People's Democratic Republic of Algeria

Ministry of Higher Education and Scientific Research

Yahia Fares University of Medea

Faculty of Sciences

Department of Mathematics and Computer Science

Master's Final Year Project

Field: Computer Science

Specialization: Applied Artificial Intelligence

Monitoring and Optimization of a Catalytic Process:

Application to an Isomerization Unit at Sonatrach

Prepared by :

GUEMANA Fella

DIFALLAH Yamina

Supervised by :

Dr. RAHIM Messaoud

Defended in June 2026 before the jury:

Dr. CHETOUANE Mohamed	President
Dr. HAYI Mohamed Yassine	Examiner
Dr. RAHIM Messaoud	Supervisor
Mme TLIBA Safa	Co-Supervisor

Academic Year : 2025 – 2026

Dedications

In the name of Allah, the Most Merciful and the Most Compassionate.

First and foremost, I offer my deepest gratitude to **Allah Almighty** for His infinite blessings, mercy, and the strength to complete this work.

I dedicate this work to **myself**, in honor of the resilience, hard work, and commitment that carried me through this academic journey.

To my beloved parents, **Mr. Difallah Mohamed** and **Mrs. Sadouki Fayza**, I express my deepest love and appreciation. Your endless support and faith in me are the foundation of every success I achieve.

To my dear brother, **Haithem**, and sister, **Acil**, thank you for your constant encouragement, presence, and affection.

A special dedication goes to my partner and best friend, **Guemana Fella**. Thank you for your unwavering support, incredible teamwork, and invaluable friendship throughout this experience.

I extend my sincere thanks to my supervisor, **Mr. Messoudi Rahim**, for his guidance, patience, and valuable expertise.

Finally, thank you to everyone who contributed, directly or indirectly, to my education and personal growth.

To all of you, I extend my deepest gratitude.

Yamina

First and foremost, I express my deepest gratitude to **Allah**, the Most Merciful and Compassionate, for granting me the strength, patience, and guidance throughout this work.

I would also like to dedicate this achievement to **myself**, in recognition of the perseverance, dedication, and hard work I have invested throughout my academic and personal journey. Every challenge I faced has strengthened my determination and contributed to my growth.

To my beloved father, **Mr. Guemana Kamel**, I extend my sincere gratitude for his endless sacrifices, unwavering support, and constant encouragement. Your trust in me and your dedication have always inspired me to strive for excellence.

To my dearest mother, **Mrs. Menadli Fethia**, words cannot adequately express the depth of my gratitude. Your unconditional love, patience, prayers, and continuous support have been my greatest source of strength. You have always believed in me, even during the most difficult moments, and I owe much of my success to you.

To my sister, **Houda**, and my brothers, **Amine and Zahredine**, thank you for your love, support, encouragement, and presence throughout my journey. Your support has always been a source of strength and motivation for me.

A special dedication goes to my best friend, **Difallah Yamina**, whose friendship, support, encouragement, and collaboration have made this experience both meaningful and memorable. Thank you for being an exceptional teammate and a true friend.

I would like to express my sincere appreciation to my academic supervisor, **Mr. MESSOUD Rahim**, for his valuable guidance, constructive advice, availability, and continuous support throughout the realization of this work. His expertise and encouragement have been invaluable to the success of this project.

I would also like to extend my heartfelt thanks to the industrial team of **Sonatrach**, especially **Mrs. Tliba Safaa** and **Mr. Takedjoud Amine**, for their support, guidance, professionalism, and the knowledge they shared with us during our internship. Their advice and encouragement greatly contributed to the successful completion of this project.

To my future self, whenever doubt arises, look back at this achievement and remember how far you have come. May this work always remind you that with faith, passion, perseverance, and effort, you are capable of overcoming challenges, turning ambitions into reality, and achieving far more than you ever imagined.

Fella

Acknowledgements

We would like to express our sincere gratitude to all those who have contributed, directly or indirectly, to the successful completion of this work.

First and foremost, we would like to extend our deepest appreciation to our academic supervisor, **Dr.MESSAOUD Rahim**, for his invaluable guidance, continuous support, insightful advice, and encouragement throughout the development of this project. His expertise and dedication have greatly contributed to the achievement of this work.

We would also like to express our heartfelt thanks to the industrial team at Sonatrach, especially **Mrs. TLIBA Safaa** and **Mr. TAKEDJOUD Amine**, for their guidance, availability, technical support, and the valuable knowledge they shared with us during our internship. Their assistance was instrumental in the successful realization of this project.

Our sincere thanks are also extended to the members of the jury for accepting to evaluate this work and for the time and attention they have devoted to reviewing it.

We would like to express our gratitude to all the professors and staff members of the Department of Mathematics and Computer Science at **Yahia Farès University of Médéa** for the quality of education, knowledge, and support they provided throughout our academic journey.

Finally, we would like to thank our families for their unconditional love, patience, encouragement, and unwavering support throughout our studies. Their confidence in us has been a constant source of motivation and strength.

To everyone who has contributed in any way to this accomplishment, we offer our heartfelt thanks and appreciation.

Abstract

The petroleum refining industry faces increasing challenges in maximizing the production of high-quality, environmentally compliant fuels while maintaining operational efficiency. Catalytic isomerization units play a crucial role in this objective by converting low-octane light naphtha into high-octane isomerate. However, the complex nonlinear behavior of the process, coupled with continuous catalyst deactivation, makes conventional optimization approaches difficult to implement and maintain.

This project presents a data-driven Digital Twin framework for the prediction and optimization of the catalytic isomerization process. The proposed approach leverages historical data collected from the Distributed Control System (DCS) to develop surrogate models capable of accurately predicting the Research Octane Number (RON) of the produced isomerate. Several machine learning and deep learning algorithms were evaluated and compared to identify the most suitable predictive model for representing the process behavior.

Building upon the predictive model, an optimization framework was developed to maximize RON by determining the optimal inlet temperature setpoints of reactors R-001 and R-002 while accounting for catalyst aging effects. The Digital Twin enables rapid what-if simulations, performance monitoring, and optimization without interfering with plant operation. By combining machine learning-based prediction with advanced optimization techniques, the proposed framework provides a practical decision-support tool for refinery engineers, facilitating improved process performance, enhanced operational efficiency, and increased economic value.

Keywords: Catalytic Isomerization, Digital Twin, Machine Learning, Process Optimization, Research Octane Number (RON), Catalyst Aging, Data-Driven Modeling, Refinery Operations.

Résumé

L'industrie du raffinage pétrolier est confrontée à des défis croissants pour maximiser la production de carburants de haute qualité conformes aux normes environnementales tout en maintenant une efficacité opérationnelle élevée. Les unités d'isomérisation catalytique jouent un rôle essentiel dans cet objectif en transformant le naphta léger à faible indice d'octane en isomérat à haut indice d'octane. Toutefois, le comportement non linéaire complexe du procédé, associé à la désactivation progressive du catalyseur, rend les approches conventionnelles d'optimisation difficiles à mettre en œuvre et à maintenir.

Ce projet propose un cadre de Jumeau Numérique (Digital Twin) basé sur les données pour la prédiction et l'optimisation du procédé d'isomérisation catalytique. L'approche développée exploite les données historiques issues du système de contrôle distribué (DCS) afin de construire des modèles de substitution capables de prédire avec précision le Research Octane Number (RON) de l'isomérat produit. Plusieurs algorithmes d'apprentissage automatique et d'apprentissage profond ont été évalués et comparés afin d'identifier le modèle prédictif le plus performant pour représenter le comportement du procédé.

À partir de ce modèle prédictif, un cadre d'optimisation a été développé pour maximiser le RON en déterminant les températures d'entrée optimales des réacteurs R-001 et R-002 tout en tenant compte du vieillissement du catalyseur. Le Jumeau Numérique permet de réaliser rapidement des simulations « what-if », d'assurer le suivi des performances et d'effectuer des optimisations sans perturber le fonctionnement de l'unité industrielle. En combinant la prédiction basée sur l'intelligence artificielle et des techniques avancées d'optimisation, le cadre proposé constitue un outil d'aide à la décision performant pour les ingénieurs de raffinage, contribuant à l'amélioration des performances du procédé, de l'efficacité opérationnelle et de la valeur économique de l'installation.

Mots-clés : Isomérisation catalytique, Jumeau Numérique, Apprentissage automatique, Optimisation des procédés, Research Octane Number (RON), Vieillissement du catalyseur, Modélisation basée sur les données, Raffinage pétrolier.

ملخص

تواجه صناعة تكرير النفط تحديات متزايدة تتمثل في تعظيم إنتاج الوقود عالي الجودة والمتوافق مع المعايير البيئية مع الحفاظ على كفاءة التشغيل. وتُعد وحدات الأزمرة الحفزية (Catalytic Isomerization) من أهم الوحدات المستخدمة لتحقيق هذا الهدف، حيث تقوم بتحويل النفثا الخفيفة منخفضة رقم الأوكتان إلى إيزوميرات عالية رقم الأوكتان. إلا أن السلوك غير الخطي المعقد للعملية، بالإضافة إلى التدهور المستمر في نشاط المحفز، يجعل أساليب التحسين التقليدية صعبة التطبيق وذات فعالية محدودة.

يقدم هذا المشروع إطار عمل قائمًا على مفهوم التوأم الرقمي (Digital Twin) للتنبؤ بعملية الأزمرة الحفزية وتحسينها اعتمادًا على البيانات. يعتمد النهج المقترح على البيانات التاريخية المستخرجة من نظام التحكم الموزع (DCS) لتطوير نماذج بديلة قادرة على التنبؤ بدقة برقم الأوكتان البحثي (RON) للإيزوميرات المنتجة. وقد تم تقييم ومقارنة عدة خوارزميات للتعلم الآلي والتعلم العميق بهدف تحديد النموذج الأكثر كفاءة في تمثيل سلوك العملية الصناعية.

وبالاعتماد على النموذج التنبؤي الأفضل أداءً، تم تطوير إطار تحسين يهدف إلى تعظيم قيمة RON من خلال تحديد درجات حرارة الدخول المثلى للمفاعلين R-001 وR-002 مع الأخذ بعين الاعتبار تأثير تقادم المحفز. يتيح التوأم الرقمي إجراء محاكاة سريعة للسيناريوهات التشغيلية المختلفة، ومراقبة الأداء، وتنفيذ عمليات التحسين دون التأثير على التشغيل الفعلي للوحدة الصناعية. ومن خلال دمج تقنيات التنبؤ المعتمدة على الذكاء الاصطناعي مع خوارزميات التحسين المتقدمة، يوفر الإطار المقترح أداة فعالة لدعم اتخاذ القرار بالنسبة لمهندسي التكرير، مما يساهم في تحسين أداء العملية، وزيادة الكفاءة التشغيلية، وزيادة القيمة الاقتصادية للمصفاة.

الكلمات المفتاحية: الأزمرة الحفزية، التوأم الرقمي، التعلم الآلي، تحسين العمليات، رقم الأوكتان البحثي ،(RON) تقادم المحفز، النمذجة المعتمدة على البيانات، تكرير النفط.

Contents

General Introduction **11**

I Host Organization and Problem Statement **13**
1 Introduction . . . 13
2 SONATRACH Organization . . . 14
3 The Algiers Refinery . . . 15
4 Sonatrach – RPC Activity . . . 15
5 Strategic Objectives of Sonatrach RPC . . . 16
6 General Overview of Petroleum Refining . . . 16
7 Light Naphtha Isomerization Unit (Unit U510) . . . 18
7.1 General Description of the Unit . . . 18
7.2 Isomerization Catalyst . . . 19
8 Critical Operating Parameters . . . 20
9 Operational Limitations of Conventional Monitoring Approaches . . . 22
10 Problem Statement and Proposed Digital Twin Architecture . . . 22
11 Conclusion . . . 23

II Artificial Intelligence for Prediction and Optimization **24**
1 Introduction . . . 24
2 Artificial Intelligence, Machine Learning, and Deep Learning . . . 25
2.1 Main Learning Paradigms in Artificial Intelligence . . . 26
2.2 Types of Problems Solved by Artificial Intelligence . . . 27
3 Artificial Intelligence for Prediction and Optimization . . . 28
3.1 Artificial Intelligence for Prediction . . . 28
3.2 Artificial Intelligence for Optimization . . . 43
4 Digital Twin . . . 53
5 State of the Art in Isomerization Optimization . . . 53
5.1 Machine Learning in Isomerization . . . 54
5.2 Evolutionary and Bayesian Algorithms in Process Optimization . . . 55
6 Conclusion . . . 55

III System Design **57**
1 Introduction . . . 57

2 Global Architecture . . . 57
2.1 Operator Workflow . . . 60
2.2 AI Engineer Workflow . . . 61
3 Detailed Design . . . 61
3.1 Layer 1: Physical System . . . 61
3.2 Layer 2: Data-Acquisition & Integration Layer . . . 64
3.3 Layer 3: Digital Twin Platform . . . 69
3.4 Layer 4: Role-Based End-User Applications Layer . . . 72
4 Conclusion . . . 75

IV Implementation and Results Discussion **77**
1 Introduction . . . 77
2 Development Tools and Environment . . . 77
2.1 Frontend Technologies . . . 77
2.2 Backend Technologies . . . 78
2.3 Machine Learning and Optimization Components . . . 78
2.4 Data Storage Strategy . . . 79
2.5 Reporting Services . . . 80
2.6 Summary . . . 80
3 Digital Twin Implementation and Evaluation . . . 81
3.1 Data Storage Implementation . . . 81
3.2 AI Service Implementation . . . 81
3.3 Prediction Models . . . 82
3.4 Optimization Solution . . . 84
3.5 Discussion: Predictive Model Selection . . . 86
3.6 Discussion: Optimization Algorithm Selection . . . 86
3.7 Decision Module . . . 87
4 Application Interface . . . 87
4.1 Live DCS Monitoring . . . 89
4.2 Simulation Optimizer . . . 90
4.3 Additional Operational Modules . . . 92
4.4 AI Engineer Interface . . . 93
5 Conclusion . . . 95

General Conclusion **96**

Bibliography **98**

List of Tables

I.1 Critical Operating Parameters of Unit U510. 19
I.2 Optimized Operational Constraints and Parameter Impacts for Unit U510. . . . 21

II.1 Overview of the eight optimization algorithms benchmarked in this study, detailing their structural classifications, operational search strategies, and core hyperparameter parameters. 44

III.1 DCS Tag Mapping Dictionary 66

IV.1 Prediction model performance summary on the held-out test set. 82
IV.2 Prediction model performance summary on the held-out test set. 82
IV.3 Cross-validation performance of predictive models. 84
IV.4 Single-objective static optimizer benchmark comparison. 85
IV.5 Optimizer effectiveness and stability over a 15-day dynamic evaluation window (Mean Best Octane $\approx$ 87.83–87.84 for all algorithms). 85

List of Figures

Figure I.1 General Organizational Chart of Sonatrach [1]. 14
Figure I.2 General schematic representation of crude oil fractional distillation cuts and related industrial applications [4]. 17

Figure II.1 Hierarchy of intelligent systems: AI encompasses ML, which in turn encompasses DL. Adapted from Russell & Norvig [7] and LeCun et al. [10]. 25
Figure II.2 Decision Tree: Example structure for RON prediction using reactor operational variables and catalyst age criteria. Adapted from Hastie et al. [20] and Géron [11]. 29
Figure II.3 Random Forest Architecture: bootstrap aggregation (bagging) of independent decision trees [26]. 30
Figure II.4 Conceptual threshold selection strategy: Exhaustive optimization in Random Forest (left) versus randomized threshold sampling in Extra Trees (right) [37]. 31
Figure II.5 Gradient Boosting: sequential residual correction across three boosting stages. Adapted from Friedman [15]. 32
Figure II.6 Visual impact of structural regularization in XGBoost: Overfitting mitigation (left) and leaf weight shrinkage via L_2 penalties (right) [16]. . 33
Figure II.7 LightGBM tree growth mechanics: Conventional balanced level-wise growth (left) versus asymmetric, optimized leaf-wise growth (right) [17]. 34
Figure II.8 CatBoost architectural innovations: Target leakage mitigation via ordered boosting (left) versus computationally efficient symmetric (oblivious) tree topology (right) [18]. 36
Figure II.9 Deep Neural Network (DNN) architecture: input layer (blue), multiple hidden layers with ReLU activations (green), and output layer (red). Adapted from Goodfellow et al. [19] and Schmidhuber [30]. 37
Figure II.10 Convolutional Neural Network (CNN) architecture: input layer (blue), stacked convolutional layers (purple), max-pooling downsampling layers (green), and dense fully connected layers (orange). Adapted from LeCun et al. [31] and Goodfellow et al. [19]. 38

Figure II.11 Multi-Layer Perceptron (MLP) topology: input features, three dense hidden layers (128-64-32) with ReLU activation functions, and a single continuous RON output node. Adapted from Goodfellow et al. [19] and Rumelhart et al. [32]. 40
Figure II.12 Recurrent Neural Network (RNN) unrolled over five consecutive time steps: temporal sequence inputs (blue), hidden state context tracking (green), and time-step outputs (red). Adapted from Goodfellow et al. [19] and Schmidhuber [30]. 41
Figure II.13 LSTM cell architecture: forget gate (red), input gate (green), candidate cell tanh (purple), and output gate (blue). The cell state $c^{(t)}$ serves as a long-range memory highway; the hidden state $h^{(t)}$ is the output at each time step. Adapted from Hochreiter & Schmidhuber [33]. 43
Figure II.14 NSGA-II algorithm flowchart configured for single-objective optimization (maximizing RON). Adapted from Deb et al. [47] and Deb [12]. . 46
Figure II.15 SPEA2 archive mechanism in single-objective tracking mode, highlighting the selection frontier of maximum-RON solutions. Adapted from Zitzler et al. [44]. 48
Figure II.16 PSO particle mechanics updating spatial positions and tracking the global swarm center gBest across an operational landscape. Adapted from Kennedy and Eberhart [38]. 49
Figure II.17 Differential Evolution structural phases: Vector difference mutation geometry (left) and binomial trial vector crossover mapping (right). Adapted from Storn and Price [39]. 50
Figure II.18 Simulated Annealing profile: Exponential decay cooling schedule (left) and temperature-dependent Metropolis acceptance probability boundaries (right). Adapted from Kirkpatrick et al. [28]. 51
Figure II.19 Optuna TPE mechanics: Historical parameter splitting into density distributions $l(x)$ and $g(x)$ (left) alongside the prioritized Expected Improvement acquisition peak (right). Adapted from Akiba et al. [43] and Bergstra et al. [42]. 52
Figure II.20 Bayesian Optimization framework: Gaussian Process surrogate mean and uncertainty bounds (left) linked to the calculated Expected Improvement acquisition function profile (right). Adapted from Jones et al. [41] and Mockus [40]. 53

Figure III.1 Conceptual Process Optimization Platform – Workflow by Layers . . 58
Figure III.2 Activity diagram describing the operational workflow of the proposed Digital Twin-based optimization system and the interactions between Operators and AI Engineers. 60
Figure III.3 Physical configuration of the isomerization process. 62

Figure III.4 Data acquisition and transmission pipeline. 64
Figure III.5 The Digital Twin system organization and interconnected modules. . . 70
Figure III.6 Continuous feedback and closed-loop model retraining workflow. . . 72
Figure III.7 Intelligent Digital Twin Optimization Platform – Operator Workspace Dashboard (Figure 3.2). 72
Figure III.8 UML Use Case Diagram for the Role-Based End-User Applications Layer (Figure 3.1). 75

Figure IV.1 ISO-OPT platform role-based secure authentication gateway. 88
Figure IV.2 Sonatrach multivariable optimization console and real-time process loop overview. 89
Figure IV.3 Live DCS process monitoring and reactor status interface. 90
Figure IV.4 Catalyst Age Configuration Interface (Step 1). 90
Figure IV.5 Real-time baseline snapshot evaluation panel (Step 2). 91
Figure IV.6 Optimization routine execution parameters view (Step 3). 91
Figure IV.7 Operator review, safety verification, and setpoint execution panel (Step 4). 92
Figure IV.8 Real Experiment laboratory data entry interface. 92
Figure IV.9 The Alarm Center alert aggregation view. 93
Figure IV.10 Model performance diagnostic metrics dashboard. 93
Figure IV.11 Model Retraining Center parameter configuration panel. 94
Figure IV.12 Database Model Training History archive registry. 94
Figure IV.13 Dataset Management control view for ingestion quality control and pipeline tracking. 95

List of Abbreviations

1D-CNN	One-Dimensional Convolutional Neural Network
2D-CNN	Two-Dimensional Convolutional Neural Network
AI	Artificial Intelligence
ANN	Artificial Neural Network
API	Application Programming Interface
B	Total number of bootstrap trees in an ensemble (Ensemble Bagging Size)
BO	Bayesian Optimization
C-001	Product Stabilization Fractionation Column
C5	Pentane fraction (Hydrocarbons containing 5 carbon atoms)
C6	Hexane fraction (Hydrocarbons containing 6 carbon atoms)
CatBoost / CAT	Categorical Boosting
CFR	Cooperative Fuel Research (Standardized testing engine for octane rating)
CNN	Convolutional Neural Network
COM	Marketing Activity (*Activité Commercialisation* - Sonatrach)
CPS	Cyber-Physical System
CV	Coefficient of Variation
DB	Database
DBMS	Database Management System
DCS	Distributed Control System
DE	Differential Evolution
DEAP	Distributed Evolutionary Algorithms in Python

DIH Deisohexanizer Column

DL Deep Learning

DNN Deep Neural Network

DT Digital Twin

E&P Exploration and Production Activity (*Activité Exploration et Production* - Sonatrach)

ET Extra Trees (Extremely Randomized Trees)

EXP Express.js Web Application Framework

GB Gradient Boosting

GBDT Gradient Boosting Decision Trees

H2/HC Hydrogen-to-Hydrocarbon Molar Ratio

HCl Hydrochloric Acid

HMI Human-Machine Interface

I/O Input / Output (Modules or Signals)

IQR Interquartile Range

ISO-OPT Isomerization Optimization Platform

JS JavaScript

L1 Lasso Regularization Penalty (L_1 absolute weight norm)

L2 Ridge Regularization Penalty (L_2 squared weight norm)

LHSV Liquid Hourly Space Velocity

LightGBM / LGBM Light Gradient Boosting Machine

LPG Liquefied Petroleum Gas

LQS Liquefaction and Separation Activity (*Activité Liquéfaction et Séparation* - Sonatrach)

LSTM Long Short-Term Memory Network

MAE Mean Absolute Error

MAPE Mean Absolute Percentage Error

ML	Machine Learning
MLP	Multi-Layer Perceptron
MSE	Mean Squared Error
Node.js	JavaScript Runtime Environment
NSGA-II	Non-dominated Sorting Genetic Algorithm II
NumPy	Numerical Python Scientific Computing Library
OPC	Open Platform Communications
OPC UA	Open Platform Communications Unified Architecture
Optuna	Hyperparameter Optimization Framework based on TPE
Pandas	Python Data Analysis and Data Manipulation Library
PDF	Portable Document Format
PID	Proportional-Integral-Derivative (Regulatory Control Loop)
PostgreSQL	Postgre Structured Query Language Relational Database Management System
Pt	Platinum
Pt/Al2O3-Cl	Platinum-based catalyst supported on Chlorinated Alumina
R-1 / R-301 / R-001	Isomerization Reactor 1 (Lead Fixed-Bed Adiabatic Reactor)
R-2 / R-302 / R-002	Isomerization Reactor 2 (Lag Fixed-Bed Adiabatic Reactor)
R2 / R^2	Coefficient of Determination (R^2)
RBAC	Role-Based Access Control
ReLU	Rectified Linear Unit (Activation Function)
REST	Representational State Transfer
RF	Random Forest
RFCC	Residue Fluid Catalytic Cracking
RMSE	Root Mean Squared Error
RON	Research Octane Number
RPC	Refining and Petrochemicals Activity (*Activité Raffinage et Pétrochimie* - Sonatrach)

RTD	Resistance Temperature Detector
SciPy	Scientific Python Library
SHAP	Shapley Additive exPlanations
skopt	Scikit-Optimize Library
SQL	Structured Query Language
TCE	Tetrachloroethylene (Organic chloride promoter agent)
TPE	Tree-structured Parzen Estimator
TRC	Pipeline Transport Activity (*Activité Transport par Canalisations* - Sonatrach)
TS	TypeScript
U510	Light Naphtha Isomerization Unit index 510 at the Algiers Refinery
UI	User Interface
UML	Unified Modeling Language
Vite	Next-Generation Frontend Build Tool
WABT	Weight Average Bed Temperature
XGBoost / XGB	Extreme Gradient Boosting
ΔP	Differential Pressure (Pressure Drop across catalyst beds)
$T_{\text{in_R1}}$	Inlet Temperature of Reactor R-001
$T_{\text{in_R2}}$	Inlet Temperature of Reactor R-002

General Introduction

The petroleum refining industry plays a vital role in meeting the global demand for transportation fuels and petrochemical feedstocks. In recent decades, increasingly stringent environmental regulations and growing market competition have compelled refineries to improve product quality while maximizing operational efficiency. Among the key quality indicators of gasoline, the Research Octane Number (RON) is particularly important, as it determines the fuel's resistance to engine knocking and directly impacts its commercial value.

Catalytic isomerization is one of the most effective refining processes for increasing gasoline octane without generating environmentally harmful compounds. By converting low-octane linear paraffins into high-octane branched isomers, isomerization units contribute significantly to the production of cleaner and higher-value gasoline. However, the operation of these units is characterized by complex nonlinear behavior influenced by numerous process variables, catalyst activity, and changing operating conditions. In addition, catalyst deactivation over time alters process performance, making the identification of optimal operating conditions a challenging task.

Traditionally, process optimization relies on operator experience, empirical correlations, or first-principles models. Although these approaches have proven useful, they often struggle to capture the complexity of modern industrial processes and may require considerable time and resources to maintain. The rapid development of Artificial Intelligence (AI), Machine Learning (ML), and Digital Twin technologies offers new opportunities for addressing these limitations through data-driven modeling and optimization.

A Digital Twin is a virtual representation of a physical system that uses historical and real-time data to reproduce, analyze, and predict process behavior. When combined with machine learning algorithms, digital twins can provide accurate process predictions and enable advanced optimization strategies without interfering with actual plant operation. Such technologies are increasingly being adopted in Industry 4.0 initiatives to improve operational efficiency, reduce costs, and support intelligent decision-making.

This work focuses on the catalytic isomerization unit U510 and proposes a data-driven Digital Twin framework for predicting and optimizing the Research Octane Number (RON). Historical data collected from the Distributed Control System (DCS) are used to develop predictive machine learning models capable of accurately representing process behavior. The resulting Digital Twin is then coupled with optimization algorithms to determine the optimal inlet tem-

perature setpoints of reactors R-001 and R-002, with the objective of maximizing RON while accounting for catalyst aging effects.

The main contribution of this work lies in the integration of machine learning-based prediction and optimization within a Digital Twin environment dedicated to catalytic isomerization. The proposed framework provides refinery engineers with a practical decision-support tool for evaluating operating scenarios, improving product quality, and enhancing process performance.

This dissertation is organized as follows. Chapter 1 presents the industrial context of the study. It introduces Sonatrach and the refinery environment, followed by a detailed description of the catalytic isomerization process. Chapter 2 provides the theoretical foundations of the proposed approach. It introduces the concepts of Artificial Intelligence, Machine Learning, Digital Twins, and optimization techniques. Chapter 3 focuses on the conception and architecture of the proposed system, describing the overall workflow. Chapter 4 presents the implementation and experimental results, detailing data preprocessing, model development, training and evaluation procedures, optimization results, and the implementation of the user interface. Finally, the conclusion summarizes the main findings and outlines potential directions for future research.

Chapter I

Host Organization and Problem Statement

1 Introduction

Petroleum refining strategically transforms crude oil into high-value products like gasoline, diesel, kerosene, and liquefied petroleum gas (LPG) [1]. Within modern refinery configurations, gasoline upgrading units are vital for meeting increasingly strict global environmental and fuel quality standards.

In this context, the Light Naphtha Isomerization Unit (Unit U510) is a critical asset. Its primary objective is to maximize the gasoline octane rating by catalytically converting low-octane normal paraffins into high-octane branched isoparaffins without increasing aromatic or lead content [3].

However, maintaining optimal performance is difficult because the process is highly sensitive to operating conditions and catalyst health. Fluctuations in reactor temperatures, hydrogen-to-hydrocarbon (H_2/HC) ratios, feed purity, and chloride injection rates directly govern catalytic activity. Legacy monitoring methods offer limited early-warning capabilities for process drift or catalyst deactivation, frequently resulting in a dropped Research Octane Number (RON), high hydrogen consumption, and inflated operating costs [4].

Concurrently, advances in Artificial Intelligence (AI) and Machine Learning (ML) offer new solutions for industrial process optimization. Predictive soft-sensing and intelligent meta-heuristics can supplement conventional control strategies, enabling real-time decision support and proactive asset management [5]. Driven by these opportunities, this thesis develops an AI-powered digital twin framework specifically tailored to Unit U510.

This chapter establishes the industrial foundation of our project. First, it introduces the Algiers Refinery and Sonatrach's Refining & Petrochemicals (RPC) activities. It then details the catalytic isomerization mechanism of Unit U510 and its critical operating parameters. Finally, it analyzes the limitations of conventional monitoring approaches to define the core problem addressed in this work.

2 SONATRACH Organization

By the year 2030, Sonatrach will look very different from the current corporate giant, which has raised numerous questions for several years now and, above all, has struggled to emerge from a prolonged period of turbulence that ultimately impacted its production. The organizational chart in Figure I.1 represents the macrostructure of SONATRACH [1]:

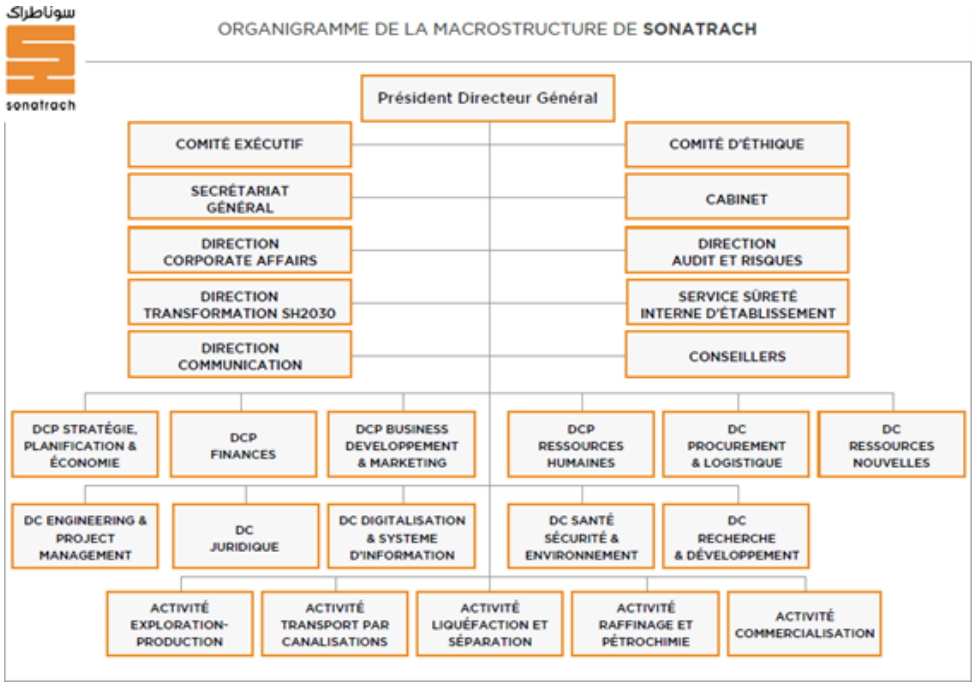


Figure I.1: General Organizational Chart of Sonatrach [1].

Each activity from the SONATRACH Activities within this enterprise carries out its core operations, develops its business portfolio, and contributes, within its respective field of expertise, to the development of the company's international operations. **Among these pillars, our research work is specifically prepared for and dedicated to the Refining and Petrochemical (RPC) Activity, focusing directly on catalytic process optimization to enhance operational efficiency.**

The operational structures are organized around the following core activities:

- **The Exploration and Production Activity (E&P):** Responsible for developing and implementing exploration, development, and exploitation policies and strategies for upstream oil and gas sectors, in line with the strategic objectives of the Company.
- **The Pipeline Transport Activity (TRC):** Responsible for developing and implementing policies and strategies regarding hydrocarbon transport via pipelines, in line with the strategic objectives of the Company.
- **The Liquefaction and Separation Activity (LQS):** Responsible for developing and implementing operating, management, and development policies and strategies for gas liquefaction and separation activities, in line with the strategic objectives of the Company.

- **The Refining and Petrochemicals Activity (RPC):** Responsible for developing and implementing operating, management, and development policies and strategies for refining and petrochemical sectors, in line with the strategic objectives of the Company.
- **The Marketing Activity (COM):** Responsible for developing and implementing marketing policies and strategies for hydrocarbons across both international and domestic markets, in line with the strategic objectives of the Company.

3 The Algiers Refinery

Algeria possesses several strategic refineries distributed across the country, notably in Algiers, Skikda, Arzew, Adrar, and Hassi Messaoud. These refineries are operated by Sonatrach, the national oil and gas company and the leading actor in the Algerian energy sector [1].

Among them, the Algiers Refinery, located in Sidi Arcine, plays a major role in supplying petroleum products to the central region of the country. Commissioned in 1964, the refinery has undergone several modernization projects aimed at increasing refining capacity, improving energy efficiency, and enhancing product quality [1].

As part of the national refinery rehabilitation program, several processing units were upgraded to produce fuels compliant with Euro 5 environmental standards. This modernization contributed to:

- Atmospheric and vacuum distillation units
- Naphtha hydrotreating units
- Catalytic reforming units
- Residue Fluid Catalytic Cracking (RFCC) unit
- Gas treatment and sulphur recovery units
- Light Naphtha Isomerization Unit (Unit U510)

The isomerization unit is singled out as a critical asset for producing premium-grade, environmentally friendly gasoline blending components, and constitutes the sole focus of the present work [3].

4 Sonatrach – RPC Activity

Sonatrach is the national company responsible for the exploration, production, transportation, refining, and commercialization of hydrocarbons in Algeria.

Its RPC activity (Refining & Petrochemicals) manages all refinery and petrochemical operations. The main objectives of this activity are:

- Improving petroleum product quality,
- Increasing production capacity,
- Optimizing refinery energy efficiency,
- Reducing operating costs,
- Meeting international environmental standards.

Today, digital transformation and artificial intelligence technologies represent strategic solutions for improving industrial process performance and refinery competitiveness.

5 Strategic Objectives of Sonatrach RPC

The main strategic objectives driving this activity across the national territory encompass [1]:

- **Product Quality Upgrading:** Continuously improving petroleum product quality to align with international market evolution.
- **Capacity Expansion:** Increasing total refining and petrochemical production capacity.
- **Energy Efficiency:** Optimizing refinery energy efficiency and reducing carbon footprint.
- **Cost Optimization:** Minimizing overall plant operating costs through advanced asset management.
- **Environmental Compliance:** Strict compliance with international environmental standards and fuel specifications.

6 General Overview of Petroleum Refining

Crude oil is a complex mixture of hydrocarbons containing paraffins, naphthenes, aromatics, sulfur compounds, and other impurities [4]. Before commercial utilization, crude oil must undergo several chemical separation and conversion processes to isolate its constituent compounds.

The first step of refining is atmospheric distillation, where crude oil is heated and separated into distinct fractions according to their boiling temperature ranges. This fundamental fractionation process is illustrated in Figure I.2:

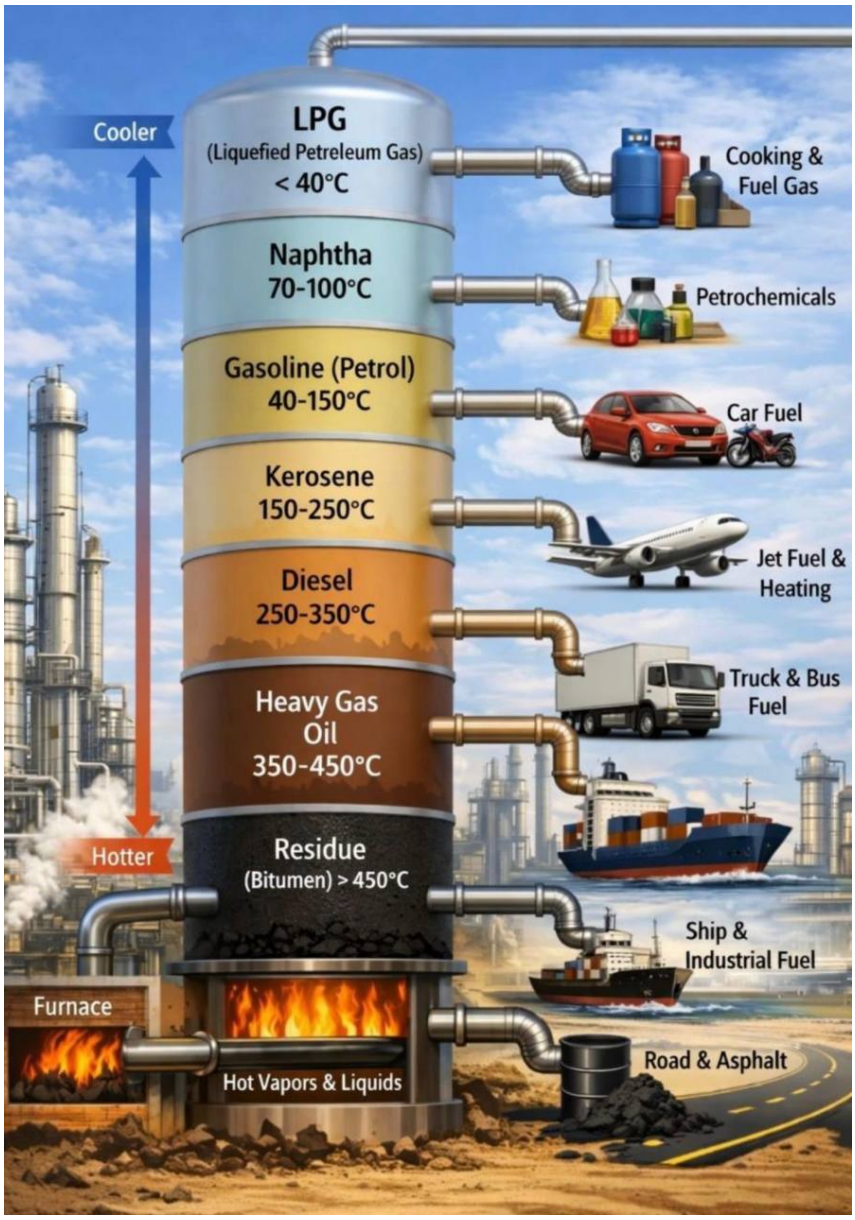


Figure I.2: General schematic representation of crude oil fractional distillation cuts and related industrial applications [4].

As illustrated in Figure I.2, crude oil separation takes place inside the atmospheric distillation tower based on the relative boiling point profiles of the hydrocarbon fractions. The primary petroleum fractions obtained during atmospheric distillation and their corresponding industrial applications include:

- **LPG:** Domestic and refinery fuel gas.
- **Naphtha:** Critical feedstock for petrochemical cracking and gasoline upgrading units (such as isomerization).
- **Gasoline:** Finished automotive fuel for internal combustion engines.
- **Kerosene:** Jet fuel for aviation turbines and localized heating applications.

- **Diesel:** Transport fuel for heavy logistics, civil hardware, and industrial combustion power blocks.
- **Heavy Fractions:** Fuel oils for maritime shipping lanes, industrial process furnaces, asphalt, and bitumen production.

The process begins with atmospheric distillation, where crude oil is heated in a furnace and fed into a distillation column. Different hydrocarbon fractions are separated based on their boiling-point ranges and withdrawn as side streams or overhead products [4].

Light naphtha, composed predominantly of C_5 and C_6 normal paraffins, is of particular strategic importance as the primary feedstock for catalytic isomerization processes. Its low intrinsic octane number (RON $\approx$ 60–70 for n-pentane and n-hexane) necessitates upgrading before blending into finished gasoline [3].

7 Light Naphtha Isomerization Unit (Unit U510)

The Light Naphtha Isomerization Unit is engineered to improve the gasoline octane number by converting low-octane normal paraffins into high-octane branched isoparaffins.

The process primarily treats hydrotreated light naphtha feed, which is composed essentially of C_5 and C_6 hydrocarbons [3]. The principle of catalytic isomerization is based on molecular skeletal rearrangement without altering the total number of carbon atoms within the treated molecules.

Typical chemical conversion pathways executed within the reactors include:

- **Pentane Fraction Conversion:** n-pentane (RON ≈ 62) $\rightarrow$ isopentane (RON ≈ 93)
- **Hexane Fraction Conversion:** n-hexane (RON ≈ 25) $\rightarrow$ isohexanes (such as 2,2-dimethylbutane, RON ≈ 92)

This chemical conversion pathway yields a high-octane isomerate product stream, which is directly routed to the refinery's downstream storage blocks for finished gasoline blending.

7.1 General Description of the Unit

Unit U510 consists of several interconnected process sections responsible for feed preparation, catalytic reaction, product separation, and stabilization.

The principal control variables and their roles are summarised in Table I.1.

Parameter	Typical Range	Impact on Process Performance
Reactor inlet temperature ($T_{\text{in_R1}}$)	103–125 °C	Controls reaction kinetics and thermodynamic equilibrium; lower temperatures favour more branched isomers.
Reactor inlet temperature ($T_{\text{in_R2}}$)	108–132 °C	Fine-tuning of secondary conversion and selectivity in R-2.
Operating pressure	20–35 bar	Maintains hydrogen partial pressure; prevents catalyst coking.
H_2/HC molar ratio	1.0–2.5 mol/mol	Suppresses coking; excess H_2 penalises yield and energy.
LHSV (Liquid Hourly Space Velocity)	1.5–3.0 h^{-1}	Determines contact time; lower LHSV improves conversion at cost of throughput.
Chloride injection rate	Process-dependent	Maintains catalyst acidity; deficiency causes irreversible activity loss.
Feed flow rate & composition	Variable	C_5/C_6 ratio and feed purity directly affect RON output and catalyst lifetime.

Table I.1: Critical Operating Parameters of Unit U510.

The overarching material flow follows a deterministic pathway: Hydrotreated light naphtha is mixed with a controlled hydrogen stream and sent directly to the reactor section under strictly regulated temperature and pressure conditions. Inside the reactors, the bifunctional platinum-based catalyst promotes the conversion of normal paraffins into highly branched isomers with significantly higher octane numbers.

Following the completion of the catalytic cycle, the reactor effluent passes through downstream separation and stabilization sections to isolate two primary streams:

- **Isomerate Product:** High-octane liquid stream directed to the downstream gasoline blending pool.
- **Off-gases:** Light hydrocarbon vapors sent to adjacent refinery treatment units or the plant fuel header.

7.2 Isomerization Catalyst

The catalyst is considered the core element of the isomerization process. Industrial isomerization units generally use platinum-based catalysts supported on chlorinated alumina (Pt/Al_2O_3–Cl), a

bifunctional catalyst that combines metallic (hydrogenation/dehydrogenation) and acid (skeletal rearrangement) functions [4]. The chloride promoter provides the Brønsted acid sites required for carbocation-mediated isomerization. Continuous chloride injection is therefore necessary to compensate for losses during operation.

This catalyst class exhibits exceptional activity at relatively low temperatures (130–180 °C), which thermodynamically favours the formation of highly branched isomers. However, it is also highly sensitive to feed contaminants [4]:

- **Sulfur compounds:** Irreversibly deactivate the platinum metal sites through chemisorption.
- **Water:** Hydrolyzes the surface chloride, destroying acid sites and causing rapid RON decline.
- **Nitrogen and oxygen compounds:** Act as basic poisons that neutralize catalyst acidity.

Progressive catalyst deactivation — manifested as a gradual decline in RON output — is an inherent feature of extended industrial campaigns. Maintaining feed purity and optimal operating conditions is therefore a primary operational objective [4].

8 Critical Operating Parameters

The performance of the isomerization unit strongly depends on the precise control and continuous regulation of several key operating parameters. The main critical variables governing the system, along with their functional roles, are structured in Table I.2 [3]:

Process Parameter	**Detailed Thermodynamic and Hydrodynamic Impact**
Reactor 1 Temperature	Controls initial reaction kinetics. Lower temperatures thermodynamically favor high-octane branched isomers (i-C_5/i-C_6), while higher temperatures risk localized thermal runaways.
Reactor 2 Temperature	Used for final equilibrium fine-tuning. Excessive heating shifts selectivity toward unwanted hydrocracking side-reactions, increasing light gas yields.
Operating Pressure	Stabilizes gas-phase velocity and maintains hydrogen partial pressure across the fixed beds, which is vital to suppress precursor adsorption and catalyst coking.
H_2/HC Molar Ratio	Regulates the hydrogenation-dehydrogenation mechanism on active platinum (Pt) sites. Deficiencies accelerate coking, while an excess penalizes liquid volume recovery.
Space Velocity (LHSV)	Dictates contact time between reactants and active sites. A lower LHSV maximizes the approach to thermodynamic equilibrium but curtails total refinery throughput.
Chloride Injection Rate	Continuously restores surface chloride ions stripped away as volatile HCl by the product vapors, preserving the structural Brønsted acidity of the alumina base.
Feed Flow Stability	Direct multiplier of the unit's hydraulic load. Sharp fluctuations destabilize the liquid-to-gas ratio, causing transient pressure drops (ΔP) across the reactor core.
Feed Contaminants	Trace moisture permanently alters the matrix through irreversible catalyst hydrolysis. Similarly, sulfur compounds temporarily poison platinum sites, causing sharp RON drops.

Table I.2: Optimized Operational Constraints and Parameter Impacts for Unit U510.

Significant or sustained variations in these parameters outside of their optimized operational windows may result in severe operational penalties, including yield reduction, lower octane numbers, catalyst deactivation, process instability, and excessive resource consumption. Optimizing these parameters simultaneously within a high-dimensional search space represents a major industrial challenge that motivates the development of advanced data-driven predictive architectures [5].

9 Operational Limitations of Conventional Monitoring Approaches

Under real industrial conditions, Unit U510 is subject to a range of operational challenges that conventional Distributed Control System (DCS) monitoring cannot fully address. The most frequently encountered issues include progressive catalyst deactivation, periodic chloride imbalances, temperature excursions, and feed quality variability [5].

The principal limitations of conventional monitoring methods are:

- **Delayed detection of process drift:** RON is not measured online in real time; laboratory analysis introduces latencies of several hours, during which suboptimal conditions may persist undetected.
- **Reactive rather than predictive management:** Operators respond to observed deviations rather than anticipating them, limiting the scope for preventive action.
- **No early-warning capability for catalyst deactivation:** The slow, progressive nature of deactivation renders it difficult to diagnose from instantaneous DCS readings alone.
- **Limited decision support:** The high dimensionality of the operating variable space makes manual identification of the optimal operating window impractical.

These limitations collectively motivate the development of an intelligent, data-driven approach to process monitoring and optimization, as clarified in the following sections [5].

10 Problem Statement and Proposed Digital Twin Architecture

The Light Naphtha Isomerization Unit (U510) is a key industrial unit in the refinery that improves gasoline quality by increasing its Research Octane Number (RON). Achieving a high and stable octane level requires continuous control of several operating variables such as reactor temperatures, pressure, hydrogen-to-hydrocarbon ratio, feed conditions, and catalyst state.

In practice, product quality (RON) is not measured online. It is obtained through laboratory analysis with a delay of several hours. This creates a gap between the real process conditions and the available quality information [4]. As a result, the unit may operate under suboptimal conditions without immediate detection.

A major additional challenge is catalyst aging (catalyst deactivation). Over time, the catalyst loses activity, which directly affects the octane level even if operating conditions remain unchanged [4]. This aging effect makes the process highly dynamic and more difficult to control, because the same settings do not always produce the same operational results.

For operators, the relationship between operating conditions, catalyst age, and octane quality is highly nonlinear and difficult to interpret manually. This limits the ability to identify optimal operating conditions in real time. Therefore, there is a clear and demonstrated need for an intelligent system that can predict octane in real time, take into account specific catalyst degradation metrics, support manual interpretation, and recommend optimal operational setpoints.

Proposed Digital Twin Architecture

To address these challenges, this work proposes an Artificial Intelligence-based Digital Twin architecture specifically tailored for the U510 Isomerization Unit [5].

By definition, a Digital Twin is a dynamic virtual model that accurately reproduces the physical and thermodynamic behavior of the real industrial unit by continuously leveraging both historical datasets and real-time operational data streams [45]. Through this dual data integration, the framework enables real-time monitoring, state estimation, and proactive process optimization.

11 Conclusion

This chapter established the industrial and scientific context of the present work. It presented the Algiers Refinery and Sonatrach's downstream operations, described the operating principle and key parameters of the Light Naphtha Isomerization Unit (U510), and identified the main limitations of conventional monitoring approaches. It also highlighted the relevance of Artificial Intelligence techniques for industrial process optimization.

The problem statement has been clearly defined, focusing on the need for a real-time Research Octane Number (RON) prediction system and an intelligent optimization framework capable of recommending optimal operating conditions under dynamic constraints, particularly catalyst aging and feed variability.

The following chapter presents the state of the art in Machine Learning and optimization methods relevant to this problem, providing the theoretical foundation for the methodology developed in the subsequent chapters.

Chapter II

Artificial Intelligence for Prediction and Optimization

1 Introduction

Artificial Intelligence (AI) has emerged as a powerful field that enables computer systems to learn from data, identify complex patterns, make accurate predictions, and support decision-making [7, 8]. By transforming large amounts of data into actionable knowledge, AI provides effective solutions for problems that are difficult to address using conventional mathematical or rule-based approaches [9].

Among the many applications of AI, prediction and optimization are two of the most important and widely used [10]. Prediction aims to estimate unknown or future values by learning the relationships between input variables and target outputs [11]. Optimization, on the other hand, seeks to determine the best set of parameters or decisions that maximize or minimize a given objective while satisfying predefined constraints [12]. The combination of predictive modeling and optimization creates intelligent systems capable of both understanding data and improving performance [13].

This chapter presents the Artificial Intelligence concepts and techniques relevant to this work. It first introduces the relationship between Artificial Intelligence, Machine Learning, and Deep Learning, followed by the main learning paradigms and the different categories of problems solved by AI [7]. The chapter then presents the predictive models and optimization algorithms employed in this study, introduces the digital twin framework, and discusses how prediction and optimization are combined to build an intelligent decision-support system [45].

2 Artificial Intelligence, Machine Learning, and Deep Learning

Within the broad field of AI, three closely related but distinct concepts are essential to understand [7]. These form a nested hierarchy expressed as:

$$\text{AI} \supset \text{ML} \supset \text{DL}$$

Artificial Intelligence (AI) refers to any method that allows machines to behave intelligently, including rule-based expert systems, search and planning algorithms, and data-driven learning [7].

Machine Learning (ML) is a subfield of AI concerned with developing algorithms that learn from data by identifying statistical patterns and relationships. These learned patterns enable models to make predictions or decisions when presented with new, unseen inputs [14]. Common machine learning models include Random Forest, Extra Trees, Gradient Boosting, XGBoost [16], LightGBM [17], and CatBoost [18].

Deep Learning (DL) is a subfield of ML employing multi-layered artificial neural networks to learn hierarchical data representations [19]. Unlike traditional machine learning methods that rely on manually engineered features, deep learning models automatically extract relevant features from raw data through multiple layers of nonlinear transformations [10]. Common deep learning architectures include Deep Neural Networks (DNNs), Convolutional Neural Networks (CNNs), and Long Short-Term Memory (LSTM) networks [**?**].

The structural classification of intelligent computing methods follows a nested hierarchy. **Figure II.1** illustrates the relationship between these fields, demonstrating how Artificial Intelligence (AI) acts as the overarching domain encompassing Machine Learning (ML), which in turn fully encompasses Deep Learning (DL) technologies.

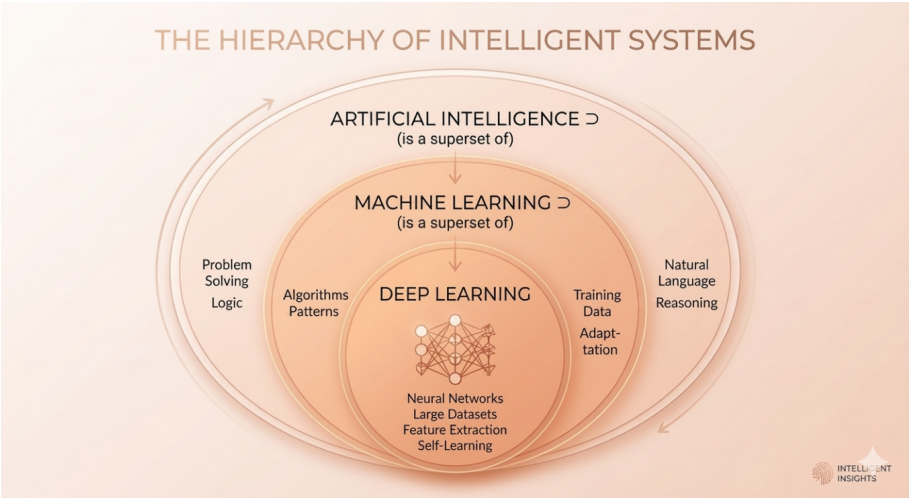


Figure II.1: Hierarchy of intelligent systems: AI encompasses ML, which in turn encompasses DL. Adapted from Russell & Norvig [7] and LeCun et al. [10].

2.1 Main Learning Paradigms in Artificial Intelligence

Artificial Intelligence systems can be categorized according to the way they learn from data [7, 14].

2.1.1 Supervised Learning

Supervised Learning uses labeled datasets, where both input variables and corresponding target outputs are known [14]. The objective is to learn a mapping function capable of predicting outputs for unseen data. It is widely used for classification and regression tasks [20].

Examples:

- Product quality prediction
- Customer churn prediction
- Medical diagnosis

Unsupervised Learning

Unsupervised Learning operates on unlabeled data and aims to discover hidden structures, patterns, or relationships within datasets without predefined outputs [21].

Examples:

- Customer segmentation
- Anomaly detection
- Data clustering

2.1.2 Semi-Supervised Learning

Semi-Supervised Learning combines a small amount of labeled data with a larger quantity of unlabeled data [22]. This approach reduces labeling costs while maintaining acceptable predictive performance.

Examples:

- Medical image classification with limited annotated scans
- Speech recognition systems trained with partially labeled audio data
- Fraud detection using a small set of labeled transactions

2.1.3 Reinforcement Learning

Reinforcement Learning trains an intelligent agent through interactions with an environment [23]. The agent learns optimal actions by receiving rewards or penalties based on its decisions.

Examples:

- Robotics
- Autonomous vehicles
- Industrial process control

2.1.4 Deep Reinforcement Learning

Deep Reinforcement Learning combines neural networks with reinforcement learning techniques [24], enabling agents to solve complex decision-making problems in high-dimensional environments.

Examples:

- Autonomous driving and vehicle navigation
- Robot motion planning and control
- Intelligent process optimization and adaptive control systems

2.2 Types of Problems Solved by Artificial Intelligence

Artificial Intelligence techniques can address various categories of real-world problems depending on the nature of the target variable and the objective of the task [7, 20].

2.2.1 Classification

Classification aims to assign observations to predefined categories or classes [20]. **Output:** Discrete labels (e.g., Normal / Abnormal).

2.2.2 Regression (Prediction)

Regression aims to predict continuous numerical values based on input variables [11]. **Output:** Continuous numerical values.

2.2.3 Clustering

Clustering groups similar observations without predefined labels [21].

2.2.4 Anomaly Detection

Anomaly Detection identifies unusual observations that significantly differ from normal patterns [25].

2.2.5 Recommendation Systems

Recommendation Systems suggest relevant items or actions to users based on historical data and behavioral patterns [27].

2.2.6 Optimization

Optimization seeks the best combination of decision variables to maximize or minimize a specific objective function while satisfying operational constraints [12, 15]. In this work, optimization is employed to determine the optimal operating conditions that maximize the octane number while respecting industrial constraints [13, 45].

2.2.7 Forecasting

Forecasting predicts future values based on historical observations and temporal trends [29].

3 Artificial Intelligence for Prediction and Optimization

Artificial Intelligence plays a major role in solving prediction and optimization problems in many scientific and industrial fields. By learning from historical data, AI models can identify complex relationships between variables and support data-driven decision-making [8, 13].

Prediction and optimization are closely related tasks. Predictive models estimate system behavior or future outcomes, while optimization algorithms use these predictions to determine the best decisions or operating conditions for achieving specific objectives [10, 12].

3.1 Artificial Intelligence for Prediction

Prediction is one of the most widely used applications of AI and ML. It aims to estimate unknown or future values by learning the relationships between input variables and target outputs from historical data [11, 14].

3.1.1 Decision Tree (Foundation)

A **Decision Tree** is a supervised machine learning algorithm used for both regression and classification tasks [20, 21]. It works by recursively partitioning the multidimensional feature space into smaller, distinct, and more homogeneous regions based on a sequence of hierarchical decision rules. At each non-leaf node, the algorithm evaluates the input features and selects the optimal split point that minimizes the local prediction error, such as the Mean Squared Error

(MSE) for regression applications [20]. This recursive partitioning process continues downward until a predefined stopping criterion is satisfied, such as reaching the maximum tree depth, a minimal sample threshold per leaf, or an insufficient reduction in variance [11, 21].

To illustrate this mechanism within the context of refining operations, Figure II.2 presents an example of a trained decision tree structure configured for Research Octane Number (RON) prediction, showcasing how reactor inlet temperatures and catalyst age features are evaluated sequentially to isolate mean RON output partitions.

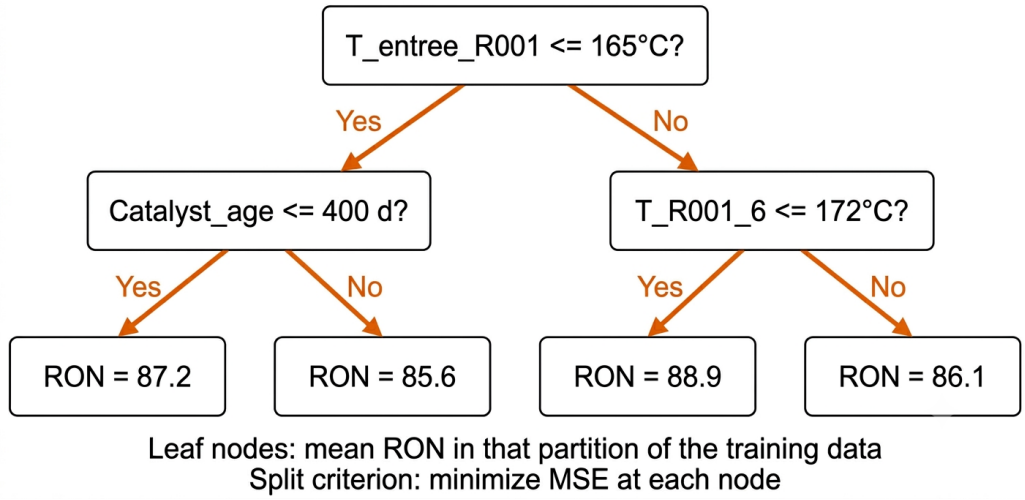


Figure II.2: Decision Tree: Example structure for RON prediction using reactor operational variables and catalyst age criteria. Adapted from Hastie et al. [20] and Géron [11].

3.1.2 Random Forest (RF)

Random Forest (RF) is an ensemble learning algorithm that combines multiple decision trees to improve predictive accuracy and generalization performance [11, 20, 21]. Introduced by Breiman (2001) [26], the algorithm constructs B independent decision trees, each trained on a distinct bootstrap sample drawn randomly with replacement from the original training dataset.

To evaluate a new input vector x, the final prediction of the ensemble is obtained by averaging the individual numerical outputs of all constituent trees for regression tasks:

$$\hat{f}(x) = \frac{1}{B} \sum_{b=1}^{B} T_b(x) \tag{II.1}$$

where $T_b(x)$ denotes the continuous prediction produced by the b-th independent decision tree.

To improve predictive robustness and mitigate overfitting, this architecture introduces a dual-level randomization strategy: bootstrapping the observation data rows, and randomly selecting a subset of features at each split node. This mechanism significantly reduces model variance through tree de-correlation while preserving a low structural prediction bias [20]. Figure II.3 details the overall parallel structural mechanics and data flow of this bootstrap aggregation (bagging) framework.

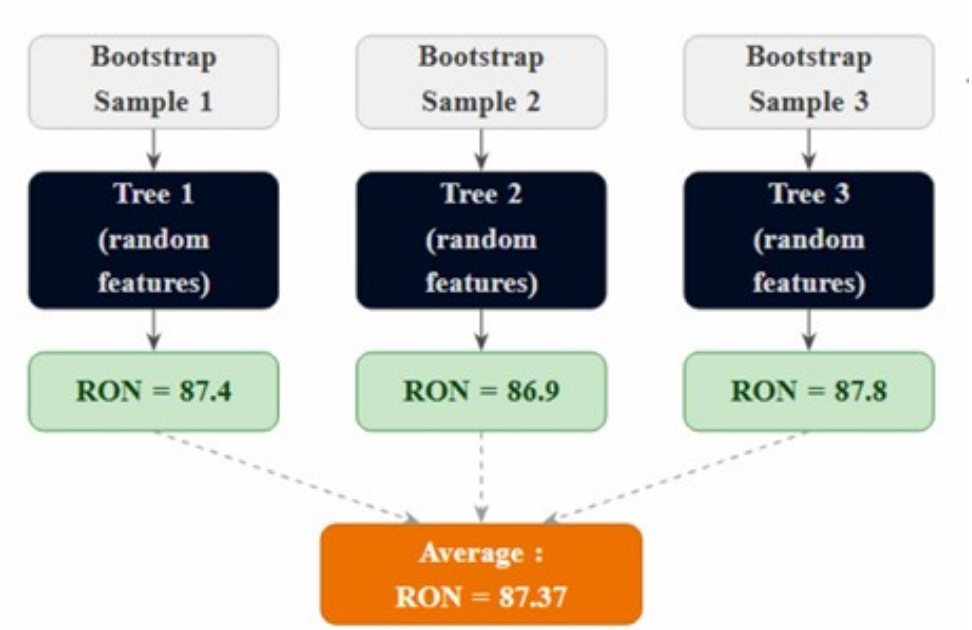


Figure II.3: Random Forest Architecture: bootstrap aggregation (bagging) of independent decision trees [26].

3.1.3 Extra Trees (ET)

Extra Trees (Extremely Randomized Trees) is an ensemble learning algorithm introduced by Geurts et al. (2006) [37] that extends the Random Forest architecture by introducing an additional level of structural randomization during tree construction [11, 20, 21]. While a standard Random Forest searches exhaustively across a feature subset for the mathematically optimal split threshold at each node, the Extra Trees algorithm draws split thresholds completely at random from a uniform distribution spanning each candidate feature's localized value range [37]. Once these random thresholds are drawn, the algorithm evaluates them and selects the best performing candidate based on a predefined information gain or variance reduction criterion.

By heavily increasing randomness in the tree fabrication process, Extra Trees generates a highly diverse ensemble of heavily de-correlated constituent trees, which significantly reduces overall model variance and minimizes the risk of overfitting to local noise [20]. Furthermore, because it entirely bypasses the computationally expensive optimization search for exact mathematical split points, Extra Trees significantly diminishes training and computational execution times, rendering it exceptionally effective for processing large, high-dimensional, and noisy industrial datasets [11, 21, 37].

To illustrate this core divergence in split mechanics, Figure II.4 provides a side-by-side conceptual comparison between Random Forest and Extra Trees in the selection of localized split thresholds.

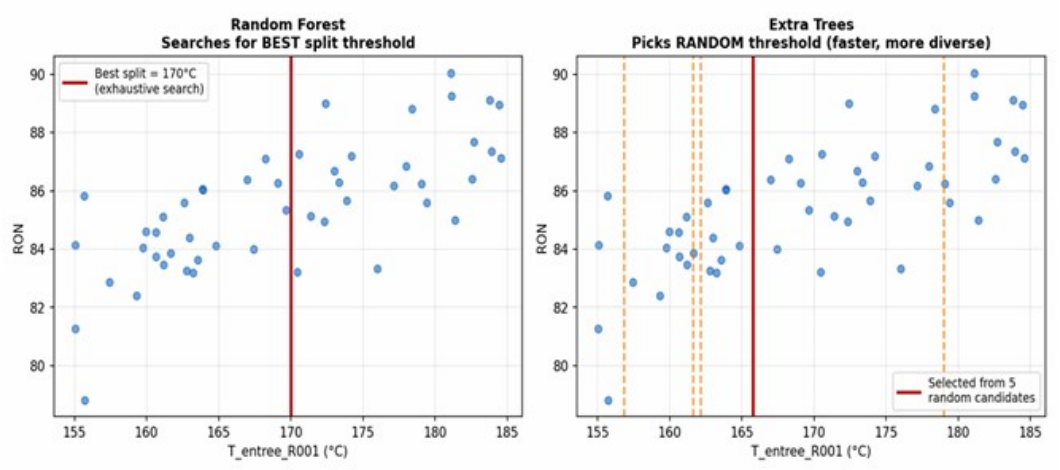


Figure II.4: Conceptual threshold selection strategy: Exhaustive optimization in Random Forest (left) versus randomized threshold sampling in Extra Trees (right) [37].

3.1.4 Gradient Boosting (GB)

Gradient Boosting [15] is an ensemble learning technique that builds a highly robust predictive model in a sequential manner by combining multiple weak learners, typically shallow decision trees. Unlike parallel ensemble methods such as Random Forest, Gradient Boosting constructs trees additively. At each iteration m, a new tree $h_m(x)$ is trained specifically to fit the negative gradient of the loss function, which directly corresponds to the localized residuals or prediction errors produced by the cumulative ensemble up to that point:

$$F_m(x) = F_{m-1}(x) + \eta \cdot h_m(x) \tag{II.2}$$

where $F_m(x)$ represents the updated model prediction at iteration m, $F_{m-1}(x)$ is the aggregated prediction of all previous trees, and $\eta \in (0, 1]$ denotes the learning rate (or shrinkage factor) controlling the contribution and step size of each newly incorporated tree [15].

By forcing each successive tree to approximate the negative gradient of the loss function, the model progressively drives down prediction errors through iterative optimization. This rigorous stage-wise optimization process enables Gradient Boosting to capture complex nonlinear relationships and intricate higher-order interactions between process variables without requiring manual feature engineering [20, 11].

To clarify this iterative process, Figure II.5 illustrates the fundamental mechanism of sequential residual correction. The initial stage begins with a simple baseline prediction (typically the mean of the target variable). Subsequent weak learners are then appended one at a time, with each new tree focusing its training capacity entirely on the remaining errors left unresolved by the existing ensemble. As more estimators are sequentially integrated, the additive prediction function becomes increasingly refined and better aligned with the true observed training data profile, leading to a steady, asymptotic minimization of the global empirical loss [15, 20].

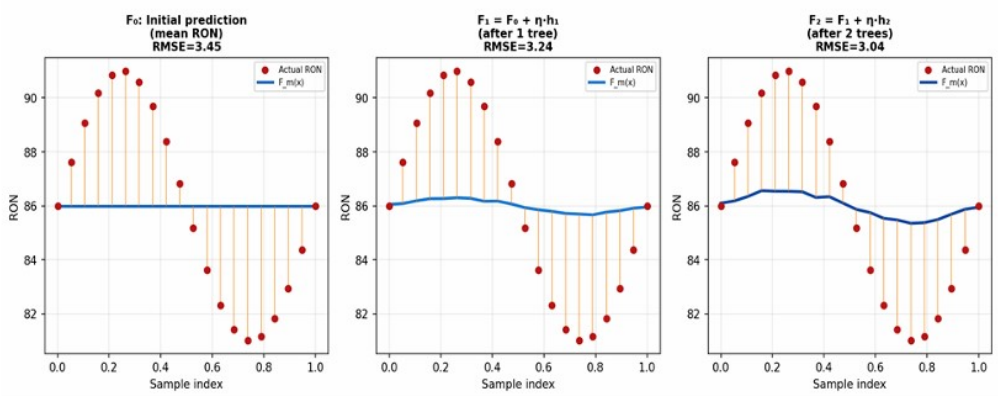


Figure II.5: Gradient Boosting: sequential residual correction across three boosting stages. Adapted from Friedman [15].

3.1.5 XGBoost (Extreme Gradient Boosting)

XGBoost (Extreme Gradient Boosting) is a highly optimized and scalable implementation of Gradient Boosting Trees that combines additive ensemble learning with advanced regularization techniques to achieve state-of-the-art predictive accuracy while maintaining excellent generalization performance [16, 15]. Building upon the classical sequential gradient boosting framework, XGBoost introduces several structural innovations, including efficient tree-pruning strategies, parallelized hardware computation, and a second-order optimization mechanism using a Taylor series expansion of the objective loss function [16].

To explicitly control model complexity and prevent overfitting, the regularized objective function minimized at each boosting iteration m is defined as follows:

$$\text{Obj}^{(m)} = \sum_{i=1}^{N} L\left(y_i, \hat{y}_i^{(m-1)} + h_m(x_i)\right) + \gamma T + \frac{\lambda}{2}\sum_{j=1}^{T} w_j^2 \tag{II.3}$$

where L represents a differentiable training loss function evaluating the true target y_i against the prediction, T denotes the total number of terminal leaf nodes in the newly added tree h_m, and w_j represents the continuous weight assigned to leaf j. The hyperparameters γ and λ act as user-defined regularization parameters penalizing tree complexity and leaf weight magnitudes, respectively [16]. By enforcing this structural penalty, the algorithm effectively curtails model variance and improves generalization behavior across noisy datasets [9, 11, 20, 21]. Furthermore, utilizing second-order gradient information (gradients and hessians) allows for much faster and more accurate optimization trajectories than traditional first-order gradient boosting approaches [16, 15].

To illustrate this structural capability, Figure II.6 outlines the explicit visual effect of L_1 and L_2 regularization within the XGBoost framework. As shown in the left panel, an unregularized model suffers heavily from high-frequency oscillations and overfits by tracking individual training data anomalies. Conversely, the regularized XGBoost model constructs a smooth, well-generalized decision boundary that captures the overarching industrial process trends. The right

panel displays the impact of L_2 weight shrinkage on individual tree leaf values, showcasing how shrinking the magnitudes of w_j suppresses over-sensitivity to noise and stabilizes overall process predictions [16, 20, 21]. Because of these robust mathematical safeguards, XGBoost has become a cornerstone machine learning tool for tabular refining data, delivering outstanding performance across numerous predictive applications [16, 11].

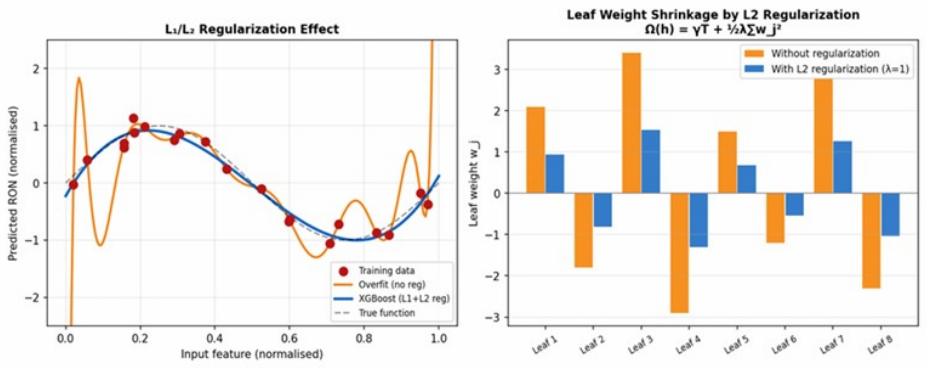


Figure II.6: Visual impact of structural regularization in XGBoost: Overfitting mitigation (left) and leaf weight shrinkage via L_2 penalties (right) [16].

3.1.6 LightGBM (Light Gradient Boosting Machine)

LightGBM (Light Gradient Boosting Machine) is an highly efficient, distributed implementation of Gradient Boosting Decision Trees (GBDT) specifically engineered to accelerate training speeds, drastically reduce memory consumption, and maintain premier predictive performance on large-scale industrial datasets [17, 15].

The primary architectural breakthrough of LightGBM lies in its node splitting methodology. Unlike conventional GBDT frameworks (such as standard Gradient Boosting or early versions of XGBoost) that employ a rigid level-wise (layer-by-layer) tree growth strategy, LightGBM adopts a leaf-wise (best-first) growth strategy. At each boosting step, the algorithm explicitly scans all available leaf nodes and exclusively expands the single leaf that yields the maximum mathematical reduction in the global objective loss function. The structural information gain associated with this localized splitting mechanism is calculated using first- and second-order gradient statistics as follows:

$$\text{Gain} = \frac{1}{2}\left[\frac{G_L^2}{H_L+\lambda} + \frac{G_R^2}{H_R+\lambda} - \frac{(G_L+G_R)^2}{H_L+H_R+\lambda}\right] - \gamma \tag{II.4}$$

where G_L and G_R represent the cumulative sums of the first-order gradients (the jacobians) of the left and right child nodes, while H_L and H_R represent the corresponding cumulative sums of the second-order gradients (the hessians). The parameters λ and γ act as user-defined regularization constraints penalizing leaf weights and structural split complexity, respectively [17]. This leaf-wise strategy achieves a much steeper reduction in empirical loss for an identical number of total leaves, resulting in significantly faster algorithmic convergence and enhanced predictive

accuracy [17, 20].

To visually clarify this paradigm shift in decision tree construction, Figure II.7 illustrates the fundamental operational difference between conventional level-wise growth and the leaf-wise strategy implemented by LightGBM.

As shown in the left panel, the level-wise approach expands all active nodes across an identical depth layer simultaneously, forcing a strictly balanced, symmetrical tree structure. While this conservative expansion controls tree depth and helps mitigate overfitting, it frequently spends computational resources expanding branches that contribute negligible information toward minimizing the global loss function.

Conversely, as demonstrated in the right panel, LightGBM's leaf-wise approach isolates the leaf node exhibiting the maximum information gain and expands only that targeted branch. Consequently, the decision tree grows asymmetrically and deeply, concentrating its model capacity on the most descriptive, highly complex regions of the continuous process feature space. This focused branch development leads to far lower training errors, optimal utilization of system hardware, and faster convergence metrics compared with traditional boosting methods [17, 20, 21].

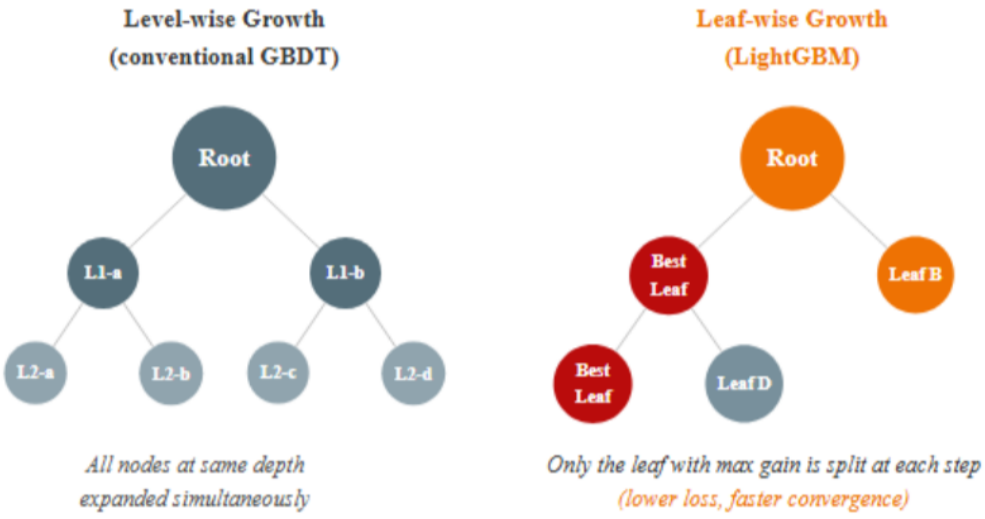


Figure II.7: LightGBM tree growth mechanics: Conventional balanced level-wise growth (left) versus asymmetric, optimized leaf-wise growth (right) [17].

3.1.7 CatBoost (Categorical Boosting)

CatBoost (Categorical Boosting) is an advanced gradient boosting framework specifically engineered to overcome the phenomena of prediction shift and target leakage, which frequently degrade accuracy and computational efficiency in tabular data applications [18].

The defining innovation of CatBoost is its **ordered boosting** mechanism. In standard gradient boosting implementations, the gradients used to train a new tree are evaluated using the exact same samples that were already utilized to construct the existing ensemble, introducing an implicit data dependency that leads to out-of-sample prediction shift. To mitigate this structural bias, CatBoost processes data sequentially according to predefined random permutations.

Under this paradigm, the residual for any given training sample at boosting iteration t is calculated using a model ensemble that has been exposed exclusively to preceding samples within that permutation:

$$r_i^{(t)} = y_i - F^{(t)}(x_i), \quad \text{where } F^{(t)} \text{ is trained on samples } \{1, \ldots, i-1\} \text{ only} \tag{II.5}$$

where $r_i^{(t)}$ represents the localized target residual for the i-th observation, y_i is its true target value, and $F^{(t)}$ denotes the aggregated model function learned strictly from historical observations indexed prior to i [18]. By completely avoiding information leakage during training, ordered boosting yields highly reliable gradient estimates, stabilizing model optimization trajectories and significantly minimizing overfitting on limited or noisy refining datasets [9, 18, 21].

Beyond ordered boosting, CatBoost achieves state-of-the-art execution speeds by employing **symmetric (oblivious) decision trees** as base predictors [18]. Unlike conventional asymmetric decision trees, oblivious trees apply an identical, uniform split criterion (same feature and same threshold) across all active nodes at any given depth level.

To visually contrast these core algorithmic innovations against classical frameworks, Figure II.8 outlines the dual mechanics of ordered boosting data flows and symmetric splitting structures.

As detailed in the left panel, conventional boosting methods allow information from the entire training partition to implicitly influence localized residual calculations, which introduces estimation bias. Conversely, CatBoost's ordered boosting strategy enforces an asymmetric historical boundary based on random sequences, ensuring that no downstream information leaks into upstream residual calculations [18, 20].

The right panel showcases the resulting symmetric tree structure. Because every node at a given depth layer shares the exact same split condition, the tree grows in a perfectly balanced manner. This highly regular topology simplifies the underlying tree traversal down to bitwise operations, which drastically accelerates model inference speeds, decreases hardware memory footprints, and provides a powerful structural safeguard against excessive tree complexity and overfitting [11, 20, 21].

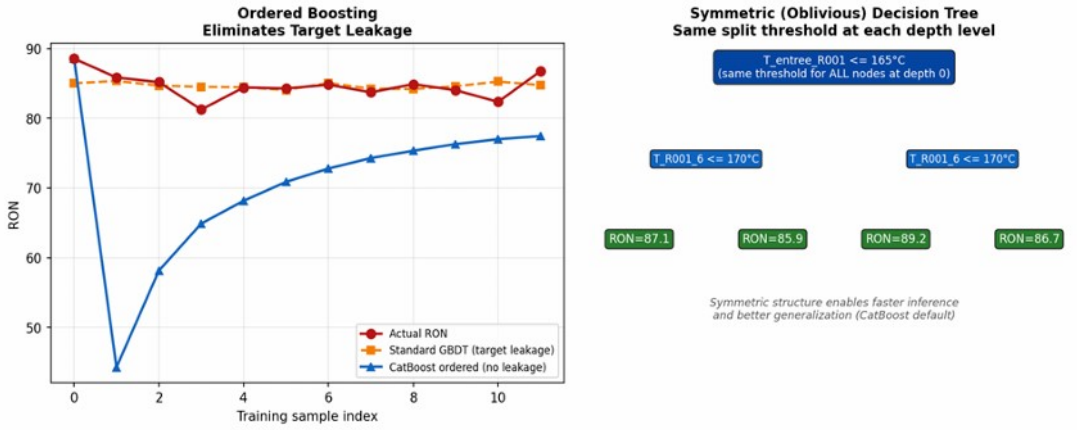


Figure II.8: CatBoost architectural innovations: Target leakage mitigation via ordered boosting (left) versus computationally efficient symmetric (oblivious) tree topology (right) [18].

3.1.8 Deep Neural Networks (DNNs)

A Deep Neural Network (DNN) is a fully connected feedforward artificial neural network composed of an input layer, an output layer, and multiple sequential hidden layers [19, 30]. DNNs extend traditional shallow neural architectures by expanding the depth of the network, which enables the model to automatically learn hierarchical, highly nonlinear abstractions and complex feature relationships directly from raw data without manual engineering [10, 19, 21]. Mathematically, each hidden layer l transforms the representation from the preceding layer by performing the following vector-matrix computation:

$$h^{(l)} = \sigma\left(W^{(l)}h^{(l-1)} + b^{(l)}\right) \tag{II.6}$$

where $W^{(l)}$ and $b^{(l)}$ denote the weight matrix and bias vector of layer l, respectively, $h^{(l-1)}$ is the activation output vector from the previous layer, and σ represents a nonlinear element-wise activation function. To ensure robust gradient flow throughout deep topologies, the Rectified Linear Unit (ReLU) activation function is widely adopted:

$$\mathrm{ReLU}(z) = \max(0, z) \tag{II.7}$$

By replacing all negative values with zero and maintaining a constant gradient of one for all positive activations, ReLU effectively mitigates the vanishing gradient problem during backpropagation, accelerating training convergence and improving efficiency in deep architectures [10, 19, 30].

To illustrate this structural framework, Figure II.9 outlines a conceptual fully connected Deep Neural Network layout tailored for processing tabular refining features.

As shown in the schematic, the blue nodes on the left represent the input layer, which receives the multi-variable process feature vector. The green nodes correspond to three fully connected (dense) hidden layers, where each individual neuron is structurally coupled to every single neu-

ron in the subsequent layer. These dense connections enable the network to capture complex, higher-order interactions among all operational parameters simultaneously, progressively constructing higher-level representations of the data as the information flows forward. Finally, the red nodes on the right represent the output layer, which maps the final dense representations into the numerical process prediction (such as real-time RON estimation). This multilayered topology enables the network to model highly complex, continuous nonlinear surfaces that remain fundamentally uncapturable by conventional linear or shallow models [19, 21, 30].

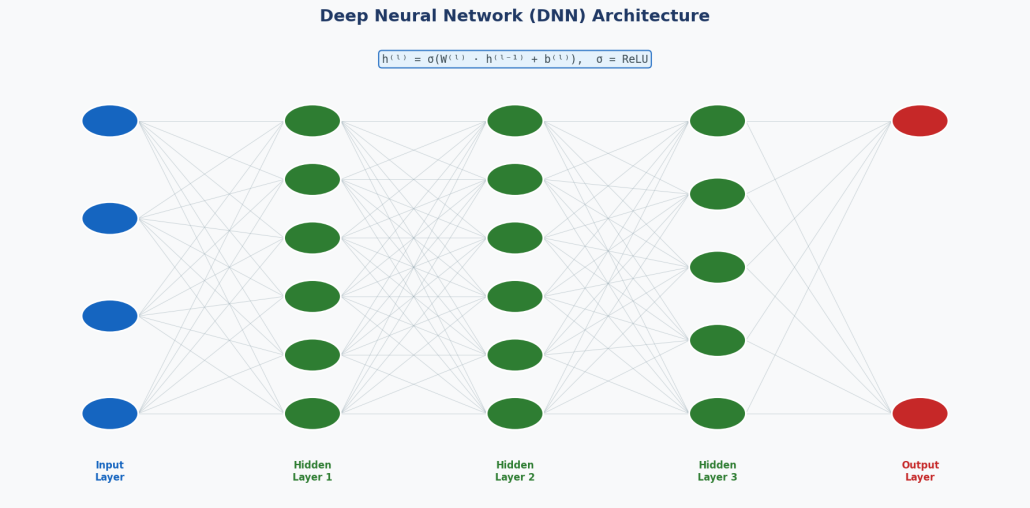


Figure II.9: Deep Neural Network (DNN) architecture: input layer (blue), multiple hidden layers with ReLU activations (green), and output layer (red). Adapted from Goodfellow et al. [19] and Schmidhuber [30].

3.1.9 Convolutional Neural Networks (CNNs)

Convolutional Neural Networks (CNNs) represent a specialized class of deep learning architectures designed to automatically and adaptively learn hierarchical spatial or temporal feature representations through localized convolutional operations [19, 10]. While originally developed to revolutionize computer vision and image recognition, CNNs have been successfully adapted for advanced pattern recognition, complex signal processing, and multivariate industrial time-series analysis tasks [10, 19, 30].

The foundational mathematical mechanism of a CNN relies on the convolution operator, where a localized kernel (or filter) slides across the input space to produce a feature map. For a 2D discrete input space, the computation at layer l is formalized as:

$$z_{i,j}^{(l)} = \sigma\left(\sum_{m=1}^{M}\sum_{n=1}^{N} w_{m,n}^{(l)} \cdot x_{i+m,\,j+n}^{(l-1)} + b^{(l)}\right) \tag{II.8}$$

where $w_{m,n}^{(l)}$ represents the weights of an $M \times N$ convolutional kernel, $x^{(l-1)}$ is the input feature map (or raw multi-variable parameter grid) from the preceding layer, $b^{(l)}$ is the localized bias term, and σ denotes the element-wise Rectified Linear Unit (ReLU) activation function.

By relying heavily on the dual principles of local connectivity and parameter sharing (where the same kernel weights are applied across the entire input field), CNNs dramatically suppress overall model variance and reduce the total number of trainable parameters compared to fully connected deep networks [19, 31].

To clarify the explicit spatial extraction workflow utilized within this framework, Figure II.10 details the specific CNN topology engineered for this study.

As illustrated in the structural diagram, the data flow begins with an input layer (blue block) that feeds the structured process variable array forward. This is followed by a stacked sequence of convolutional layers (purple blocks) that progressively scale up in filter density, deploying 32, 64, and 128 filters respectively. Each successive convolutional layer extracts increasingly abstract, high-order geometric or temporal feature combinations from the refining unit dataset.

Interspersed after the second convolutional stage, a max-pooling layer (green block) with a stride of 2 is integrated to systematically downsample the dimensions of the active feature maps while strictly preserving the maximum local activation values. This pooling operation effectively curbs computational execution times and reinforces the translation invariance and structural robustness of the learned features against transient variations or high-frequency noise in the refinery sensor streams [19, 31]. Finally, these compressed feature extractions are flattened and channeled into fully connected layers (orange blocks) to map the abstract representations directly to the final target estimation.

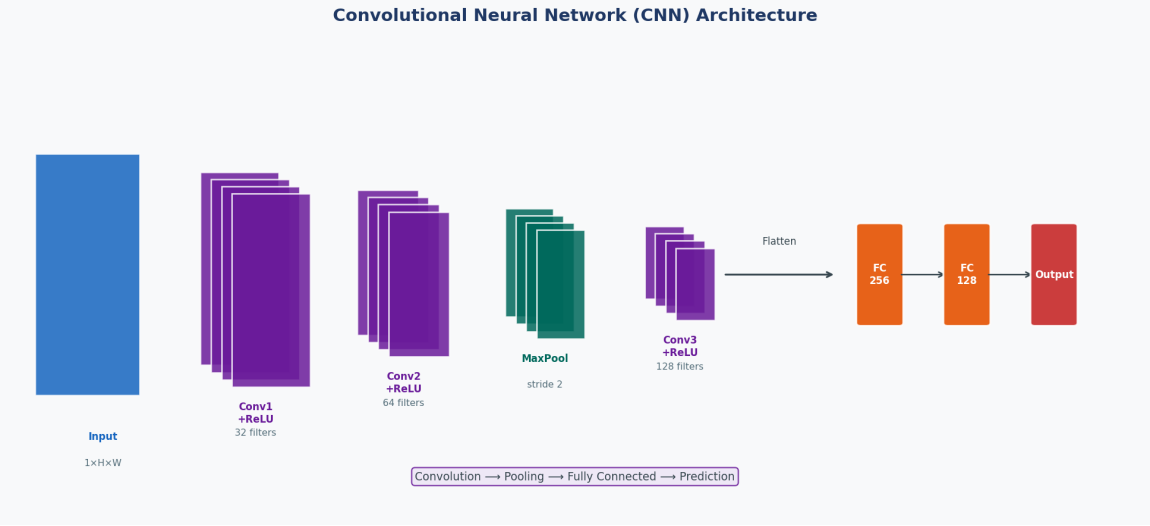


Figure II.10: Convolutional Neural Network (CNN) architecture: input layer (blue), stacked convolutional layers (purple), max-pooling downsampling layers (green), and dense fully connected layers (orange). Adapted from LeCun et al. [31] and Goodfellow et al. [19].

3.1.10 Multi-Layer Perceptron (MLP)

A **Multi-Layer Perceptron (MLP)** is a foundational class of feedforward artificial neural networks consisting of an input layer, one or more dense hidden layers, and an output layer, where each individual neuron is fully connected to all neurons in the subsequent layer [14, 19, 21].

Backed by the universal approximation theorem, MLPs serve as powerful mathematical estimators capable of modeling complex, highly nonlinear continuous mappings directly from structured datasets without structural a priori assumptions [19, 21, 30].

For a network configured with L hidden layers, the deterministic forward propagation and transformation of information at any given hidden layer l is formalized as:

$$h^{(l)} = \sigma\left(W^{(l)}h^{(l-1)} + b^{(l)}\right) \tag{II.9}$$

where $W^{(l)}$ and $b^{(l)}$ denote the weight matrix and bias vector associated with layer l, respectively, $h^{(l-1)}$ represents the continuous activation vector exiting the previous layer, and σ represents the localized element-wise activation function. In modern deep architectures, the Rectified Linear Unit (ReLU) has become the standard activation choice due to its high computational simplicity and its inherent ability to maintain non-saturating gradients, which effectively alleviates the vanishing gradient problem during backpropagation routines [10, 19, 30].

To update the system parameters (W and b), empirical loss gradients are computed via the Backpropagation algorithm [32] and mapped to an advanced first-order optimization routine, such as the Adam (Adaptive Moment Estimation) optimizer [34], which dynamically computes individual adaptive learning rates using estimates of the first and second moments of the gradients.

To visually demonstrate this fully connected mapping scheme, Figure II.11 illustrates the explicit network topology deployed within this framework for refinery process state evaluation.

The data flow initiates at the input layer (blue nodes), which is populated by the multivariable operational feature space. These inputs are subsequently fed forward through three stacked hidden layers (green nodes) containing 128, 64, and 32 neurons, respectively. Because every single node retains a physical weight connection to all nodes in its adjacent layer, the topology maps comprehensive feature interactions simultaneously, refining the signal down to a singular terminal output node (red node) that yields the real-time Research Octane Number (RON) prediction [19, 21, 30].

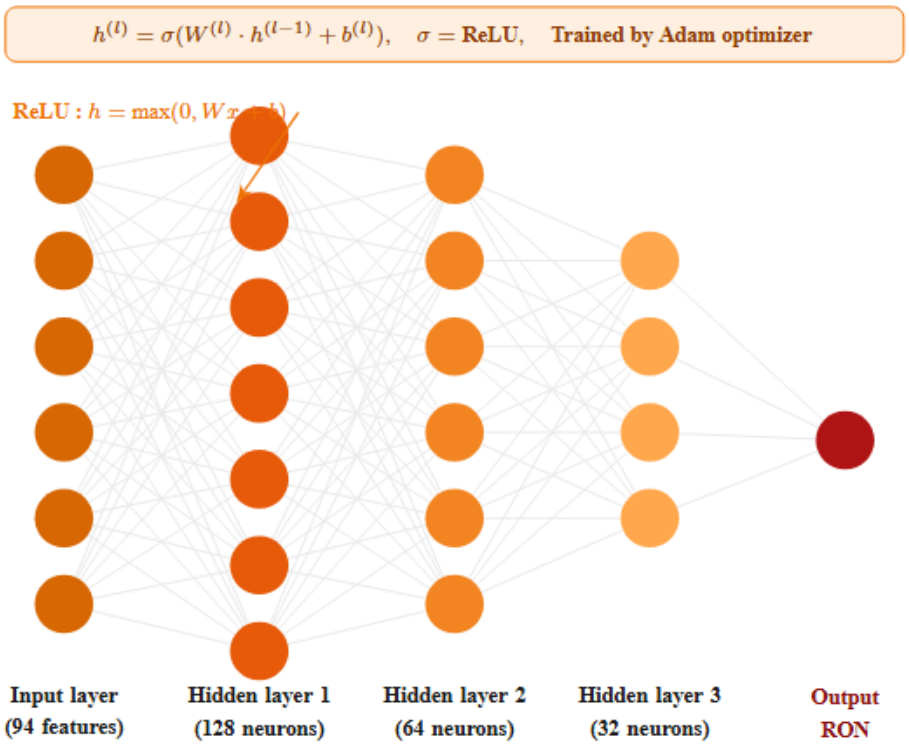


Figure II.11: Multi-Layer Perceptron (MLP) topology: input features, three dense hidden layers (128-64-32) with ReLU activation functions, and a single continuous RON output node. Adapted from Goodfellow et al. [19] and Rumelhart et al. [32].

3.1.11 Recurrent Neural Networks (RNNs)

Recurrent Neural Networks (RNNs) constitute a specialized class of deep neural architectures explicitly designed to process sequential, ordered, and temporal data by maintaining an internal recurrent hidden state that dynamically evolves over time [19, 21, 30]. Given a multi-variable operational time-series input sequence denoted as $x^{(1)}, x^{(2)}, \ldots, x^{(T)}$, the network processes each sequence element sequentially. At any specific time step t, the localized hidden state $h^{(t)}$ acts as an internal memory buffer and is computed recursively by incorporating the current observation with historical context:

$$h^{(t)} = \tanh\left(W_h h^{(t-1)} + W_x x^{(t)} + b\right) \tag{II.10}$$

The continuous prediction or sequence output vector $y^{(t)}$ at that identical time step is subsequently mapped via the following linear transformation:

$$y^{(t)} = W_y h^{(t)} + b_y \tag{II.11}$$

where W_h denotes the recurrent weight matrix mapping historical context, W_x represents the input weight matrix, W_y is the terminal output weight matrix, and b, b_y correspond to the lo-

calized bias vectors [19, 21, 30]. Through this recursive cyclical feedback loop, RNNs capture long-range contextual information and capture temporal trends distributed across consecutive time frames, rendering them highly effective for industrial sequence modeling, time-series forecasting, and transient signal analysis [19, 33, 30].

To conceptually clarify this recursive feedback mechanism and explain how gradients are tracked during training, Figure II.12 illustrates the unrolled computational representation of an RNN across five consecutive time steps.

As shown in the graph, the unrolling procedure translates the cyclical recurrent loop into a directed, sequential acyclic structure. Each individual block corresponds to an active hidden state layer $h^{(t)}$ at a specific epoch, which simultaneously ingests the localized input vector $x^{(t)}$ (represented by the blue blocks) and the structural information vector propagated forward from the immediate historical hidden state $h^{(t-1)}$ (represented by the green blocks).

This combined vector undergoes the transformation detailed in Equation (II.10) to generate the current hidden representation, which produces the real-time continuous estimation $\hat{y}^{(t)}$ (represented by the red blocks) while concurrently pushing the updated context vector forward to the succeeding time step $t+1$. This unrolled view highlights the parameter-sharing constraint of recurrent architectures: the exact same weight matrices (W_h, W_x, W_y) are shared uniformly at every single time step throughout the temporal horizon. This structural constraint dramatically suppresses model complexity, minimizes parameters, and forces the network to learn translation-invariant temporal features across the operational timeline [19, 30].

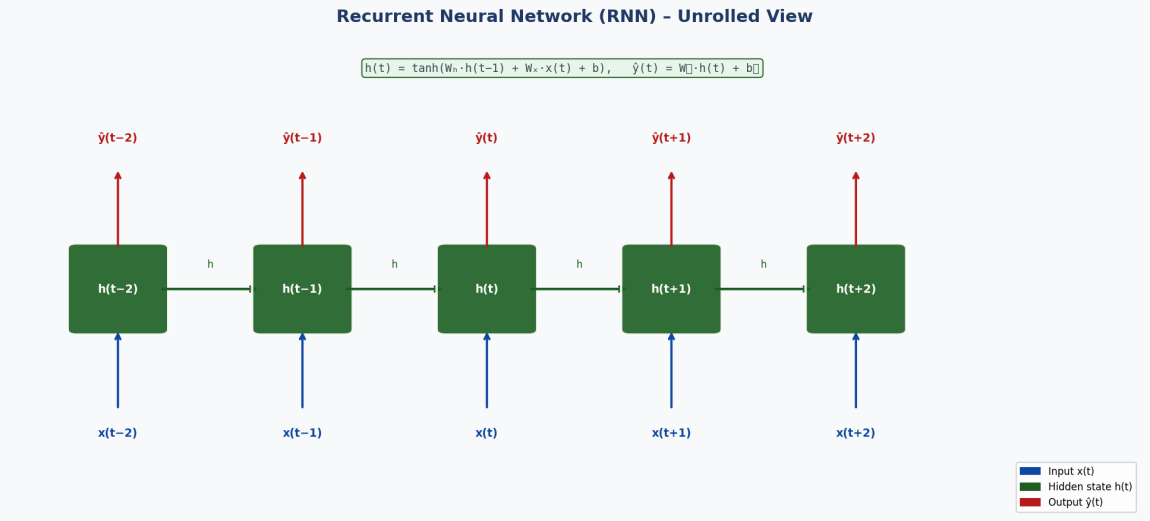


Figure II.12: Recurrent Neural Network (RNN) unrolled over five consecutive time steps: temporal sequence inputs (blue), hidden state context tracking (green), and time-step outputs (red). Adapted from Goodfellow et al. [19] and Schmidhuber [30].

3.1.12 Long Short-Term Memory Network (LSTM)

The Long Short-Term Memory (LSTM) network is an advanced variant of recurrent neural architectures originally introduced by Hochreiter and Schmidhuber [33]. It was specifically engineered to overcome the severe mathematical limitations of the vanishing and exploding gradient

problems that restrict conventional RNNs from learning long-range temporal dependencies. By incorporating an explicit internal memory cell along with a specialized set of regulatory gating mechanisms, LSTMs selectively store, update, and retrieve information across extended time horizons without losing vital historical context [19, 30, 33].

The architecture maintains two distinct continuous internal states: a hidden state vector $h^{(t)}$, which represents the short-term working output of the network, and a cell state vector $c^{(t)}$, which serves as a highly protected, long-term memory highway running through the sequence [19, 21, 30]. This dual-state mechanism allows LSTMs to preserve stability and mitigate gradient degradation during Backpropagation Through Time (BPTT) operations on noisy industrial time-series profiles [10, 19, 32, 33].

At any given time step t, an individual LSTM cell ingests the current feature input vector $x^{(t)}$ along with the historical states $h^{(t-1)}$ and $c^{(t-1)}$. The interaction within the cell structure is governed by four synchronized component operations:

$$f^{(t)} = \sigma\left(W_f \cdot \left[h^{(t-1)}, x^{(t)}\right] + b_f\right) \tag{II.12}$$

$$i^{(t)} = \sigma\left(W_i \cdot \left[h^{(t-1)}, x^{(t)}\right] + b_i\right) \tag{II.13}$$

$$\tilde{c}^{(t)} = \tanh\left(W_c \cdot \left[h^{(t-1)}, x^{(t)}\right] + b_c\right) \tag{II.14}$$

$$c^{(t)} = f^{(t)} \odot c^{(t-1)} + i^{(t)} \odot \tilde{c}^{(t)} \tag{II.15}$$

$$o^{(t)} = \sigma\left(W_o \cdot \left[h^{(t-1)}, x^{(t)}\right] + b_o\right) \tag{II.16}$$

$$h^{(t)} = o^{(t)} \odot \tanh\left(c^{(t)}\right) \tag{II.17}$$

where $f^{(t)}$, $i^{(t)}$, and $o^{(t)}$ represent the activation vectors of the forget, input, and output gates, respectively, while $\tilde{c}^{(t)}$ denotes the newly generated candidate cell state [19, 33]. The operator $\odot$ represents the element-wise Hadamard product.

In this structural formulation, the forget gate determines which portions of historical memory are discarded or kept, the input gate modulates how much new descriptive data alters the internal memory bank, and the output gate isolates which pieces of the updated memory are actively exposed through the hidden state output layer [19, 30, 33].

To demonstrate this sequence-tracking mechanism visually, Figure II.13 illustrates the precise mathematical data flow and internal structure of a single LSTM computational unit.

The upper linear horizontal pathway maps the cell state $c^{(t)}$, acting as the primary long-range memory channel traversing uninterrupted through the unit. The forget gate (highlighted in red) selectively removes irrelevant historical parameters from the previous memory buffer via a multiplicative operation. Concurrently, the input gate (green) and candidate cell block (purple) collaborate via a joint addition to insert fresh operational details.

The updated cell state $c^{(t)}$ is then transmitted directly to the succeeding time step $t+1$. Simultaneously, the output gate (blue) filters the current memory matrix, scaling it to generate the active hidden state vector $h^{(t)}$, which serves as the local output layer while being pushed forward as historical context for the next cycle. This gating framework ensures the stable propagation

of gradient values through deep sequences, preserving long-term insights while dynamically adjusting to local changes [19, 30, 33].

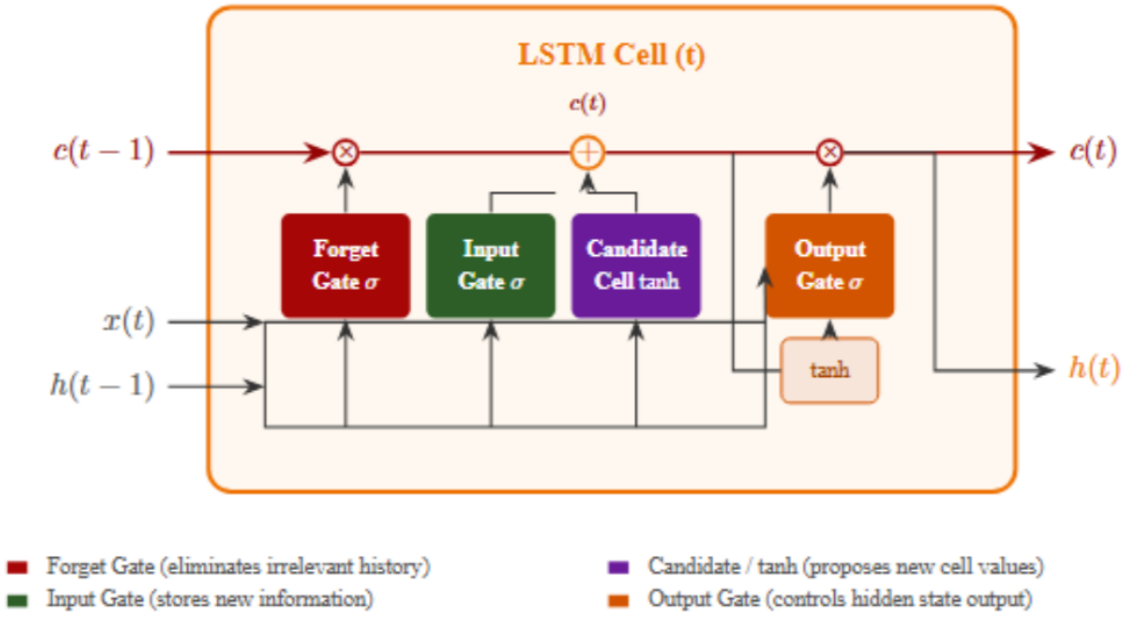


Figure II.13: LSTM cell architecture: forget gate (red), input gate (green), candidate cell tanh (purple), and output gate (blue). The cell state $c^{(t)}$ serves as a long-range memory highway; the hidden state $h^{(t)}$ is the output at each time step. Adapted from Hochreiter & Schmidhuber [33].

3.2 Artificial Intelligence for Optimization

Artificial Intelligence has significantly enhanced optimization by providing powerful techniques capable of solving complex problems that are difficult to address using traditional mathematical approaches. Unlike conventional optimization methods, AI-based algorithms can efficiently explore large and nonlinear search spaces [12, 39].

3.2.1 Optimization Problem Formulation

Optimization is the process of systematically identifying the best possible solution within a feasible set defined by constraints and variable bounds [35]. Formally:

$$\begin{aligned} \text{maximize} \quad & f(\mathbf{x}) \\ \text{subject to:} \quad & \mathbf{x}_L \leq \mathbf{x} \leq \mathbf{x}_U \end{aligned}$$

where $\mathbf{x}$ is the vector of decision variables, $f(\mathbf{x})$ is the objective function, and $\mathbf{x}_L$, $\mathbf{x}_U$ are the lower and upper bounds.

3.2.2 Classical Optimization Techniques

Before the emergence of AI-based methods, industrial optimization problems were commonly addressed using classical mathematical techniques.

- **Gradient-Based Optimization:** Methods such as Steepest Descent and Sequential Quadratic Programming (SQP) use derivative information to guide the search. Their performance deteriorates when the objective function is highly nonlinear or non-differentiable.
- **Grid Search:** Evaluates the objective function over a predefined set of parameter combinations. Computational cost increases rapidly with the number of decision variables.
- **Response Surface Methodology (RSM):** Approximates the relationship between process variables and responses using polynomial models.
- **Nelder–Mead Simplex Method:** A derivative-free technique that iteratively updates a simplex. Suitable for low-dimensional problems but may converge to local optima.

3.2.3 Overview of Optimization Algorithms Benchmarked

To systematically isolate the most robust optimization paradigm for the refinery unit's predictive framework, a comprehensive benchmarking suite consisting of eight distinct artificial intelligence and metaheuristic algorithms is deployed. These techniques span across evolutionary computation, swarm intelligence, stochastic thermodynamic heuristics, and sequential Bayesian architectures, providing a diverse exploratory range over the parameter search space.

Table II.1 provides a structured architectural overview of the benchmarked candidates, classifying each algorithm by its foundational family, operational search strategy, target objective capability, and primary hyperparameter configuration.

Algorithm	Family	Search Strategy	Objective Type	Key Hyperparameters
NSGA-II	Evolutionary	Population-based	Multi-objective	Pop. size, n_gen, CR, F
NSGA-III	Evolutionary	Population-based	Multi-objective	Pop. size, ref. points
SPEA2	Evolutionary	Archive-based	Multi-objective	Archive size, k-NN
PSO	Swarm	Velocity update	Single-objective	w, c_1, c_2, swarm size
Diff. Evol.	Evolutionary	Population-based	Single-objective	F, CR, pop. size
Sim. Anneal.	Metaheuristic	Single solution	Single-objective	T_0, α, n_iter
Optuna (TPE)	Bayesian	Sequential model	Single-objective	n_trials, γ
BayesOpt (GP)	Bayesian	Sequential model	Single-objective	n_iter, init_points, κ

Table II.1: Overview of the eight optimization algorithms benchmarked in this study, detailing their structural classifications, operational search strategies, and core hyperparameter parameters.

As summarized in the taxonomy matrix, population-based and archive-based evolutionary frameworks (NSGA-II, NSGA-III, and SPEA2) are primarily maintained to address conflicting multi-objective criteria by building explicit Pareto-optimal fronts. Conversely, methods utilizing dynamic velocity trajectories (PSO), localized mutation vectors (Differential Evolution), or temperature-controlled stochastic transitions (Simulated Annealing) are configured to tackle highly non-convex, single-objective error minimization surfaces. Finally, sequential model-based optimization techniques relying on Tree-structured Parzen Estimators (Optuna TPE) and Gaussian Process surrogates (BayesOpt GP) are integrated to evaluate computationally expensive objective trajectories with maximum sample efficiency.

NSGA-II — Non-dominated Sorting Genetic Algorithm II The Non-dominated Sorting Genetic Algorithm II (NSGA-II) [47] is a population-based evolutionary framework applied here in single-objective refinement mode to maximize the unit's target Research Octane Number (RON). The algorithm evolves an initial group of N candidate solutions over successive generations using stochastic genetic operators and rigid selection criteria.

The exploration of the parameter search space relies primarily on **Simulated Binary Crossover (SBX)**. Given two parent vectors x^1 and x^2, the generated offspring individuals c^1 and c^2 are computed element-wise as:

$$c_i^1 = 0.5\left[(1+\beta)x_i^1 + (1-\beta)x_i^2\right] \tag{II.18}$$

$$c_i^2 = 0.5\left[(1-\beta)x_i^1 + (1+\beta)x_i^2\right] \tag{II.19}$$

where $\beta \geq 0$ is a distribution parameter controlling child-to-parent spatial proximity. Fine-grained local exploitation is subsequently handled via **Polynomial Mutation**, expressed as:

$$x_i' = x_i + \delta_q \cdot (x_{U,i} - x_{L,i}) \tag{II.20}$$

where $x_{U,i}$ and $x_{L,i}$ define the operational upper and lower bounds of the feature space, and δ_q is an entry drawn from a polynomial probability distribution [36]. Preservation of top-tier parameters is secured via **Elitism**, where parent and offspring populations are merged into a unified set of size $2N$ before sorting, ensuring that the highest-quality solutions are preserved from one generation to the next [47, 12].

Figure II.14 outlines this complete structural workflow. The cycle initializes by randomly instantiating N candidate parameters within operational bounds. After calculating local objective fitness inputs, the set undergoes non-dominated sorting and crowding distance assignments to enforce population diversity. Binary tournament selection then maps optimal parents to Equations (II.18)–(II.20), before trimming the merged $2N$ matrix back down to N elitist members. This evolutionary routine repeats until an asymptotic convergence threshold is satisfied.

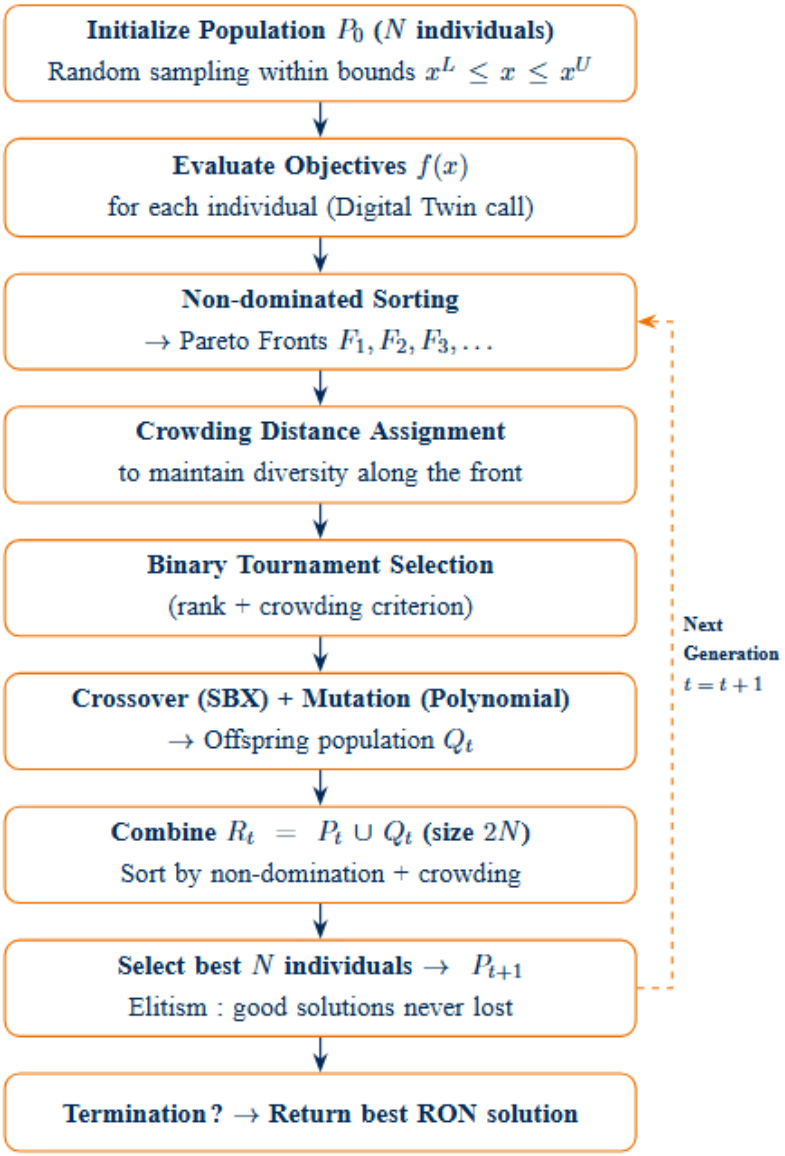


Figure II.14: NSGA-II algorithm flowchart configured for single-objective optimization (maximizing RON). Adapted from Deb et al. [47] and Deb [12].

NSGA-III — Non-dominated Sorting Genetic Algorithm III NSGA-III [48] heavily modifies the structural selection framework of the traditional NSGA-II architecture by entirely replacing the localized crowding distance metric with an advanced, reference-point-based selection grid. To systematically preserve population diversity across dense optimization layers, structured reference points z are distributed uniformly across a hyper-spherical plane using the Das-Dennis binomial formulation:

$$H = \binom{M + p - 1}{p} \tag{II.21}$$

where H represents the cumulative number of reference coordinates generated, M defines the total number of objective functions, and p denotes the total number of divisions executed along each feature axis [48]. Candidate populations are explicitly mapped to these reference vec-

tors via perpendicular distance normalization, forcing the search trajectory to maintain an even spread across non-convex spaces.

SPEA2 — Strength Pareto Evolutionary Algorithm 2 The Strength Pareto Evolutionary Algorithm 2 (SPEA2) [44] relies on an isolated, external archive environment to store and continuously update the finest solutions discovered during runtime. When configured in single-objective operational mode, this archive acts as a strict repository for tracking maximum-RON candidates. The mathematical fitness $F(i)$ assigned to an individual member i within the current population matrix is formalized as:

$$F(i) = R(i) + D(i), \quad \text{where } R(i) = \sum_{j \in \mathcal{A}_t : j \succ i} S(j), \quad \text{and } D(i) = \frac{1}{\sigma_i^k + 2} \tag{II.22}$$

where $R(i)$ is the raw fitness component, the notation $j \succ i$ denotes that solution j mathematically dominates solution i, and $S(j)$ represents the overall strength value of candidate j (the total number of population individuals it directly dominates). To prevent stagnation, the density estimation parameter $D(i)$ calculates the spatial distance σ_i^k separating the target individual from its k-th nearest neighbor in the feature space, maximizing archive diversity [44].

Figure II.15 provides a visual representation of this optimization loop. The blue entries map candidate solutions scattered across the exploratory landscape, while the red entries track active members locked within the external archive. While a broad array of transient parameters are generated across successive generations, only the most robust profiles are permitted into the archive based on the joint quality-density evaluation of Equation (II.22). The dashed red line captures the progressive advancement of the archived frontier, focusing search energy into high-performing regions while preventing premature convergence.

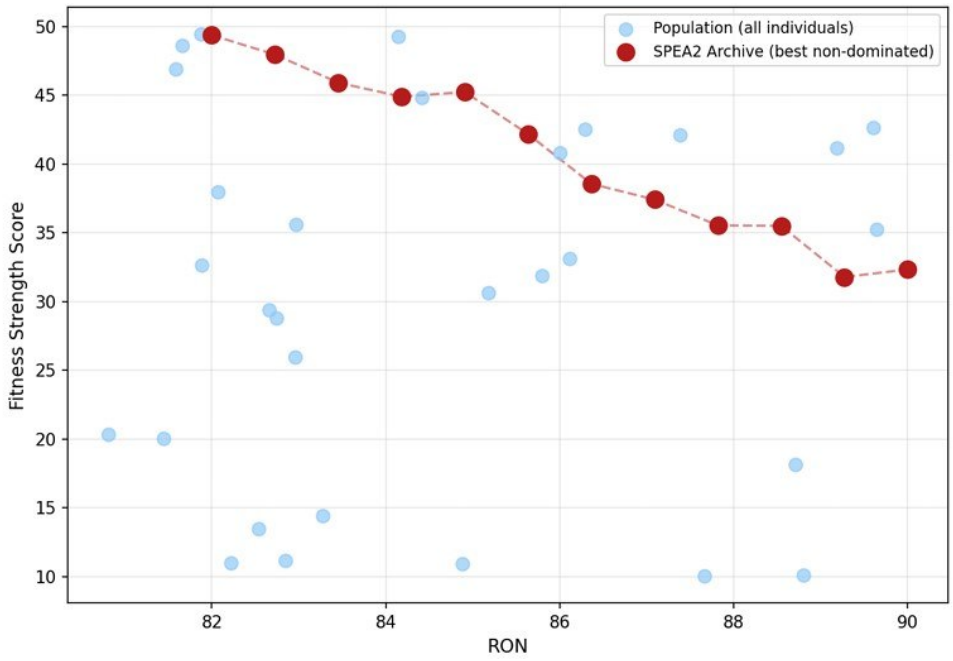


Figure II.15: SPEA2 archive mechanism in single-objective tracking mode, highlighting the selection frontier of maximum-RON solutions. Adapted from Zitzler et al. [44].

Particle Swarm Optimization (PSO) Particle Swarm Optimization (PSO) [38] models a collaborative, derivative-free searching methodology inspired by the decentralized flocking behaviors of animal swarms. Each candidate solution is represented as a single particle i retaining a dynamic velocity vector v_i and a spatial coordinate vector x_i. At every discrete time step t, these vectors are updated according to the following kinematic rules:

$$v_i(t+1) = w \cdot v_i(t) + c_1 r_1 \big(\text{pBest}_i - x_i(t)\big) + c_2 r_2 \big(\text{gBest} - x_i(t)\big) \tag{II.23}$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \tag{II.24}$$

where $w \in (0, 1]$ represents the inertia weight parameter dampening rapid direction changes, c_1 and c_2 denote the cognitive and social acceleration coefficients respectively, and r_1, r_2 are independent stochastic vectors sampled uniformly from the interval $[0, 1]$ [38]. The vector pBest_i isolates the historically optimal coordinate position discovered exclusively by particle i, whereas gBest preserves the global optimum coordinate identified by the entire collective swarm up to iteration t.

To visually interpret this momentum-driven search framework, Figure II.16 overlays a five-particle active tracking scenario atop a contour topology mapping the process loss landscape. The directional arrows denote the updating velocity trajectories as individual particles balance their internal historical memory (pBest) against the global attraction pulls of the swarm center (gBest, marked by the red star). This collaborative search loop enables efficient navigation through rough, non-linear valleys toward the absolute global minimum.

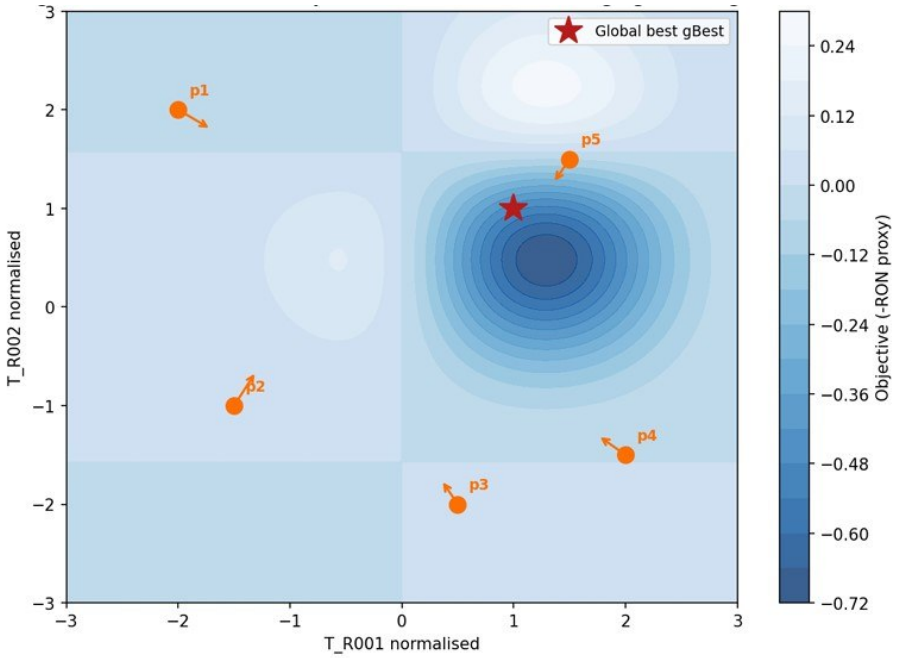


Figure II.16: PSO particle mechanics updating spatial positions and tracking the global swarm center gBest across an operational landscape. Adapted from Kennedy and Eberhart [38].

Differential Evolution (DE) Differential Evolution (DE) [39] is a robust stochastic parallel parameter search algorithm that relies on vector differences to drive its mutation and crossover cycles. At each iteration, the algorithm constructs an intermediate mutant vector v by combining three mutually isolated, randomly sampled population vectors $(x_{r_1}, x_{r_2}, x_{r_3})$. A binomial crossover is then applied to build a terminal trial vector u_j:

$$v = x_{r_1} + F \cdot (x_{r_2} - x_{r_3}) \tag{II.25}$$

$$u_j = \begin{cases} v_j & \text{if rand}_j < \text{CR} \quad \text{or} \quad j = j_{\text{rand}} \\ x_{t,j} & \text{otherwise} \end{cases} \tag{II.26}$$

$$x_t \leftarrow u \quad \text{if } f(u) \geq f(x_t) \tag{II.27}$$

where $F \in [0.5, 1.0]$ represents the mutation scaling factor controlling differential step lengths, $\text{CR} \in [0, 1]$ denotes the user-defined crossover probability, and x_t defines the original target vector. The final step implements a greedy selection mechanism where the trial vector replaces the target vector only if it registers an improved objective value [39].

Figure II.17 contrasts these twin operational pillars. The left diagram displays the spatial mutation mechanism, showcasing how scaling the difference vector $(x_{r_2} - x_{r_3})$ and appending it to the base member x_{r_1} generates a dynamic exploratory trajectory. The right diagram isolates the binomial crossover stage, where elements are selectively harvested from either the mutant vector or the parent target vector based on the threshold condition of Equation (II.26).

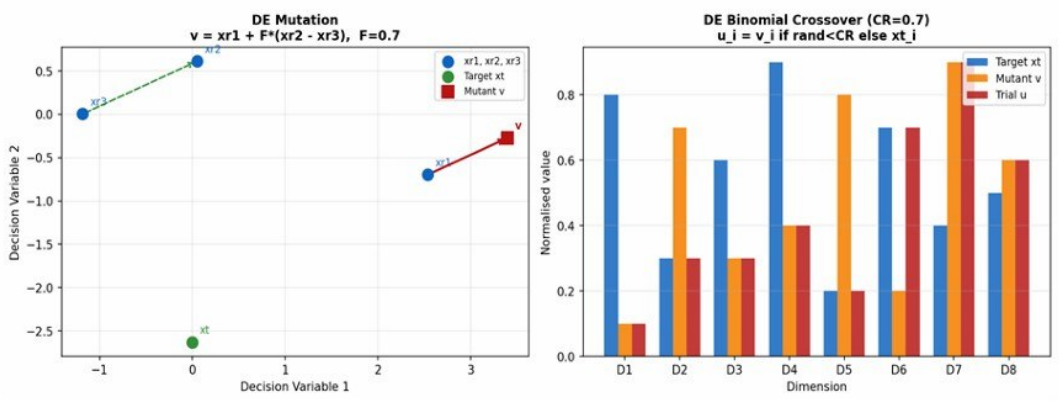


Figure II.17: Differential Evolution structural phases: Vector difference mutation geometry (left) and binomial trial vector crossover mapping (right). Adapted from Storn and Price [39].

Simulated Annealing (SA) Simulated Annealing (SA) [28] is a single-solution metaheuristic optimization paradigm derived from metallurgical thermodynamics, modeling the controlled cooling of a material to reach a minimum-energy crystalline structure. To escape local minima, SA introduces a probabilistic acceptance criterion that allows the model to accept inferior solutions early in the search. The mathematical probability $P(\text{accept})$ of migrating from a current state x to a randomized neighbor state x' is regulated by the Metropolis function:

$$P(\text{accept}) = 1 \quad \text{if } f(x') > f(x) \tag{II.28}$$

$$P(\text{accept}) = \exp\left(-\frac{\Delta E}{T(k)}\right) \quad \text{if } f(x') \leq f(x) \tag{II.29}$$

$$T(k) = T_0 \cdot \alpha^k, \quad \text{where } \alpha = \left(\frac{T_{\text{end}}}{T_0}\right)^{1/n_{\text{iter}}} \tag{II.30}$$

where $\Delta E = |f(x') - f(x)|$ isolates the local loss degradation, $T(k)$ is the system temperature at epoch k, T_0 is the initial temperature configuration, and $\alpha \in (0, 1)$ defines the exponential cooling coefficient over n_{iter} scheduled runs [28].

Figure II.18 presents these dual operational mechanics. The left graph tracks the cooling schedule, demonstrating how the system temperature drops exponentially over time. At high initial temperatures, the Boltzmann probability term in Equation (II.29) approaches unity, allowing wide exploration across the solution space. The right diagram presents the Metropolis acceptance probability curve, showing how the likelihood of accepting unfavorable moves decreases as the temperature drops, shifting the algorithm from global exploration to local exploitation.

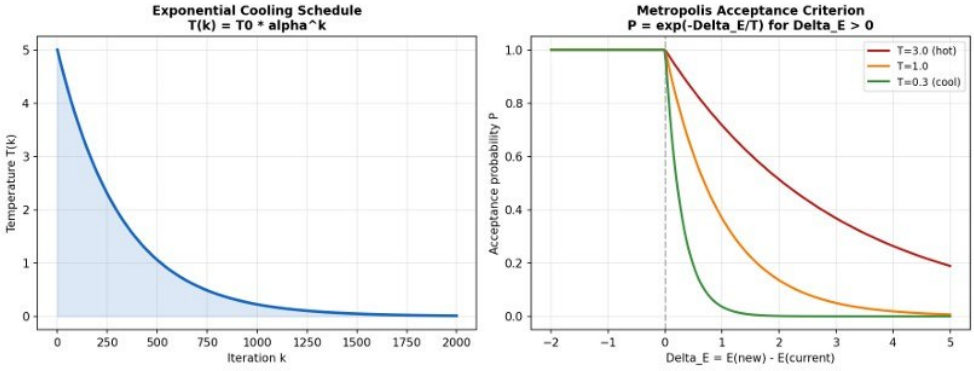


Figure II.18: Simulated Annealing profile: Exponential decay cooling schedule (left) and temperature-dependent Metropolis acceptance probability boundaries (right). Adapted from Kirkpatrick et al. [28].

Optuna — Tree-structured Parzen Estimator (TPE) Optuna [43] deploys a sequential model-based hyperparameter optimization workflow powered by the Tree-structured Parzen Estimator (TPE) algorithm [42]. Instead of computing the standard posterior probability distribution $p(y \mid x)$ directly, TPE applies Bayes' rule by modeling the parameter space using two independent density kernels bounded by an objective performance threshold y^*:

$$l(x) = p(x \mid y < y^*), \quad \text{and} \quad g(x) = p(x \mid y \geq y^*) \tag{II.31}$$

where $l(x)$ forms a kernel density estimate constructed exclusively from the top-performing parameter configurations, while $g(x)$ pools the remaining lower-performing configurations [42, 43]. The optimization trajectory selects subsequent candidate points by maximizing the Expected Improvement (EI), which is mathematically proportional to the ratio between these two density kernels:

$$\mathrm{EI}(x) \propto \frac{l(x)}{g(x)} \tag{II.32}$$

TPE identifies optimal evaluation points where the probability of belonging to the high-performing distribution $l(x)$ maximizes relative to the unpromising distribution $g(x)$ [42].

This spatial splitting mechanism is illustrated in Figure II.19. The left diagram displays the historical evaluation points divided into the high-performing $l(x)$ cluster and the lower-performing $g(x)$ cluster based on the threshold y^*. The right panel maps the corresponding probability density curves alongside the calculated Expected Improvement (EI) function. The peak of this curve identifies the next optimal parameter value suggested for evaluation, balancing exploratory data harvesting with focused local refinement.

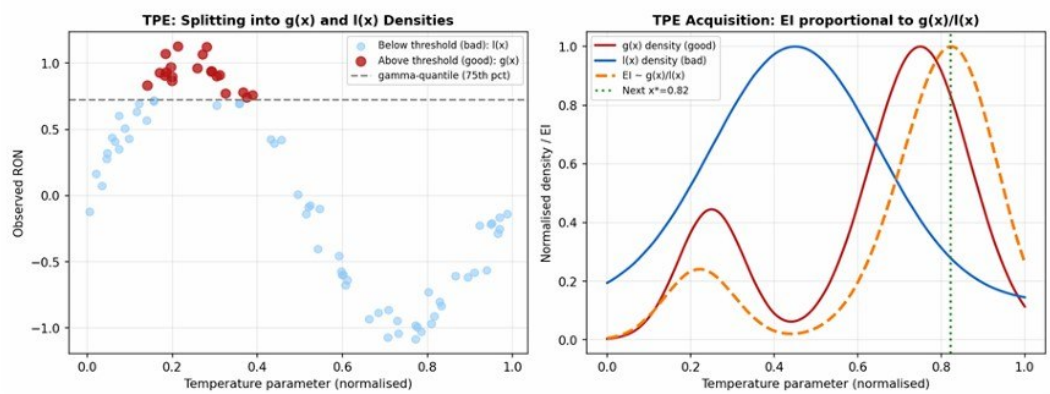


Figure II.19: Optuna TPE mechanics: Historical parameter splitting into density distributions $l(x)$ and $g(x)$ (left) alongside the prioritized Expected Improvement acquisition peak (right). Adapted from Akiba et al. [43] and Bergstra et al. [42].

Bayesian Optimization with Gaussian Process (BayesOpt) BayesOpt [40, 41] is a powerful sequential model-based optimization framework designed to optimize computationally expensive black-box objective functions with high sample efficiency. The algorithm models the underlying objective surface using a non-parametric Gaussian Process ($\mathcal{GP}$) surrogate model, defined as:

$$f(x) \sim \mathcal{GP}\left(\mu(x),\, k(x, x')\right) \tag{II.33}$$

where $\mu(x)$ represents the prior mean function and $k(x, x')$ defines the covariance kernel function (such as the Matérn or Radial Basis Function kernel). Given a set of n collected observations $\mathbf{y}$, the updated posterior predictive distribution at any candidate point x yields a localized mean $\mu_n(x)$ and variance $\sigma_n^2(x)$ computed via matrix inversion:

$$\mu_n(x) = \mathbf{k}^T \left(\mathbf{K} + \sigma_n^2 \mathbf{I}\right)^{-1} \mathbf{y} \tag{II.34}$$

$$\sigma_n^2(x) = k(x, x) - \mathbf{k}^T \left(\mathbf{K} + \sigma_n^2 \mathbf{I}\right)^{-1} \mathbf{k} \tag{II.35}$$

where $\mathbf{K}$ is the $n \times n$ kernel covariance matrix evaluated across all historical data points, $\mathbf{k}$ is the cross-covariance vector linking the new candidate x to past points, and σ_n^2 captures experimental observation noise. To select the next evaluation point, the algorithm maximizes an acquisition function, such as the Upper Confidence Bound (UCB):

$$x_{t+1} = \arg\max_x \left[\mu_n(x) + \kappa \cdot \sigma_n(x)\right] \tag{II.36}$$

where the hyperparameter $\kappa \geq 0$ serves as a balance weight controlling the exploration–exploitation trade-off [40, 41].

Figure II.20 presents this sequential optimization loop. The left diagram shows the active $\mathcal{GP}$ surrogate model fitted across historical observations. The solid blue curve maps the predicted mean $\mu_n(x)$, while the shaded region represents the surrounding uncertainty interval $\sigma_n(x)$. Unexplored regions exhibit a wide uncertainty band, whereas areas near existing data points display

narrow confidence intervals. The right diagram isolates the matching Expected Improvement (EI) acquisition function. The peak of this function determines the exact position of the next experimental evaluation point (x_{t+1}), targeting high-uncertainty or high-performance zones.

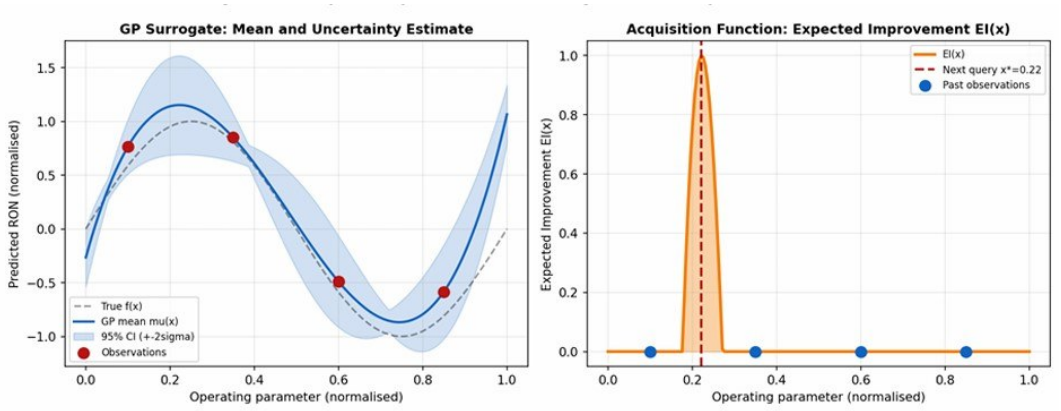


Figure II.20: Bayesian Optimization framework: Gaussian Process surrogate mean and uncertainty bounds (left) linked to the calculated Expected Improvement acquisition function profile (right). Adapted from Jones et al. [41] and Mockus [40].

4 Digital Twin

A Digital Twin (DT) is a dynamic virtual representation of a physical asset, process, or system that is continuously updated using data from its real-world counterpart [45]. By combining physical knowledge, historical data, real-time measurements, and advanced analytical models, a digital twin enables the monitoring, simulation, prediction, and optimization of system behavior in a virtual environment.

Digital twins have become an important component of Industry 4.0 and are widely used in manufacturing, energy, transportation, healthcare, and process industries. Their ability to replicate the behavior of physical systems allows engineers and operators to evaluate different operating scenarios, predict future performance, detect anomalies, and support decision-making without directly affecting the real system [46].

One of the main advantages of digital twins is their ability to perform "what-if" analyses and optimization studies in a safe and cost-effective manner. As advances in Artificial Intelligence, Machine Learning, and Industrial Internet of Things (IIoT) technologies continue, digital twins are expected to play an increasingly important role in intelligent and autonomous industrial systems [45, 46].

5 State of the Art in Isomerization Optimization

The optimization of catalytic isomerization units represents a critical frontier in modern petroleum refining, where maximizing the Research Octane Number (RON) must be carefully balanced

against complex thermodynamic limitations, catalyst deactivation pathways, and strict operating margins. To contextualize the technological milestones governing this domain, this section provides a critical, comprehensive overview of contemporary literature, tracing the structural transition from classical deterministic modeling to modern data-driven and metaheuristic optimization architectures.

5.1 Machine Learning in Isomerization

The catalytic isomerization process plays a crucial role in modern refineries by converting low-octane linear paraffins into high-octane branched isomers, thereby substantially improving gasoline blending pools while meeting increasingly stringent environmental and emission regulations. However, the operational performance of an industrial isomerization unit depends heavily on an array of highly interacting, multi-variable parameter streams, including multi-stage reactor temperatures, operating pressure, fluctuating feed compositions, hydrogen-to-hydrocarbon (H_2/HC) ratios, and transient catalyst poisoning states. Managing these variables efficiently and simultaneously is essential for maximizing product quality, increasing unit profitability, reducing furnace energy consumption, and extending the operational lifetime of expensive noble-metal catalysts.

The increasing availability of high-fidelity historical process data and distributed computational resources has significantly accelerated the adoption of Artificial Intelligence (AI) techniques in refinery automation. According to Venkatasubramanian [13], AI enables data-driven modeling approaches capable of capturing highly nonlinear, non-equilibrium process behaviors that remain exceptionally difficult to describe or solve using conventional, first-principles kinetic or thermodynamic models.

Among the earliest machine learning approaches systematically applied to catalytic isomerization were Multi-Layer Perceptrons and standard Artificial Neural Networks (ANNs) [49]. These pioneering models successfully demonstrated the capacity to map nonlinear industrial processes and generate reliable target estimations directly from historical operating logs. Such predictive capabilities provide an essential proactive safeguard, enabling plant operators to accurately anticipate complex chemical process responses before executing physical setpoint changes, thereby reducing down-time risks and improving operational decision-making.

The subsequent development of ensemble learning methods further elevated predictive accuracy and enhanced model robustness against data noise. Bagging and boosting tree variants—specifically Random Forest [26], Gradient Boosting [15], XGBoost [16], LightGBM [17], and CatBoost [18]—have demonstrated outstanding performance across complex petrochemical workflows due to their inherent structural capability to handle deep non-linear interactions, noisy sensor measurements, collinear variables, and large-scale industrial tabular datasets.

More recently, Deep Learning techniques such as Deep Neural Networks (DNNs) and Long Short-Term Memory (LSTM) recurrent networks have attracted significant research interest for industrial process modeling [10, 19]. Their ability to automatically extract high-level, hierar-

chical patterns from raw datasets and capture long-range temporal dependencies makes them particularly suitable for modeling the highly dynamic, time-varying behaviors inherent to continuous refinery processes.

5.2 Evolutionary and Bayesian Algorithms in Process Optimization

While advanced machine learning models provide highly accurate forward predictions of complex process behaviors, they remain purely descriptive; robust optimization algorithms are required to actively navigate these learned surfaces and determine the precise operating conditions that yield peak process performance. In catalytic isomerization units, optimization routines aim to identify the most favorable, cost-effective operating setpoints while strictly satisfying an intricate matrix of technical, safety, equipment, and environmental constraints.

Evolutionary Algorithms (EAs) and swarm-based metaheuristics have become widely used in chemical process optimization due to their unique structural capacity to solve highly nonlinear, non-convex, and multi-objective problems without requiring analytical gradient information or risking localized stagnation on rough error surfaces [12]. Among the most influential approaches, the Non-dominated Sorting Genetic Algorithm II (NSGA-II) stands as a foundational benchmark for multi-objective optimization [47]. It has been successfully applied across numerous refinery processes where product quality, energy efficiency, and economic performance must be optimized simultaneously to construct a representative Pareto-optimal front.

Differential Evolution (DE) [39] has also demonstrated excellent performance in chemical engineering applications due to its algorithmic simplicity, parameter robustness, and strong global exploration capabilities when operating over continuous decision variables. In parallel, for scenarios where evaluating the objective function requires querying computationally expensive models, Bayesian Optimization has gained substantial popularity. By utilizing Gaussian Process surrogates or Tree-structured Parzen Estimators, Bayesian frameworks can isolate high-quality global solutions using a strictly limited number of objective evaluations, thereby drastically minimizing computational overhead and optimization execution times [40, 43].

The structural integration of these machine learning predictive models with advanced optimization algorithms has ultimately led to the development of intelligent, automated decision-support systems capable of continuously tracking and improving refinery operations. By facilitating optimal product quality, higher operational throughput, minimized energy consumption, enhanced catalyst utilization, and lower production costs, these integrated frameworks form a core component of modern smart refinery initiatives and industrial digital twin architectures [13, 45].

6 Conclusion

This chapter presented the theoretical foundations of Artificial Intelligence for prediction and optimization. It introduced the fundamental concepts of AI, Machine Learning, and Deep Learn-

ing, as well as the main learning paradigms and the different categories of problems addressed by AI techniques [7, 19].

The chapter also reviewed the principal predictive models used in data-driven applications, ranging from ensemble learning methods such as Random Forest [26], Extra Trees [37], Gradient Boosting [15], XGBoost [16], LightGBM [17], and CatBoost [18], to deep learning architectures including DNNs, CNNs, MLPs, RNNs, and LSTMs [19, 33]. In addition, the concepts of optimization, optimization problem formulation, and the most widely used optimization algorithms were discussed [12, 47, 39, 40, 43].

The next chapter presents the proposed methodology and overall system architecture, describing the different stages involved in data processing, predictive modeling, optimization, and digital twin development.

Chapter III

System Design

1 Introduction

Catalytic isomerization units operate under complex and highly nonlinear conditions influenced by multiple interacting process variables, catalyst aging, and delayed quality measurements [66]. These challenges limit the effectiveness of conventional manual operation and traditional control approaches [66].

To overcome these limitations, this work proposes an Intelligent Digital Twin-Based Optimization System for the Sonatrach U510 Naphtha Isomerization Unit. The proposed solution integrates industrial data acquisition, machine learning models, digital twin technology, and optimization algorithms within a unified cyber-physical framework to predict product quality and identify optimal operating conditions.

The system is organized into four interconnected layers: the Physical System Layer, the Data Acquisition and Integration Layer, the Digital Twin system Layer, and the End-User Application Layer. Together, these layers transform industrial process data into actionable recommendations that support operational decision-making and process optimization.

This chapter presents the overall design of the proposed solution, describes its four-layer architecture, and explains the interactions between its components that enable intelligent prediction, optimization, and decision support.

2 Global Architecture

The proposed Intelligent Digital Twin-Based Optimization system is designed as a modular, scalable, and industrial-grade architecture for real-time prediction and optimization of catalytic isomerization processes [66]. The system integrates industrial sensing infrastructure, data engineering pipelines, machine learning surrogate models, and optimization algorithms into a unified decision-support ecosystem capable of transforming raw operational data into optimal operating recommendations [66].

The architecture is organized into four interconnected layers, each responsible for a specific function within the overall optimization workflow as shown in Figure III.1.

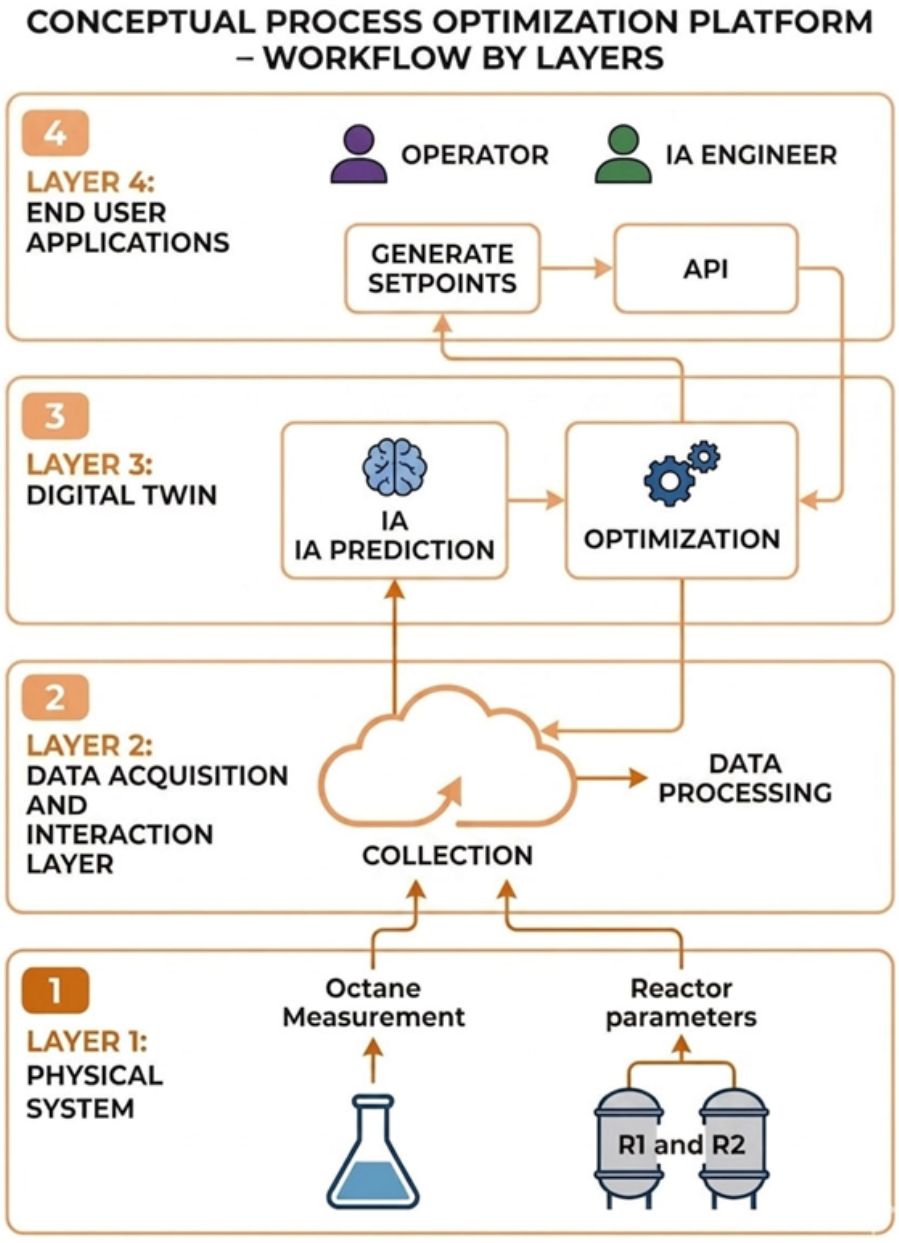


Figure III.1: Conceptual Process Optimization Platform – Workflow by Layers

The global architecture of the proposed intelligent optimization platform for Sonatrach's Unit U510 (Naphtha Isomerization Unit) is engineered as a hierarchical, closed-loop cyber-physical system (CPS) structured across four highly integrated layers.

The operational workflow begins at the Physical System (Layer 1), where high-frequency field sensors monitor continuous process variables across Reactors R-1 and R-2, supplemented by asynchronous daily laboratory analysis of product Research Octane Number (RON). These raw inputs are captured by the Data-Acquisition & Integration Layer (Layer 2) through the plant's Distributed Control System (DCS) and OPC UA communication servers, which route the

telemetry to an Industrial Data Historian [77, 78]. This middleware layer synchronizes the multi-rate streams, filters high-frequency instrument noise, and appends programmatic `Catalyst_ID` and `Catalyst_Age` metrics to track catalyst degradation over time.

The resulting structured parameter vectors are then passed to the Digital Twin Platform (Layer 3), which serves as the computational core. This layer uses high-performance ML models as fast numerical surrogates to predict unit performance metrics in milliseconds, providing instant objective evaluations for a single-objective evolutionary optimization engine. The optimization algorithm resolves the conflicting parameters of octane maximization, decoding the selected strategy into ideal operating temperature setpoints for Reactor 1 (R_1) and Reactor 2 (R_2).

Finally, the Role-Based User Applications Layer (Layer 4) translates these complex back-end operations into intuitive human-machine interfaces, allowing the Operator to validate and apply the recommended temperature targets directly to configure the DCS controllers, while enabling AI Engineers to track model accuracy and trigger automated retraining cycles when process drift is detected.

To complement the architectural view presented in Figure III.1 that illustrates the operational workflow of the proposed intelligent optimization system, Figure III.2 describes the interactions between its two primary user roles: the Operator and the AI Engineer.

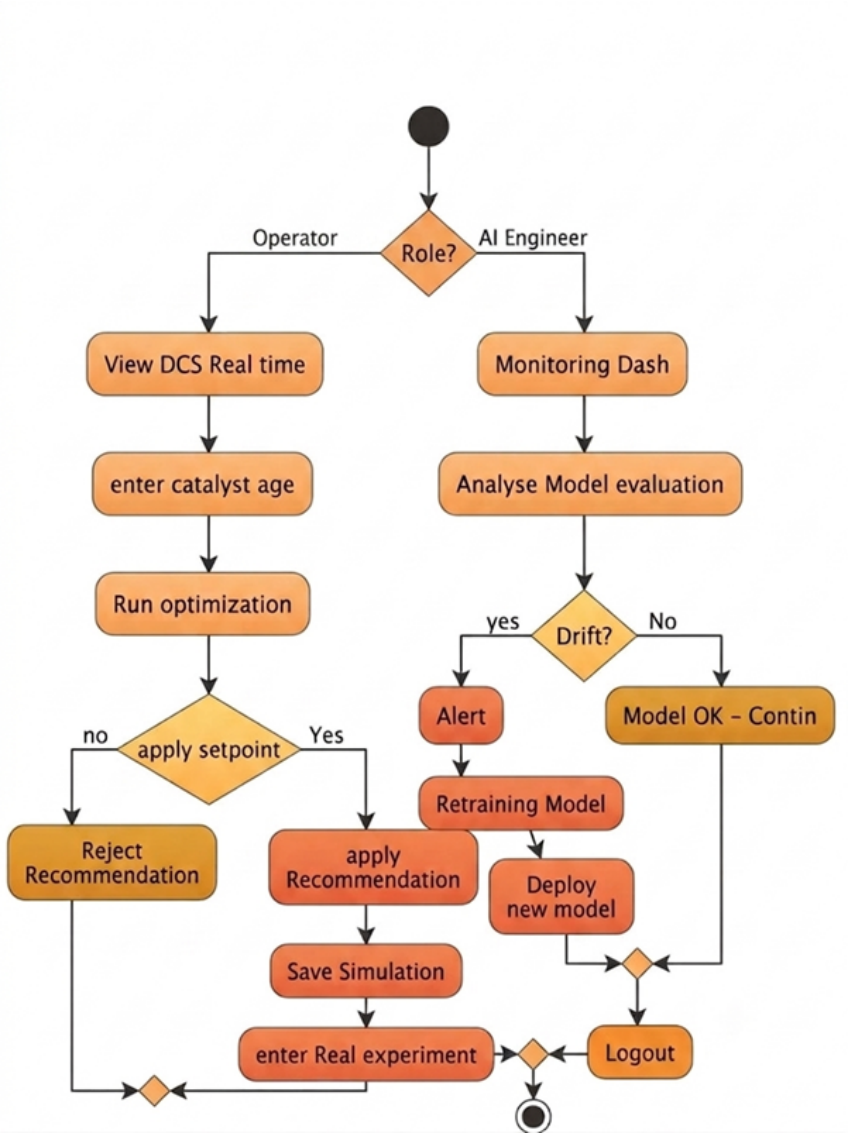


Figure III.2: Activity diagram describing the operational workflow of the proposed Digital Twin-based optimization system and the interactions between Operators and AI Engineers.

The activity begins when a user accesses the platform and selects a role. The workflow then follows two parallel paths depending on the responsibilities of the selected actor:

2.1 Operator Workflow

The operator interacts directly with the optimization system through the plant supervision interface.

- First, real-time process measurements are retrieved from the Distributed Control System (DCS) [73].
- The operator then enters the current catalyst age, which is an important process parameter influencing reactor performance and product quality.

- Once the required operational information is available, the optimization module is executed.
- The Digital Twin evaluates multiple operating scenarios and generates recommended temperature setpoints designed to improve the predicted Research Octane Number (RON).
- The operator reviews the proposed recommendation and decides whether to apply it: If the recommendation is rejected, the workflow terminates without modifying the operating conditions. If the recommendation is accepted, the optimized setpoints are applied to the process.
- The corresponding simulation results are stored in the system database, and the real operational outcome can later be recorded as a new experiment for future analysis and model improvement.

2.2 AI Engineer Workflow

The AI Engineer is responsible for monitoring and maintaining the predictive performance of the Digital Twin models.

- Through the monitoring dashboard, the engineer analyzes model evaluation metrics and continuously tracks prediction quality.
- A decision point is then reached to determine whether model drift has occurred.
- If no drift is detected, the model is considered valid and continues operating normally.
- If drift is detected, an alert is generated and a retraining procedure is initiated using the latest available operational data.
- After retraining, the updated model is validated and deployed to replace the previous version. The new model then becomes available for future optimization cycles.

3 Detailed Design

After seeing the global architecture, this section presents the different layers of our system.

3.1 Layer 1: Physical System

The foundational layer of the architecture comprises the actual industrial refinery hardware, metallurgical process boundaries, and the hydrocarbon chemical streams undergoing molecular transformation; it constitutes the tangible physical environment that the digital system monitors, models, and optimizes, as shown in Figure III.3.

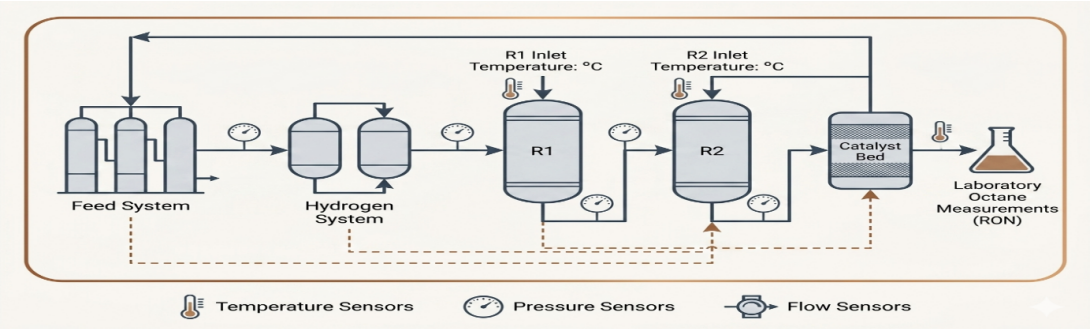


Figure III.3: Physical configuration of the isomerization process.

The physical configuration and its core thermodynamic parameters consist of the following components:

3.1.1 Feedstock Infrastructure and Preheating System

The process originates with the *Feed System*, which introduces the liquid hydrocarbon *feedstock*—typically light straight-run Naphtha rich in low-octane linear paraffins (n-pentane and n-hexane)—into the optimization perimeter under specific baseline characteristics. Before entering the catalytic zone, this stream is combined with a continuous stream of recycle hydrogen gas and routed through a charge heater to achieve the required phase vaporization and activation temperature.

3.1.2 Multi-Stage Isomerization Reactors

The vaporized feedstock is routed sequentially through a multi-stage reaction section, specifically *Reactor R-301* and *Reactor R-302*, operating in a series configuration.

Definition of a Chemical Reactor: In chemical process engineering, a reactor is a highly specialized, pressure-rated process vessel designed to contain, control, and sustain chemical reactions under strictly regulated thermodynamic conditions [70].

Process Dynamics: In this architecture, R-301 and R-302 function as fixed-bed adiabatic reactors [71]. The primary objective is to facilitate the structural rearrangement of straight-chain alkanes into high-branched *isomers* (such as isopentane and dimethylbutanes) without altering their molecular weight. Because the hydroisomerization reaction is slightly exothermic and limited by *thermodynamic equilibrium* constraints, splitting the reaction across a multi-stage configuration allows for intermediate temperature control, optimizing both the reaction kinetics and the ultimate chemical conversion efficiency [71, 70].

3.1.3 The Heterogeneous Catalyst Bed

Housed securely within the internal structural grids of both reactors, the *Catalyst Bed* provides the active surface matrix required to lower the activation energy of the hydroisomerization path-

way. This architecture typically utilizes a highly active bifunctional catalyst, such as platinum supported on chlorinated alumina or zeolites [71].

Technical Operational Variables: The behavior of the catalyst bed is highly sensitive to fluctuations in the *Weight Average Bed Temperature (WABT)* and the *Liquid Hourly Space Velocity (LHSV)*—an industrial metric defining the ratio of the volumetric liquid feed flow rate per hour to the total volume of catalyst inside the vessel [71].

Deactivation Mechanics: Over prolonged operational cycles, the catalyst suffers from irreversible *sintering* and reversible *coking* (the progressive accumulation of carbonaceous deposits on the active platinum sites) [71]. This phenomenon of catalyst deactivation causes process drift, forcing operators to step up the WABT to maintain steady-state conversion levels, a variable that the digital twin must dynamically approximate.

3.1.4 The Primary Sensor Array

Continuous physical monitoring of the thermodynamic and hydrodynamic states of the plant is executed via a comprehensive network of inline field instruments.

Definition of an Industrial Sensor: An industrial sensor (or primary sensing element) is an instrumental transducer designed to continuously detect a specific physical or chemical property of a process medium and convert that phenomenon into a standardized electrical signal (e.g., a 4–20 mA analog current or digital fieldbus protocol) suitable for *telemetry* and remote supervisory control [68].

Sensor classifications within the array include:

- **Temperature Sensors:** Consisting of high-precision Thermocouples and Resistance Temperature Detectors (RTDs) strategically distributed across the reactor inlets, internal catalyst bed layers, and outlets to capture the exothermic profile and protect against thermal runaway.
- **Pressure Sensors:** Utilizing electronic differential pressure (ΔP) transmitters to monitor pressure drops across the packed catalyst bed (a critical diagnostic indicator for bed compaction or channeling) as well as total vessel operating constraints.
- **Flow Sensors:** Comprising differential pressure orifice plates or Coriolis mass flow meters to measure the overall volumetric *throughput* of the feedstock and regulate the molar hydrogen-to-hydrocarbon (H_2/HC) ratio required to suppress coke deposition.

3.1.5 Laboratory Octane Measurements (RON)

The definitive target performance metric of the physical system is the product quality of the stabilized *isomerate* stream, quantified by its *Research Octane Number (RON)*.

Technical Challenge: RON is a standardized metric that measures a fuel's antidetonation and knock resistance within a spark-ignition engine [72]. In an industrial refinery, determining RON traditionally requires manual sample extraction followed by analytical testing via downstream laboratory gas chromatography or standardized Cooperative Fuel Research (C CFR) testing engines.

The Latency Problem: Because laboratory analyses are discrete and suffer from an inherent multi-hour dead-time delay, RON cannot be integrated directly into real-time closed-loop control. This latency creates the operational requirement for the data-driven surrogate models in Layer 3 to act as an online "soft sensor".

3.2 Layer 2: Data-Acquisition & Integration Layer

Layer 2 functions as the operational middleware and communication bridge between the physical plant environment (Layer 1) and the digital computing ecosystem (Layer 3).



Figure III.4: Data acquisition and transmission pipeline.

The transmission pipeline relies on three interconnected industrial computing technologies to guarantee safe, lossless, and secure data extraction:

3.2.1 Distributed Control System (DCS)

The immediate aggregation point for all real-time operational telemetry is the plant's Distributed Control System (DCS) [73]. The DCS consists of a network of decentralized micro-processor controllers that execute deterministic regulatory control loops (such as Proportional-Integral-Derivative or PID algorithms) at the microsecond level [73]. Field sensor signals cross physical barrier lines and terminate at specialized input/output (I/O) modules. Here, raw electronic voltages or currents are digitized and bound to specific instrumentation tags, isolating the underlying field hardware from the rest of the enterprise network.

3.2.2 Open Platform Communications (OPC) Interface

To establish data interoperability across historically proprietary vendor systems, an OPC (Open Platform Communications) infrastructure is deployed as the primary software integration layer. Specifically utilizing the OPC Unified Architecture (OPC UA) standard, this component abstracts complex hardware differences into an object-oriented information model [77]. The OPC

Server continuously maps data points from the DCS memory buffers, creating secure, read-only server-client communication channels that stream process parameters to outside analytical servers without degrading core plant control stability.

3.2.3 Industrial Data Historian

Because refinery control networks are optimized for immediate real-time execution rather than long-term memory allocation, telemetry is passed into an *Industrial Data Historian* [78]. The Historian is a high-performance database engine specifically designed to ingest high-frequency, time-series data streams [78]. It utilizes advanced data compression algorithms to store immense volumes of historical tag data with highly granular time-stamping, serving as the definitive repository for long-term analytical extraction.

3.2.4 Case Study Dataset Profiling (Sonatrach Asset)

The data used to develop and validate the intelligent optimization platform is compiled directly from a production refinery unit operated by *Sonatrach* [1].

- Sampling Resolution: Telemetry tags are historically sampled at fixed *15-minute intervals*.
- Dataset Volume: The raw operational snapshot contains exactly *223,291 observations*.
- Tag Mapping Dictionary: Each individual tracking tag mapped via the DCS-OPC pipeline represents a specific critical process parameter within the catalytic isomerization system, as detailed in the technical schema below (Table III.1):

DCS Tag Identifier	Physical Parameter Description	Metric Engineering Units
FI0011	Hydrogen Flow Rate	kg/h
FI0017	Tetrachloroethylene (TCE) Flow Rate	l/h
FIC0003	Combined Hydrocarbon Feed Flow Rate	m^3/h
FIC0016	Recycle Gas Volumetric Flow Rate	m^3/h (Standard)
PI0035	Reactor 1 (R_1) Inlet Operational Pressure	kg/cm^2
TIC0024	Reactor 1 (R_1) Total Inlet Temperature	°C
TI0025 to TI0036	Reactor 1 (R_1) Internal Bed Thermocouples (1 to 12)	°C
TI0037	Reactor 1 (R_1) Outlet Temperature	°C
PI0040	Reactor 2 (R_2) Inlet Operational Pressure	kg/cm^2
TIC0039	Reactor 2 (R_2) Total Inlet Temperature	°C
TI0040 to TI0051	Reactor 2 (R_2) Internal Bed Thermocouples (1 to 12)	°C
TI0052	Reactor 2 (R_2) Outlet Temperature	°C
PDI0051A	Reactor 1 (R_1) Total Differential Pressure Drop (ΔP)	kg/cm^2
PDI0052A	Reactor 2 (R_2) Total Differential Pressure Drop (ΔP)	kg/cm^2

Table III.1: DCS Tag Mapping Dictionary

To facilitate the data-driven modeling process, the raw telemetry extracted from the DCS and Historian databases is structured into a flattened tabular format [75]. Based on the operational perimeter, the dataset contains the following critical columns representing the key thermodynamic and hydrodynamic variables of the isomerization unit:

Temporal & Identification Features These features establish the chronological order and life-cycle state of the physical system:

- `DateTime`: The exact timestamp of the operational observation, sampled at 15-minute intervals.
- `Catalyst_ID`: An engineered cyclical index (ranging from 1 to 8) representing distinct operational runs or virtual catalyst structures.
- `Catalyst_Age`: A continuous, engineered metric tracking the operational day count of active catalyst deployment to model degradation.

Hydrodynamic Flow Variables (Feedstock & Additives) These variables dictate the throughput and chemical balance entering the catalytic zone:

- `Flow_Cha`: The Combined Hydrocarbon Feed Flow Rate, representing the volumetric input of light straight-run naphtha.

- `Flow_H2`: The Hydrogen Flow Rate, critical for establishing the required H_2/HC molar ratio to suppress coke deposition.
- `Flow_TCE`: The Tetrachloroethylene flow rate, used as a chloriding agent to maintain catalyst activity.

Reactor 1 (R-301) Thermodynamic Profile These columns monitor the first stage of the multi-stage fixed-bed adiabatic reaction system:

- `P_in_R1`: Reactor 1 Inlet Operational Pressure.
- `T_in_R1`: Reactor 1 Total Inlet Temperature.
- `T_R_001` to `T_R_001.11`: The array of 12 internal bed thermocouples distributed across the catalyst layers to capture the exothermic thermal profile of Reactor 1.
- `T_sortie_F` (Outlet R1): The final outlet temperature exiting Reactor 1.

Reactor 2 (R-302) Thermodynamic Profile These columns monitor the second stage of the reaction, ensuring ultimate chemical conversion efficiency:

- `P_in_R2`: Reactor 2 Inlet Operational Pressure.
- `T_in_R2`: Reactor 2 Total Inlet Temperature.
- `T_R_002` to `T_R_002.11`: The array of 12 internal bed thermocouples distributed across Reactor 2.
- `T_sortie` (Outlet R2): The final stabilized outlet temperature exiting Reactor 2.

Target Performance Metric

- `Octane`: The Research Octane Number (RON) obtained via laboratory sample extraction.

3.2.5 Asynchronous Temporal Data Fusion

A primary challenge in refining data architecture is the severe sampling rate mismatch between continuous automated telemetry and target product quality metrics [76]. The DCS-OPC pipeline records process temperatures, pressures, and flow streams continuously every 15 minutes [74]. In contrast, the true performance metric—the research octane number (RON)—is generated manually through offline laboratory chemical analysis once per day.

To train a robust machine learning surrogate model, these multi-tier sources must undergo *asynchronous data fusion* [76]. The preprocessing engine executes an integration workflow that matches the daily laboratory timestamp with its corresponding 15-minute continuous process vectors. Rather than applying forward-filling mechanisms that introduce data leakage, the engine synchronizes the daily analytical targets back to the specific steady-state operational windows of the unit, creating a unified dataset suitable for predictive modeling.

3.2.6 Process-Informed Data Augmentation & Train/Test Partitioning

Because industrial datasets often lack explicit tracking variables for structural plant changes, two new process-informed features are programmatically generated and appended to the dataset:

Catalyst ID Generation (`Catalyst_ID`): To model performance variations across alternate catalyst life cycles or distinct batch configurations, the system implements a cyclical indexing principle. The observations are assigned a rotating index ranging from 1 to 8. This variable simulates the behavior of eight distinct operational runs or virtual catalyst structures.

Catalyst Age Tracking (`Catalyst_Age`): Because the 15-minute sampling frequency generates exactly 96 distinct telemetry records over a standard 24-hour period, the system tracks catalyst degradation by incrementing the operational age by 1 unit for every block of 96 consecutive records. This establishes a continuous metric representing the operational day count of active catalyst deployment.

To validate the model's ability to generalize to future operational horizons, a robust *Chronological Cross-Validation Strategy* is applied:

- **Training Dataset:** Compiled from observations bound to `Catalyst_ID` 1 through 6, allowing the model to learn baseline kinetics across multiple cyclical variations.
- **Evaluation/Testing Dataset:** Compiled from observations bound to `Catalyst_ID` 7 and 8, testing the system's ability to maintain predictive accuracy on unseen catalyst lifecycles.
- **Theoretical Blind Test Window:** The *final 15 calendar days* of the continuous dataset are completely isolated and held back from all training loops. This subset serves as a continuous, time-ordered blind test to verify the digital twin's predictive stability over real-time operational horizons.

3.2.7 Operational Data Preprocessing

To transform the raw integrated matrix into pristine inputs for the machine learning models, data is systematically processed through a deterministic four-stage engineering pipeline [75]:

Datetime Parsing and Chronological Sequencing

The pipeline scans all object-type and string columns to automatically identify, parse, and standardize timestamp entries into localized datetime structures. Once the uniform time vectors are established, the entire dataset is sorted in strict chronological order. This step ensures that subsequent time-series splits do not introduce data leakage or look-ahead bias during training.

Deduplication and Structural Consolidation

Due to network lag, server restarts, or redundant database scripts, the raw logs may contain exact duplicate rows. The pipeline performs structural de-duplication by identifying and removing identical multi-variable entries sharing the same timestamp, preserving data integrity.

Resilient Outlier Detection via Lenient IQR Capping

Industrial instrumentation data is highly prone to brief signal spikes, sensor calibration loss, and environmental interference [68]. To filter these anomalies without removing valid extreme states, the system utilizes a *Winsorization method* based on the Interquartile Range (IQR) [79]:

$$\text{IQR} = Q_3 - Q_1 \tag{III.1}$$

Where Q_1 is the 25th percentile and Q_3 is the 75th percentile of the feature distribution. Any observation falling outside the modified limits is capped at the boundary thresholds:

$$\text{Lower Bound} = Q_1 - 3 \times \text{IQR} \tag{III.2}$$

$$\text{Upper Bound} = Q_3 + 3 \times \text{IQR} \tag{III.3}$$

The multiplier of 3 is intentionally selected instead of the traditional 1.5 standard [79]. In chemical refining, rapid process shifts—such as unit start-ups, raw feedstock switches, or planned operational adjustments—generate extreme data points that are physically accurate and vital for training. A strict 3.0 threshold filters erroneous sensor spikes while preserving these critical physical conditions. Crucially, tracking identifiers (`Catalyst_ID`, `Catalyst_Age`) and the target variable (`Octane`) are excluded from this capping routine to preserve their analytical properties.

3.3 Layer 3: Digital Twin Platform

The Digital Twin Platform constitutes the intelligence core and decision-making brain of the proposed architecture [45]. While Layers 1 and 2 collect and organize operational data from the physical plant, Layer 3 transforms these raw measurements into actionable knowledge through prediction, simulation, optimization, and decision support [66].

The Digital Twin maintains a dynamic virtual representation of the U510 Naphtha Isomerization Unit, continuously mirroring its operational state and evaluating potential operating strategies before they are applied to the real process, as structured in Figure III.5.

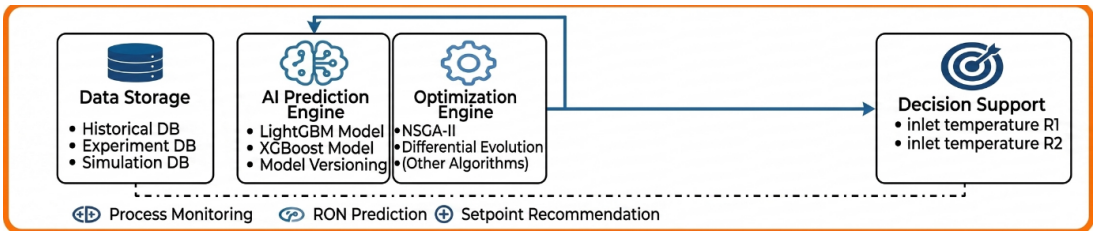


Figure III.5: The Digital Twin system organization and interconnected modules.

The Digital Twin system is organized into three interconnected modules:

3.3.1 Data Storage System

The Data Storage System serves as the memory of the Digital Twin and stores all information required for prediction, optimization, and continuous learning.

- **Historical Database (Historical DB):** Stores long-term operational records collected from the refinery. These data are used for machine learning model training, validation, and periodic retraining.
- **Experiment Database (Experiment DB):** Records optimization recommendations, operator decisions, applied setpoints, and laboratory measurements. This database creates a traceable history of operational experiments and system performance.
- **Simulation Database (Simulation DB):** Stores thousands of virtual operating scenarios evaluated during optimization. These records enable rapid comparison of alternative operating conditions without affecting the physical plant.

3.3.2 AI Prediction and Optimization Engine

The AI Prediction and Optimization Engine represents the analytical center of the Digital Twin and serves as the primary decision-making component of the system. It contains the machine learning surrogate models developed in this work, including MLP, LSTM, XGBoost, LightGBM, CatBoost, Random Forest, and Extra Trees models [16, 17, 18].

During the prediction phase, the engine retrieves historical process observations and laboratory RON measurements from the Historical Database. Once deployed, the trained models receive real-time process variables from Layer 2 and estimate the expected Research Octane Number (RON) within milliseconds.

To maintain long-term predictive accuracy, newly collected laboratory measurements, operator actions, and operational outcomes stored in the Experiment Database are continuously used to monitor model performance, detect process drift, and trigger automated retraining procedures whenever prediction accuracy degrades.

The optimization subsystem integrates advanced algorithms [67]. During each optimization cycle, candidate operating conditions are generated and evaluated using the machine learning

surrogate models rather than the physical plant itself. The resulting virtual evaluations are stored in the Simulation Database, which acts as a high-speed repository for thousands of hypothetical operating scenarios. This database enables the optimization algorithms to compare alternative reactor temperature setpoints, track search progress, and avoid repeated evaluations of identical operating conditions.

The optimization engine ultimately identifies the operating conditions that maximize the predicted RON while satisfying process constraints [67]. The best candidate solution is then forwarded to the Decision Support Module, which converts the optimization result into actionable reactor setpoint recommendations for plant operators.

3.3.3 Decision Module

The Decision module converts optimization results into practical industrial recommendations. After evaluating thousands of candidate operating conditions, the module selects the best solution and generates recommended setpoints for Reactor 1 and Reactor 2. These recommendations are presented to plant personnel through the application layer, where operators can review, validate, and apply them to the real process. The resulting plant response is subsequently fed back into the Digital Twin, allowing continuous learning and adaptation.

Closed-Loop Learning Mechanism A key characteristic of the proposed architecture is its closed-loop operation, as structurally mapped in Figure III.6. Once recommendations are implemented, new process measurements and laboratory RON values are collected and stored in the databases. These data are then used to evaluate model performance, detect process drift, and trigger automatic retraining when necessary. Through this continuous feedback cycle, the Digital Twin evolves alongside the physical plant, maintaining prediction accuracy, adapting to catalyst aging effects, and ensuring sustained optimization performance throughout the operational lifecycle of the refinery unit.

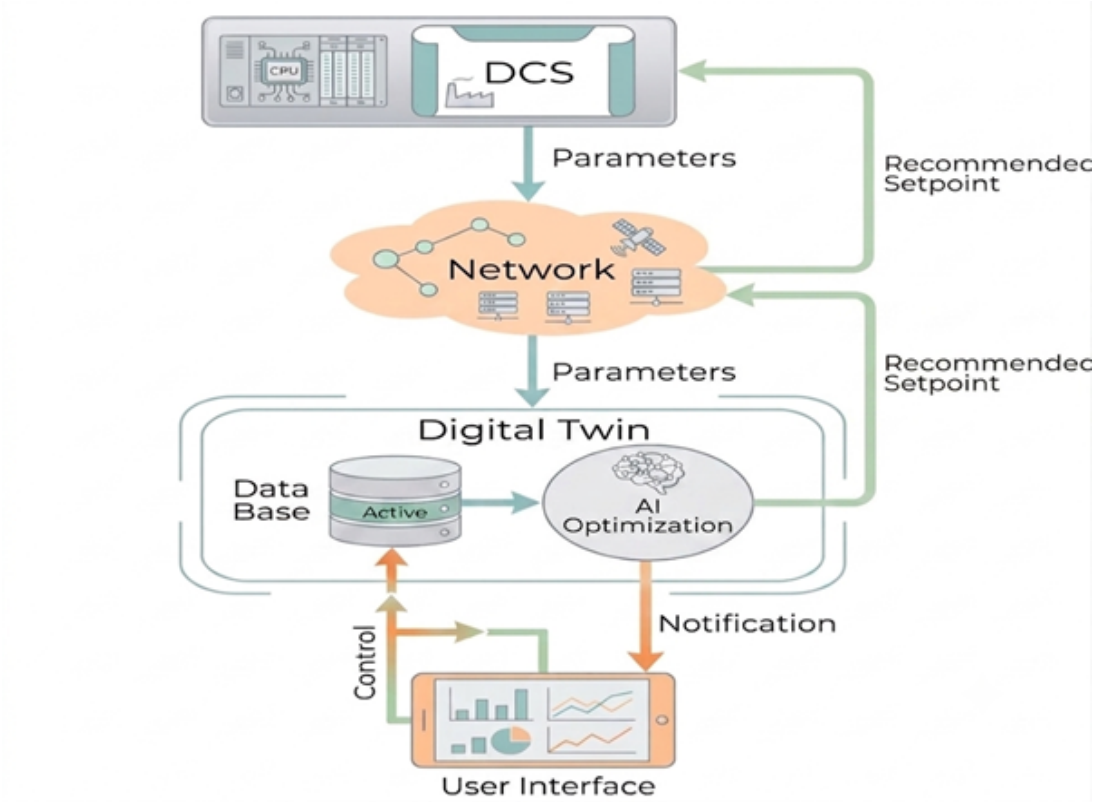


Figure III.6: Continuous feedback and closed-loop model retraining workflow.

3.4 Layer 4: Role-Based End-User Applications Layer

Layer 4 represents the human-machine interface (HMI), the operational control center, and the visualization frontend of the intelligent optimization platform [69]. While the underlying layers handle high-frequency data ingestion (Layer 2) and heavy computational modeling or optimization loops (Layer 3), Layer 4 abstracts these complex back-end mechanisms into intuitive, role-based, and actionable user interfaces. It serves as the definitive bridge where data-driven insights are translated into real-world human decisions on the refinery floor.

To fulfill these requirements for the primary user, a production-grade graphical user interface (GUI) was developed to serve as the operator's primary optimization workspace, as illustrated in Figure III.7.

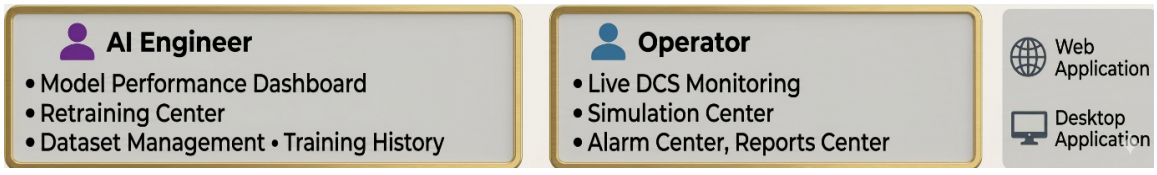


Figure III.7: Intelligent Digital Twin Optimization Platform – Operator Workspace Dashboard (Figure 3.2).

3.4.1 The Operator Subsystem & Dashboard UI Elements

The Plant Operator represents the primary industrial user responsible for maintaining steady-state unit operations, maximizing corporate product quality targets, and protecting the mechanical integrity of the isomerization reactor vessels (R_1 and R_2) [73]. Within Layer 4, the Operator interacts with the unified workspace panel (Figure III.7), which is structured into four functional zones:

- **Configuration and Catalyst Life-Cycle Management Panel:** Located in the upper section of the dashboard, this panel enables the operator to manage data-entry use cases. It features a dropdown selector for the active `Catalyst_ID` (ranging from Run 1 to Run 8) and an integer input field for tracking the continuous `Catalyst_Age` in operational days. This input provides critical contextual deactivation parameters directly to the underlying digital twin [71].

- **DCS Telemetry and Process Parameters Readout:** This section displays the localized process conditions synchronized from the Layer 2 data pipelines [74]. It tracks live hydrodynamic flow variables, including Hydrocarbon Charge Flow (`Flow_Cha` in m^3/h), Hydrogen Gas Flow (`Flow_H2` in kg/h), and Tetrachloroethylene additive injection (`Flow_TCE` in l/h). It also provides full visibility into the current reactor bed thermodynamic states, showing the exact operational temperatures (°C) and inlet/outlet pressures (kg/cm^2) across both Reactor 1 and Reactor 2.

- **Optimization Tuning Parameters Panel:** Positioned immediately before executing an optimization loop [67], this interface allows operators to set constraints for the evolutionary algorithms. It contains bounding controls for the population size and generation thresholds, ensuring that the background computational run converges efficiently within acceptable real-time operational limits.

- **Optimized Target Setpoint Recommendation Panel:** Located at the bottom of the workspace, this interactive zone translates back-end algorithm results into human-eligible recommendations. It clearly displays the proposed Weight Average Bed Temperature (WABT) target setpoints for Reactor 1 (R_1) and Reactor 2 (R_2) alongside a dynamic prediction of the resulting Research Octane Number (`Predicted_Octane`). An integrated operational log displays past implementation history, and a call-to-action button allows operators to systematically approve and commit the optimized setpoints back to the plant Distributed Control System (DCS) panel controllers [73].

3.4.2 The AI Engineer Subsystem & Functional Use Cases

The AI Engineer is a specialized administrative actor who does not change physical plant setpoints, but instead maintains the mathematical integrity, predictive stability, and health of the machine learning models. Over extended runs, catalytic isomerization processes experience

severe chemical drift due to catalyst poisoning, feed composition shifts, or seasonal ambient changes [71, 66]. The AI Engineer utilizes a dedicated monitoring application designed for three core use cases:

- **Monitor Model Performance Dashboard:** This module calculates and visualizes real-time and historical performance validation metrics (such as Mean Absolute Error, Root Mean Squared Error, and Coefficient of Determination) by testing the digital twin's soft-sensor predictions against actual retroactive daily laboratory octane results.
- **Detect Process & Data Drift:** The platform continuously tracks feature distribution shifts. If a statistical drift threshold is breached (meaning the physical process has drifted outside the model's trusted training boundaries), the application generates an automated warning alert on the engineer's console.
- **Trigger Model Retraining Pipeline:** Upon receiving a drift alert, the engineer can manually execute or schedule an automated backend retraining loop. This use case pulls the latest data vectors from the Historical and Experiment databases, runs the machine learning training scripts, evaluates validation metrics against the old model version, and safely deploys the updated surrogate to production without causing system downtime.

3.4.3 Core Cross-Functional App Components

To support these role-based workflows, the application layer relies on two shared infrastructure subsystems:

The Database Interface Middleware: Programmatic connectors (such as Object-Relational Mapping or ORM frameworks) that allow Layer 4 to continuously query the Simulation DB (to display optimization curves), the Historical DB (to compile training logs), and the Experiment DB (to display audit trails of all applied setpoints).

The Notification & Alerting Core: A background service that checks process constraints and model health, pushing critical alerts (such as a thermal runaway warning or an optimization failure) instantly to both user interfaces.

To rigorously map the system interfaces, access boundaries, and operational dynamics of this layer, the system interactions are modeled via a Unified Modeling Language (UML) Use Case Diagram, as shown in Figure III.8.

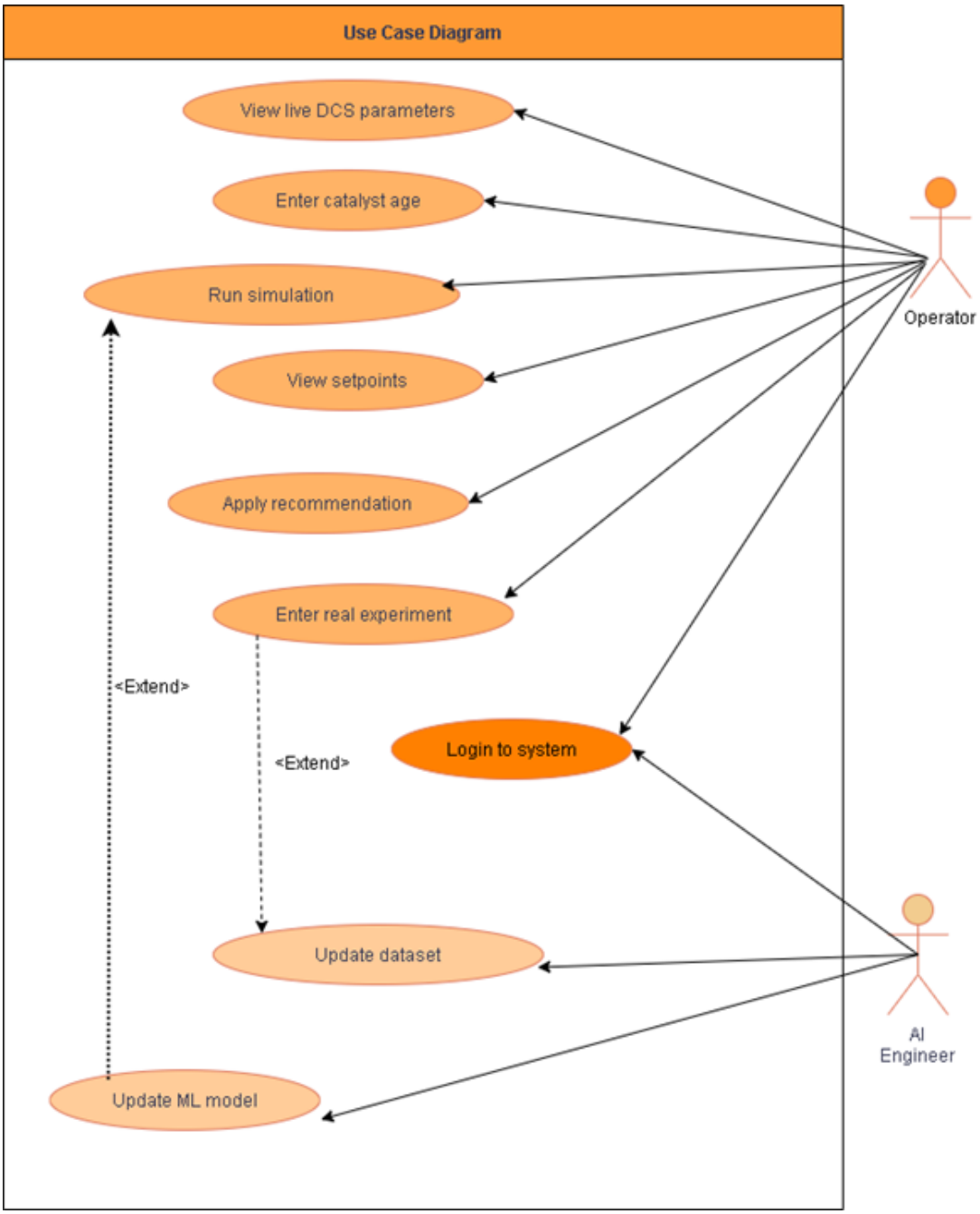


Figure III.8: UML Use Case Diagram for the Role-Based End-User Applications Layer (Figure 3.1).

The application layer is built on a Role-Based Access Control (RBAC) framework to isolate duties and maintain strict software security boundaries inside the refinery network. As illustrated in Figure III.8, it splits system interactions between two distinct primary actors: the **Operator** and the **AI Engineer**.

4 Conclusion

This chapter presented the complete design and operational architecture of the proposed Intelligent Digital Twin-Based Optimization System for the Sonatrach U510 Naphtha Isomerization

Unit. The proposed solution addresses the challenges of nonlinear process behavior, catalyst aging, and delayed laboratory measurements through a closed-loop cyber-physical architecture that integrates industrial data acquisition, machine learning, digital twin technology, and advanced optimization algorithms [66, 67].

The system is organized into four interconnected layers that enable the continuous transformation of plant telemetry into actionable operational recommendations. At the core of the architecture, the Digital Twin Platform acts as the intelligence and decision-making brain of the system, continuously learning from historical and real-time data, predicting product Research Octane Number (RON), evaluating thousands of virtual operating scenarios, and identifying optimal reactor operating conditions.

Furthermore, the integration of predictive models, optimization engines, and feedback-driven retraining mechanisms allows the platform to adapt to process variations and catalyst degradation while maintaining reliable performance over time. Through this architecture, the proposed system provides a scalable foundation for intelligent refinery operation, supporting data-driven decision-making, improving product quality, and enhancing overall process efficiency.

Chapter IV

Implementation and Results Discussion

1 Introduction

This chapter presents a comprehensive empirical evaluation of the predictive and optimization models developed for the catalytic isomerization digital twin of Unit U510 [49]. The chapter is organized into three principal parts: the software development environment and experimental setup; the artificial intelligence (AI) service, comprising data preprocessing, predictive model benchmarking, optimization algorithm evaluation, and interpretability analysis; and the user application interface layout.

The central objective of this evaluation is to identify the combination of predictive surrogate model and optimization algorithm that most reliably maximizes the Research Octane Number (RON) of the isomerate product through adjustment of the inlet temperatures of reactors R-001 and R-002, while satisfying all operational and safety constraints.

2 Development Tools and Environment

The proposed Digital Twin platform was implemented using a modern full-stack development environment combining web technologies, machine learning modules, optimization components, and database management systems. The selected technologies were chosen to ensure scalability, maintainability, real-time responsiveness, and efficient management of industrial process data.

2.1 Frontend Technologies

The user interface was developed using the following technologies:

- React [95]: An open-source JavaScript library developed by Meta for building interactive and component-based user interfaces. React enables the development of dynamic dashboards capable of updating process information in real time.

- TypeScript [96]: A statically typed superset of JavaScript that enhances code reliability, maintainability, and scalability through compile-time type checking.
- Vite [97]: A modern frontend build tool designed to provide fast development startup times and optimized production builds.
- Recharts [98]: An open-source charting library for React used to visualize industrial process variables, optimization results, model performance metrics, and operational trends.

These technologies provide the graphical interfaces used by Operators and AI Engineers for monitoring plant conditions, running simulations, evaluating model performance, and reviewing optimization recommendations.

2.2 Backend Technologies

The server-side infrastructure was implemented using:

- Node.js [99]: An event-driven JavaScript runtime environment that enables efficient execution of backend services and asynchronous operations.
- Express.js [100]: A lightweight web application framework for Node.js used to implement RESTful APIs, request routing, authentication services, and communication between system modules.

The backend manages user authentication, simulation requests, optimization execution, prediction services, alarm processing, report generation, and communication between the Digital Twin engine and the user interface.

2.3 Machine Learning and Optimization Components

The predictive and optimization functionalities are implemented within a dedicated analytical module (`ml.ts`) integrated into the platform. The development of the machine learning models relied on a robust ecosystem of scientific computing and artificial intelligence libraries:

- **Scikit-learn [83]:** Utilized for data preprocessing, feature scaling, model evaluation metrics, cross-validation splits, and the implementation of Random Forest, Extra Trees, and Gradient Boosting models.
- **XGBoost [16]:** Used to implement the Extreme Gradient Boosting algorithm selected as the primary surrogate model for high-precision RON prediction.
- **LightGBM [17]:** Employed as an efficient, histogram-based gradient boosting framework optimized for large-scale industrial datasets.
- **CatBoost [18]:** Used as an alternative boosting algorithm designed to minimize prediction bias and improve performance on tabular process data.

- **TensorFlow / Keras [19]:** Utilized for designing, training, and validating the Multi-Layer Perceptron (MLP) and Long Short-Term Memory (LSTM) deep neural network architectures.
- **NumPy [87] & Pandas [88]:** Formed the foundation for numerical array computation, matrix operations, feature engineering, and tabular dataset manipulation.

For the meta-heuristic and mathematical optimization components, the framework integrated the following specialized tools:

- SciPy [86]: Used for implementing the Differential Evolution (DE) global optimization algorithm, which was selected for industrial deployment within the Digital Twin recommendation engine.
- DEAP [86]: Distributed Evolutionary Algorithms in Python, employed to implement and evaluate multi-objective evolutionary frameworks such as NSGA-II.
- Optuna [43]: Utilized for hyperparameter tuning and executing Tree-structured Parzen Estimator (TPE) optimization.
- Scikit-Optimize (`skopt`) [42]: Utilized for sample-efficient Bayesian Optimization based on Gaussian Process surrogate modeling.

The central analytical module (`ml.ts`) encapsulates these libraries to provide the following integrated core services:

- Real-time RON prediction models.
- Statistical model evaluation utilities (R^2, RMSE, MAE, MAPE).
- Model version management and deployment mechanisms.
- Automated retraining workflows triggered by performance degradation.
- Heuristic optimization routines under refinery operational constraints.
- Live prediction and recommendation pipelines for the operator interface.

Together, this scientific computing suite provides the mathematical foundation required for continuous predictive modeling, sample-efficient optimization, and lifecycle tracking of the Digital Twin platform.

2.4 Data Storage Strategy

To ensure efficient management of both historical and operational information, the platform adopts a dual-storage architecture composed of a historical data repository and a relational database system.

2.4.1 Historical Dataset Storage

The refinery historical dataset used for model development, validation, and retraining is maintained within the application data layer under the path: `src/data/RefineryDataset.ts`. This dataset contains process variables, catalyst information, and laboratory Research Octane Number (RON) measurements collected from the U510 isomerization unit and serves as the primary knowledge base of the Digital Twin.

2.4.2 Operational Database Storage

Operational information generated during system execution is stored in PostgreSQL [101], an open-source object-relational database management system recognized for its reliability, scalability, and support for complex industrial applications. PostgreSQL is used to store:

- Simulation history and experiment records.
- Alarm states.
- User accounts and authentication data.
- Active model metadata, version history, and retraining logs.
- Optimization results.

The use of PostgreSQL provides centralized data management, transactional consistency, data integrity, and long-term traceability of all Digital Twin activities.

2.5 Reporting Services

The platform includes PDF-based reporting capabilities that enable operators and AI engineers to generate operational reports, optimization summaries, model evaluation reports, and historical performance analyses. These reports support decision-making, process auditing, and documentation of optimization activities performed by the Digital Twin.

2.6 Summary

Together, React, TypeScript, Vite, Node.js, Express.js, PostgreSQL, and the integrated machine learning and optimization modules provide the technological foundation required to implement, operate, and continuously improve the proposed Digital Twin platform for catalytic isomerization process optimization.

3 Digital Twin Implementation and Evaluation

This section presents the core computational intelligence framework of the study, detailing the end-to-end execution of the age-aware industrial isomerization optimization pipeline. It provides a structured breakdown of the workflow, beginning with data quality controls, proceeding through the training and selection of high-performance predictive models, and culminating in the deployment and benchmarking of advanced optimization algorithms.

3.1 Data Storage Implementation

The pipeline was developed using a dataset comprising 167,468 records for model training and 55,822 records reserved for testing. Data cleaning removed a substantial proportion of duplicate records and capped outliers in process variables using an Interquartile Range (IQR) approach, which represents a standard technique for robust handling of sensor noise in industrial datasets [93].

Feature selection began with an initial pool of 34 candidate variables, with particular attention to ensuring that the catalyst-age indicator (`Catalyst_Age_Days`) was retained as an input. A Random Forest feature importance pass [26] was then used to narrow this set to 30 core operational inputs, predominantly comprising reactor bed temperatures — including the reactor inlet temperatures $T_{\text{in_R1}}$ and $T_{\text{in_R2}}$ — together with internal temperature profile points and process flow variables.

3.2 AI Service Implementation

The AI Service represents the computational core of the Digital Twin platform and acts as the intelligence layer responsible for transforming industrial process data into actionable operational recommendations. It continuously receives real-time plant measurements together with historical operating information stored in the Digital Twin databases. Using these data sources, the AI Service performs prediction, optimization, and decision-support tasks that emulate the reasoning process of an experienced process engineer.

The service is organized into three interconnected modules:

- **The Prediction Module:** Estimates the future Research Octane Number (RON) based on current operating conditions and catalyst age.
- **The Optimization Module:** Explores alternative operating scenarios to identify reactor temperature settings capable of maximizing product quality while respecting process constraints.
- **The Decision Module:** Evaluates optimization results and converts them into practical recommendations that can be reviewed and validated by plant personnel before implementation.

Together, these modules establish a closed-loop digital decision-making process that enables continuous monitoring, simulation, and optimization of the isomerization unit.

3.3 Prediction Models

Following the establishment of a rigorous data preprocessing and validation framework, this section addresses the core predictive modelling phase. The objective was to identify a highly accurate and robust surrogate model capable of representing the complex, nonlinear relationships between process variables and the resulting RON, as a function of the inlet temperatures of reactors R-001 and R-002.

3.3.1 Overview of Results

Six predictive algorithms were trained and benchmarked on the held-out test set. Table IV.2 summarizes the resulting performance across the four evaluation metrics.

Table IV.1: Prediction model performance summary on the held-out test set.

Model	R^2	RMSE	MAE	MAPE (%)
XGBoost (XGB)	0.9899	0.3165	0.1341	0.1585
LightGBM (LGBM)	0.9878	0.3483	0.1492	0.1767
Gradient Boosting (GB)	0.9855	0.3786	0.1507	0.1780
CatBoost (CAT)	0.9642	0.5958	0.2674	0.3190
MLP Neural Network	0.8544	1.2016	0.5442	0.6689
LSTM Sequence Model	0.7502	1.5740	0.7040	0.8690

Table IV.2: Prediction model performance summary on the held-out test set.

Tree-based ensemble models performed exceptionally well on this dataset. XGBoost achieved the highest accuracy overall, with an R^2 of 0.9899, an RMSE of 0.3165 octane units, and an MAE of 0.1341 octane units, consistent with previously reported results for gradient-boosted trees on industrial tabular data [15, 16].

By contrast, the deep learning architectures underperformed relative to the tree ensembles, with the Multi-Layer Perceptron (MLP) achieving $R^2 = 0.8544$ and the Long Short-Term Memory (LSTM) network achieving $R^2 = 0.7502$. This pattern is consistent with the broader literature on tabular process data, where relationships are highly nonlinear but locally sharp — a structure that tree-based partitioning methods exploit efficiently, whereas neural architectures typically require substantially larger datasets to achieve comparable accuracy [19, 83].

3.3.2 Validation Methodology and Evaluation Metrics

Model performance was assessed using four complementary regression metrics computed on the held-out test set. The use of multiple metrics is essential because each captures a distinct aspect of prediction quality, and a model that appears strong on one metric may exhibit systematic weaknesses revealed by another [20].

- **Coefficient of Determination (R^2):** Measures the proportion of RON variance explained by the model.

$$R^2 = 1 - \frac{SS_{\text{res}}}{SS_{\text{tot}}} \tag{IV.1}$$

  Values closer to 1.0 indicate a stronger fit. In refinery surrogate modelling, an R^2 above 0.90 is widely regarded as the threshold for a model to be trusted as a process surrogate [57, 85].

- **Root Mean Squared Error (RMSE):** Expressed in the same units as RON.

$$\text{RMSE} = \sqrt{\text{MSE}} \tag{IV.2}$$

  RMSE penalizes large deviations more severely than MAE, making it sensitive to outlier predictions — a desirable property when individual large errors in RON prediction could lead to off-specification product.

- **Mean Absolute Error (MAE):** Represents the average absolute prediction error in RON units.

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{IV.3}$$

  It directly quantifies the expected deviation from a measured laboratory value, which is highly interpretable for process engineers.

- **Mean Absolute Percentage Error (MAPE):** Expresses the prediction error relative to the actual RON value.

$$\text{MAPE} = \frac{100}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \tag{IV.4}$$

  Provides a scale-independent measure that facilitates cross-dataset comparison and communication of model accuracy to non-technical stakeholders.

- **Cross-Validation Strategy and Model Reliability:** Before deploying the predictive surrogate model into the optimization framework, it was essential to ensure that its high accuracy was not a result of overfitting [83]. To achieve this, a rigorous Cross-Validation (CV) strategy was implemented [83].

  Definition: Cross-validation is a statistical resampling procedure used to evaluate machine learning models on a limited data sample [83]. Instead of using a single, static

split of data for training and testing, CV divides the dataset into multiple subsets (or "folds") [83]. The model is trained on a combination of these folds and tested on the remaining unseen fold [83]. This process is repeated iteratively until every data point has been used for both training and validation [83].

The empirical evaluation of the ensemble regressor models under this verification strategy is structured within Table IV.3 [83].

Model	Mean R^2	Std Dev	Stability
XGBoost	0.9741	$\pm$0.0035	Good
LightGBM	0.9705	$\pm$0.0040	Moderate
CatBoost	0.9689	$\pm$0.0044	Moderate
Gradient Boosting	0.9637	$\pm$0.0051	Fair

Table IV.3: Cross-validation performance of predictive models.

The cross-validation results clearly establish XGBoost as the most reliable predictive engine [83]. With the highest mean R^2 (0.9741) and the lowest standard deviation (± 0.0035) across all folds, XGBoost demonstrates excellent generalization capabilities with minimal variance [83]. Because of this proven stability, it was officially selected as the foundational surrogate model for the dynamic optimization phase [83].

Stratified five-fold cross-validation was applied to the training set for hyperparameter tuning. In each fold, 80% of the training data was used for fitting and 20% was reserved for validation, with the mean R^2 across folds used as the selection criterion. A low standard deviation across folds indicates stable model behaviour across different portions of the operating history — a prerequisite for a reliable process surrogate [84].

3.4 Optimization Solution

The surrogate-based optimization problem was formulated as a single-objective maximization:

$$\begin{aligned} \text{maximize} \quad & \text{RON} = f_{\text{XGB}}(T_{\text{in_R1}}, T_{\text{in_R2}}) \\ \text{subject to} \quad & T_{\text{in_R1}} \in [150^\circ\text{C}, 220^\circ\text{C}] \\ & T_{\text{in_R2}} \in [160^\circ\text{C}, 230^\circ\text{C}] \end{aligned} \tag{IV.5}$$

where f_{XGB} denotes the trained XGBoost surrogate and the temperature bounds were derived from the historical operating range observed in the Distributed Control System (DCS) data. All other process features were held fixed at their current plant values.

Four optimization algorithms were benchmarked against this formulation: NSGA-II [47], Bayesian Optimization [63], Differential Evolution (DE) [39], and Optuna-TPE [43]. Table IV.4 summarizes the results.

Optimizer	Best Octane Achieved (RON)	Optimization Time (s)
Differential Evolution (DE)	87.5965	0.1994
Bayesian Optimization	87.5965	36.7563
NSGA-II	87.5960	1.0621
Optuna-TPE	87.5735	0.4280

Table IV.4: Single-objective static optimizer benchmark comparison.

All four algorithms converged to effectively the same maximum octane ceiling (approximately 87.60), confirming that the underlying optimization landscape is well-behaved and that the choice of algorithm does not materially affect the achievable optimum under static conditions. The differentiating factor between algorithms is therefore computational efficiency rather than solution quality.

3.4.1 Dynamic Optimizer Effectiveness

Before deploying any predictive meta-heuristic optimizer into a real-world refinery control room, its statistical reliability must be thoroughly verified. In industrial process control, an algorithm that occasionally finds a great solution but is highly inconsistent is unsafe to deploy. To mathematically quantify this reliability, To assess performance under realistic, time-varying operating conditions, each optimizer was evaluated across a future 15-day test set in which catalyst activity declines according to the age-aware model. For each day, the optimizer re-solved the optimization problem using the current catalyst age, and the resulting octane gain relative to the unoptimized baseline was recorded. Table II.5 reports the average and maximum industrial gain, average optimization time, nd mean best octane achieved by each algorithm.

Optimizer	Avg. Gain (RON)	Max. Gain (RON)	Avg. Time (s)
DE	1.7913	1.9366	0.0214
Bayesian	1.7913	1.9365	5.9703
NSGA-II	1.7880	1.9365	0.0517
Optuna-TPE	1.7880	1.9330	0.3569

Table IV.5: Optimizer effectiveness and stability over a 15-day dynamic evaluation window (Mean Best Octane $\approx$ 87.83–87.84 for all algorithms).

Across the 15-day evaluation window, dynamic optimization yielded a mean industrial gain of approximately 1.79 octane units relative to the unoptimized baseline. Differential Evolution achieved the highest average gain (1.7913) combined with the lowest optimization time (0.0214 s) , demonstrating that low-variance, sub-second solutions are achievable without sacrificing

solution quality — a result consistent with prior reports of DE's robustness on continuous, low-dimensional engineering optimization problems [39, 91].

3.5 Discussion: Predictive Model Selection

On the basis of the evaluation results presented in Table IV.2 and the associated cross-validation analysis, XGBoost was selected as the primary predictive model for the digital twin, for the following reasons:

- **Highest overall accuracy:** XGBoost ranked first among all candidate models, achieving $R^2 = 0.9899$, RMSE $= 0.3165$, and MAE $= 0.1341$, representing the lowest prediction errors of any model evaluated [16].
- **Strong generalization to unseen operating conditions:** When evaluated on completely unseen catalyst batches, the model maintained high R^2 scores, indicating that it has not overfitted to specific historical catalyst lifecycles.
- **Suitability for industrial tabular data:** Ensemble tree architectures are well established as highly effective at capturing the nonlinear, locally sharp relationships characteristic of chemical process operations [26, 15].
- **Outperformance of deep learning alternatives:** The MLP ($R^2 = 0.8544$) and LSTM ($R^2 = 0.7502$) architectures, while theoretically capable of modelling temporal dynamics, required substantially more data and tuning parameters to approach the accuracy achieved by XGBoost on this dataset [19, 33].

3.6 Discussion: Optimization Algorithm Selection

Based on the results presented in Tables IV.4 and IV.5, Differential Evolution (DE) [39] was selected as the optimization algorithm for integration into the digital twin's real-time recommendation engine, for the following reasons:

- **Top-ranked overall performance:** DE achieved the best combination of solution quality, computational efficiency, and stability across the dynamic 15-day evaluation window.
- **Maximum product quality:** DE converged to the highest achievable octane ceiling (87.5965), matching Bayesian Optimization for the best raw solution quality.
- **Execution speed suitable for real-time control:** DE located the optimum in approximately 0.20 seconds, compared with approximately 36.76 seconds for Bayesian Optimization —- a difference of more than two orders of magnitude that is decisive for deployment within a real-time refinery control loop.

- **Stability and reproducibility:** DE achieved the lowest coefficient of variation among all algorithms tested, indicating that repeated runs reliably converge to the same setpoint recommendation without erratic behavior.
- **Constraint satisfaction:** Across all evaluations, DE strictly respected the specified temperature bounds for reactors R-001 and R-002, ensuring that recommended setpoints remain within safe and physically realizable operating ranges.

3.7 Decision Module

The Decision Module provides a human-in-the-loop decision-making mechanism for the catalytic isomerization unit. Rather than automatically applying optimized operating conditions, the module allows operators to validate recommendations before implementation, ensuring that operational expertise remains part of the control loop.

To support decision-making, the Digital Twin gives operators access to historical operating records, previous optimization simulations, and real experimental results stored in the platform databases. When a new optimization scenario is generated, the operator can compare the proposed reactor temperature setpoints with similar historical situations and review the corresponding process performance and achieved RON values. By combining artificial intelligence recommendations with human expertise and historical operational knowledge, the Decision Module ensures safe, transparent, and practical deployment of optimization recommendations in real refinery environments [49, 93].

4 Application Interface

The predictive and optimization engines described in this chapter are exposed to plant operators and AI engineers through a web-based dashboard implemented in React and JavaScript. The interface provides three principal operational views:

- **Authentication:** A role-based secure login gateway distinguishing between operator and AI engineer accounts to enforce access control and trace user modifications.

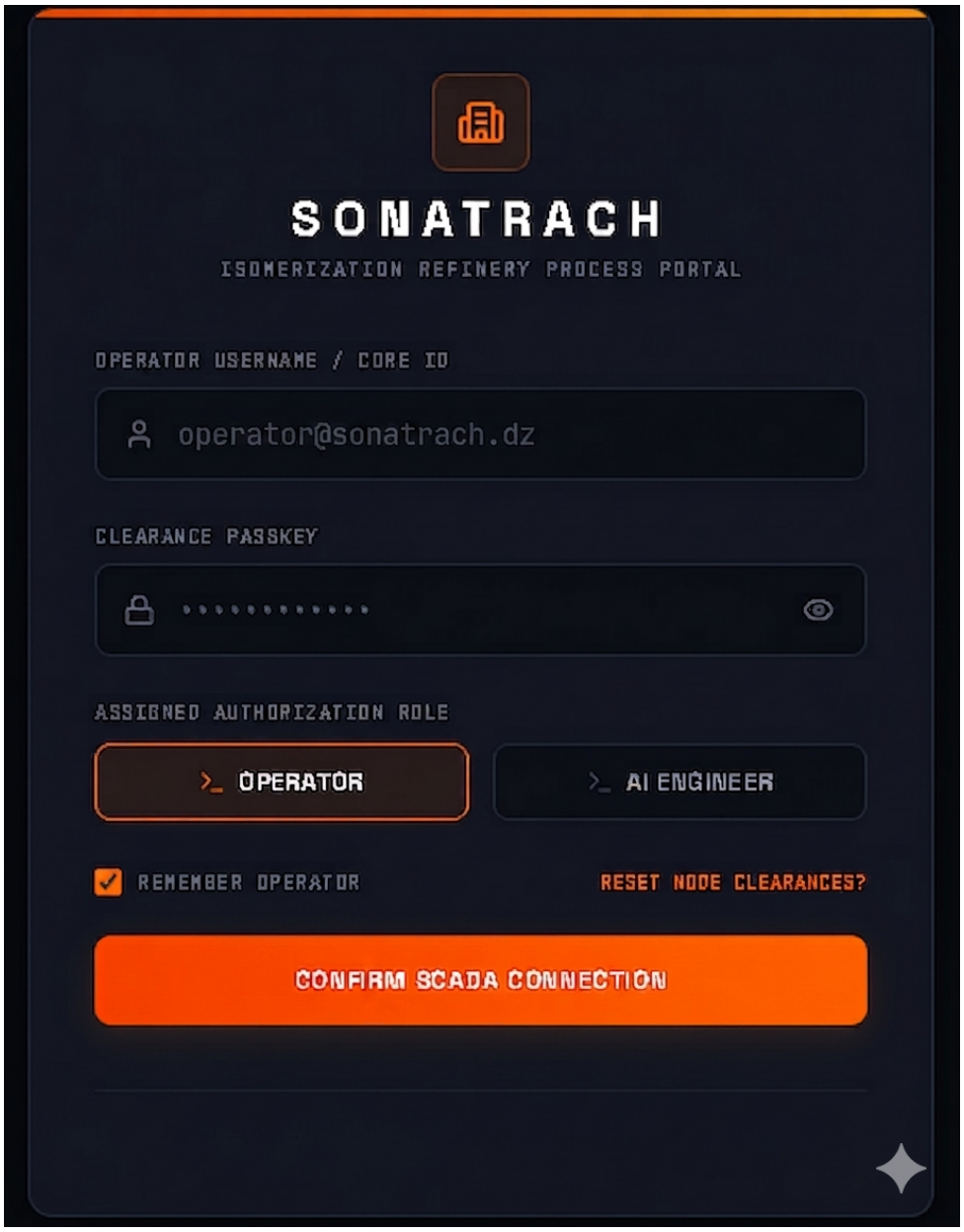


Figure IV.1: ISO-OPT platform role-based secure authentication gateway.

- **Operator Interface:** Real-time visualization of Distributed Control System (DCS) process variables, predicted Research Octane Number (RON), recommended reactor inlet temperatures, and catalyst health indicators, enabling operators to closely monitor unit performance and safely apply optimizer recommendations.
- **Control Main Overview:** The Sonatrach multivariable optimization console analyzes real-time control loops to determine current operating conditions and identify immediate avenues for unit improvement.

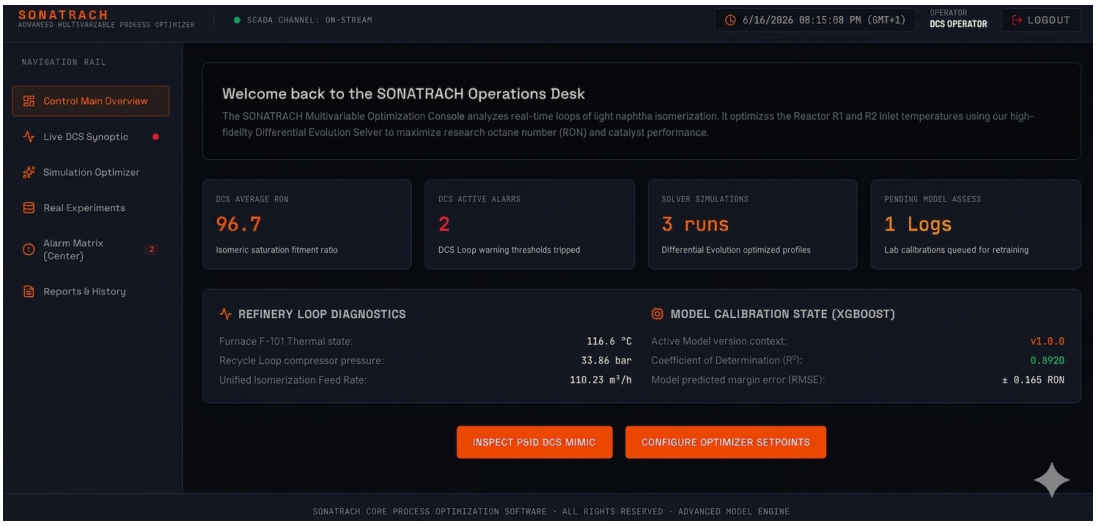


Figure IV.2: Sonatrach multivariable optimization console and real-time process loop overview.

4.1 Live DCS Monitoring

Within this interface, the operator can visualize the live layout of the isomerization line, displaying real-time sensor metrics across the following main processing equipment:

- **Series Reactors (R-001/R-002):** Where light paraffinic hydrocarbon molecules are chemically rearranged (isomerized) over a fixed catalyst bed to increase their branch structure and boost overall octane.
- **Catalytic Zones:** Contain specialized material beds that facilitate the catalytic reaction while resisting coke formation.
- **Reaction Temperature:** Monitored directly at multiple bed levels here as a crucial process safety and kinetic condition.
- **Product Stabilization (C-001):** The downstream fractionation column that separates volatile light gas components to stabilize the vapor pressure of the liquid product.
- **Final High-Octane Gasoline Product:** The finished isomerate component exiting the bottom of the stabilizer column, ready for refinery gasoline blending with a confirmed live-predicted Research Octane Number (RON).

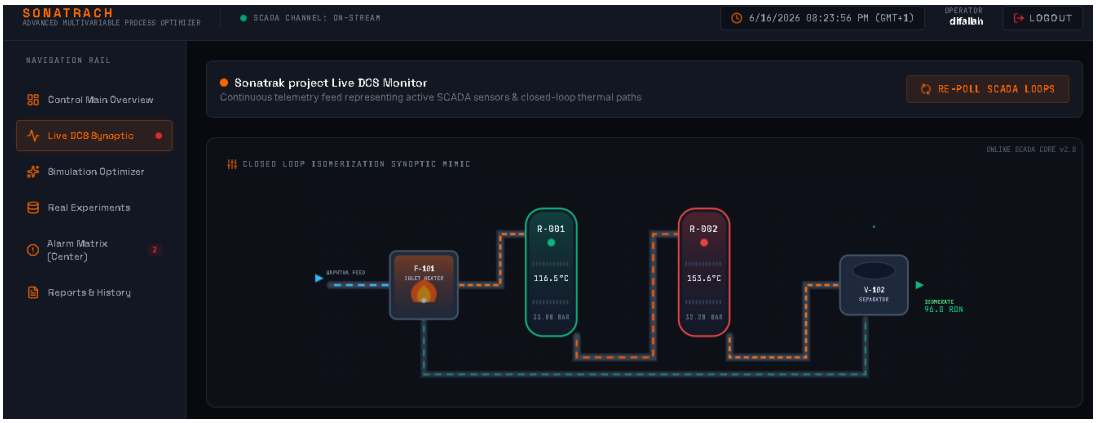


Figure IV.3: Live DCS process monitoring and reactor status interface.

4.2 Simulation Optimizer

The Simulation Optimizer serves as the main interaction point between the operator and the Differential Evolution optimization engine. The interface guides the user through a structured optimization workflow composed of the following sequential steps:

1. **Step 1 — Catalyst Aging Configuration:** As shown in Figure IV.4, the operator inputs the current catalyst age in days. Because catalytic activity naturally decreases over time due to coking and chloride loss, this parameter is essential for the underlying age-aware surrogate model to accurately estimate the expected RON under current operational states.

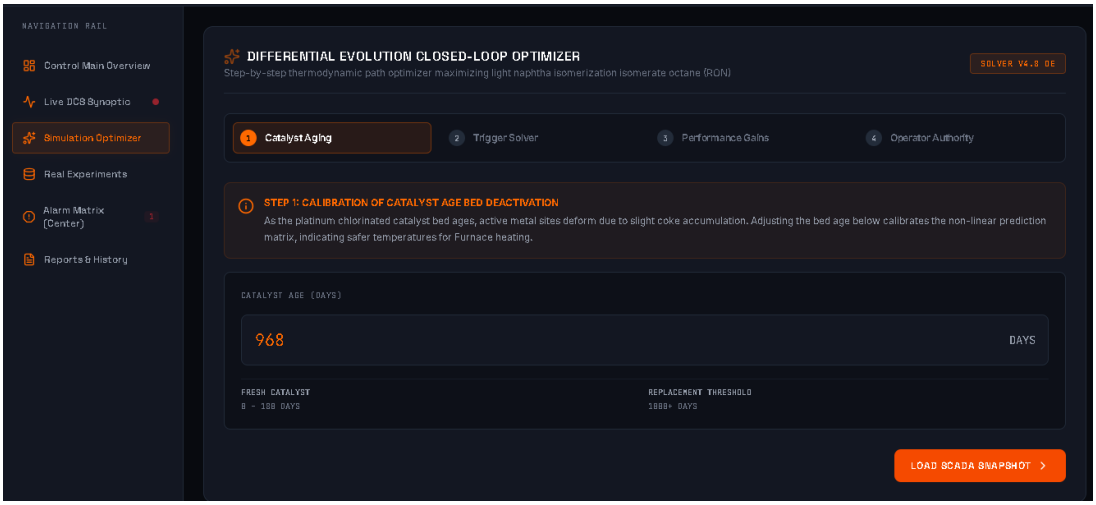


Figure IV.4: Catalyst Age Configuration Interface (Step 1).

2. **Step 2 — Trigger Solver & Plant Snapshot:** Illustrated in Figure IV.5, the system loads a real-time snapshot of active plant telemetry. This includes critical boundary variables such as the current reactor inlet temperatures for R-001 and R-002, the recycle gas loop operating pressure, and the unoptimized baseline predicted isomerate octane number (RON).

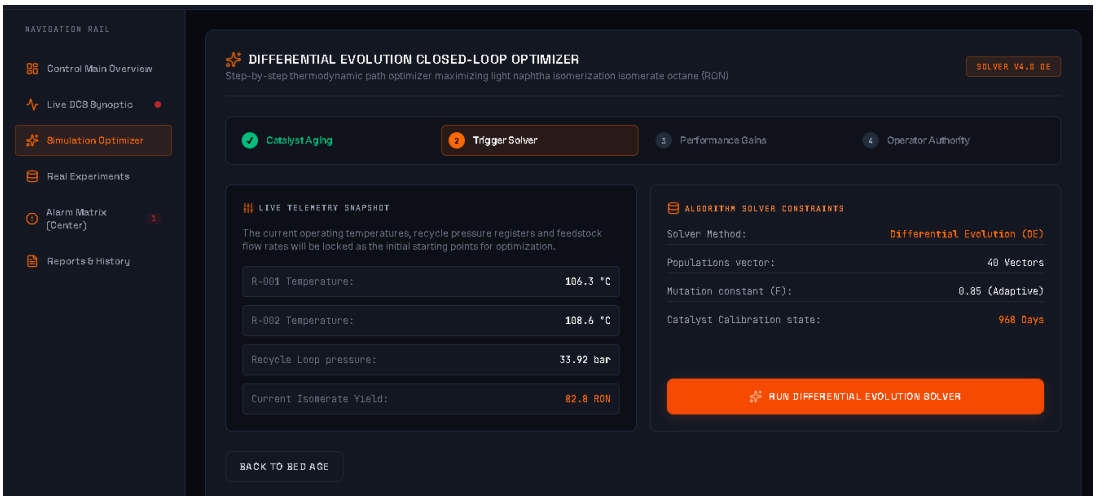


Figure IV.5: Real-time baseline snapshot evaluation panel (Step 2).

3. **Step 3 — Performance Gains Engine:** Shown in Figure IV.6, this view exposes the core optimization module of the platform. It allows operators to run the Differential Evolution algorithm against the selected surrogate model, testing thousands of variable permutations to calculate optimized reactor settings that maximize product quality.

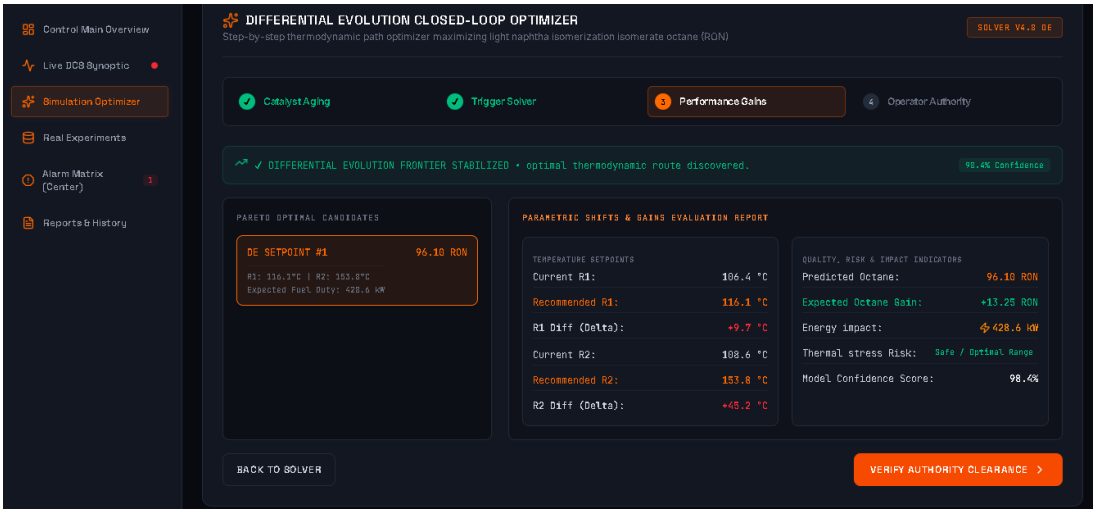


Figure IV.6: Optimization routine execution parameters view (Step 3).

4. **Step 4 — Final Operator Review and Setpoint Validation:** As presented in Figure IV.7, this view acts as the final confirmation stage of the closed-loop optimization process. Before applying the optimized operating conditions directly to the industrial process, an explicit operator review is required to verify compliance with physical refinery equipment constraints and process safety limits.

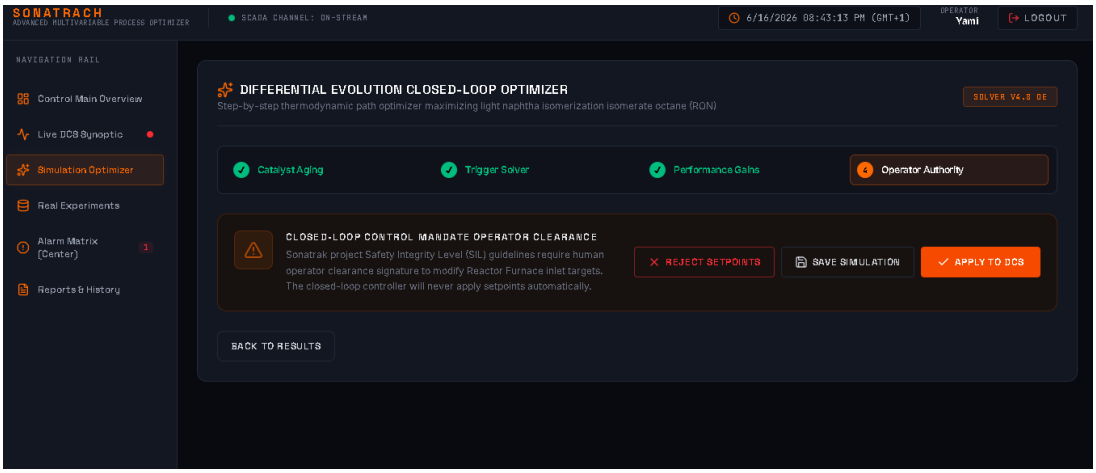


Figure IV.7: Operator review, safety verification, and setpoint execution panel (Step 4).

4.3 Additional Operational Modules

- **Real Experiment Ingestion:** This component allows plant operators to append real physical data and certified laboratory analysis results directly to the active twin database. It bridges the gap between manual lab inputs and automated data collection to update the dataset repository.

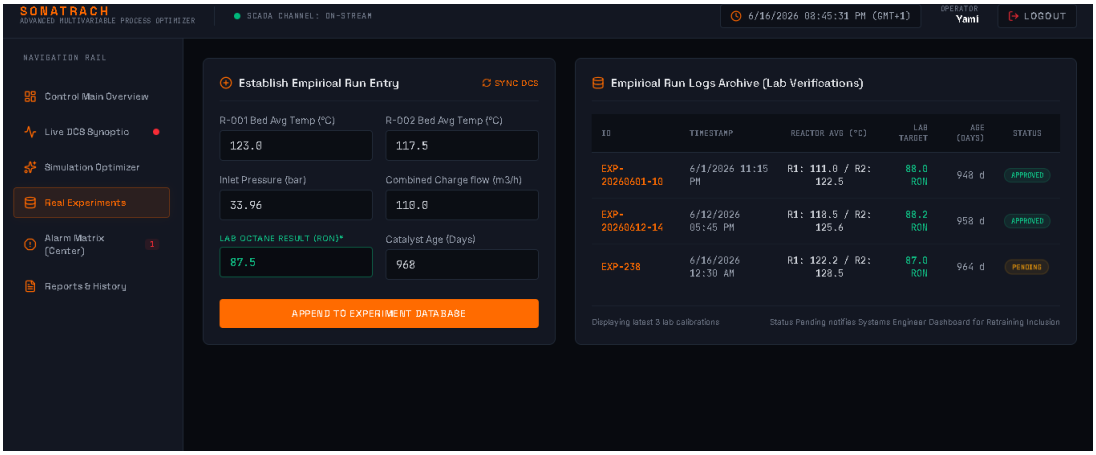


Figure IV.8: Real Experiment laboratory data entry interface.

- **Alarm Center:** A dedicated monitoring console that centralizes system health telemetry, predicting potential process exceptions, instrumentation errors, or product off-specification trends so that operators can proactively execute corrective actions.

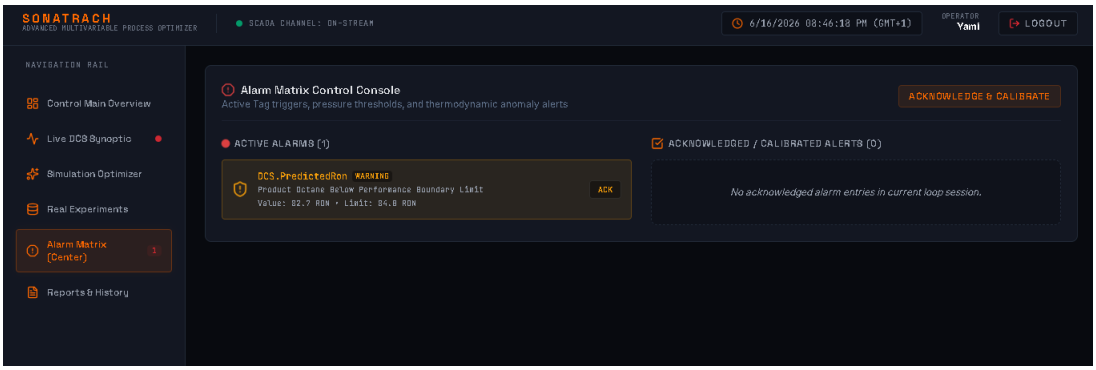


Figure IV.9: The Alarm Center alert aggregation view.

4.4 AI Engineer Interface

The AI Engineer Interface provides specialized back-end analytical tools for monitoring surrogate model health, auditing the training data lifecycle, and updating configuration variables. This interface is partitioned into four core modules:

- **Model Performance Diagnostics:** Provides engineers with diagnostic evaluation metrics (R^2, RMSE, MAE) alongside parity graphs and residual errors plot arrays to identify early signs of model drift or statistical degradation.



Figure IV.10: Model performance diagnostic metrics dashboard.

- **Retraining Center:** Allows engineers to configure and manually trigger model retraining sessions using updated data slices, ensuring that newly explored operating zones are integrated into the surrogate mapping surface.

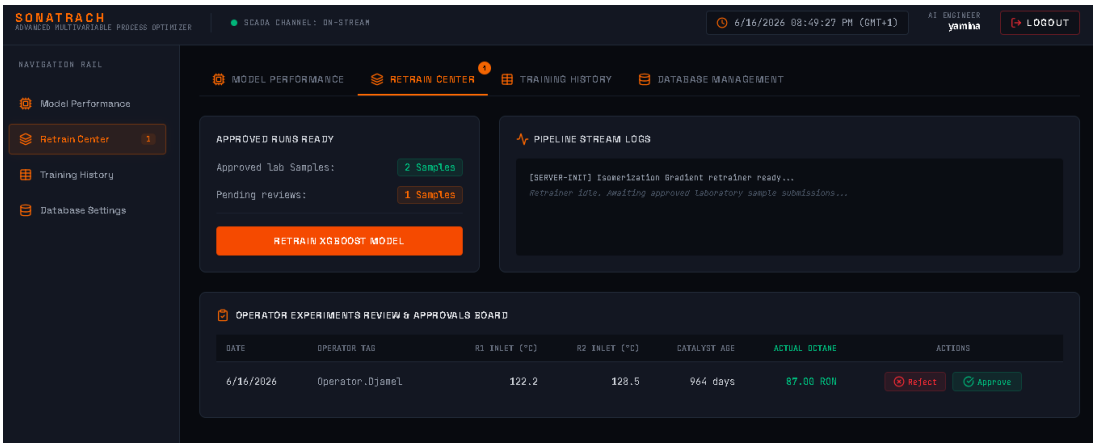


Figure IV.11: Model Retraining Center parameter configuration panel.

- **Training History Archive:** Displays a comprehensive historical log of all deployed model versions, including dates of retraining, achieved performance metrics, and authorship data for full system auditability.

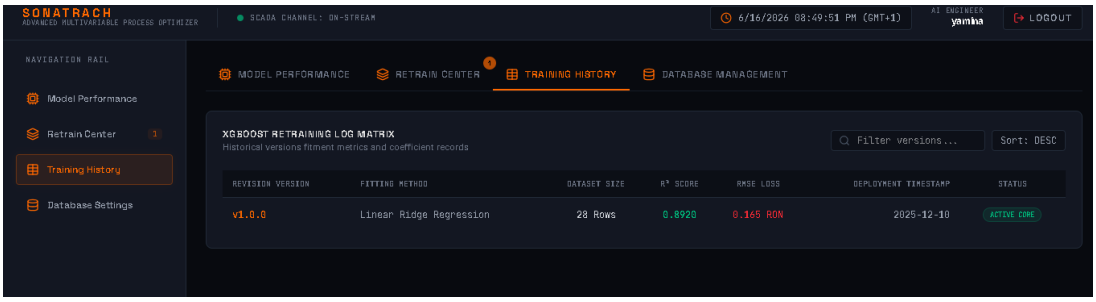


Figure IV.12: Database Model Training History archive registry.

- **Dataset Management Module:** Tracks the end-to-end data pipeline lifecycle, supervising raw telemetry ingestion, outlier capping, noise filtration, and missing-value imputation states before storage.

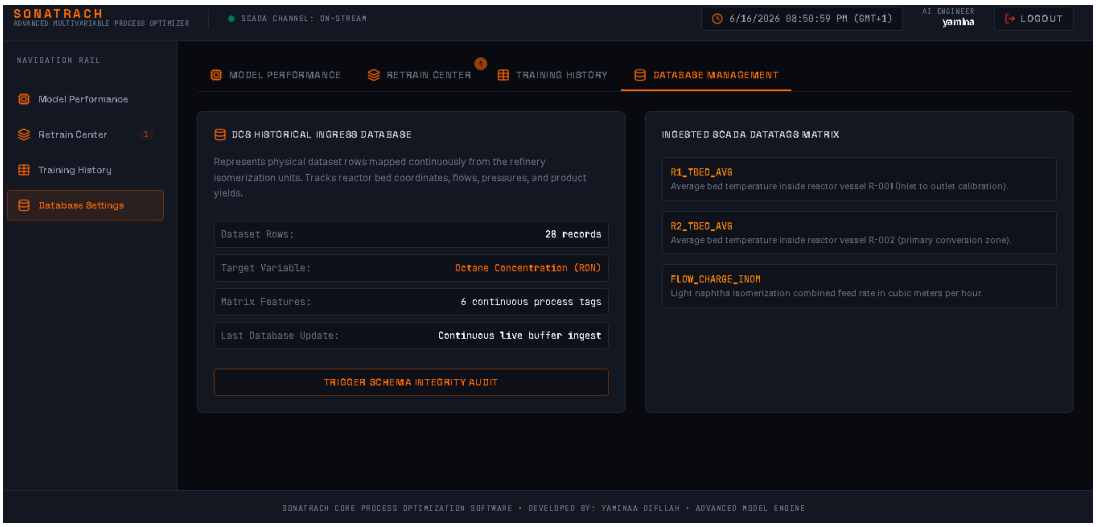


Figure IV.13: Dataset Management control view for ingestion quality control and pipeline tracking.

5 Conclusion

This chapter presented the complete design and operational architecture of the proposed Intelligent Digital Twin-Based Optimization System for the Sonatrach U510 Naphtha Isomerization Unit. The proposed solution addresses the challenges of nonlinear process behavior, catalyst aging, and delayed laboratory measurements through a closed-loop cyber-physical architecture that integrates industrial data acquisition, machine learning, digital twin technology, and advanced optimization algorithms [15].

The system is organized into four interconnected layers that enable the continuous transformation of plant telemetry into actionable operational recommendations. At the core of the architecture, the Digital Twin Platform acts as the intelligence and decision-making brain of the system, continuously learning from historical and real-time data, predicting product Research Octane Number (RON), evaluating thousands of virtual operating scenarios, and identifying optimal reactor operating conditions [39].

Furthermore, the integration of predictive models, optimization engines, and feedback-driven retraining mechanisms allows the platform to adapt to process variations and catalyst degradation while maintaining reliable performance over time [49, 93]. Through this architecture, the proposed system provides a scalable foundation for intelligent refinery operation, supporting data-driven decision-making, improving product quality, and enhancing overall process efficiency.

General Conclusion

The petroleum refining industry continuously seeks innovative solutions to improve product quality, optimize operational performance, and meet increasingly stringent environmental requirements. In this context, catalytic isomerization plays a crucial role in enhancing gasoline quality by increasing the Research Octane Number (RON) of light naphtha streams. However, the complex and nonlinear nature of the process, combined with catalyst aging and changing operating conditions, makes the identification of optimal operating parameters a challenging task.

This work presented a data-driven Digital Twin framework for the prediction and optimization of the U510 catalytic isomerization unit. The proposed approach leveraged historical data collected from the Distributed Control System (DCS) to model the relationship between process variables and product octane. Several machine learning and deep learning techniques were investigated and evaluated in order to identify the most accurate predictive model for representing process behavior.

Based on the selected predictive model, a Digital Twin was developed to provide a virtual representation of the process capable of rapidly estimating RON under different operating conditions. This virtual environment was then coupled with advanced optimization algorithms to determine the optimal inlet temperature setpoints of reactors R-001 and R-002. By restricting the optimization problem to these key operational variables, the proposed framework provides practical and actionable recommendations that can be directly used by refinery operators to improve process performance.

The results demonstrated the effectiveness of combining Artificial Intelligence, Machine Learning, Digital Twin technology, and optimization techniques for industrial process improvement. The developed system enables rapid prediction, scenario analysis, and optimization without interfering with plant operation, thereby supporting more informed and data-driven decision-making.

In addition to the predictive and optimization modules, a user-friendly interface was developed to facilitate interaction with the Digital Twin and provide intuitive access to predictions and optimization results. This interface enhances the practical applicability of the proposed solution and contributes to its potential integration into industrial environments.

Overall, the proposed framework represents a significant step toward the digital transformation of refinery operations and illustrates the potential of AI-driven solutions for intelligent

process optimization. The outcomes of this study demonstrate that Digital Twin technology, supported by Machine Learning and optimization algorithms, can serve as a powerful decision-support tool for improving operational efficiency, maximizing product quality, and generating added economic value in modern petroleum refineries.

Future Works

Future work may focus on integrating real-time DCS data streams, implementing continuous online optimization, incorporating additional process constraints, and extending the framework to other refinery units and industrial processes.

Bibliography

[1] Sonatrach, “Annual Report — Refining & Petrochemicals Activity,” General Directorate of Refining, Algiers, Algeria, 2023.

[2] Sonatrach Refining Division, “U510 Naphtha Isomerization Unit Technical Specification Sheet and Catalyst Run History,” Sonatrach Internal Operational Repository, 2022.

[3] UOP LLC, “Penex™ / DIH Isomerization Process: Design and Operating Manual,” Honeywell UOP, Des Plaines, IL, USA, 2019.

[4] M. Rase, *Chemical Reactor Design for Process Plants: Volume 2 — Case Studies and Design Data*, Wiley, New York, USA, 1977.

[5] Z. Ge, Z. Song, S. X. Ding, and B. Huang, “Data mining and analytics in the process industry: The role of machine learning,” *IEEE Access*, vol. 5, pp. 20590–20616, 2017, doi: 10.1109/ACCESS.2017.2756872.

[6] J. H. Gary, G. E. Handwerk, and M. J. Kaiser, *Petroleum Refining: Technology and Economics*, 5th ed., CRC Press, Taylor & Francis Group, 2007.

[7] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 4th ed., Pearson, 2020.

[8] M. I. Jordan and T. M. Mitchell, “Machine learning: Trends, perspectives, and prospects,” *Science*, vol. 349, no. 6245, pp. 255–260, 2015, doi: 10.1126/science.aaa8415.

[9] P. Domingos, “A few useful things to know about machine learning,” *Communications of the ACM*, vol. 55, no. 10, pp. 78–87, 2012.

[10] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, 2015, doi: 10.1038/nature14539.

[11] A. Géron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*, 3rd ed., O’Reilly Media, 2022.

[12] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*, John Wiley & Sons, 2001.

[13] V. Venkatasubramanian, "The promise of artificial intelligence in chemical engineering," *AIChE Journal*, vol. 65, no. 2, pp. 466–478, 2019, doi: 10.1002/aic.16489.

[14] T. M. Mitchell, *Machine Learning*, McGraw-Hill, 1997.

[15] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *Annals of Statistics*, vol. 29, no. 5, pp. 1189–1232, 2001.

[16] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, 2016, pp. 785–794, doi: 10.1145/2939672.2939785.

[17] G. Ke et al., "LightGBM: A highly efficient gradient boosting decision tree," *Advances in Neural Information Processing Systems*, vol. 30, pp. 3146–3154, 2017.

[18] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, "CatBoost: Unbiased boosting with categorical features," *Advances in Neural Information Processing Systems*, vol. 31, pp. 6638–6648, 2018.

[19] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT Press, 2016.

[20] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, 2nd ed., Springer, 2009.

[21] C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006.

[22] O. Chapelle, B. Schölkopf, and A. Zien, Eds., *Semi-Supervised Learning*, MIT Press, 2006.

[23] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed., MIT Press, 2018.

[24] V. Mnih et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, pp. 529–533, 2015, doi: 10.1038/nature14236.

[25] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Computing Surveys*, vol. 41, no. 3, pp. 1–58, 2009, doi: 10.1145/1541880.1541882.

[26] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[27] F. Ricci, L. Rokach, and B. Shapira, Eds., *Recommender Systems Handbook*, 2nd ed., Springer, 2015.

[28] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.

[29] R. J. Hyndman and G. Athanasopoulos, *Forecasting: Principles and Practice*, 3rd ed., OTexts, 2021. [Online]. Available: `https://otexts.com/fpp3/`

[30] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85–117, 2015, doi: 10.1016/j.neunet.2014.09.003.

[31] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[32] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986, doi: 10.1038/323533a0.

[33] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[34] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. ICLR 2015*, 2015.

[35] J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd ed., Springer, 2006.

[36] K. Deb and R. B. Agrawal, "Simulated binary crossover for continuous search space," *Complex Systems*, vol. 9, no. 2, pp. 115–148, 1995.

[37] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," *Machine Learning*, vol. 63, no. 1, pp. 3–42, 2006, doi: 10.1007/s10994-006-6226-1.

[38] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. ICNN'95*, vol. 4, pp. 1942–1948, 1995.

[39] R. Storn and K. Price, "Differential evolution — a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, Dec. 1997, doi: 10.1023/A:1008202821328.

[40] J. Mockus, "On Bayesian methods for seeking the extremum," in *Optimization Techniques IFIP Technical Conference*, Springer, 1975, pp. 400–404.

[41] D. R. Jones, M. Schonlau, and W. J. Welch, "Efficient global optimization of expensive black-box functions," *Journal of Global Optimization*, vol. 13, no. 4, pp. 455–492, 1998.

[42] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, "Algorithms for hyper-parameter optimization," *Advances in Neural Information Processing Systems*, vol. 24, pp. 2546–2554, 2011.

[43] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," in *Proc. 25th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, 2019, pp. 2623–2631.

[44] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength Pareto evolutionary algorithm," TIK-Report 103, ETH Zurich, 2001.

[45] M. Grieves and J. Vickers, "Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems," in *Transdisciplinary Perspectives on Complex Systems*, F. J. Kahlen, S. Flumerfelt, and A. Alves, Eds., Springer, 2017, pp. 85–113.

[46] A. Rasheed, O. San, and T. Kvamsdal, "Digital twin: Values, challenges and enablers from a modeling perspective," *IEEE Access*, vol. 8, pp. 21980–21912, 2020.

[47] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[48] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach," *IEEE Trans. Evolutionary Computation*, vol. 18, no. 4, pp. 577–601, 2014.

[49] G. Zahedi, S. Mohammadzadeh, and M. Morari, "Artificial neural network simulation of n-heptane and iso-octane dehydrogenation in a fluidized bed reactor," *Chemical Engineering & Technology*, vol. 31, no. 12, pp. 1784–1789, 2008.

[50] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.

[51] A. Bhosekar and M. Ierapetritou, "Advances in surrogate based modeling, feasibility analysis, and optimization: A review," *Computers & Chemical Engineering*, vol. 108, pp. 250–267, 2018.

[52] L. T. Biegler and I. E. Grossmann, "Retrospective on optimization," *Computers & Chemical Engineering*, vol. 28, no. 8, pp. 1169–1192, 2004.

[53] J. Blank and K. Deb, "pymoo: Multi-objective optimization in Python," *IEEE Access*, vol. 8, pp. 89497–89509, 2020.

[54] A. R. Conn, K. Scheinberg, and L. N. Vicente, *Introduction to Derivative-Free Optimization*, SIAM, 2009.

[55] S. Das and P. N. Suganthan, "Differential evolution: A survey of the state-of-the-art," *IEEE Trans. Evolutionary Computation*, vol. 15, no. 1, pp. 4–31, 2011.

[56] I. Das and J. E. Dennis, "Normal-boundary intersection: A new method for generating the Pareto surface in nonlinear multicriteria optimization problems," *SIAM Journal on Optimization*, vol. 8, no. 3, pp. 631–657, 1998.

[57] A. Forrester, A. Sobester, and A. Keane, *Engineering Design via Surrogate Modelling*, John Wiley & Sons, 2008.

[58] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, 1989.

[59] J. H. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, 1975.

[60] K. Miettinen, *Nonlinear Multiobjective Optimization*, Kluwer Academic Publishers, 1999.

[61] A. D. Nandasana, A. K. Ray, and S. K. Gupta, "Application of the non-dominated sorting genetic algorithm (NSGA) in chemical reaction engineering," *Industrial & Engineering Chemistry Research*, vol. 42, no. 20, pp. 4838–4852, 2003.

[62] F. Nogueira, "Bayesian Optimization: Open source constrained global optimization tool for Python," 2014. [Online]. Available: `https://github.com/fmfn/BayesianOptimization`

[63] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. de Freitas, "Taking the human out of the loop: A review of Bayesian optimization," *Proc. IEEE*, vol. 104, no. 1, pp. 148–175, 2016.

[64] J. Snoek, H. Larochelle, and R. P. Adams, "Practical Bayesian optimization of machine learning algorithms," *Advances in Neural Information Processing Systems*, vol. 25, pp. 2951–2959, 2012.

[65] E. Zitzler and L. Thiele, "Multiobjective optimization using evolutionary algorithms," in *Parallel Problem Solving from Nature*, Springer, 1998, pp. 292–301.

[66] A. Bhatt, "Machine learning-based predictive modeling of refinery operations: A comprehensive review," *Journal of Petroleum Science and Engineering*, vol. 217, p. 110918, 2022.

[67] R. Garg, S. Singh, and A. Mishra, "Multi-objective optimization of catalytic cracking process using evolutionary algorithms," *Chemical Engineering Research and Design*, vol. 168, pp. 245–258, 2021.

[68] L. H. Chiang, E. L. Russell, and R. D. Braatz, *Fault Detection and Diagnosis in Industrial Systems*, Springer, London, 2001.

[69] M. Rüßmann et al., "Industry 4.0: The future of productivity and growth in manufacturing industries," Boston Consulting Group, Apr. 2015.

[70] H. S. Fogler, *Elements of Chemical Reaction Engineering*, 5th ed., Prentice Hall, Boston, MA, 2016.

[71] J. Ancheyta, *Modeling and Simulation of Catalytic Reactors for Petroleum Refining*, John Wiley & Sons, Hoboken, NJ, 2011.

[72] W. W. Pulkrabek, *Engineering Fundamentals of the Internal Combustion Engine*, 2nd ed., Pearson Prentice Hall, Upper Saddle River, NJ, 2004.

[73] D. E. Seborg, T. F. Edgar, D. A. Mellichamp, and F. J. Doyle III, *Process Dynamics and Control*, 4th ed., John Wiley & Sons, Hoboken, NJ, 2016.

[74] S. Mackay, E. Wright, D. Reynders, and J. Park, *Practical Industrial Data Networks: Design, Installation and Troubleshooting*, Newnes, Oxford, 2004.

[75] J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques*, 3rd ed., Morgan Kaufmann, Waltham, MA, 2011.

[76] S. Narasimhan and C. Jordache, *Data Reconciliation: Principles and Techniques*, Gulf Professional Publishing, Boston, MA, 2000.

[77] W. Mahnke, S. Leitner, and T. Damm, *OPC Unified Architecture*, Springer-Verlag, Berlin, 2009.

[78] R. Brooks, *Introduction to Industrial Data Historians: Storing and Utilizing Time-Series Process Data*, TechPress, London, 2018.

[79] J. W. Tukey, *Exploratory Data Analysis*, Addison-Wesley, Reading, MA, 1977.

[80] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014.

[81] L. Prechelt, "Early stopping — but when?" in *Neural Networks: Tricks of the Trade*, G. Montavon, G. B. Orr, and K.-R. Müller, Eds., Springer, Berlin, 1998, pp. 55–69.

[82] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," *Advances in Inference and Information Processing Systems*, vol. 30, pp. 4765–4774, 2017.

[83] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[84] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning*, Springer, New York, NY, 2013.

[85] N. V. Queipo et al., "Surrogate-based analysis and optimization," *Progress in Aerospace Sciences*, vol. 41, no. 1, pp. 1–28, 2005, doi: 10.1016/j.paerosci.2005.02.001.

[86] P. Virtanen et al., "SciPy 1.0: Fundamental algorithms for scientific computing in Python," *Nature Methods*, vol. 17, pp. 261–272, 2020, doi: 10.1038/s41592-019-0686-2.

[87] C. R. Harris et al., "Array programming with NumPy," *Nature*, vol. 585, pp. 357–362, 2020, doi: 10.1038/s41586-020-2649-2.

[88] W. McKinney, "Data structures for statistical computing in Python," in *Proc. 9th Python in Science Conf.*, 2010, pp. 56–61.

[89] J. D. Hunter, "Matplotlib: A 2D graphics environment," *Computing in Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007, doi: 10.1109/MCSE.2007.55.

[90] M. L. Waskom, "seaborn: Statistical data visualization," *Journal of Open Source Software*, vol. 6, no. 60, p. 3021, 2021, doi: 10.21105/joss.03021.

[91] S. J. Qin and T. A. Badgwell, "A survey of industrial model predictive control technology," *Control Engineering Practice*, vol. 11, no. 7, pp. 733–764, 2003, doi: 10.1016/S0967-0661(02)00186-7.

[92] M. Črepinšek, S.-H. Liu, and M. Mernik, "Exploration and exploitation in evolutionary algorithms: A survey," *ACM Computing Surveys*, vol. 45, no. 3, p. 35, 2013, doi: 10.1145/2480308.2480312.

[93] D. C. Montgomery, E. A. Peck, and V. G. G. Vining, *Introduction to Linear Regression Analysis*, 5th ed., Wiley, New York, NY, 2012.

[94] L. Fortuna, S. Graziani, A. Rizzo, and M. G. Xibilia, *Soft Sensors for Monitoring and Control of Industrial Processes*, Springer, London, 2007, doi: 10.1007/978-1-84628-480-9.

[95] Facebook Inc., "React: A JavaScript Library for Building User Interfaces," Meta Open Source, 2026. [Online]. Available: `https://react.dev`

[96] Microsoft Corporation, "TypeScript: Typed Superset of JavaScript for Scale," Developer Tools Division, 2026. [Online]. Available: `https://www.typescriptlang.org`

[97] Vite Team, "Vite: Next Generation Frontend Tooling," Open Source Ecosystem, 2026. [Online]. Available: `https://vitejs.dev`

[98] Recharts Team, "Recharts: A Redefined Chart Library Built with React and D3," Open Source Architecture, 2026. [Online]. Available: `https://recharts.org`

[99] OpenJS Foundation, "Node.js Asynchronous JavaScript Runtime Environment," Linux Foundation Frameworks, 2026. [Online]. Available: `https://nodejs.org`

[100] Express.js Team, "Express: Fast, Unopinionated, Minimalist Web Framework for Node.js," Node.js Foundation, 2026. [Online]. Available: `https://expressjs.com`

[101] PostgreSQL Global Development Group, "PostgreSQL: The World's Most Advanced Open Source Relational Database," Object-Relational Engine, 2026. [Online]. Available: `https://www.postgresql.org`