021), pp. 1–20.

[62] T. Karhela, A. Villberg, and H. Niemistö. “Open ontology-based integration platform for modeling and simulation in engineering”. In: *International Journal of Modeling, Simulation, and Scientific Computing* 3.02 (2012), p. 1250004.

[63] E. Moreau, C. Vogel, and M. Barry. “A paradigm for democratizing artificial intelligence research”. In: *Innovations in Big Data Mining and Embedded Knowledge* (2019), pp. 137–166.

[64] T. R. Wanasinghe, L. Wroblewski, B. K. Petersen, R. G. Gosine, L. A. James, O. De Silva, G. K. Mann, and P. J. Warrian. “Digital twin for the oil and gas industry: Overview, research trends, opportunities, and challenges”. In: *IEEE Access* 8 (2020), pp. 104175–104197.

[65] *Linde Virtual Furnace*. `https://www.linde-engineering.com/en/process-plants/petrochemical-plants/linde-virtual-furnace/index.html/`. Accessed: 2023-09-08.

[66] B. Balaskó. “From Crude Oil to Steam Cracker: How Petrochemical Naphtha Quality Drives the Simultaneous Refinery and Petrochemicals Optimization”. In: *World Petroleum Congress*. WPC. 2017, D023S008R001.

[67] *Europe Steam Cracker Capacities*. `https://www.petrochemistry.eu/about-petrochemistry/petrochemicals-facts-and-figures/cracker-capacity/`. Accessed: 2023-09-08.

[68] D. ASTM. “5134-98: Standard Test Method for Detailed Analysis of Petroleum Naphthas through n-Nonane by Capillary Gas Chromatography”. In: *1993 Annual Book of ASTM Standards* 5 (1998), p. 03.

[69] M. A. Hafeez, A. Procacci, A. Coussement, and A. Parente. “Challenges and opportunities for the application of digital twins in hard-to-abate industries: A review”. In: *Resources, Conservation and Recycling* 209 (2024), p. 107796.

[70] G. S. Martínez, S. A. Sierla, T. A. Karhela, J. Lappalainen, and V. Vyatkin. "Automatic generation of a high-fidelity dynamic thermal-hydraulic process simulation model from a 3D plant model". In: *IEEE Access* 6 (2018), pp. 45217–45232.

[71] L. Wang, T. Deng, Z. Zheng, and Z.-J. M. Shen. "Explainable modeling in digital twin". In: *2021 Winter Simulation Conference (WSC)*. IEEE. 2021, pp. 1–12.

[72] A. C. Dimian, C. S. Bildea, and A. A. Kiss. "Introduction in process simulation". In: *Computer Aided Chemical Engineering*. Vol. 35. Elsevier, 2014, pp. 35–71.

[73] I. Žliobaitė, M. Pechenizkiy, and J. Gama. "An overview of concept drift applications". In: *Big Data Analysis: New Algorithms for a New Society* (2016), pp. 91–114.

[74] S. Mäkinen, H. Skogström, E. Laaksonen, and T. Mikkonen. "Who needs MLOps: What data scientists seek to accomplish and how can MLOps help?" In: *2021 IEEE/ACM 1st Workshop on AI Engineering-Software Engineering for AI (WAIN)*. IEEE. 2021, pp. 109–112.

[75] J. A. Caballero and I. E. Grossmann. "An algorithm for the use of surrogate models in modular flowsheet optimization". In: *AIChE Journal* 54.10 (2008), pp. 2633–2650.

[76] N. Quirante and J. A. Caballero. "Large scale optimization of a sour water stripping plant using surrogate models". In: *Computers & Chemical Engineering* 92 (2016), pp. 143–162.

[77] S. Jog, D. Vázquez, L. F. Santos, J. A. Caballero, and G. Guillén-Gosálbez. "Hybrid analytical surrogate-based process optimization via Bayesian symbolic regression". In: *Computers & Chemical Engineering* 182 (2024), p. 108563.

[78] M. Bubel, P. O. Ludl, T. Seidel, N. Asprion, and M. Bortz. “A modular approach for surrogate modeling of flowsheets”. In: *Chemie Ingenieur Technik* 93.12 (2021), pp. 1987–1997.

[79] F. Boukouvala and M. G. Ierapetritou. “Surrogate-based optimization of expensive flowsheet modeling for continuous pharmaceutical manufacturing”. In: *Journal of Pharmaceutical Innovation* 8 (2013), pp. 131–145.

[80] A. L. Jaimes, S. Z. Martınez, C. A. C. Coello, et al. “An introduction to multiobjective optimization techniques”. In: *Optimization in Polymer Processing* 1 (2009), p. 29.

[81] Y. Cui, Z. Geng, Q. Zhu, and Y. Han. “Multi-objective optimization methods and application in energy saving”. In: *Energy* 125 (2017), pp. 681–704.

[82] W. Gong, Q. Duan, J. Li, C. Wang, Z. Di, Y. Dai, A. Ye, and C. Miao. “Multi-objective parameter optimization of common land model using adaptive surrogate modeling”. In: *Hydrology and Earth System Sciences* 19.5 (2015), pp. 2409–2425.

[83] E. Covantes, E. Fernández, and J. Navarro. “Handling the multiplicity of solutions in a MOEA based PDA-THESEUS framework for multi-criteria sorting”. In: *Foundations of Computing and Decision Sciences* 41.4 (2016), pp. 213–235.

[84] P. Meert, F. Pereira, and P. Willems. “Surrogate modeling-based calibration of hydrodynamic river model parameters”. In: *Journal of Hydro-environment Research* 19 (2018), pp. 56–67.

[85] B. A. Khuwaileh and P. J. Turinsky. “Surrogate based model calibration for pressurized water reactor physics calculations”. In: *Nuclear Engineering and Technology* 49.6 (2017), pp. 1219–1225.

[86] M. Chen, O. A. Abdalla, A. Izady, M. R. Nikoo, and A. Al-Maktoumi. "Development and surrogate-based calibration of a CO2 reservoir model". In: *Journal of Hydrology* 586 (2020), p. 124798.

[87] K. McBride and K. Sundmacher. "Overview of surrogate modeling in chemical process engineering". In: *Chemie Ingenieur Technik* 91.3 (2019), pp. 228–239.

[88] P. Borisut and A. Nuchitprasittichai. "Adaptive Latin Hypercube Sampling for a Surrogate-Based Optimization with Artificial Neural Network". In: *Processes* 11.11 (2023), p. 3232.

[89] M. D. McKay, R. J. Beckman, and W. J. Conover. "A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code". In: *Technometrics* 21.2 (1979), pp. 239–245. ISSN: 00401706. URL: http://www.jstor.org/stable/1268522 (visited on 07/20/2025).

[90] J. C. Helton and F. J. Davis. "Latin hypercube sampling and the propagation of uncertainty in analyses of complex systems". In: *Reliability Engineering & System Safety* 81.1 (2003), pp. 23–69.

[91] I. M. Sobol. "Distribution of points in a cube and approximate evaluation of integrals". In: *USSR Computational mathematics and mathematical physics* 7 (1967), pp. 86–112.

[92] J. H. Halton. "On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals". In: *Numerische Mathematik* 2.1 (1960), pp. 84–90.

[93] D. C. Montgomery. *Design and analysis of experiments*. John wiley & sons, 2017.

[94] G. E. Box and D. W. Behnken. "Some new three level designs for the study of quantitative variables". In: *Technometrics* 2.4 (1960), pp. 455–475.

[95] R. Alizadeh, J. K. Allen, and F. Mistree. “Managing computational complexity using surrogate models: a critical review”. In: *Research in Engineering Design* 31.3 (2020), pp. 275–298.

[96] D. D. Saurette, A. Biswas, A. W. Gillespie, et al. “Determining minimum sample size for the conditioned Latin hypercube sampling algorithm”. In: *Pedosphere* 34.3 (2024), pp. 530–539.

[97] L. Dias, A. Bhosekar, and M. Ierapetritou. “Adaptive sampling approaches for surrogate-based optimization”. In: *Computer Aided Chemical Engineering*. Vol. 47. Elsevier, 2019, pp. 377–384.

[98] Y. Guo, P. Nath, S. Mahadevan, and P. Witherell. “Active learning for adaptive surrogate model improvement in high-dimensional problems”. In: *Structural and Multidisciplinary Optimization* 67.7 (2024), p. 122.

[99] K. Crombecq, D. Gorissen, D. Deschrijver, and T. Dhaene. “A novel hybrid sequential design strategy for global surrogate modeling of computer experiments”. In: *SIAM Journal on Scientific Computing* 33.4 (2011), pp. 1948–1974.

[100] S. S. Garud, I. A. Karimi, and M. Kraft. “Smart sampling algorithm for surrogate model development”. In: *Computers & Chemical Engineering* 96 (2017), pp. 103–114.

[101] A. Cozad, N. V. Sahinidis, and D. C. Miller. “Learning surrogate models for simulation-based optimization”. In: *AIChE Journal* 60.6 (2014), pp. 2211–2227.

[102] J. H. Friedman. “Greedy function approximation: a gradient boosting machine”. In: *Annals of statistics* (2001), pp. 1189–1232.

[103] N. Quirante, J. Javaloyes-Antón, and J. A. Caballero. “Hybrid simulation-equation based synthesis of chemical processes”. In: *Chemical Engineering Research and Design* 132 (2018), pp. 766–784.

[104] H. Tong. “Functional linear regression with Huber loss”. In: *Journal of Complexity* 74 (2023), p. 101696.

[105] R. T. Marler and J. S. Arora. “The weighted sum method for multi-objective optimization: new insights”. In: *Structural and Multidisciplinary Optimization* 41 (2010), pp. 853–862.

[106] T. Lofthouse. “The Taguchi loss function”. In: *Work Study* 48.6 (1999), pp. 218–223.

[107] F. Alsharify, Z. Al-Khafaji, et al. “A Review of Optimization Techniques: Applications and Comparative Analysis”. In: *Iraqi Journal For Computer Science and Mathematics* 5.2 (2024), pp. 122–134.

[108] J. Kennedy and R. Eberhart. “Particle swarm optimization”. In: *Proceedings of ICNN’95-International Conference on Neural Networks*. Vol. 4. IEEE. 1995, pp. 1942–1948.

[109] J. G. Speight. *Fouling in refineries*. Gulf Professional Publishing, 2015.

[110] T. J. Santner, B. J. Williams, W. I. Notz, and B. J. Williams. *The design and analysis of computer experiments*. Vol. 1. Springer, 2003.

[111] G. Rennen, B. Husslage, E. R. Van Dam, and D. Den Hertog. “Nested maximin Latin hypercube designs”. In: *Structural and Multidisciplinary Optimization* 41.3 (2010), pp. 371–395.

[112] C. M. Bishop and N. M. Nasrabadi. *Pattern recognition and machine learning*. Vol. 4. 4. Springer, 2006.

[113] J. Zhang, S. Chowdhury, and A. Messac. “An adaptive hybrid surrogate model”. In: *Structural and Multidisciplinary Optimization* 46.2 (2012), pp. 223–238.

[114] B. Rodič. “Industry 4.0 and the new simulation modelling paradigm”. In: *Organizacija* 50.3 (2017), p. 193.

[115] M. Hadigol and A. Doostan. “Least squares polynomial chaos expansion: A review of sampling strategies”. In: *Computer Methods in Applied Mechanics and Engineering* 332 (2018), pp. 382–407.

[116] A. Bhosekar and M. Ierapetritou. “Advances in surrogate based modeling, feasibility analysis, and optimization: A review”. In: *Computers & Chemical Engineering* 108 (2018), pp. 250–267.

[117] L. Dias, A. Bhosekar, and M. Ierapetritou. “Adaptive sampling approaches for surrogate-based optimization”. In: *Computer aided chemical engineering*. Vol. 47. Elsevier, 2019, pp. 377–384.

[118] B. Palotai, G. Kis, J. Abonyi, and Á. Bárkányi. “Surrogate-based flowsheet model maintenance for Digital Twins”. In: *Digital Chemical Engineering* 15 (2025), p. 100228.

[119] G. Symeonidis, E. Nerantzis, A. Kazakis, and G. A. Papakostas. “Mlops-definitions, tools and challenges”. In: *2022 IEEE 12th Annual Computing and Communication Workshop and Conference (CCWC)*. IEEE. 2022, pp. 0453–0460.

[120] A. Gepperth and B. Hammer. “Incremental learning algorithms and applications”. In: *European symposium on artificial neural networks (ESANN)*. 2016.

[121] L. T. Meyer, M. Schouler, R. A. Caulk, A. Ribes, and B. Raffin. “Training deep surrogate models with large scale online learning”. In: *International Conference on Machine Learning*. PMLR. 2023, pp. 24614–24630.

[122] B. Hammer and M. Toussaint. “Special issue on autonomous learning”. In: *KI-Künstliche Intelligenz* 29.4 (2015), pp. 323–327.

[123] D. Honeycutt, M. Nourani, and E. Ragan. “Soliciting human-in-the-loop user feedback for interactive machine learning reduces user trust and impressions

of model accuracy". In: *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*. Vol. 8. 2020, pp. 63–72.

[124] J. Montiel, M. Halford, S. M. Mastelini, G. Bolmier, R. Sourty, R. Vaysse, A. Zouitine, H. M. Gomes, J. Read, T. Abdessalem, et al. "River: machine learning for streaming data in python". In: *Journal of Machine Learning Research* 22.110 (2021), pp. 1–8.

[125] V. Losing, B. Hammer, and H. Wersing. "Incremental on-line learning: A review and comparison of state of the art algorithms". In: *Neurocomputing* 275 (2018), pp. 1261–1274.

[126] S. Hoi, D. Sahoo, J. Lu, and P. Zhao. "A comprehensive survey". In: *Neurocomputing, Online Learning* (2021).

[127] R. Yang, T. Yang, Z. Yan, T. Krajnik, and Y. Ruichek. "Preventing Catastrophic Forgetting in Continuous Online Learning for Autonomous Driving". In: *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2024, pp. 5505–5512.

[128] X. Ying. "An overview of overfitting and its solutions". In: *Journal of physics: Conference series*. Vol. 1168. IOP Publishing. 2019, p. 022022.

[129] D. Brignac, N. Lobo, and A. Mahalanobis. "Improving replay sample selection and storage for less forgetting in continual learning". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023, pp. 3540–3549.

[130] R. J. Hyndman and A. B. Koehler. "Another look at measures of forecast accuracy". In: *International journal of forecasting* 22.4 (2006), pp. 679–688.

[131] R. Tiwari, K. Killamsetty, R. Iyer, and P. Shenoy. "Gcr: Gradient coreset based replay buffer selection for continual learning". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 99–108.

[132] Z. Mai, R. Li, H. Kim, and S. Sanner. "Supervised contrastive replay: Revisiting the nearest class mean classifier in online class-incremental continual learning". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021, pp. 3589–3599.

[133] K. R. M. Fernando and C. P. Tsokos. "Dynamically weighted balanced loss: class imbalanced learning and confidence calibration of deep neural networks". In: *IEEE Transactions on Neural Networks and Learning Systems* 33.7 (2021), pp. 2940–2951.

[134] S. Sen, K. P. Singh, and P. Chakraborty. "Dealing with imbalanced regression problem for large dataset using scalable Artificial Neural Network". In: *New Astronomy* 99 (2023), p. 101959.

[135] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, et al. "Overcoming catastrophic forgetting in neural networks". In: *Proceedings of the national academy of sciences* 114.13 (2017), pp. 3521–3526.

[136] Y. Lin, L. Chen, H. Huang, C. Ma, X. Han, and S. Cui. "Task-aware sampling layer for point-wise analysis". In: *IEEE Transactions on Visualization and Computer Graphics* 30.10 (2022), pp. 6612–6624.

[137] J. Kozal and M. Wozniak. "Increasing depth of neural networks for life-long learning". In: *Information Fusion* 98 (2023), p. 101829.

[138] Y. Gal and Z. Ghahramani. "Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning". In: *Proceedings of The 33rd International Conference on Machine Learning*. Ed. by M. F. Balcan and K. Q. Weinberger. Vol. 48. Proceedings of Machine Learning Research. New York, New York, USA: PMLR, 20–22 Jun 2016, pp. 1050–1059. URL: `https://proceedings.mlr.press/v48/gal16.html`.

[139] J. M. Hernández-Lobato and R. Adams. "Probabilistic backpropagation for scalable learning of bayesian neural networks". In: *International conference on machine learning*. PMLR. 2015, pp. 1861–1869.

[140] R. Dwivedi, D. Dave, H. Naik, S. Singhal, R. Omer, P. Patel, B. Qian, Z. Wen, T. Shah, G. Morgan, et al. "Explainable AI (XAI): Core ideas, techniques, and solutions". In: *ACM computing surveys* 55.9 (2023), pp. 1–33.

[141] F. H. Riddick and Y.-T. Lee. "Core Manufacturing Simulation Data (CMSD): A standard representation for manufacturing simulation-related information". In: *Fall Simulation Interoperability Workshop (Fall SIW)*. SISO. Orlando, Florida, USA, 2010, pp. 1–9.

[142] I. IEC. "62264-1: Enterprise-Control System Integration–Part 1: Models and Terminology". In: *IEC: Geneva, Switzerland* (2013).

[143] *ISO 15531-44:2017 Industrial automation systems and integration — Industrial manufacturing management data — Part 44: Information modelling for shop floor data acquisition*. `https://www.iso.org/standard/71064.html`. Accessed: 2023-01-01.

[144] *QIF (Quality Information Framework): 2021 Definitive Guide*. `https://www.capvidia.com/blog/qif-quality-information-framework-definitive-guidel`. Accessed: 2023-01-01.

[145] R. Pechter. "Conformance standard for the predictive model markup language". In: *Proceedings of the 4th International Workshop on Data Mining Standards, Services, and Platforms*. 2006, pp. 6–13.

[146] *Portable Format for Analytics (PFA)*. `https://dmg.org/pfa/`. Accessed: 2023-01-01.

[147] *Open Software. Open ideas. Open your architecture and expand.* `https://oagi.org/`. Accessed: 2023-01-01.

[148] M. J. Pratt. “Introduction to ISO 10303—The STEP standard for product data exchange”. In: *Journal of Computing and Information Science in Engineering* 1.1 (2001), pp. 102–103.

[149] *B2MML*. `https://mesa.org/topics-resources/b2mml/`. Accessed: 2023-01-01.