Physics-guided time-series machine learning for dynamic modeling, control, and resilience in process systems

by

Amirsalar Bagheri

M.S., Kansas State University, 2024

AN ABSTRACT OF A DISSERTATION

submitted in partial fulfillment of the
requirements for the degree

DOCTOR OF PHILOSOPHY

Department of Chemical Engineering
Carl R. Ice College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2026

Abstract

Real-time predictive process control and optimization rely fundamentally on accurate dynamic process models. Across chemical, agricultural, and food processing systems, several strategies have been extensively studied for constructing such models. First-principles, white-box models are among the principal approaches, derived from mass, energy, and momentum balances that yield systems of partial differential equations (PDEs) or ordinary differential equations (ODEs). Using these models for real-time purposes, however, faces certain limitations, including unknown parameters and high computational costs. Moreover, as process systems become increasingly integrated, the governing equations grow limited in their ability to describe the full system dynamics, introducing unmodeled physics and discrepancies that adversely affect real-time performance. Time-series machine learning (ML) modeling has been introduced to address these challenges; however, purely black-box models, in which a learning-based model is developed from process data alone, suffer in turn from limited generalizability and extrapolation capability.

Bridging the gap between these two modeling approaches, this dissertation explores different methods of integrating process domain knowledge with advanced time-series ML techniques, constructing knowledge-guided modeling frameworks that can be embedded within real-time control and optimization schemes and further employed for process safety and resilient operation. Depending on the form in which process knowledge is available, ranging from high-fidelity simulations and reduced-order physical models to expert assessments, imaging, and real-time sensor measurements, dedicated frameworks are developed, including surrogate models for model predictive control, hybrid and physics-informed architectures, multimodal modeling, and ML-based cyberattack detection and recovery. The proposed frameworks are evaluated through several case studies in chemical, agricultual and food process systems.

Physics-guided time-series machine learning for dynamic modeling, control, and resilience in process systems

by

Amirsalar Bagheri

M.S., Kansas State University, 2024

A DISSERTATION

submitted in partial fulfillment of the
requirements for the degree

DOCTOR OF PHILOSOPHY

Department of Chemical Engineering
Carl R. Ice College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2026

Approved by:

Major Professor
Dr. Davood B. Pourkargar

Copyright

© Amirsalar Bagheri 2026.

Abstract

Real-time predictive process control and optimization rely fundamentally on accurate dynamic process models. Across chemical, agricultural, and food processing systems, several strategies have been extensively studied for constructing such models. First-principles, white-box models are among the principal approaches, derived from mass, energy, and momentum balances that yield systems of partial differential equations (PDEs) or ordinary differential equations (ODEs). Using these models for real-time purposes, however, faces certain limitations, including unknown parameters and high computational costs. Moreover, as process systems become increasingly integrated, the governing equations grow limited in their ability to describe the full system dynamics, introducing unmodeled physics and discrepancies that adversely affect real-time performance. Time-series machine learning (ML) modeling has been introduced to address these challenges; however, purely black-box models, in which a learning-based model is developed from process data alone, suffer in turn from limited generalizability and extrapolation capability.

Bridging the gap between these two modeling approaches, this dissertation explores different methods of integrating process domain knowledge with advanced time-series ML techniques, constructing knowledge-guided modeling frameworks that can be embedded within real-time control and optimization schemes and further employed for process safety and resilient operation. Depending on the form in which process knowledge is available, ranging from high-fidelity simulations and reduced-order physical models to expert assessments, imaging, and real-time sensor measurements, dedicated frameworks are developed, including surrogate models for model predictive control, hybrid and physics-informed architectures, multimodal modeling, and ML-based cyberattack detection and recovery. The proposed frameworks are evaluated through several case studies in chemical, agricultual and food process systems.

Table of Contents

List of Figures xi

List of Tables xvii

Acknowledgements xix

Dedication xx

1 Introduction 1

1.1 Time-Series Prediction: A General Problem 1

1.2 Time-Series Modeling in Process Systems Engineering 3

1.2.1 Model reduction approaches 5

1.2.2 Grey-box (hybrid) modeling approaches 5

1.2.3 Fully data-driven (black-box) modeling approaches 7

1.3 Problem Statement and Motivation 9

1.4 Research Objectives and Case Studies 9

1.5 Contributions 12

2 Materials and Methods 16

2.1 Machine Learning Architectures for Dynamic Process Modeling 16

2.1.1 Extreme Gradient Boosting (XGBoost) 17

2.1.2 Light Gradient Boosting Machine (LightGBM) 18

2.1.3 Feed Forward Neural Networks (FNNs) 19

2.1.4 Recurrent Neural Networks (RNNs) 20

2.1.5 Sequence-to-Sequence (Seq2Seq) Models 24

2.1.6 Self-Attention Decoder-Only Transformers . . . 25
2.1.7 Physics-Informed and Hybrid Modeling . . . 26
2.2 Computer Vision Methods for Process State Characterization . . . 28
2.2.1 Canny Edge Detection . . . 28
2.2.2 Hough Transform . . . 29
2.2.3 Gabor Filters for Structure Analysis . . . 30
2.3 K-means spatial clustering . . . 30
2.4 Evaluation Metrics . . . 31
2.4.1 Regression Metrics: NRMSE, MAE, and Mean Bias . . . 31
2.4.2 Classification Metrics: ROC, AUC, F1, Accuracy, Precision, and Recall 32
2.5 Jaccard index . . . 32
2.6 Explained variance (EV) . . . 33
2.7 Model Predictive Control (MPC) . . . 33
2.8 Moving Horizon Estimation (MHE) . . . 34
2.9 Closed-loop performance metrics . . . 35

3 Surrogate-Based Real-Time Model Predictive Control of Ammonia Synthesis and Adsorption Processes . . . 37
3.1 Introduction . . . 37
3.2 Multiscale modeling of the integrated ammonia synthesis and adsorption process 39
3.3 Problem Statement . . . 40
3.4 High-fidelity simulation of the multiscale model . . . 41
3.5 Integrated surrogate modeling of the reactor–adsorber system . . . 43
3.5.1 Centralized surrogate modeling . . . 43
3.5.2 Integrated surrogate modeling with synchronized co-training . . . 44
3.6 Surrogate-based MPC of the integrated process . . . 46
3.7 Integrated Surrogate Model Evaluations . . . 48
3.8 Closed-loop MPC results . . . 50

3.9 Conclusion . . . 52

4 Hybrid and Physics-informed Time-Series Modeling for Soil Water Content (SWC) 55

4.1 Introduction . . . 55

4.2 Kansas Mesonet Dataset and Data Preprocessing . . . 57

4.3 Antecedent Precipitation Index (API) Model . . . 59

4.4 Spatial Clustering of Kansas Mesonet Stations . . . 61

4.5 Centralized and Distributed Hybrid Uncertainty Quantification . . . 61

4.6 Centralized and Distributed Physics-Informed SWC Models . . . 65

4.7 Results and Discussions . . . 67

4.7.1 Spatial Clustering Results . . . 68

4.7.2 Hybrid Uncertainty Quantification Results . . . 69

4.7.3 Physics-Informed Neural Network Results . . . 72

4.7.4 Overall Comparison . . . 75

4.8 Conclusion . . . 78

5 Multimodal Time-Series Modeling of Plant-Based Meat Extrusion . . . 80

5.1 Introduction . . . 80

5.2 Extrusion Process Description . . . 83

5.2.1 Ingredients and formulation . . . 84

5.2.2 Extrusion operational conditions . . . 84

5.2.3 Image acquisition and visual analysis . . . 85

5.3 Integration of Time-Series ML and CV for Extrusion Modeling . . . 86

5.3.1 Data collection . . . 86

5.3.2 Preprocessing and feature engineering . . . 87

5.4 CV methods for structure quantification . . . 90

5.4.1 Unsupervised learning CV method . . . 90

5.4.2 Semi-supervised learning CV method . . . 92

5.5 Integrated time-series framework for extrusion dynamics and structure modeling 93
5.6 Results and Discussion . . . 95
5.6.1 CV-based structure quantification results . . . 96
5.6.2 Time-series ML integration with structure quantification . . . 99
5.7 Conclusions . . . 104

6 Transformer-Based Monitoring for Cyber-Resilient Operation of Highly Integrated Chemical Process Systems . . . 105
6.1 Introduction . . . 105
6.2 Representative sensor-level adversarial attacks in process Systems . . . 109
6.2.1 Min-max attacks . . . 109
6.2.2 Noise-based attacks . . . 110
6.2.3 Surge attacks . . . 110
6.2.4 Set difference attacks . . . 111
6.3 Cyberattack Detection Methods . . . 111
6.3.1 Multi-Label classification and binary cross-entropy loss function . . . 112
6.4 Time-series classification frameworks for attack detection and identification . 113
6.4.1 Centralized end-to-end (ETE) transformer . . . 114
6.4.2 Centralized two-tier framework . . . 115
6.4.3 Community-based distributed frameworks: DETE and D2Tier . . . 116
6.5 Post-attack state reconstruction using extended Kalman filters . . . 122
6.6 Case Study: Integrated Process of Benzene Alkylation with Ethylene . . . 124
6.6.1 Data generation and preprocessing . . . 126
6.6.2 Model implementation and training . . . 129
6.6.3 Centralized detection and identification performance . . . 132
6.6.4 Distributed detection and identification performance . . . 138
6.6.5 Post-attack state reconstruction performance . . . 145
6.7 Comparative evaluation of centralized and distributed frameworks . . . 153

6.8 Conclusions 157

7 Conclusions and Future Directions 159

7.1 Concluding Remarks 159

7.2 Future Motivations: Toward Automated Reasoning for Knowledge Integration 161

Bibliography 164

A Appendix A: Supplementary Figures for Chapter 6 186

A.1 Confusion Matrices for Proposed Frameworks 186

A.2 Post-attack EKF-based state reconstruction 189

A.2.1 Centralized Methods 189

A.2.2 Distributed Methods 196

List of Figures

1.1 Conceptual framework of the dissertation 10

2.1 Feed Forward neural Network (FNN) 20
2.2 RNN structure 21
2.3 An LSTM Cell 22
2.4 A GRU Cell 24
2.5 Decoder-only Transformer Architecture 27

3.1 Schematic of the integrated ammonia synthesis and adsorption process, comprising the Haber–Bosch PBR, intermediate heat exchanger, and TSA adsorption–regeneration columns. 40
3.2 Subsystem decomposition of the integrated ammonia synthesis and adsorption process 45
3.3 Closed-loop architecture of the surrogate-based MPC for the integrated ammonia synthesis and adsorption process 47
3.4 Open-loop predictions of the centralized and integrated surrogate models against the high-fidelity ground truth for two test operating conditions. . . . 49
3.5 NRMSE comparison of the centralized and integrated surrogate models for the three key process outputs across the test dataset. 49
3.6 Closed-loop trajectories of the manipulated inputs (left column) and process outputs (right column) under the Integrated-surrogate MPC. 53
3.7 Closed-loop trajectories of the manipulated inputs (left column) and process outputs (right column) under the centralized-surrogate MPC. 53

4.1 Schematic representation of the water cycle. 56
4.2 Kansas MesoNet Distributed Stations 58
4.3 Inertia as a function of the number of clusters K for the Kansas Mesonet stations. 62
4.4 Hybrid UQ Model Training scheme 63
4.5 Hybrid UQ Model Inference scheme 64
4.6 Silhouette score distributions for the candidate cluster numbers. 68
4.7 Clustering results under different candidate values of K. 69
4.8 2D and 3D projections of the final clustering with $K^* = 6$. 70
4.9 Time-series plots for the UQ-based LSTM models. 71
4.10 Time-series plots for the UQ-based Transformer models. 71
4.11 Parity plots for the UQ-based models. 72
4.12 Error distribution plots for the UQ-based models. 73
4.13 Time-series plots for the physics-informed LSTM models. 74
4.14 Time-series plots for the physics-informed Transformer models. 74
4.15 Parity plots for the physics-informed models. 75
4.16 Error distribution plots for the physics-informed models. 75
4.17 NRMSE box plots of the SWC models across stations. 76
4.18 Mean bias box plots of the SWC models across stations. 77
4.19 Joint NRMSE–Moran's I trade-off across all model configurations. 78

5.1 Schematic of the twin-screw extrusion system used in the study. 83
5.2 Interconnected network of extrusion process variables. 88
5.3 Unsupervised CV algorithm for structure quantification. 91
5.4 Effect of tuning Canny edge detection parameters on the performance of the unsupervised CV method. 92
5.5 Semi-supervised CV algorithm for structure quantification. 93
5.6 Schematics of the integrated time-series framework for structure score prediction. 94

5.7 Confusion matrices for the CV-based methods. 97
5.8 Comparison of ROC curves for CV-based methods. 98
5.9 Comparison of structure scores predicted by the unsupervised and semi-supervised methods with expert-labeled scores across feed compositions and time-series data. 99
5.10 XGBoost model time-series prediction for WG. 100
5.11 XGBoost model time-series prediction for PPI. 101
5.12 XGBoost model time-series prediction for SPI. 101
5.13 LightGBM model time-series prediction for WG. 101
5.14 LightGBM model time-series prediction for PPI. 102
5.15 LightGBM model time-series prediction for SPI. 102

6.1 ETE centralized transformer for cyberattack detection and sensor identification. 114
6.2 Two-tier framework schematics. 116
6.3 DETE transformer framework for community-level attack detection and identification in highly integrated process systems. 119
6.4 D2Tier framework for community-level attack detection and identification in highly integrated process systems. 121
6.5 Benzene alkylation process schematic. 126
6.6 MPC-MHE loop under cyber attack 127
6.7 Dataset distribution across different labels. 128
6.8 ROC curve and AUC values for the performance of the CETE transformer framework. 133
6.9 ROC curve and AUC values for the two-tier framework. 136
6.10 Community-level ROC curves for the distributed ETE and two-tier frameworks. 139
6.11 ROC curves at the network level for the DETE and D2Tier frameworks. . . 142
6.12 Closed-loop detection and reconstruction under a single surge attack using the ETE framework. 147

6.13 Closed-loop detection and reconstruction under a single surge attack using a two-tier framework. 147
6.14 Closed-loop detection and reconstruction under multi-sensor surge attacks using the ETE transformer framework. 148
6.15 Closed-loop detection and reconstruction under multi-sensor surge attacks using the two-tier framework. 149
6.16 Closed-loop detection and reconstruction under a single min–max attack using the DETE framework. 150
6.17 Closed-loop detection and reconstruction under a single min–max attack using the D2Tier framework. 151
6.18 Closed-loop detection and reconstruction under a multi-sensor min–max attack using the DETE framework. 152
6.19 Closed-loop detection and reconstruction under a multi-sensor min–max attack using the D2Tier framework 152

A.1 Confusion matrix using CETE transformer. 187
A.2 Confusion matrix using C2tier framework. 187
A.3 Network-level confusion matrix for the DETE framework. 188
A.4 Network-level confusion matrix for the D2tier framework. 189
A.5 Closed-loop simulation results for detection and state reconstruction under single min-max attacks using the CETE transformer. 190
A.6 Closed-loop simulation results for detection and state reconstruction under single min-max attacks using the C2tier framework. 190
A.7 Closed-loop simulation results for detection and state reconstruction under multi min-max attacks using the CETE transformer. 191
A.8 Closed-loop simulation results for detection and state reconstruction under multi min-max attacks using the C2tier framework. 191

A.9 Closed-loop simulation results under single noise attacks using the CETE transformer. 192
A.10 Closed-loop simulation results under single noise attacks using the C2tier framework. 192
A.11 Closed-loop simulation results under multi-noise attacks using the CETE transformer. 193
A.12 Closed-loop simulation results under multi-noise attacks using the C2tier framework. 193
A.13 Closed-loop simulation results under single set difference attacks using the CETE transformer. 194
A.14 Closed-loop simulation results under single set difference attacks using the C2tier framework. 194
A.15 Closed-loop simulation results under multi-set difference attacks using the CETE transformer. 195
A.16 Closed-loop simulation results under multi-set difference attacks using the C2tier framework. 195
A.17 Closed-loop detection and reconstruction under a single surge attack using the DETE framework. 196
A.18 Closed-loop detection and reconstruction under a single surge attack using the D2tier framework. 197
A.19 Closed-loop detection and reconstruction under a multi-surge attack using the DETE framework. 197
A.20 Closed-loop detection and reconstruction under a multi-surge attack using the D2tier framework. 198
A.21 Closed-loop detection and reconstruction under a single noise attack using the DETE framework. 198

A.22 Closed-loop detection and reconstruction under a single noise attack using the D2tier framework. 199
A.23 Closed-loop detection and reconstruction under a multi-noise attack using the DETE framework. 199
A.24 Closed-loop detection and reconstruction under a multi-noise attack using the D2tier framework. 200
A.25 Closed-loop detection and reconstruction under a single set-difference attack using the DETE framework. 200
A.26 Closed-loop detection and reconstruction under a single set-difference attack using the D2tier framework. 201
A.27 Closed-loop detection and reconstruction under a multi-set-difference attack using the DETE framework. 201
A.28 Closed-loop detection and reconstruction under a multi-set-difference attack using the D2tier framework. 202

List of Tables

3.1 Normalized ISC of the reactor manipulated inputs under the centralized- and Integrated-surrogate MPC. 51

3.2 Normalized ISE of the regulated outputs under the centralized- and Integrated-surrogate MPC. 51

3.3 DPI of the centralized- and Integrated-surrogate MPC, aggregating the normalized ISC and ISE contributions of Tables 3.1–3.2. 52

4.1 Overall quantitative metrics for each model configuration averaged across all 62 Kansas Mesonet stations on the test set. 79

5.1 Evaluation metrics for CV methods: Binary classification and structure score prediction. 98

5.2 Evaluation metrics for XGBoost and LightGBM across all feed compositions. 103

6.1 Attack configurations used for data generation. 127

6.2 Community detection for temperature sensors. 131

6.3 CETE transformer detection performance across different attack types. . . . 133

6.4 CETE transformer detection metrics across multi-attack scenarios. 134

6.5 CETE transformer identification performance across different attack types. . 134

6.6 CETE transformer identification performance across different attack scenarios. 134

6.7 Overall detection performance of the two-tier framework across attack types. 136

6.8 Detection metrics by number of attacked sensors for the two-tier framework. 137

6.9 Overall identification performance of the two-tier framework across attack types. 137

6.10 Identification metrics by number of attacked sensors for the two-tier framework. 138

6.11 Community-level comparison of detection performance. 140
6.12 Comparison of community-level identification performance across methods. . 140
6.13 Network-level detection performance comparison. 142
6.14 Network-level identification performance comparison. 143
6.15 Network-level detection performance by attack type. 143
6.16 Network-level identification performance by attack type. 144
6.17 Network-level attack detection performance as a function of attack cardinality. 144
6.18 Network-level attack identification performance as a function of attack cardinality. 145
6.19 EKF-based reconstruction metrics under single-sensor attacks for the centralized frameworks. 149
6.20 EKF-based reconstruction metrics under multi-sensor attacks for the centralized frameworks. 150
6.21 Post-attack reconstruction performance under single-sensor attacks for the distributed frameworks. 153
6.22 Post-attack reconstruction performance under coordinated multi-sensor attacks for the distributed frameworks. 154
6.23 Centralized vs. distributed network-level detection performance. 155
6.24 Centralized vs. distributed network-level identification performance. 155
6.25 Average reconstruction performance across attack types under single-sensor attacks. 156
6.26 Average reconstruction performance across attack types under coordinated multi-sensor attacks. 156

Acknowledgments

I would like to express my sincere gratitude to the Kansas State University Graduate School for its continued support throughout my doctoral studies, and to the Tim Taylor Department of Chemical Engineering for providing the academic environment, resources, and facilities that made this research possible. The department's commitment to fostering rigorous, collaborative research has been instrumental in shaping both this dissertation and my development as a researcher.

I am profoundly grateful to my advisor, Dr. Davood Pourkargar, for his invaluable guidance, mentorship, and unwavering support throughout this journey. His insights have not only shaped the technical direction of this work but have also illuminated my growth as an independent researcher. I would also like to thank my doctoral committee members, Dr. Antony, Dr. Amama, and Dr. Caragea, for their thoughtful feedback, critical questions, and guidance, which have strengthened this dissertation. Finally, I owe my deepest gratitude to my family and friends, whose love, patience, and encouragement sustained me through every stage of this work. Their support has been a constant source of strength, and this dissertation would not have been possible without them.

Dedication

This dissertation is dedicated to my parents, whose unconditional love, sacrifice, and belief in me have been the foundation of every achievement in my life. Their unwavering support and encouragement gave me the strength and confidence to pursue this path, even across the many miles that have separated us during these years of study. Every page of this work carries a piece of the values, patience, and perseverance they instilled in me.

This dissertation is also dedicated to my sister, whose constant love, encouragement, and belief in me have meant more than words can express. Even across the distance, her presence has been a source of comfort, strength, and motivation throughout this journey, and I am deeply grateful to share this milestone with her.

Chapter 1

Introduction

1.1 Time-Series Prediction: A General Problem

Time-series prediction, also known as dynamic system modeling, concerns predicting a system's future behavior given its past states and current input signals. The desire to predict what happens in the future under different input scenarios, and to analyze the range of possible outcomes that follow, has long been a standing challenge across a wide range of application domains, precisely because most systems of practical interest, whether financial, environmental, or industrial, do not evolve independently at each time step but instead carry forward information from their past contextual states.

In financial markets, the time-series prediction challenge lies in forecasting stock prices over short- and long-term horizons, given a wide range of economic, political, and market-sentiment inputs. For instance, several studies have focused on using machine learning methods such as recurrent neural networks (RNNs) and Long Short-Term Memory (LSTM) architectures to forecast stock prices from historical trading data, motivated by the highly stochastic and non-stationary nature of asset price movements, which limits the effectiveness of classical linear time-series models[1;2].

In urban transportation systems, the time-series prediction challenge is forecasting short-term traffic flow from historical volume patterns that are affected by recurring daily and

weekly cycles, as well as irregular disruptions such as accidents or road work. For instance, several studies have used machine learning methods, such as LSTM networks, to forecast traffic volumes from historical sensor data, motivated by the inability of classical statistical traffic models to capture the nonlinear and non-stationary fluctuations arising from complex urban mobility patterns[3;4].

In the energy sector, accurately predicting short-term electricity demand is a long-standing challenge for grid operators, since generation must be scheduled in advance while consumption itself fluctuates with weather, time-of-day effects, and variable consumer behavior. Machine learning-based load forecasting methods have been extensively surveyed in this context, with several studies reporting that data-driven models can capture the nonlinear temporal dependencies underlying residential and grid-level consumption patterns more effectively than classical statistical approaches, providing the accuracy needed to support generation scheduling and maintain real-time supply-demand balance[5;6].

Across these examples, the common limitations are the same: the governing dynamics of the system are either unknown, analytically intractable, or computationally demanding to resolve on the timescales required for decision-making.

For decades, these obstacles limited time-series prediction to linear or shallow statistical models, such as autoregressive and moving-average methods, which struggle to capture long-range temporal dependencies and nonlinear behavior. The emergence of learning-based methods, particularly RNNs and, more recently, self-attention-based transformer architectures, has substantially expanded the capacity of nonlinear temporal correlations for time-series prediction. RNNs introduced the ability to propagate information across time steps through internal memory states, allowing models to learn temporal correlations directly from sequential data rather than from a pre-specified functional form[7]. Transformer architectures further extended this capability by replacing recurrence with self-attention mechanisms, enabling models to weigh the relevance of distant past observations directly and to be trained more efficiently in parallel[8]. This methodological progress has occurred alongside, and has been substantially enabled by, advances in computing hardware, including graphics processing units (GPUs) and specialized neural network accelerators, which have made the training

of large, deep sequence models computationally practical[9]. Together, these developments have unlocked the capacity to tackle time-series prediction problems that were previously intractable for data-driven methods, particularly those involving long memory horizons, high-dimensional inputs, and strongly nonlinear dynamics.

Regardless of the specific architecture used, time-series prediction problems of this kind are commonly formulated using a general discrete-time state-space representation,

$$x_{t+1} = f(x_t, u_t), \tag{1.1a}$$

$$y_t = h(x_t), \tag{1.1b}$$

where x_t denotes the internal state of the system at time t, u_t denotes any external or manipulated inputs, y_t denotes the observed output, and $f(\cdot)$ and $h(\cdot)$ denote the generally nonlinear and possibly unknown state transition and observation functions, respectively. Learning-based time-series models can be viewed as data-driven approximations of $f(\cdot)$ and $h(\cdot)$, learned directly from historical input-output sequences rather than derived from first principles. As discussed in the following section, this state-space perspective aligns naturally with the mathematical description of dynamic processing systems, where the state transition function is traditionally derived from first-principles mass, energy, and momentum balances.

1.2 Time-Series Modeling in Process Systems Engineering

In process systems engineering (PSE), the time-series prediction challenge is more complex, as the aim is to predict the future state of a process system that typically involves a thermophysical transformation. This challenge, often referred to as dynamic process modeling, is the first step in constructing any process control and optimization system.

The mathematical representation of such dynamic behavior is traditionally derived from

first principles, or so-called white-box modeling, by formulating conservation (mass, energy, and momentum) balances over the relevant control volumes of the process. In its most general form, a conservation balance for a field variable $\xi(x, y, z, t)$ (e.g., species concentration, temperature, or momentum) can be written as

$$\frac{\partial \xi}{\partial t} + \nabla \cdot (\mathbf{v}\xi) = \nabla \cdot (D\nabla\xi) + r(\xi, u), \tag{1.2}$$

where $\xi(x, y, z, t)$ is the field variable of interest evaluated at spatial coordinates (x, y, z) and time t, $\mathbf{v}(x, y, z, t)$ denotes the bulk velocity field, D denotes the relevant diffusivity or conductivity, and $r(\xi, u)$ denotes the net generation or consumption term, such as reaction kinetics, which is itself a function of the state and any manipulated or disturbance inputs u. When ξ represents momentum itself, Eq. (1.2) recovers the general form of the Navier-Stokes equations, coupling the velocity field $\mathbf{v}$ to pressure, viscous, and body-force terms, and is typically solved jointly with the continuity and energy balances to fully describe the transport phenomena within the process[10;11].

Depending on the simplifying assumptions applied to a given process, such as neglecting spatial gradients (lumped modeling), assuming steady-state behavior in one or more directions, or reducing the coordinate system based on geometry, Eq. (1.2) reduces to either a system of partial differential equations (PDEs) or a system of ordinary differential equations (ODEs), for well-mixed, lumped-parameter systems[12]. Obtaining a closed-form analytical solution for these governing equations is generally difficult. Instead, it requires numerical solution techniques, such as finite difference, finite element, or finite volume methods, discretized in time and in the appropriate coordinate system (Cartesian, cylindrical, or spherical) for the geometry of the process under consideration, to obtain a numerically tractable dynamic model of the system[13].

Despite the practicality of these numerical methods, their real-time use requires significant computational capacity, hindering the application of white-box models for process control and optimization. This computational burden becomes a serious constraint for high-dimensional, integrated process systems, where repeatedly solving the governing PDEs or

ODEs within the short sampling intervals required by real-time optimization and control is often impractical[14–18].

To address this limitation, prior work in the field has proposed replacing or augmenting white-box models with computationally cheaper alternatives that approximate the underlying process dynamics. Recent works span several complementary directions.

1.2.1 Model reduction approaches

Model reduction approaches construct lower-dimensional approximations of the original governing equations by projecting the full-order dynamics onto a reduced subspace, preserving the underlying physical structure while substantially reducing simulation cost. These methods typically proceed by identifying a low-dimensional basis, often obtained through proper orthogonal decomposition or balanced truncation, that captures the dominant modes of the system's behavior, and then projecting the original high-dimensional ODE or PDE system onto this basis to obtain a reduced-order model with substantially fewer states. Because the reduced model is derived directly from the original governing equations rather than learned purely from data, it retains a degree of physical interpretability and extrapolation reliability that purely data-driven approaches often lack. However, the accuracy of the reduced model is inherently tied to the accuracy of the basis used, and its performance could drop outside the parameter or operating range over which the reduction was performed, which limits its applicability to process systems that operate over wide-ranging or highly nonlinear operating conditions[19–21].

1.2.2 Grey-box (hybrid) modeling approaches

Grey-box, or hybrid, modeling approaches retain part of the first-principles structure of Eqs. (1.2) while embedding data-driven components to represent the portions of the dynamics that are unknown, uncertain, or computationally expensive to resolve directly[22]. Two closely related lines of prior work fall under this category. Hybrid semi-parametric models, which explicitly combine a mechanistic submodel with a nonparametric data-driven submodel, and

physics-informed approaches, which instead embed the governing equations directly into the training objective of an otherwise data-driven model.

Hybrid semi-parametric models

In this setting, a mechanistic submodel, derived from mass, energy, or momentum balances, is combined with a nonparametric submodel, typically a neural network or other flexible regression model, that is trained to capture the residual dynamics not accounted for by the mechanistic component, such as unmeasured reaction kinetics or transport coefficients[22]. This structure allows the resulting model to retain the interpretability and extrapolation behavior of the mechanistic submodel in regions where the physics is well understood, while relying on the data-driven submodel only where necessary, which has been shown to reduce the amount of training data required relative to a fully data-driven model of comparable accuracy[23]. Hybrid semi-parametric modeling has been applied extensively to process monitoring, control, optimization, and scale-up tasks in the chemical and biochemical process industries[17] and forms the basis of the physics-informed hybrid modeling framework developed later in this dissertation.

Physics-informed approaches

Physics-informed approaches instead embed the governing conservation laws directly into the training objective of a machine learning model, most commonly a neural network, constraining the learned solution to satisfy the known PDEs or ODEs of the system while retaining the flexibility of a data-driven function approximator[24]. Rather than discretizing the governing equations over a mesh, as in traditional numerical methods, physics-informed neural networks (PINNs) incorporate the residual of the governing equations, along with any known initial and boundary conditions, as additional terms in the network's loss function, allowing the equations to be enforced at arbitrary collocation points in the domain without requiring a fixed spatial discretization[25]. This framework has since been extended well beyond its original formulation to a broader class of physics-informed machine learning methods that

integrate physical priors into learning architectures in a variety of ways, spanning forward simulation, inverse parameter estimation, and the discovery of hidden or unmodeled physics from data[26–29]. Because the resulting model is explicitly constrained to remain consistent with the underlying physics, physics-informed approaches are particularly well suited to dynamic processes for which the governing balance equations are known but computationally expensive to solve, or are only partially known due to uncertain parameters or unmodeled phenomena[30].

1.2.3 Fully data-driven (black-box) modeling approaches

Black-box approaches, often known as data-driven approaches, consider scenarios where abundant process data is available but obtaining a solvable, numerically or analytically, first-principles model is not feasible. These methods rely on fully exploiting the observed process data using regression techniques or, more recently, machine learning algorithms, ranging from shallow learning models to deep learning-based neural networks, to approximate the state transition function $f(\cdot)$ of Eq. (1.1a) directly[31;32].

Data-driven modeling in process systems engineering predates the recent adoption of deep learning by several decades. It has traditionally relied on classical regression and system identification techniques, such as partial least squares, principal component regression, and autoregressive models with exogenous inputs (ARX/NARX), to construct input-output models of process dynamics directly from historical operating data[33]. These classical techniques remain widely used, particularly for soft-sensing and process monitoring applications, but are generally limited in their ability to capture strongly nonlinear dynamics or long-range temporal dependencies without extensive manual feature engineering[34].

The emergence of deep learning-based time-series models has substantially expanded the capabilities of black-box modeling by removing much of this reliance on hand-engineered features. Among these, recurrent neural network (RNN) architectures, and LSTM networks in particular, are especially well-suited candidates for dynamic process modeling, since their internal memory states are explicitly designed to propagate information across time steps,

allowing them to learn temporal correlations directly from sequential process data rather than from a pre-specified functional form[7]. This property has made RNN-based architectures a common choice for constructing black-box surrogate models of nonlinear process dynamics for use in real-time optimization and control applications[35;36].

This flexibility, however, comes at the cost of reduced interpretability and a greater reliance on the quantity and quality of the available training data, since black-box models offer no inherent guarantee of consistency with physical conservation laws outside the range of conditions represented in the training set[14]. Despite these limitations, black-box time-series models have been widely adopted across process systems engineering wherever sufficient process data is available, owing to their comparatively low development cost and their ability to capture complex nonlinear dynamics without requiring detailed first-principles knowledge of the process[14;15].

From the perspective of machine learning, and time-series modeling in particular, each of these directions ultimately reformulates dynamic process modeling as a sequence learning problem. Given historical measurements of the process state and inputs, a time-series ML model, whether reduced-order, physics-informed, hybrid, or fully data-driven, is trained to predict how the state evolves forward in time, in place of, or in conjunction with, the governing conservation equations derived in the previous section.

Across all four of these modeling paradigms, model reduction, grey-box, and black-box approaches, time-series ML-based dynamic models have found broad application throughout process systems engineering, extending beyond offline prediction accuracy. In process control and optimization, surrogate models built using these approaches have been embedded directly within Model Predictive Control (MPC) and Moving Horizon Estimation (MHE) frameworks to enable real-time, computationally tractable control of nonlinear processes[37;38]. In cybersecurity, these same modeling paradigms have been used to detect deviations from expected process behavior, forming the basis of ML-based frameworks for identifying falsified sensor measurements and maintaining resilient, economically stable operation under active cyberattack scenarios[39;40].

1.3 Problem Statement and Motivation

The purpose of this dissertation is to develop knowledge-guided time-series modeling frameworks for chemical, agricultural, and food processing systems. These frameworks are designed to be capable of leveraging whatever physical knowledge of a process is available, regardless of the form that knowledge takes, to construct a more physically consistent dynamic model than a purely data-driven approach would otherwise allow. Physical understanding of a process is rarely available in a single, uniform form; rather, it exists in different layers, ranging from high-fidelity simulations, to simplified physical models, to expert judgment and unconventional measurements, to raw sensor data, depending on how well characterized the process is and how the process itself is currently monitored. This dissertation is motivated by the observation that existing time-series ML approaches tend to treat these layers of knowledge separately and inconsistently, either discarding available physical knowledge in favor of a fully data-driven surrogate, or, conversely, relying on physical models that are too detailed to embed efficiently into a learning architecture (Figure 1.1). Rather than adopting a single modeling strategy, this dissertation is motivated by unifying these layers of knowledge infusion, distilled surrogate knowledge, reduced-order physical models, expert and multimodal feedback, and real-time sensor measurements, under a common, physics-guided time-series ML framework, with each layer addressed in turn through a dedicated research objective and case study.

1.4 Research Objectives and Case Studies

The first layer of knowledge infusion addressed is the case where an existing high-fidelity model, such as a computational fluid dynamics (CFD) or multiscale simulation, is available but too computationally expensive to deploy within a real-time optimization or control loop. Rather than treating the resulting surrogate model as a purely data-driven fit to observed input-output data, the first objective of this dissertation is to derive a time-series surrogate model directly from the mass, energy, and momentum balance principles embedded within

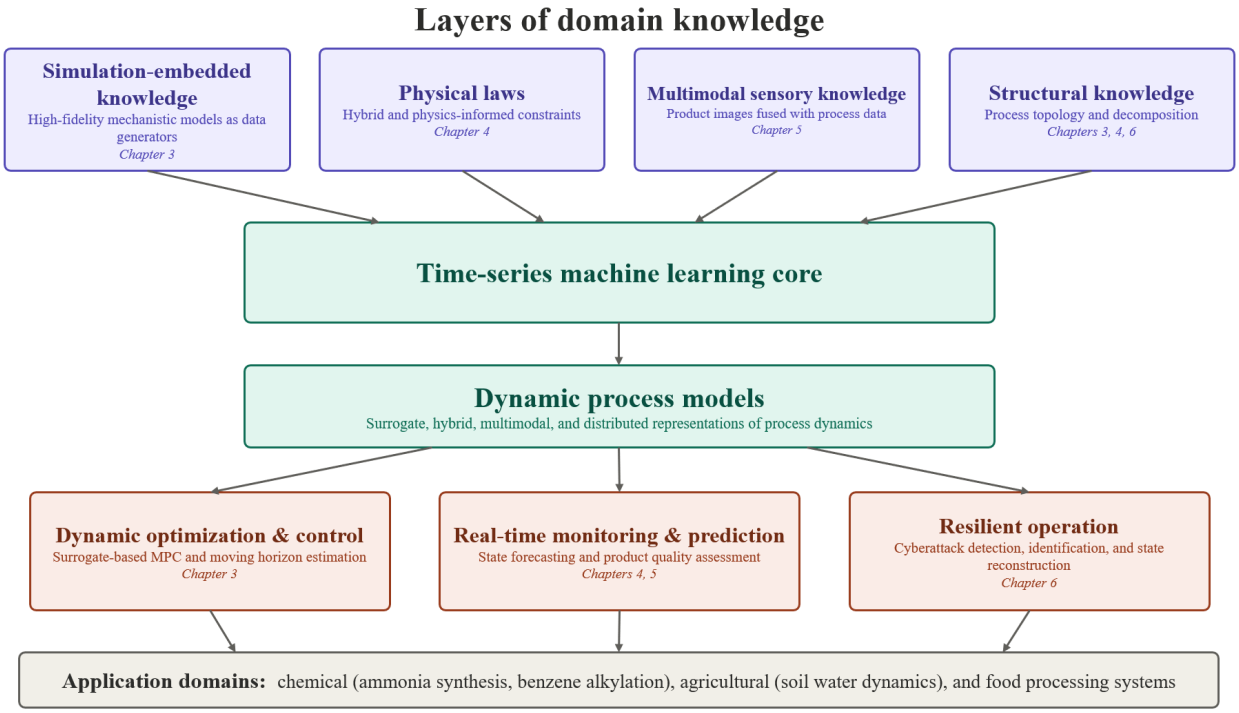

Figure 1.1: *Conceptual framework of the dissertation*

the CFD or multiscale model itself, such that the surrogate acts as an accurate, physically grounded proxy of the high-fidelity model for real-time control applications, rather than an independently learned black-box approximation. This objective is evaluated through the dynamic optimization and predictive control of an *ammonia synthesis and adsorption* process (Chapter 3), where LSTM-based surrogate models, derived from the offline high-fidelity CFD simulations of the process, are integrated into an MPC framework to enable real-time control at a fraction of the computational cost of the original high-fidelity model.

The second layer of knowledge infusion addressed is the case where a simplified, reduced-order physical model of the process is available, but is known to be inaccurate, deviating from the true process behavior in ways that are not captured by the reduced-order equations alone. The second objective of this dissertation is to embed this reduced-order physical knowledge into hybrid and physics-informed neural network (PINN) architectures by learning the residual between the reduced-order model's predictions and the true observed process behavior, rather than discarding the reduced-order model or relying on it directly. This objective is evaluated through a *soil water content (SWC)* case study (Chapter 4), where a simplified,

reduced-order hydrological model of soil water dynamics is combined with residual learning and physics-informed neural network architectures, in both centralized and distributed settings, to correct for the reduced-order model's inaccuracies under limited measurement conditions.

The third layer of knowledge infusion addressed is the case where the underlying process is poorly understood, no physical model, reduced-order or otherwise, is available, and the only accessible process knowledge takes the form of expert feedback together with sensor and imagery data. The third objective of this dissertation is to develop a physics- and data-guided multimodal time-series framework that combines expert-level feedback with image and sensor data as the available forms of process knowledge, in the absence of any governing physical model. This objective is evaluated through a multimodal time-series framework for *plant-based meat extrusion* (Chapter 5), where process time-series data, product imaging, and expert quality assessments are combined to link extrusion process parameters to product quality and enable real-time monitoring and control of a process whose thermophysical transformations remain poorly characterized.

The fourth layer of knowledge infusion addressed is the case where process knowledge is available in the form of the active controller design and real-time sensor measurements themselves, used to detect, localize, and recover from deliberate disruptions to the process rather than to characterize its nominal dynamics. The fourth objective of this dissertation is to combine this controller and sensor-level knowledge with time-series classification techniques to detect cyberattacks, localize the affected subsystem, and recover the process to safe operation in highly integrated systems. This objective is evaluated through a centralized and distributed cyberattack detection, identification, and reconstruction framework applied to a *benzene-alkylation* process (Chapter 6), a highly integrated process used as a representative benchmark for evaluating the scalability of the proposed detection and recovery architecture.

The remainder of this dissertation is organized as follows. Chapter 2 presents the materials and methods common across the dissertation, including the ML architectures, MPC/MHE formulations, computer vision techniques, and physics-informed/hybrid modeling approaches used throughout. Chapter 3 presents the integrated surrogate-based MPC framework de-

veloped for the ammonia synthesis and adsorption process. Chapter 4 presents the hybrid, physics-informed, and distributed modeling framework developed for the soil water content case study. Chapter 5 presents the multimodal time-series framework developed for plant-based meat extrusion. Chapter 6 presents the centralized and distributed cyberattack detection, identification, and reconstruction framework developed for the benzene-alkylation process.

1.5 Contributions

Each of the presented frameworks and case studies has specific contributions toward the overall objective of the dissertation and the specific objectives within each case study. Below is the complete list of the publications:

[1] Bagheri, Amirsalar, Shirin Sheikhi Zadeh, Sajid Alavi, and Davood Babaei Pourkargar. "An Integrated Time-Series Modeling and Computer Vision Framework for Predictive Structure Characterization of Extruded Plant-Based Meat Products." *Artificial Intelligence in Agriculture* (2026). doi:10.1016/j.aiia.2026.03.010

[2] Bagheri, Amirsalar, Thiago Oliveira Cabral, Shirin Sheikhi Zadeh, Sajid Alavi, and Davood Babaei Pourkargar. "Learning-Based Digital Twin for Quality Control in Food Manufacturing: A Case Study in Extrusion-Based Plant Protein Texturization." In *Digital Twins in Manufacturing: Applications and Case Studies*, Elsevier. Accepted for publication, 2026.

[3] Bagheri, Amirsalar, Amirmohammad Ebrahimi, and Davood B. Pourkargar. "Self-Attention Transformer Architectures for Cyberattack Detection and Secure State Reconstruction of Integrated Chemical Process Systems." *Chemical Engineering Research and Design* (2025). doi:10.1016/j.cherd.2025.06.038

[4] Bagheri, Amirsalar, Thiago Oliveira Cabral, and Davood B. Pourkargar. "Integrated Learning-Based Estimation and Nonlinear Predictive Control of an Ammonia Synthesis Reactor." *AIChE Journal* 71, no. 5 (2025): e18732. doi:10.1002/aic.18732

[5] Bagheri, Amirsalar, Andres Patrignani, Behzad Ghanbarian, and Davood B. Pourkargar. "A Hybrid Time-Series and Physics-Informed Machine Learning Framework to Predict Soil Water Content." *Engineering Applications of Artificial Intelligence* 144 (2025): 110105. doi:10.1016/j.engappai.2025.110105

[6] Bagheri, Amirsalar, Thiago O. Cabral, and Davood B. Pourkargar. "Learning-Based Estimation and Predictive Control of an Ammonia Synthesis Reactor." In *2025 American Control Conference (ACC)*, pp. 2311–2316. IEEE, 2025. doi:10.23919/ACC63710.2025.11107844

[7] Bagheri, Amirsalar, Andres Patrignani, Behzad Ghanbarian, and Davood B. Pourkargar. "A Physics-Informed Machine Learning Approach to Predict Soil Water Content for Agricultural Decision-Making." In *2024 American Control Conference (ACC)*, pp. 2–7. IEEE, 2024. doi:10.23919/ACC60939.2024.10644411

[8] Cabral, Thiago O., Amirsalar Bagheri, and Davood B. Pourkargar. "Learning-Based Model Predictive Control of an Ammonia Synthesis Reactor." In *2024 American Control Conference (ACC)*, pp. 45–50. IEEE, 2024. doi:10.23919/ACC60939.2024.10644317

[9] Oliveira Cabral, Thiago, Amirsalar Bagheri, and Davood B. Pourkargar. "Learning-Based Model Reduction and Predictive Control of an Ammonia Synthesis Process." *Industrial & Engineering Chemistry Research* 63, no. 23 (2024): 10325–10342. doi:10.1021/acs.iecr.4c00340

Alongside peer-reviewed publications below is the list of conference presentations:

[1] Bagheri, Amirsalar, Shirin Sheikhi Zadeh, Sajid Alavi, and Davood Babaei Pourkargar. "Multimodal Machine Learning for Predictive Structural Characterization of Plant-Based Meat Products in Food Extrusion Processes. (124c)" In *2025 AIChE Annual Meeting*, AIChE, 2025. https://aiche.confex.com/aiche/2025/meetingapp.cgi/Paper/716369

[2] Bagheri, Amirsalar, Amirmohammad Ebrahimi, and Davood Babaei Pourkargar. "A Data-Driven Transformer-Based Framework for Cyber-Process Incident Detection and

State Reconstruction in Highly Integrated Process Systems. (243d)" In *2025 AIChE Annual Meeting*, AIChE, 2025.
https://aiche.confex.com/aiche/2025/meetingapp.cgi/Paper/715827

[3] Bagheri, Amirsalar, Thiago Oliveira Cabral, and Davood Babaei Pourkargar. "Nonlinear Model Predictive Control of Ammonia Synthesis and Separation Process Using Integrated Surrogate Modeling. (391d)" In *2025 AIChE Annual Meeting*, AIChE, 2025.
https://aiche.confex.com/aiche/2025/meetingapp.cgi/Paper/717712

[4] Bagheri, Amirsalar, Thiago Oliveira Cabral, and Davood Babaei Pourkargar. "Time-Series Integrated Surrogate Modeling for Control of Ammonia Synthesis and Adsorption Processes. (711a)" In *2024 AIChE Annual Meeting*, AIChE, 2024.
https://aiche.confex.com/aiche/2024/prelim.cgi/Paper/693876

[5] Bagheri, Amirsalar, and Davood Babaei Pourkargar. "A Physics-Informed Deep Learning Approach to Predict Soil Water Content for Agricultural Decision-Making.(579a)" In *2024 AIChE Annual Meeting*, AIChE, 2024.
https://aiche.confex.com/aiche/2024/meetingapp.cgi/Paper/688362

[6] Bagheri, Amirsalar. "Data-Driven and Physics-Informed Machine Learning Applications in Real-Time Decision Making and Predictive Control. (364y)" In *2024 AIChE Annual Meeting*, AIChE, 2024.
https://aiche.confex.com/aiche/2024/prelim.cgi/Paper/697139

[7] Ebrahimi, Amirmohammad, Amirsalar Bagheri, and Davood Babaei Pourkargar. "A Distributed Data-Driven Predictive Modeling Approach for Cyber-Process Incident Identification Using Spectral Community Detection. (375d)" In *2024 AIChE Annual Meeting*, AIChE, 2024. https://aiche.confex.com/aiche/2024/meetingapp.cgi/Paper/693898

[8] Bagheri, Amirsalar, Thiago Oliveira Cabral, and Davood Babaei Pourkargar. "Learning-Based Data Reconstruction and Predictive Modeling of an Ammonia Synthesis Process for State Estimation and Control Applications. (207g)" In *2023 AIChE Annual Meeting*, AIChE, 2023.

https://aiche.confex.com/aiche/2023/prelim.cgi/Paper/671548

Chapter 2

Materials and Methods

Numerical time-series ML architectures serve as the core modeling tool across all case studies presented in this dissertation. Depending on the case study, both shallow learning and deep learning time-series models are used, spanning data-driven, hybrid, multimodal, and physics-informed settings, and implemented in either centralized or distributed configurations. Computer vision techniques have additionally been employed as a secondary component to the dynamic time-series models for the plant-based meat extrusion case study, where the time-series image analysis complements the numerical time-series ML component. MPC and MHE have further been employed in Chapters 3 and 6. As the control and estimation layer for the integrated surrogate model, and as a component supporting the cyberattack detection framework for the highly integrated benzene-alkylation system.

2.1 Machine Learning Architectures for Dynamic Process Modeling

The nonlinear, time-varying behavior exhibited by the process systems considered in this dissertation, ranging from the reactor-adsorber dynamics of ammonia synthesis to the thermophysical transformations occurring during food extrusion, motivates the use of time-series machine learning (ML) architectures capable of learning temporal dependencies directly from

process data. Depending on the case study, this dissertation employs feedforward, recurrent, and attention-based deep learning architectures, as well as shallow tree-based learners, selected according to the nature of the available data and the computational demands of the downstream control or monitoring task. This section reviews these architectures in order of increasing capacity to represent temporal dependencies.

2.1.1 Extreme Gradient Boosting (XGBoost)

XGBoost is a highly efficient and scalable ML algorithm that builds decision trees sequentially, with each new tree trained to minimize a specified loss function. It relies on a second-order Taylor series approximation of the loss to improve both convergence speed and computational efficiency[41]. Its robustness on small, noise-contaminated datasets has been demonstrated in prior work on food safety risk assessment and other small-sample learning applications, underscoring its suitability for data-limited settings[42;43].

The optimization objective in XGBoost is given by:

$$\mathcal{L}(t) = \sum_{i=1}^{n} l\big(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)\big) + \Omega(f_t) \tag{2.1}$$

where $l(\cdot)$ is the loss function, typically mean squared error (MSE) for regression tasks, y_i is the true value of the i-th data point, and $\hat{y}_i^{(t-1)}$ is its predicted value at the previous iteration $t-1$. The term $f_t(x_i)$ represents the tree added at iteration t, and $\Omega(f_t)$ is a regularization term that limits model complexity to prevent overfitting:

$$\Omega(f_t) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2 \tag{2.2}$$

where T is the number of leaves in the tree and w_j is the weight of leaf j. Here, γ penalizes trees with more leaves to discourage excessive depth, while λ applies L_2 regularization to the leaf weights to improve generalization.

New splits are added to each tree using a greedy algorithm that selects the split maxi-

mizing the gain in the objective function:

$$\text{Gain} = \frac{1}{2}\left[\frac{G_L^2}{H_L+\lambda} + \frac{G_R^2}{H_R+\lambda} - \frac{(G_L+G_R)^2}{H_L+H_R+\lambda}\right] - \gamma \tag{2.3}$$

where G and H denote, respectively, the first- and second-order gradients of the loss function with respect to the predictions. This split-selection strategy allows XGBoost to capture nonlinear relationships in the data while remaining resistant to overfitting. In time-series modeling, XGBoost generates future predictions by using lagged and engineered temporal features as inputs[44], allowing it to capture complex dependencies and, through feature importance analysis, identify the process variables most relevant to the prediction task.

2.1.2 Light Gradient Boosting Machine (LightGBM)

LightGBM is an advanced implementation of gradient boosting that uses a histogram-based algorithm to speed up training and lower memory usage[45]. Rather than growing trees level-wise as conventional gradient boosting methods do, LightGBM follows a leaf-wise growth strategy that expands the leaf contributing the most to the loss first. This targeted growth improves accuracy by concentrating model refinement where the predictions need it most. As in XGBoost, the LightGBM objective seeks to minimize the following loss function:

$$\mathcal{L}(t) = \sum_{i=1}^{n} l(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \Omega(f_t) \tag{2.4}$$

where $f_t(x_i)$ is the tree added at iteration t. What sets LightGBM apart is its histogram-based partitioning of the data: continuous feature values are grouped into discrete bins, which lowers the computational cost of evaluating candidate splits:

$$\text{Gain}_{\text{hist}} = \max_k \left(\frac{G_k^2}{H_k+\lambda}\right) \tag{2.5}$$

where k indexes the bin, G_k is the gradient, and H_k is the second-order gradient (Hessian) associated with that bin.

Its leaf-wise growth strategy further boosts performance by permitting deeper, more flexible trees:

$$H(t+1) = H(t) + \sum_{j=1}^{L} \frac{G_j}{H_j + \lambda} \tag{2.6}$$

where $H(t)$ is the accumulated Hessian and L is the number of leaves. Because it always expands the leaf offering the greatest loss reduction, LightGBM converges faster and generalizes better than level-wise methods, especially on large datasets. This efficiency and scalability make LightGBM well suited to time-series modeling and texture quantification. Relative to XGBoost, which handles missing data robustly and performs well in recursive multi-step forecasting, LightGBM's leaf-wise growth is particularly effective on sparse and imbalanced data while still guarding against overfitting. In extrusion processes, both models are used to establish quantitative relationships between operational parameters and texture measurements, sharpening the dynamic representation of system behavior across varying operating conditions[46].

2.1.3 Feed Forward Neural Networks (FNNs)

FNNs are a class of neural networks in which the connections from each layer to the next are unidirectional. They are the simplest form of neural network architecture, designed to recognize patterns and make decisions without holding any hidden state. The primary components of FNNs are layers: an input layer, one or more hidden layers, and an output layer. Each neuron in a layer is connected to every neuron of the subsequent layer through weights and biases[47].Figure 2.1 shows the architecture of FNNs

Mathematically, the output y of a neuron can be represented as:

$$y = f\left(\sum_i w_i x_i + b\right) \tag{2.7}$$

Here, x_i denotes the input values, w_i the weights associated with the corresponding connections, b the bias term, and f the activation function.

The activation function, such as the sigmoid or ReLU, introduces nonlinearity into the

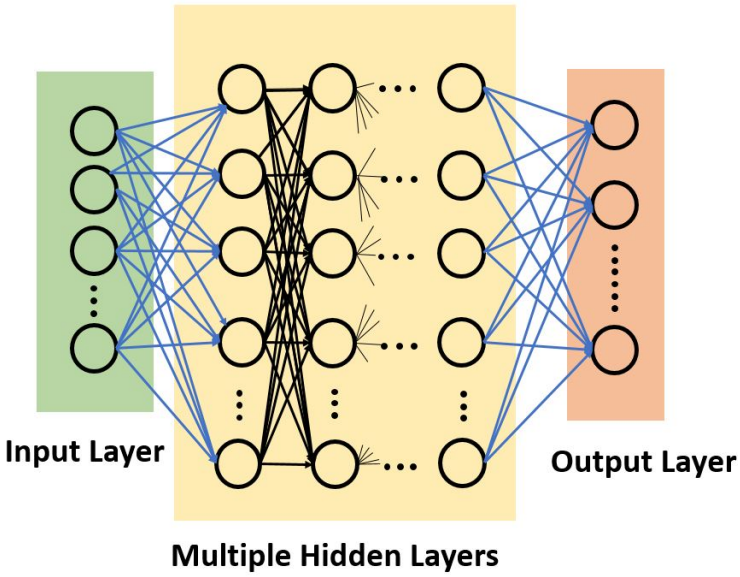

Figure 2.1: *Feed Forward neural Network (FNN)*

model, enabling it to capture nonlinear relationships in the dataset[48].

Training an FNN involves adjusting the weights and biases to minimize the difference between the predicted output and target values. This is achieved using the backpropagation algorithm in conjunction with gradient descent. Backpropagation computes the gradient of the loss function with respect to each weight by applying the chain rule, which is then used by gradient descent to update the weights and biases, minimizing the loss[49].

2.1.4 Recurrent Neural Networks (RNNs)

Although FNNs are efficient and fast, they are not well suited to recognizing patterns in sequential data. Time-series problems and natural language processing tasks are both examples of sequential problems, where each value depends on the values that came before it in the sequence. RNNs, another class of neural networks, are designed to handle this kind of data by introducing a recursive connection from each neuron back to itself, generating a hidden state at every step. The central idea behind RNNs is that this hidden state carries forward information from previous steps in the sequence, allowing the network to retain a memory of what it has already seen.

Vanilla RNNs

The vanilla Recurrent Neural Network (RNN) is the simplest form of RNN architecture. It processes sequential data by maintaining an internal state, or memory, that carries hidden information forward and allows previous outputs to inform future ones. This basic design, however, struggles with long-term dependencies due to the vanishing (or exploding) gradient problem, in which gradients shrink or grow exponentially as they are propagated back over long sequences, making it difficult for the network to learn correlations between distant time steps[48]. Even so, vanilla RNNs have been applied successfully to tasks such as language modeling and time-series forecasting, showing that they can model sequential data effectively when dependencies are short-range. Their simplicity and the intuitive way they mimic sequential memory make them a natural starting point from which more advanced architectures, such as LSTMs and GRUs, were later developed[47]. Figure 2.2 illustrates the structure of a vanilla RNN.

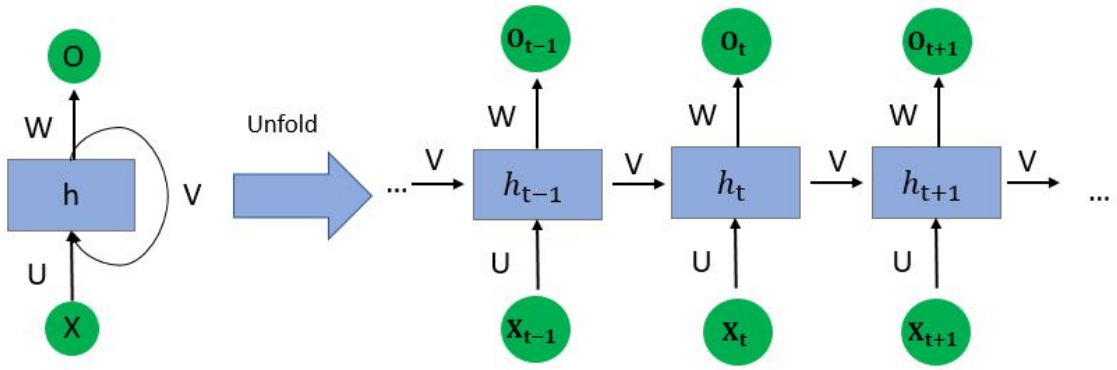

Figure 2.2: *RNN structure*

The hidden state h_t at time t is computed as:

$$h_t = f(W_{hh}h_{t-1} + W_{xh}x_t + b_h) \tag{2.8}$$

where x_t is the input at time t, W_{hh} and W_{xh} are the weight matrices associated with the hidden-to-hidden and input-to-hidden connections, respectively, and b_h is the bias term.

Long Short-Term Memories (LSTMs)

The limitations vanilla RNNs face in learning long sequences motivated the development of LSTMs, a more advanced RNN variant built specifically to capture long-term dependencies. LSTMs introduce memory cells capable of retaining information over extended periods. Their central idea is the cell state, together with three gates (input, forget, and output) that regulate how information flows into, through, and out of the cell[47;50]. The input gate controls how much new information enters the memory cell, the forget gate determines what existing information is discarded, and the output gate governs what is passed forward to the next time step[48]. Figure 2.3 shows the architecture of an LSTM cell.

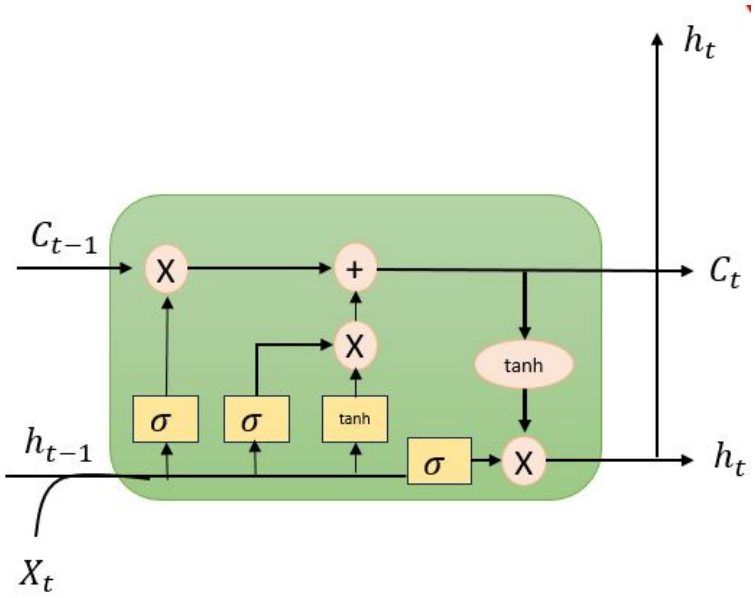

Figure 2.3: *An LSTM Cell*

The information flow through these gates is described mathematically as follows. Given

an input vector x_t, a previous cell state c_{t-1}, and a previous hidden state h_{t-1}:

$$\text{Forget gate:} \quad f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{2.9}$$

$$\text{Input gate:} \quad i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{2.10}$$

$$\tilde{c}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{2.11}$$

$$\text{Cell state update:} \quad c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \tag{2.12}$$

$$\text{Output gate:} \quad o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{2.13}$$

$$h_t = o_t \odot \tanh(c_t) \tag{2.14}$$

where h_{t-1} is the hidden state from the previous time step, σ is the sigmoid function, $\odot$ denotes element-wise multiplication, and the W matrices and b vectors are the trainable parameters of the LSTM.

LSTM networks have several limitations. Their multi-gate, recurrent architecture makes them computationally demanding and slow to train, particularly on large datasets. They can also struggle with very long sequences, where even their gating mechanisms are insufficient to preserve relevant information, leading to degraded performance. In addition, achieving good performance typically requires extensive hyperparameter tuning, which can be both time-consuming and dependent on domain expertise[47;48].

Gated Recurrent Units (GRUs)

GRUs were introduced to address the architectural complexity of LSTMs and the computational cost associated with training them, offering a simpler and more efficient alternative. They achieve this by combining the forget and input gates into a single update gate and merging the cell state with the hidden state[47]. Despite this simplification, GRUs retain the ability to capture long-term dependencies in sequential data through a reduced gating mechanism composed of just two gates: the update gate and the reset gate. The update gate governs how much of the previous state is retained, while the reset gate controls how much new input information is incorporated, allowing GRUs to model sequences effectively

while typically training faster and requiring fewer parameters than LSTMs. Chung et al. demonstrated the effectiveness of GRUs across a range of applications, from general sequence modeling to speech signal processing, highlighting both their versatility and their performance[51]. Figure 2.4 shows the architecture of a GRU cell.

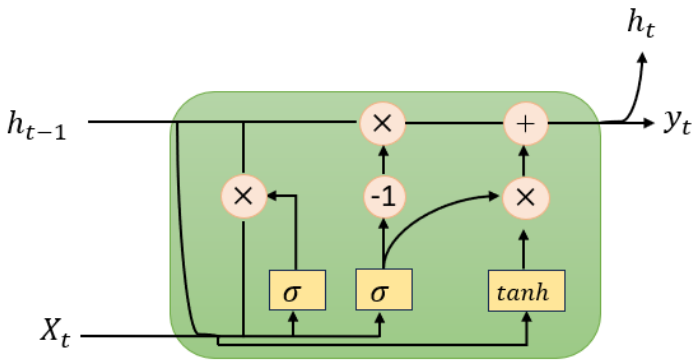

Figure 2.4: *A GRU Cell*

The computations within a GRU cell can be expressed as:

$$\text{Update Gate:} \quad z_t = \sigma(W_z \cdot [h_{t-1}, x_t]) \tag{2.15}$$

$$\text{Reset Gate:} \quad r_t = \sigma(W_r \cdot [h_{t-1}, x_t]) \tag{2.16}$$

$$\text{Candidate Hidden State:} \quad \tilde{h}_t = \tanh(W \cdot [r_t * h_{t-1}, x_t]) \tag{2.17}$$

$$\text{Final Hidden State:} \quad h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t \tag{2.18}$$

where h_{t-1} is the previous hidden state at time step $t-1$, z_t is the update gate vector, r_t is the reset gate vector, $\tilde{h}_t$ is the candidate hidden state, h_t is the resulting new hidden state, and the W matrices and b vectors are the trainable parameters of the GRU.

2.1.5 Sequence-to-Sequence (Seq2Seq) Models

Seq2Seq models were originally developed for natural language processing but have since been widely adopted for time-series forecasting, owing to their ability to map an input sequence

onto an output sequence of a different length. the encoder compresses the input sequence into a fixed-length context vector, and the decoder then reconstructs the output sequence step by step[52;53]. The context vector z is obtained as:

$$z, h_{t-1}^{\text{dec}} = f_{\text{enc}}(x_1, x_2, \ldots, x_T) \tag{2.19}$$

where x_t denotes the input time-series values and f_{enc} is the encoder network. The decoder subsequently uses this context vector to generate the output sequence:

$$y_t^{\text{dec}} = f_{\text{dec}}(h_{t-1}^{\text{dec}}, y_{t-1}, z) \tag{2.20}$$

where h_t^{dec} is the decoder's hidden state at time t.

2.1.6 Self-Attention Decoder-Only Transformers

This sequential dependency of RNNs prevents parallelization across the input sequence and becomes a practical bottleneck when working with long historical windows[54]. Beyond this computational constraint, compressing the entire input history into a single fixed-length context vector also limits the model's ability to selectively recall relevant past states when generating a forecast. The self-attention mechanism overcomes both issues by computing pairwise relationships across all time steps at once, letting any position in the sequence attend directly to any other without passing information through intermediate recurrent states[8;55].

Given an input sequence represented as $\mathbf{X} \in \mathbb{R}^{T \times d_{\text{model}}}$, the self-attention mechanism projects each token into queries $\mathbf{Q}$, keys $\mathbf{K}$, and values $\mathbf{V}$ through learned linear projections, and computes the attention output as:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}}\right)\mathbf{V} \tag{2.21}$$

where d_k is the key dimension, and the scaling factor $\sqrt{d_k}$ keeps large dot products from

pushing the softmax into regions where gradients are close to zero[8]. Multi-head attention extends this by running H independent attention operations in parallel, each attending to a different aspect of the temporal structure, and then concatenating their outputs into a single unified representation[55].

A transformer layer wraps this multi-head attention operation with two further components: a position-wise feedforward network that transforms each token independently through a nonlinear mapping, and residual connections with layer normalization applied around both the attention and feedforward sublayers[8]. Because self-attention is permutation-invariant and has no inherent sense of order, a positional encoding is added to the input embeddings so that temporal position information is preserved before the sequence enters the attention layers[55].

For autoregressive time-series forecasting, the decoder-only architecture keeps only the decoder stack from the standard encoder-decoder transformer and enforces causality through a triangular mask on the attention weight matrix, which prevents each position from attending to future tokens (Figure 2.5). This turns the model into a left-to-right conditional model, where the prediction at time t depends only on observations up through $t-1$. The resulting architecture processes the full observed history in parallel during training while generating predictions autoregressively at inference time, and it is more parameter-efficient than its encoder-decoder counterpart[55].

2.1.7 Physics-Informed and Hybrid Modeling

Residual Learning-Based Hybrid Models

Residual (or series) hybrid models correct a first-principles model by learning the mismatch between its predictions and observed process data, rather than replacing the first-principles model outright. Given a nominal first-principles model f_{phys} and process data (x_i, y_i), a neural network $f_{\text{NN}}(\cdot;\theta)$ is trained to approximate the residual:

$$\hat{y} = f_{\text{phys}}(x) + f_{\text{NN}}(x;\theta), \qquad \theta^* = \arg\min_{\theta} \sum_{i=1}^{n} \left\| y_i - f_{\text{phys}}(x_i) - f_{\text{NN}}(x_i;\theta) \right\|^2 \tag{2.22}$$

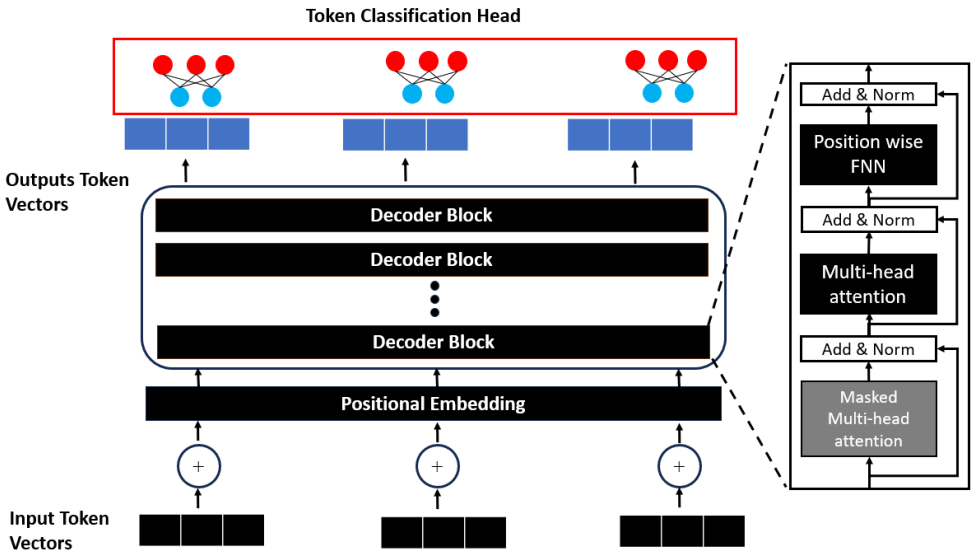

Figure 2.5: *Decoder-only Transformer Architecture*

This formulation constrains the learning task to the (typically smaller and smoother) unmodeled dynamics, improving data efficiency and extrapolation relative to a purely data-driven model trained on y_i directly[56].

Physics-Informed Neural Networks (PINNs)

Rather than correcting a mechanistic model's output, PINNs embed the governing equations directly into the neural network's training objective[24]. For a governing PDE of the general form $\mathcal{N}[u](x,t) = 0$, a neural network $u_\theta(x,t)$ is trained to jointly satisfy the observed data and the PDE residual, evaluated at collocation points (x_c, t_c) sampled from the domain:

$$\mathcal{L}(\theta) = \underbrace{\frac{1}{n}\sum_{i=1}^{n} \left\| u_\theta(x_i,t_i) - u_i \right\|^2}_{\mathcal{L}_{\text{data}}} + \lambda \underbrace{\frac{1}{n_c}\sum_{j=1}^{n_c} \left\| \mathcal{N}[u_\theta](x_{c,j},t_{c,j}) \right\|^2}_{\mathcal{L}_{\text{physics}}} \tag{2.23}$$

where $\mathcal{L}_{\text{data}}$ penalizes deviation from measured data, $\mathcal{L}_{\text{physics}}$ penalizes violation of the governing equation via automatic differentiation of u_θ, and λ weights the physics constraint relative to the data-fitting term[24]. Because the physical law itself acts as a regularizer, PINNs can achieve accurate predictions with less training data and improved generalization outside the training distribution compared to unconstrained networks[25].

Depending on the presented case studies, the formulations of the hybrid and PINNs vary and will be discussed in the next chapters.

2.2 Computer Vision Methods for Process State Characterization

Computer vision (CV) methods have been employed in this dissertation to support the second research objective, which is developing a multimodal process model by integrating image processing with numerical time-series data. In this framework, CV serves as a means of quantifying product texture from image data, providing an additional data modality that complements the numerical time-series measurements collected during process operation. This section discusses each CV component used to achieve this objective.

2.2.1 Canny Edge Detection

Canny edge detection is a multi-stage algorithm that locates edges by identifying regions of an image where intensity changes rapidly. The first stage applies Gaussian filtering to smooth the image and suppress noise, obtained by convolving the input image $I(x, y)$ with a Gaussian kernel $G(x, y)$:

$$I_G(x, y) = I(x, y) * G(x, y) \tag{2.24}$$

where $*$ denotes convolution. Sobel operators are then used to compute the intensity gradient, approximating the partial derivatives of the smoothed image in the x- and y-directions:

$$G_x = \frac{\partial I_G}{\partial x}, \quad G_y = \frac{\partial I_G}{\partial y} \tag{2.25}$$

From these two gradient components, the gradient magnitude $M(x, y)$ and direction $\theta(x, y)$ at each pixel are obtained as:

$$M(x, y) = \sqrt{G_x^2 + G_y^2}, \quad \theta(x, y) = \tan^{-1}\left(\frac{G_y}{G_x}\right) \tag{2.26}$$

Non-maximum suppression is then applied along the gradient direction to thin these edges down to their local maxima. In the final stage, hysteresis thresholding removes weak, spurious edges: pixels with a gradient magnitude above an upper threshold T_{high} are classified as strong edges, those below a lower threshold T_{low} are discarded outright, and pixels falling between the two thresholds are kept only if they connect to a strong edge, which preserves the continuity of detected edges[57].

2.2.2 Hough Transform

The Hough transform is a feature-extraction technique that detects geometric shapes, such as lines and circles, by mapping points from Cartesian space into a parameter space[58]. For line detection, the standard Hough transform represents each line in polar coordinates as:

$$\rho = x\cos(\theta) + y\sin(\theta) \tag{2.27}$$

where ρ is the perpendicular distance from the origin to the line and θ is the angle of the line's normal. Votes are accumulated in this parameter space to identify the lines that appear most prominently in the image. Circular features are detected in an analogous way through the Hough circle transform, which parameterizes each circle by its center coordinates (a, b) and radius r:

$$(x - a)^2 + (y - b)^2 = r^2 \tag{2.28}$$

2.2.3 Gabor Filters for Structure Analysis

Gabor filters are used in this study to extract meaningful structural features from extrusion image samples, allowing for a detailed analysis of spatial patterns such as edges, orientations, and structural variations. A Gabor filter is a linear filter defined as:

$$G(x, y; \lambda, \theta, \psi, \sigma, \gamma) = \exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right) \cos\left(2\pi \frac{x'}{\lambda} + \psi\right) \tag{2.29}$$

where the rotated coordinates x' and y' are given by:

$$x' = x\cos\theta + y\sin\theta \tag{2.30}$$

$$y' = -x\sin\theta + y\cos\theta \tag{2.31}$$

Here, λ is the wavelength of the sinusoidal component and sets the scale at which the filter operates, while θ specifies the orientation of the features it detects. The phase offset ψ shapes the symmetry of the filter's response, σ controls the spatial extent of the Gaussian envelope, and γ sets the spatial aspect ratio, determining the filter's elongation[59].

2.3 K-means spatial clustering

K-means is an unsupervised learning algorithm that partitions n observations into K disjoint clusters, $\{C_1, C_2, \ldots, C_K\}$, by minimizing the total within-cluster sum of squares (WCSS)[60]:

$$J = \sum_{k=1}^{K} \sum_{\mathbf{x}_i \in C_k} \|\mathbf{x}_i - \boldsymbol{\mu}_k\|^2 \tag{2.32}$$

where $\mathbf{x}_i$ is the feature vector for observation i and $\boldsymbol{\mu}_k$ is the centroid of cluster C_k. During training, each observation is assigned to the nearest centroid using Euclidean distance, and each centroid is updated to the mean of its assigned observations. This process repeats until the cluster assignments no longer change. Because the clustering features may have different units and magnitudes, all variables are standardized to zero mean and unit variance before

clustering:

$$\tilde{x}_{i,j} = \frac{x_{i,j} - \bar{x}_j}{\sigma_j} \tag{2.33}$$

where $x_{i,j}$ is the value of feature j for observation i, $\bar{x}_j$ is the mean of feature j, and σ_j is its standard deviation. The number of clusters, K, is determined using the elbow method, in which the optimal value K^* is identified as the point where additional clusters yield only marginal reductions in $J(K)$[61;62].

2.4 Evaluation Metrics

2.4.1 Regression Metrics: NRMSE, MAE, and Mean Bias

The developed time-series model performance is assessed using the normalized root mean squared error (NRMSE), mean absolute error (MAE), and mean bias, computed over n predictions as:

$$\text{NRMSE} = \frac{1}{y_{\max} - y_{\min}} \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}, \qquad \text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{2.34}$$

$$\text{Mean Bias} = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i) \tag{2.35}$$

where y_i and $\hat{y}_i$ denote the true and predicted values, respectively, and $y_{\max}, y_{\min}$ are the extrema of the observed data used for normalization. NRMSE enables comparison of prediction error across variables with different scales and units, while MAE reports the average magnitude of error in the original units of the variable[63]. Mean bias indicates the direction of systematic over- or under-prediction and complements NRMSE and MAE, which are insensitive to error sign[64].

2.4.2 Classification Metrics: ROC, AUC, F1, Accuracy, Precision, and Recall

Classification performance is evaluated using accuracy, precision, recall, and the F1 score, computed from the number of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN):

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}, \qquad \text{Precision} = \frac{TP}{TP + FP}, \qquad \text{Recall} = \frac{TP}{TP + FN} \tag{2.36}$$

$$F_1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{2.37}$$

Precision and recall together capture the trade-off between false alarms and missed detections, and the F1 score summarizes this trade-off as their harmonic mean, which is particularly informative under class imbalance[65]. The Receiver Operating Characteristic (ROC) curve plots the true positive rate against the false positive rate across classification thresholds, and the Area Under the Curve (AUC) summarizes this curve in a single value representing the model's ability to discriminate between classes independent of a fixed threshold[66].

2.5 Jaccard index

The Jaccard Index evaluates the overlap between predicted and actual sets of attacked sensors:

$$\text{Jaccard Index} = \frac{|D \cap G|}{|D \cup G|} \tag{2.38}$$

where D is the set of predicted attacked sensors, and G is the ground truth. This metric penalizes false positives and negatives, making it suitable for multi-label attack localization. Values close to 1 denote strong agreement, while lower values indicate overprediction or missed detections.

2.6 Explained variance (EV)

Explained variance (EV) quantifies the proportion of variance in the true signal captured by the reconstructed output:

$$\mathrm{EV} = 1 - \frac{\sum_{i=1}^{N}(\bar{y}_i - \tilde{y}_i)^2}{\sum_{i=1}^{N}(\bar{y}_i - \bar{y}_{\mathrm{avg}})^2} \tag{2.39}$$

where $\bar{y}_{\mathrm{avg}}$ is the mean of the true values. EV ranges from $-\infty$ to 1, with values closer to 1 indicating better reconstruction. Unlike RMSE, which focuses on absolute error, EV emphasizes the model's ability to capture dynamic trends in the system.

2.7 Model Predictive Control (MPC)

Model predictive control (MPC) is an advanced control strategy that determines control actions by solving an optimization problem at every time step, seeking to minimize a cost function over a finite prediction horizon subject to the system's dynamics and constraints. For nonlinear systems, this optimization typically takes the form of a nonlinear quadratic cost:

$$J(U) = \int_{t}^{t+N} \left(x(\tau)^T Q x(\tau) + u(\tau)^T R u(\tau)\right) d\tau \tag{2.40}$$

where $x(\tau)$ is the system state at time τ, $u(\tau)$ is the control input, Q and R are positive definite weighting matrices, and N is the prediction horizon. This integral accumulates cost over the horizon, with the two quadratic terms penalizing deviations in the state and excessive control effort, respectively. The corresponding MPC optimization problem is:

$$\min_{U} \int_{t}^{t+N} \left(x(\tau)^T Q x(\tau) + u(\tau)^T R u(\tau)\right) d\tau \tag{2.41}$$

subject to the nonlinear system dynamics:

$$\dot{x}(\tau) = f(x(\tau), u(\tau)), \quad \tau \in [t, t+N] \tag{2.42}$$

along with any additional constraints imposed on the states and control inputs. Solving this problem yields a sequence of control actions over the horizon N, of which only the first is applied to the system; the entire optimization is then re-solved at the next time step using updated state measurements or estimates, a strategy known as receding horizon control. This ability to explicitly handle multivariable dynamics, constraints, and nonlinearities within a single optimization framework makes MPC particularly well suited to complex process control applications[67].

2.8 Moving Horizon Estimation (MHE)

Moving horizon estimation (MHE) is the estimation-side counterpart to MPC. Rather than selecting future control actions over a forward-looking horizon, MHE reconstructs the current system state by solving an optimization problem over a backward-looking window of past measurements, re-solved and shifted forward at every time step[67]. Given a moving window of length N containing the most recent measurements $y(\tau)$ and inputs $u(\tau)$, MHE estimates the state trajectory $\hat{x}(\tau)$ that best explains the observed data by minimizing a cost analogous in structure to the MPC objective in Eq. (**??**), but penalizing measurement and process-noise residuals rather than state deviation and control effort:

$$J(\hat{X}) = \Gamma\big(\hat{x}(t-N)\big) + \int_{t-N}^{t} \Big(\big\| y(\tau) - h(\hat{x}(\tau)) \big\|_{Q_e}^2 + \big\| w(\tau) \big\|_{R_e}^2 \Big)\, d\tau \tag{2.43}$$

where $h(\cdot)$ is the measurement model, $w(\tau)$ is the process disturbance acting on the system dynamics, Q_e and R_e are positive definite weighting matrices penalizing measurement and process noise, respectively, and $\Gamma(\cdot)$ is the arrival cost, a summary term that approximates the information contained in measurements taken before the current window that would otherwise be discarded when the horizon shifts forward. The corresponding MHE optimization problem is:

$$\min_{\hat{X}} \; \Gamma\big(\hat{x}(t-N)\big) + \int_{t-N}^{t} \Big(\big\| y(\tau) - h(\hat{x}(\tau)) \big\|_{Q_e}^2 + \big\| w(\tau) \big\|_{R_e}^2 \Big)\, d\tau \tag{2.44}$$

subject to the same nonlinear system dynamics used in the MPC formulation:

$$\dot{\hat{x}}(\tau) = f(\hat{x}(\tau), u(\tau)) + w(\tau), \quad \tau \in [t - N, t] \tag{2.45}$$

and any known bounds on the state and disturbance variables. Solving this problem yields an estimate $\hat{x}(t)$ of the current system state, which is then passed to the MPC layer in place of a direct state measurement whenever full-state feedback is unavailable . As with MPC, the estimation window is shifted forward and the optimization re-solved at every time step, allowing MHE to continually incorporate new measurements while bounding the computational burden of the estimation problem[67].

2.9 Closed-loop performance metrics

Evaluating and comparing the closed-loop performance of MPC frameworks requires quantitative metrics that capture both the quality of setpoint tracking and the effort exerted by the controller to achieve it. Three complementary metrics are employed throughout this dissertation: the integral of squared error (ISE), the integral of squared control (ISC), and the dynamic performance index (DPI).

The ISE quantifies the cumulative tracking performance of a regulated output $v(t)$ with respect to its targeted steady-state value v_{ss} over the closed-loop simulation horizon,

$$\mathrm{ISE}_v = \sum_{t=1}^{T} \left(\frac{v(t) - v_{ss}}{v_{ss}} \right)^2 \Delta t, \tag{2.46}$$

where T is the number of sampling instants and Δt the sampling interval. Owing to the squared error, the ISE penalizes large deviations disproportionately and is therefore dominated by transient excursions such as overshoot or slow initial convergence; a lower ISE indicates that the controller drives the output to its setpoint faster and with smaller deviations. The normalization by the steady-state value renders the metric dimensionless, enabling aggregation and comparison across outputs of different physical scales and units.

The ISC analogously quantifies the cumulative control effort exerted through a manipulated input $w(t)$,

$$\mathrm{ISC}_w = \sum_{t=1}^{T} \left(\frac{w(t) - w_{ss}}{w_{ss}} \right)^2 \Delta t. \tag{2.47}$$

A lower ISC reflects smoother and less energetic actuation, which translates to reduced actuator wear and, in process systems where the manipulated inputs represent material or energy duties, directly to lower operational expenditure. Together, the two metrics expose the fundamental trade-off of feedback control: tight tracking (low ISE) is typically purchased with aggressive control action (high ISC), and vice versa.

To adjudicate this trade-off with a single scalar measure, the individual contributions are aggregated into the dynamic performance index,

$$\mathrm{DPI} = \sum_{w \in \mathcal{U}} \mathrm{ISC}_w + \sum_{v \in \mathcal{Y}} \mathrm{ISE}_v, \tag{2.48}$$

where $\mathcal{U}$ and $\mathcal{Y}$ denote the sets of manipulated inputs and regulated outputs, respectively. The DPI thus mirrors the quadratic objective of the MPC itself, representing the realized closed-loop cost accumulated over the entire simulation; among controllers evaluated under identical scenarios, the one attaining the lowest DPI achieves the most favorable overall balance between tracking performance and control effort[68].

Chapter 3

Surrogate-Based Real-Time Model Predictive Control of Ammonia Synthesis and Adsorption Processes

3.1 Introduction

Ammonia, the second most produced chemical compound, is crucial for manufacturing fertilizers, pesticides, plastics, fabrics, explosives, and dyes[69]. Its versatile properties make it indispensable for agricultural applications, energy transportation, and as a potential fuel source[70;71], thereby supporting a sustainable chemical-energy supply chain. Ammonia is synthesized through the Haber-Bosch process, a catalytic reversible reaction between hydrogen and nitrogen under high pressure and temperature. The source of hydrogen determines whether the ammonia is classified as blue or green[72].

Ammonia production has evolved significantly due to growing fertilizer demand and technological advancements. The non-energy use of natural gas in ammonia production accounts for about 4% of the global natural gas supply, with annual production reaching 183 million tons. This production is expected to grow by 3–4 times over the next 25 years, driven by demand for hydrogen transport and power generation[73]. The role of ammonia in global food

security and its renewed interest as an energy vector in the hydrogen economy have also been investigated[74]. However, most ammonia production currently relies on fossil fuels, primarily natural gas, exposing production costs to volatile fuel markets and rendering the process economically vulnerable to energy price fluctuations and emerging carbon pricing schemes[75–77]. Managing this dependence on fossil feedstocks has therefore become an economic imperative as much as an environmental one, motivating alternative production pathways; among these, green ammonia, synthesized using renewable electricity, offers a route to decouple production costs from fossil fuel markets while leveraging the declining costs of renewable energy[78].

To meet the growing demand for ammonia while addressing economic concerns, energy-efficient production and separation processes are essential. In particular, understanding the coupled dynamics of ammonia synthesis in industrial packed-bed reactors (PBRs) and downstream ammonia recovery is crucial for optimizing process efficiency through precise process control. Although the Haber–Bosch process has been used extensively, deeper insights into reactor dynamics remain necessary. Moreover, conventional condensation-based ammonia recovery is energy-intensive at low conversions, motivating the integration of adsorption-based separation, such as temperature swing adsorption (TSA), directly with the synthesis reactor[79;80]. Real-time optimization and control of such integrated reactor–adsorber systems require a robust process model that can predict temporal dynamics, which are influenced by the interplay of surface reactions, thermodynamics, adsorption equilibria, and transport phenomena across interconnected units. These complex interactions create challenges for real-time decision-making, especially in managing reactor hotspot temperatures, species concentrations, and adsorption column temperatures.

Building on previous studies on multiscale modeling of ammonia synthesis and adsorption[68;81;82], and given that such high-fidelity models face significant computational limitations for real-time predictive control, this chapter investigates the development of time-series surrogate models trained on high-fidelity simulations of the multiscale model. In the absence of sufficient operational data from the reactor and adsorber, these offline simulations serve as a first layer of physical interpretation embedded in the surrogate models, generating a multivariate time-series dataset that captures the nonlinear dynamics of the integrated process.

Conceptualizing the process dynamics as a multivariate autoregressive time-series problem, we train sequence-to-sequence LSTM models (introduced in Chapter 2), as dynamic surrogate models representing the process system as a fast replica in the MPC closed-loop.

The proposed integrated surrogate modeling system is developed from both centralized and Integrated perspectives. In the centralized approach, a single LSTM model is trained to replicate the system-wide behavior of the entire reactor–adsorber process, whereas in the Integrated approach, dedicated subsystem models are co-trained simultaneously with synchronized information sharing, ensuring the surrogate predictions respect the physical boundaries between the process units. The developed surrogate models are then incorporated within an MPC framework, in which an iterative dynamic optimization problem is solved to determine the optimal manipulated inputs, guiding the process outputs toward a targeted steady state through closed-loop simulations.

3.2 Multiscale modeling of the integrated ammonia synthesis and adsorption process

Dynamic optimization and optimal control hinge on an accurate process model that describes the system's nonlinear behavior. In this case study, the process consists of an ammonia synthesis packed-bed reactor (PBR) integrated with a temperature swing adsorption (TSA) unit for downstream ammonia recovery, in which the reactor effluent is conditioned through an intermediate heat exchanger before entering the adsorption column, while a second column undergoes regeneration (Figure 3.1). A detailed multiscale model of this integrated system is developed based on heat, momentum, and mass balances, coupling continuum-scale transport in the porous beds with a discrete molecular description of the surface reaction kinetics, which serve as the link between the two scales. Within the adsorption column, intraparticle mass transport is resolved at the pellet scale, and the ammonia uptake is described by a Langmuir-type adsorption equilibrium with a temperature-dependent equilibrium constant. Combined with computational fluid dynamics (CFD) tools, this multiscale model constitutes

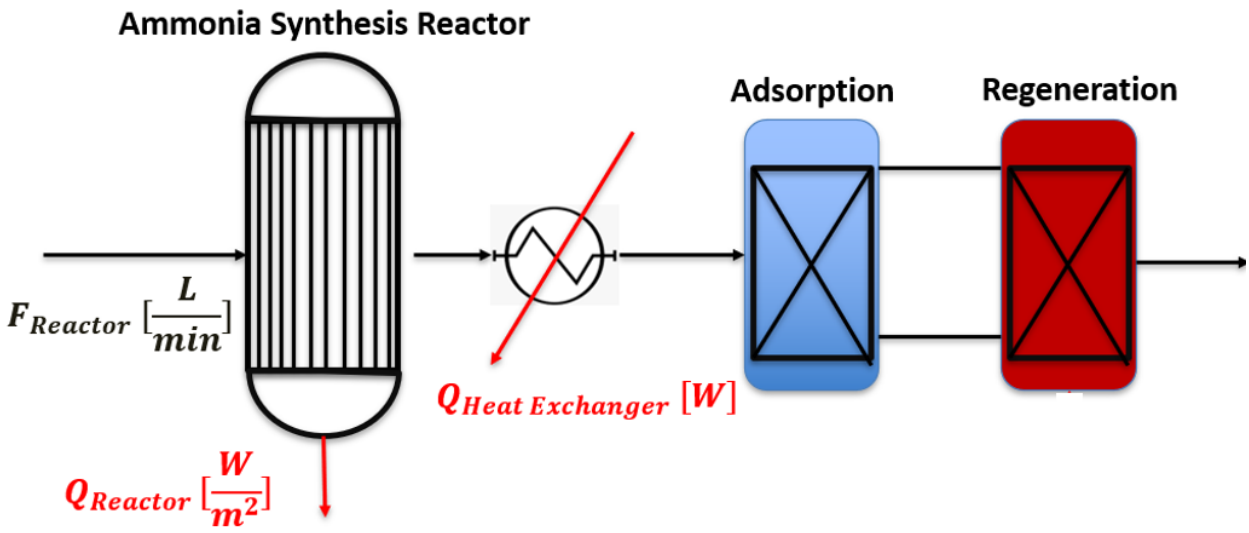

Figure 3.1: *Schematic of the integrated ammonia synthesis and adsorption process, comprising the Haber–Bosch PBR, intermediate heat exchanger, and TSA adsorption–regeneration columns.*

a digital twin of the integrated ammonia synthesis and adsorption process that replicates the behavior of the real system. The implementation of digital twins facilitates a deeper understanding of systems, prediction of failures before they occur, and testing of solutions in a virtual environment, thereby reducing costs and downtime while improving performance[?] .

Remark 1: The development of the multiscale model is a separate topic within our research group. We abstain from detailing the multiscale model here, as this chapter focuses on time-series surrogate modeling of the integrated process using LSTMs; the full details can be found in our recent papers[68;81;82].

3.3 Problem Statement

The developed multiscale model of the integrated ammonia synthesis and adsorption process is solved using the finite-element method (FEM) incorporated in COMSOL Multiphysics[83]. Despite the robustness and accuracy of this method, its slow real-time resolution and high computational demand cannot contend with the required response time of a real-time dynamic optimization algorithm. These limitations primarily stem from the high computational

cost of solving large-scale problems, which require significant memory and processing power, especially for three-dimensional models and nonlinear material behavior[84]. Additionally, the iterative nature of optimization processes and the detailed simulations required at each step exacerbate these computational challenges. For the integrated process, these limitations become even more restrictive, as coupling the synthesis reactor with the downstream heat exchanger and temperature swing adsorption unit enlarges the computational domain and expands the set of operational degrees of freedom to four manipulated inputs: the reactor feed flow rate ($F_{Reactor}$), the heat removed from the reactor ($Q_{Reactor}$), the heat exchanger duty ($Q_{Heat\,Exchanger}$), and the heat supplied to the adsorber ($Q_{Adsorber}$). As a result, the multiscale model is impractical for real-time decision-making[81].

Time-series machine learning offers a means to bridge the computational gap between multiscale modeling and real-time control and optimization. Accordingly, LSTM-based surrogate models trained on offline high-fidelity simulations of the multiscale model replace the FEM solver within the dynamic optimization layer. For the synthesis reactor, the control objective is to regulate the maximum reactor temperature (T_{max}) and the outlet ammonia concentration (C_{NH_3}) by manipulating the removed heat ($Q_{Reactor}$) and the feed flow rate ($F_{Reactor}$); for the integrated process, the objective is extended to maximizing ammonia adsorption while minimizing the heat exchanger duty, with $Q_{Heat\,Exchanger}$ and $Q_{Adsorber}$ serving as additional manipulated inputs. In both cases, the surrogate model is employed for MPC optimization while the multiscale model serves as the plant for MPC implementation, ensuring the optimization layer remains computationally tractable without sacrificing the fidelity of the closed-loop evaluation.

3.4 High-fidelity simulation of the multiscale model

Gathering sufficient process data is crucial for training the surrogate models, as capturing the full range of nonlinear behaviors within the process data is key to ensuring accurate optimization performance. Given the absence of operational data from the integrated process, this dataset is generated through offline high-fidelity simulations of the multiscale model executed

in COMSOL Multiphysics across a comprehensive array of input conditions. Four process inputs are varied to excite the coupled reactor–adsorber dynamics: the reactor feed flow rate ($F_{Reactor}$ [L/min]), the heat removed from the reactor ($Q_{Reactor}$ [W/m^2]), the heat exchanger duty ($Q_{Heat\,Exchanger}$ [W]), and the heat supplied to the adsorber ($Q_{Adsorber}$ [W/m^2]). A total of 150 simulations are conducted under different combinations of these input values, each simulating 1000 minutes of process operation, yielding a dataset that encompasses the dynamic responses of the integrated system.

Although 500-minute simulations at a uniform unit sampling interval would be expected to produce approximately 1.5×10^5 datapoints, the raw dataset contains roughly 9×10^5 datapoints. This discrepancy arises from the adaptive time-stepping of the FEM solver, which refines the temporal resolution in regions of steep dynamics and coarsens it elsewhere. The resulting non-uniform timesteps produce an irregular time-series dataset, which is incompatible with the fixed-interval sequence structure required by the LSTM-based surrogate models and disturbs the synchronization between the interconnected surrogate subsystems. To address this, the Gappy Data Regeneration (GDR) approach, detailed in the previous work[68], is employed to nonlinearly interpolate the offline simulations onto a uniform timestep ($\Delta t = 1$), yielding a regular multivariate time-series dataset of 1.5×10^5 datapoints suitable for surrogate model training.

The regularized dataset comprises 144 distinct dynamic simulations of the integrated system, corresponding to different combinations of the heat-related inputs, and is partitioned into training, validation, and test sets containing 70%, 20%, and 10% of the simulations, respectively. Importantly, the partitioning is performed at the simulation level, where each open-loop simulation is assigned to a single set in its entirety rather than being segmented across sets, which preserves the temporal consistency of the sequences and eliminates the risk of data leakage between training and evaluation. Furthermore, the simulations are allocated such that both slow and rapid dynamic regimes are represented in all three sets, preventing any dynamic regime from being unrepresented during training and ensuring that the validation and test evaluations reflect the full range of process behaviors.

3.5 Integrated surrogate modeling of the reactor–adsorber system

The integrated reactor–adsorber system is modeled using sequence-to-sequence LSTM architectures that map the trajectories of the manipulated inputs to the temporal evolution of the process outputs. Two surrogate modeling strategies are developed and compared for this purpose: a centralized strategy and an integrated strategy.

3.5.1 Centralized surrogate modeling

In the centralized strategy, a single sequence-to-sequence LSTM model is trained on the high-fidelity dataset to replicate the system-wide behavior, treating the reactor, heat exchanger, and adsorption–regeneration columns as a unified process unit. Let the manipulated input vector of the integrated process be

$$u(t) = [\, F_{Reactor}(t),\ Q_{Reactor}(t),\ Q_{Heat\,Exchanger}(t)]^{\top}\,, \tag{3.1}$$

and let $y(t)$ denote the vector of process outputs of the integrated system, comprising the reactor's maximum temperature, outlet ammonia concentration, total absorbed ammonia, and outlet ammonia concentration from the column.

$$y(t) = [\, T_{Max,Reactor}(t),\ C_{OutletNH3,Reactor}(t),\ M_{Ave,NH3Absorbed}(t)\,]^{\top}\,, \tag{3.2}$$

Given a memory window of length k, the centralized surrogate model f_{θ_c}, parameterized by the LSTM weights θ_c, predicts the output sequence over the prediction horizon m as

$$\hat{y}(t+1), \ldots, \hat{y}(t+m) = f_{\theta_c}(u(t-k+1), \ldots, u(t+m),\ y(t-k+1), \ldots, y(t))\,, \tag{3.3}$$

where the encoder summarizes the past input–output sequences into a latent context and the decoder unrolls the predictions across the horizon. The model parameters are obtained

by minimizing the mean squared prediction error over the N training sequences,

$$\mathcal{L}(\theta_c) = \frac{1}{Nm} \sum_{i=1}^{N} \sum_{j=1}^{m} \left\| y^{(i)}(t+j) - \hat{y}^{(i)}(t+j) \right\|_2^2 . \tag{3.4}$$

While conceptually straightforward, this monolithic representation must capture all process interactions within a single set of parameters θ_c, and its accuracy degrades as the input–output dimensionality of the integrated system grows.

3.5.2 Integrated surrogate modeling with synchronized co-training

In the Integrated strategy, the process is decomposed into two subsystems following its physical topology (Figure 3.2): Subsystem 1 comprises the synthesis reactor, driven by $u_1(t) = [\, F_{Reactor}(t),\, Q_{Reactor}(t) \,]^\top$, while Subsystem 2 comprises the heat exchanger and the adsorption–regeneration columns, driven by $u_2(t) = [\, Q_{Heat\,Exchanger}(t),\, Q_{Adsorber}(t) \,]^\top$. A dedicated sequence-to-sequence LSTM model is assigned to each subsystem,

$$\hat{y}_1(t+1), \ldots, \hat{y}_1(t+m) = f_{\theta_1}(u_1,\, y_1)\,, \tag{3.5}$$

$$\hat{y}_2(t+1), \ldots, \hat{y}_2(t+m) = f_{\theta_2}(u_2,\, y_2,\, \hat{z})\,, \tag{3.6}$$

where y_1 and y_2 denote the output vectors of the respective subsystems, and

$$z(t) = [\, T_{Out_{Reactor}}(t),\; C_{NH_3,Reactor}(t) \,]^\top \tag{3.7}$$

is the vector of boundary variables of the physical stream connecting the two units, predicted by the reactor model and communicated to the downstream model at the designated synchronization points. This decomposition allows each model to focus on the distinct dynamics of its respective process unit, mitigating the dimensionality burden faced by the centralized representation.

Crucially, the two subsystem models are not trained in isolation. Instead, the end-to-end

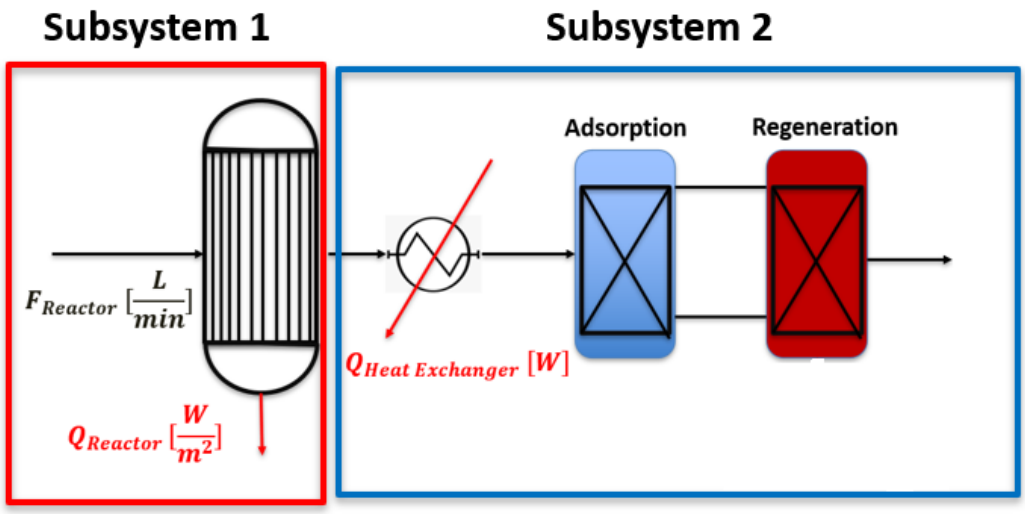

Figure 3.2: *Subsystem decomposition of the integrated ammonia synthesis and adsorption process*

system is trained simultaneously by minimizing the coupled loss

$$\mathcal{L}(\theta_1, \theta_2) = \mathcal{L}_1(\theta_1) + \mathcal{L}_2(\theta_2), \tag{3.8}$$

where $\mathcal{L}_1$ and $\mathcal{L}_2$ are the mean squared prediction errors of Subsystem 1 and Subsystem 2, respectively, and the boundary predictions $\hat{z}$ generated by the reactor model are propagated into the downstream model during training. Through this synchronized information sharing, prediction errors at the subsystem interface are penalized within the end-to-end training, constraining the coupled surrogate models to respect the process boundaries and preventing the decomposed representation from drifting toward physically inconsistent predictions while preserving the interdependencies between the reactor and the downstream separation.

Both the centralized and Integrated surrogate models are trained using the ADAM optimizer[85] with adaptive learning rates, minimizing the mean squared error loss over the regularized time-series dataset.

3.6 Surrogate-based MPC of the integrated process

The developed integrated surrogate models are subsequently embedded within an MPC framework to regulate the reactor–adsorber system (Figure 3.3). At each sampling interval t, the controller solves the finite-horizon dynamic optimization problem

$$\min_{u(t+1),\ldots,u(t+m)} J_{Total} = \underbrace{\left(x_r^\top P_r\, x_r + u_r^\top Q_r\, u_r\right)}_{\text{Reactor}} + \underbrace{\left(x_c^\top P_c\, x_c + u_c^\top Q_c\, u_c\right)}_{\text{Adsorber}} \tag{3.9}$$

$$\text{s.t.} \quad \hat{y}(t+1), \ldots, \hat{y}(t+m) = f_\theta(u,\, y)\,, \tag{3.10}$$

$$u_{\min} \le u(t+j) \le u_{\max}, \quad j = 1, \ldots, m, \tag{3.11}$$

where x_r and x_c denote the deviations of the reactor and adsorber outputs from their targeted steady states, $u_r = [\,F_{Reactor},\, Q_{Reactor}\,]^\top$ and $u_c = [\,Q_{Heat\,Exchanger}]^\top$ are the corresponding manipulated input deviations, and P_r, Q_r, P_c, and Q_c are positive definite weighting matrices that balance the competing control objectives: regulating the maximum reactor temperature, stabilizing the outlet ammonia concentration, and maximizing ammonia adsorption while minimizing the heat exchanger duty. The prediction constraint (3.10) is enforced through the integrated surrogate models, whose single-inference sequence-to-sequence predictions span the full prediction horizon m, thereby removing the computational bottleneck of embedding the FEM-based multiscale model in the optimization loop. The dynamic optimization problem is formulated and solved in Pyomo[86], with the surrogate model inference performed in Keras/TensorFlow, and the two environments exchanging predictions and decision variables at every control step.

A practical challenge arises in the implementation phase of the closed-loop simulations. Applying the optimal inputs directly to the FEM-based multiscale model at every sampling instant is computationally prohibitive, where the high-fidelity simulation of the integrated system is intensive enough that it would prolong each MPC iteration far beyond the intended sampling interval, defeating the purpose of the surrogate-based optimization. To circumvent

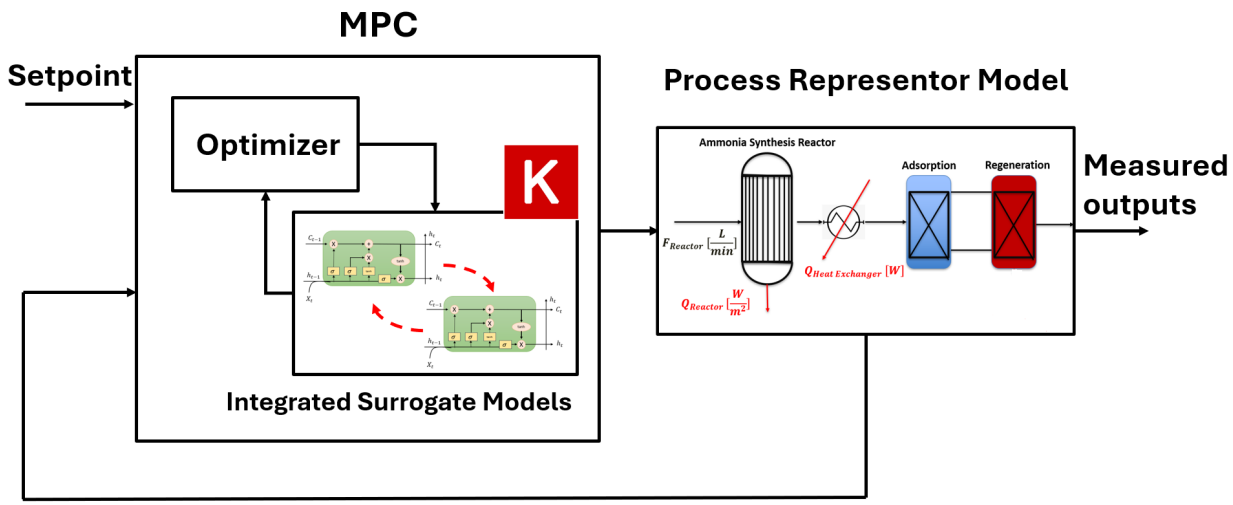

Figure 3.3: *Closed-loop architecture of the surrogate-based MPC for the integrated ammonia synthesis and adsorption process*

this, the closed-loop simulations employ a neural network replica of the process in place of the multiscale model. This replica is the GDR network used for reconstructing the offline simulation data ; as such, it is not a dynamic model but a static input–output mapping of the high-fidelity simulations, returning the process outputs associated with the applied inputs at each control step. Following the receding-horizon principle, only the first element of the optimal input sequence is applied to the process replica, the resulting outputs are fed back to the controller, and the optimization is repeated at the next sampling interval.

It must be acknowledged that substituting the process with an FNN-based replica limits the closed-loop evaluation: the replica does not independently verify the surrogate models against the full multiscale dynamics, and full implementation of the framework on the high-fidelity process model, or ultimately on an experimental system, remains necessary for verification purposes and is deferred to future work. Within this scope, the contribution of the present chapter is to establish the feasibility of integrated surrogate modeling based on high-fidelity simulation data as a means of breaking the computational bottleneck that precludes real-time dynamic optimization of the integrated ammonia synthesis and adsorption process.

3.7 Integrated Surrogate Model Evaluations

The predictive performance of the centralized and integrated surrogate models is assessed through open-loop simulations on test conditions, evaluating each model's ability to reproduce the dynamic responses of the three key process outputs: the maximum reactor temperature ($T_{max}^{Reactor}$), the outlet ammonia concentration of the reactor ($C_{NH_3}^{Reactor}$), and the total amount of ammonia adsorbed ($N_{NH_3}^{Absorbed}$). The prediction accuracy is quantified using NRMSE, enabling a consistent comparison across outputs with different scales and units.

Figure 3.4 illustrates representative open-loop predictions of both surrogate strategies against the high-fidelity ground truth under two test operating conditions. Both models capture the dominant process dynamics, including the rapid initial temperature rise toward the reaction equilibrium and the subsequent settling of the outlet concentration. However, the integrated model tracks the ground-truth trajectories closely across the entire simulation window, while the centralized model exhibits noticeable deviations: it underpredicts the maximum reactor temperature at later times, displays oscillatory drift in the outlet ammonia concentration, and underestimates the cumulative ammonia uptake in the adsorption column. The adsorption-side error is particularly consequential, as the cumulative uptake directly reflects the separation performance that the MPC framework is ultimately intended to maximize.

The aggregate comparison in Figure 3.5 confirms these observations across the test set. The integrated surrogate model outperforms its centralized counterpart on all three outputs. Notably, the largest errors of both models occur in the maximum reactor temperature, reflecting the sharp thermal dynamics of the exothermic synthesis reaction. Yet, it is also the output for which the subsystem decomposition yields improvement. These results demonstrate that decomposing the integrated process into physically motivated subsystems, coupled through synchronized co-training, effectively mitigates the curse of dimensionality faced by the centralized representation while preserving the interdependencies between the reactor and the downstream separation. Accordingly, the integrated surrogate model is adopted as the prediction model within the MPC framework evaluated in the following section.

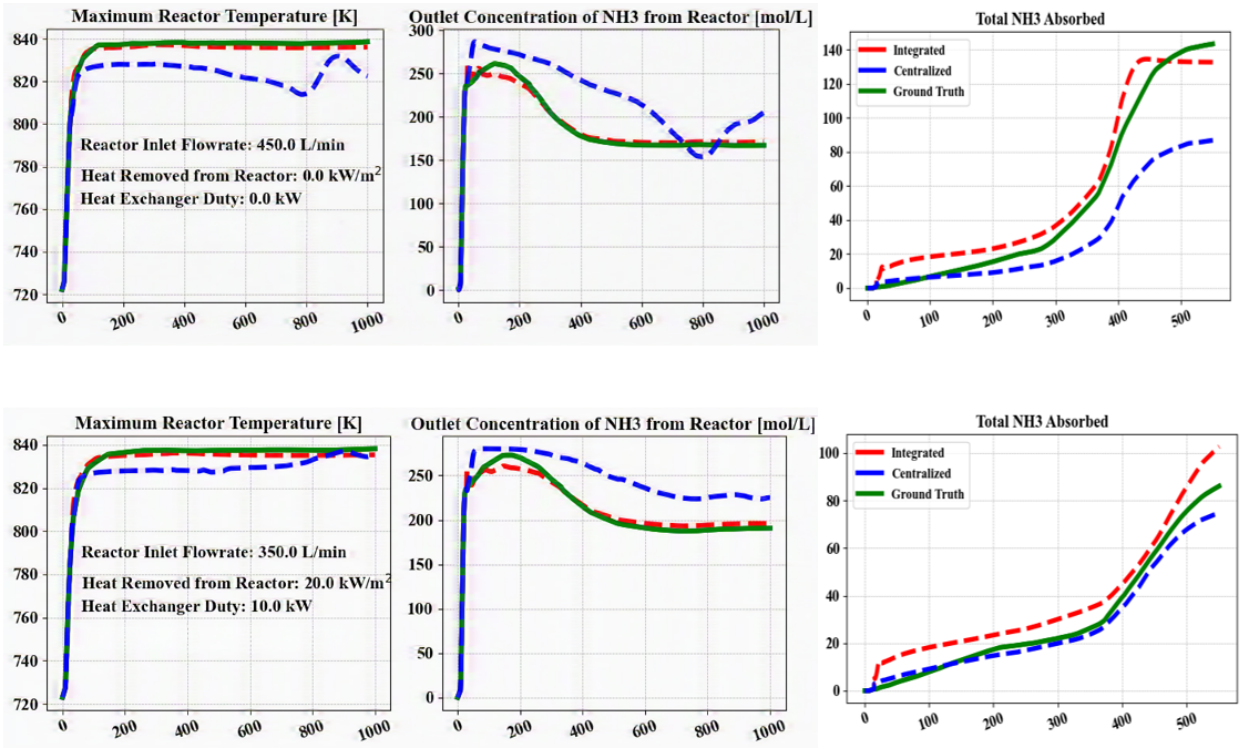

Figure 3.4: *Open-loop predictions of the centralized and integrated surrogate models against the high-fidelity ground truth for two test operating conditions.*

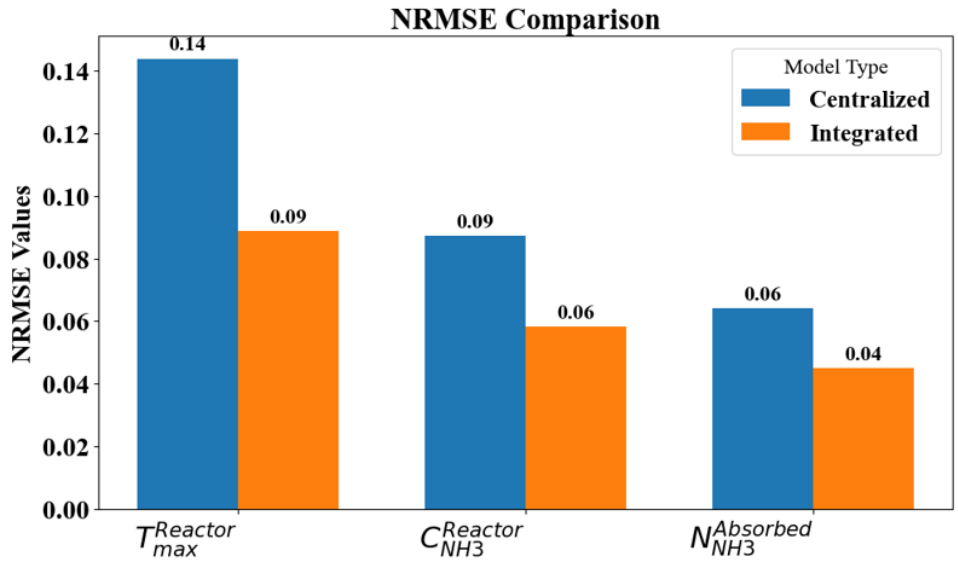

Figure 3.5: *NRMSE comparison of the centralized and integrated surrogate models for the three key process outputs across the test dataset.*

3.8 Closed-loop MPC results

The closed-loop performance of the surrogate-based MPC framework is evaluated for both the centralized and Integrated surrogate models under identical conditions: a prediction horizon of $m = 5$, a sampling interval of one minute, and 500 closed-loop iterations, corresponding to 500 minutes of operation. The targeted steady state is defined by $F_{Reactor,ss} = 400$ L/min, $q_{0,ss} = 30$ kW/m^2, and a heat exchanger duty setpoint of $Q_{HE,sp} = 20$ kW, corresponding to $T_{max,ss} = 831.56$ K and $C_{NH_3,ss} = 252.97$ mol/m^3. Both controllers start from the same initial conditions and use the same setpoints and objective weights, so the two surrogate strategies can be compared fairly. Figures 3.6 and 3.7 show the closed-loop trajectories of the manipulated inputs and process outputs under each controller, with the regulated variables plotted as normalized deviations from their steady-state values.

Both controllers steer the process to the targeted operation. The maximum reactor temperature and the outlet ammonia concentration reach narrow bands around their setpoints within roughly 100 minutes and stay there for the rest of the simulation. The heat exchanger duty is held at its 20 kW setpoint for most of the operation, which conditions the reactor effluent and sustains ammonia uptake in the adsorption column; the duty is reduced only near the end of the simulation, once the cumulative uptake levels off. As a result, the total ammonia absorbed grows steadily under both controllers, showing that the surrogate-based optimization finds input policies that regulate the reactor while serving the separation objective.

The trajectories also show persistent low-amplitude oscillations in both the inputs and the outputs. A likely contributor is the use of the static neural network replica in place of the multiscale model during the implementation phase of the closed-loop simulations. Because the replica is a static input–output mapping rather than a dynamic model, it does not filter out small mismatches between the surrogate predictions and the replica outputs; instead, these mismatches reappear at every sampling instant. Implementing the closed loop on the full multiscale model is expected to dampen these oscillations.

The two controllers are compared using the normalized ISC of the reactor inputs and the

normalized ISE of the regulated outputs, combined into the DPI, as defined in Section 2.9. The heat exchanger duty and the total ammonia absorbed are left out of these metrics: they serve the adsorption-maximization objective rather than reactor regulation, and their performance is judged directly from the duty trajectory and the cumulative uptake in Figures 3.6 and 3.7.

Table 3.1 reports the control effort of the two reactor inputs. For both controllers, most of the effort goes into the inlet flow, which is moved aggressively during the start-up transient; the Integrated-surrogate MPC uses slightly more flow effort than the centralized one. The reactor heat removal shows the largest difference between the two strategies: the Integrated-surrogate MPC keeps this input nearly constant at its nominal value (ISC of 0.010), while the centralized-surrogate MPC moves it far more (1.120), mostly during a large excursion in the initial transient.

Table 3.1: *Normalized ISC of the reactor manipulated inputs under the centralized- and Integrated-surrogate MPC.*

Manipulated input	Centralized ISC	Integrated ISC
Inlet flow, $F_{Reactor}$	25.430	28.176
Reactor heat removal, q_0	1.120	0.010

Table 3.2 reports the tracking performance of the regulated outputs. The centralized-surrogate MPC tracks both outputs slightly more closely, with lower ISE for the maximum reactor temperature and the outlet ammonia concentration. For both controllers, the concentration error is about five times larger than the temperature error, which matches the mentioned start-up transient of the outlet concentration seen in the trajectories.

Table 3.2: *Normalized ISE of the regulated outputs under the centralized- and Integrated-surrogate MPC.*

Regulated output	Centralized ISE	Integrated ISE
Maximum reactor temperature, T_{max}	0.143	0.172
Outlet NH_3 concentration, C_{NH_3}	0.781	0.890

Table 3.3 gives the aggregate comparison through the DPI. The two controllers perform very similarly overall: the centralized-surrogate MPC reaches a DPI of 27.47 against 29.25

for the Integrated-surrogate MPC, a difference of about 6% that comes almost entirely from the inlet flow effort. The breakdown across Tables 3.1–3.2 shows that neither controller is better in every respect. The centralized-surrogate MPC tracks the outputs slightly more closely and uses less inlet flow effort. In contrast, the Integrated-surrogate MPC keeps the reactor heat removal nearly constant at its nominal value and absorbs slightly more ammonia by the end of the simulation.

Table 3.3: *DPI of the centralized- and Integrated-surrogate MPC, aggregating the normalized ISC and ISE contributions of Tables 3.1–3.2.*

Controller	DPI
Centralized-surrogate MPC	27.47
Integrated-surrogate MPC	29.25

Notably, the two controllers end up so close in closed loop, even though the Integrated model is clearly more accurate in open loop. The reason is feedback, where at every sampling instant, the receding-horizon implementation corrects the prediction errors of the centralized model, which narrows the gap between the two strategies. Both surrogate strategies are therefore viable prediction models for real-time dynamic optimization of the integrated process, and both break the computational bottleneck of embedding the FEM-based multiscale model in the control loop. The Integrated strategy remains the stronger choice, offering higher model accuracy, a nearly constant reactor heat policy, and a better starting point for extending the framework to larger integrated systems.

3.9 Conclusion

This chapter presented one of the approaches of integrating physics-based domain knowledge with surrogate models, addressing the computational bottleneck of real-time predictive control in integrated process systems. In this approach, the physical knowledge enters the surrogate models through the data itself: rather than relying on operational measurements, the LSTM-based sequence-to-sequence models were trained on offline high-fidelity simulations of the multiscale model, so that the governing transport, reaction, and adsorption

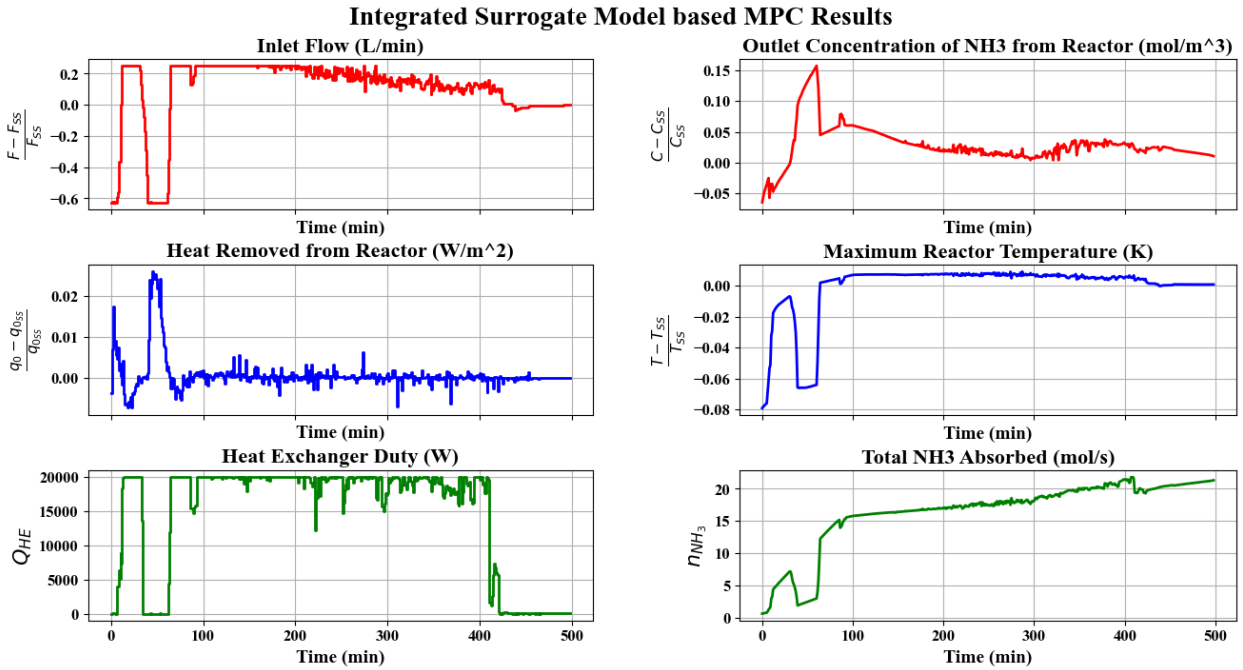

Figure 3.6: *Closed-loop trajectories of the manipulated inputs (left column) and process outputs (right column) under the Integrated-surrogate MPC.*

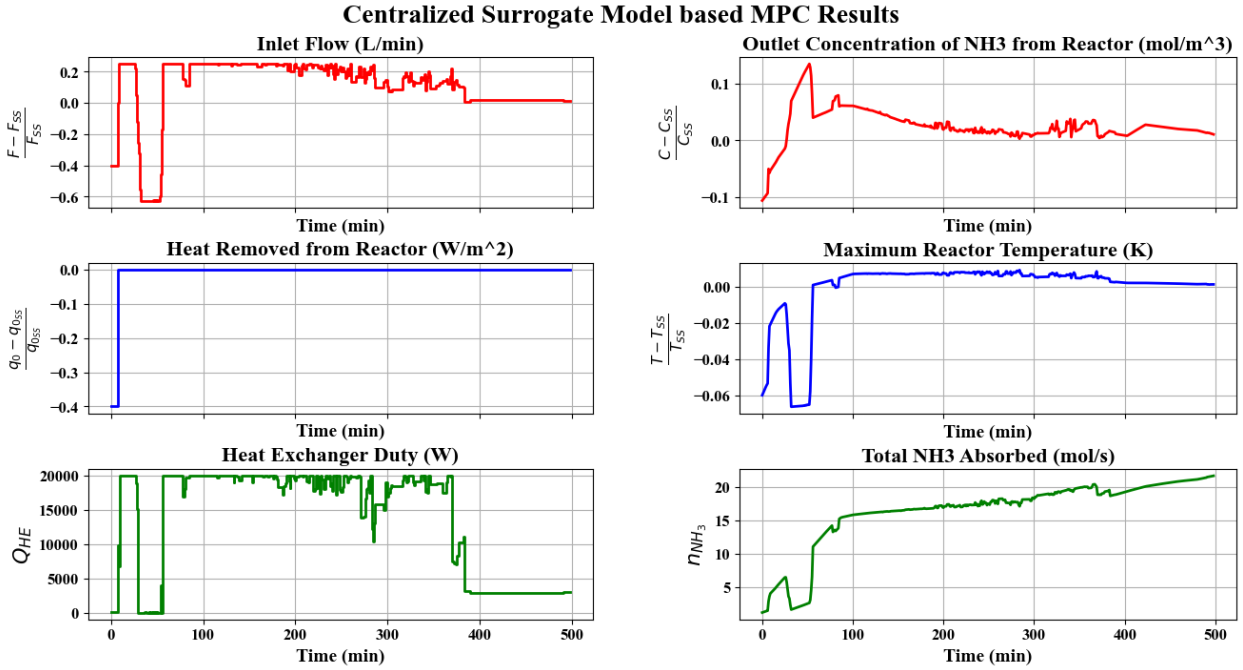

Figure 3.7: *Closed-loop trajectories of the manipulated inputs (left column) and process outputs (right column) under the centralized-surrogate MPC.*

phenomena are embedded in the learned dynamics. Two surrogate strategies were developed and compared for the integrated ammonia synthesis and adsorption process: a centralized model covering the entire process, and a distributed model that decomposes the process into reactor and heat exchanger–adsorber subsystems, co-trained with synchronized information sharing so that the coupled models respect the physical boundaries between the units. The distributed model reproduced the process dynamics more accurately in open loop, reducing the NRMSE of all three key outputs by roughly one third relative to the centralized model. Both surrogate models were then embedded in a Pyomo-based MPC framework to regulate the reactor while maximizing ammonia uptake in the adsorption column, and the two controllers performed comparably in closed loop, as feedback compensates for the prediction errors of the less accurate centralized model; the distributed-surrogate MPC nevertheless maintained a nearly constant reactor heat policy and achieved a slightly higher final ammonia uptake. These results establish the feasibility of simulation-driven surrogate modeling as a computationally tractable route to real-time dynamic optimization of coupled reaction–separation processes, while full implementation on the high-fidelity multiscale model, and ultimately on an experimental system, remains necessary to verify the framework.

Chapter 4

Hybrid and Physics-informed Time-Series Modeling for Soil Water Content (SWC)

4.1 Introduction

Ongoing changes in climatic patterns pose substantial challenges to the agricultural sector, demanding adaptive strategies for sustainable farming[87]. Intensified weather anomalies, including more frequent and severe droughts and floods, have become prevalent due to shifting climate dynamics[88;89]. These environmental disruptions significantly affect soil water content (SWC), a critical factor influencing plant growth and the uptake of water and nutrients within crop canopies[90]. SWC is an essential variable across multiple domains, including agronomy, hydrology, and engineering. In agronomy, it informs irrigation scheduling and in-season management decisions[91;92]. In hydrology, SWC is pivotal in rainfall-runoff partitioning and drought assessment[93;94]. From an engineering perspective, SWC is a crucial variable that controls soil compaction and shear resistance, is essential for constructing stable structures such as buildings and roads, and influences vertical heat transfer[95–97].

Predictive modeling of SWC relies on a thorough understanding of soil hydrology and

the physical processes that govern water movement in soils. The soil water cycle, depicted in Figure 4.1, is essential for accurately modeling SWC. This figure demonstrates water mass conservation within soil layers by accounting for inputs from precipitation and irrigation, as well as outputs through evaporation and transpiration. Additionally, water movement through soil layers is influenced by soil hydraulic properties, such as water retention and hydraulic conductivity[98;99].

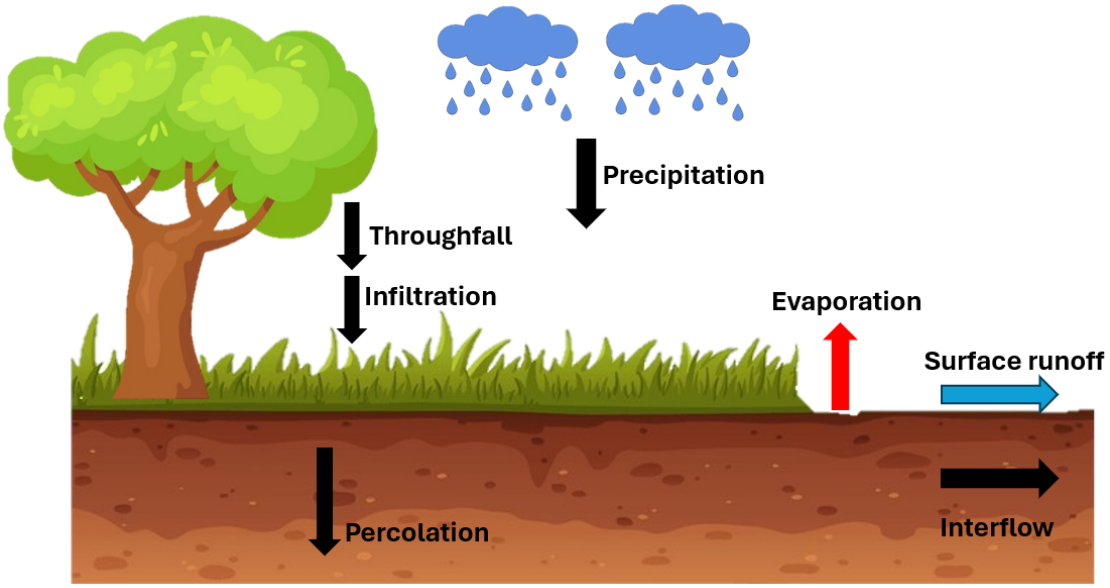

Figure 4.1: *Schematic representation of the water cycle.*

Traditional SWC predictive models based on soil physics often face challenges due to the large number of required parameters, extensive data needs on soil physical properties, and significant computational demands. For instance, models based on Richards' equation[100] employ Darcy's law and conservation of mass to describe the nonlinear relationship between soil hydraulic properties and water content[101]. Since such models are based on a nonlinear partial differential equation (PDE), solving it under various boundary and initial conditions necessitates advanced numerical techniques, complicating real-time analysis[102]. Additionally, key soil properties, such as hydraulic conductivity and water retention characteristics, exhibit considerable spatial and temporal variabilities, further limiting the applicability of Richards' equation[103;104].

In scenarios with limited information about soil physics but abundant soil moisture observations, machine learning (ML) models offer a viable alternative for predicting soil water dynamics[105]. ML algorithms can learn complex relationships among soil properties and SWC from available data. Once trained, these models enable real-time SWC prediction due to their fast inference capabilities. However, their performance depends heavily on access to extensive training datasets. Such datasets can be obtained from environmental monitoring networks[106] or remote sensing soil moisture products[107]. These datasets provide valuable inputs for developing robust data-driven ML models. Recent advancements in SWC sensors have enabled the development of modern monitoring networks known as mesoscale networks (Mesonets). These networks consist of automated weather and soil monitoring stations distributed across a region, providing continuous, high-resolution data. Mesonets collect real-time measurements of soil water content, precipitation, temperature, and other relevant variables. Examples of such networks include the Oklahoma Mesonet[108], the New York State Mesonet[109], the West Texas Mesonet[110], and the Kansas Mesonet[106].

Advances in time-series ML-based modeling approaches have substantially improved the predictive modeling of SWC[111]. However, the exclusive reliance of such methods solely on available data constrains their ability to capture underlying physical processes governing SWC dynamics. These limitations can reduce the generality of models, particularly under conditions not well-represented in the training dataset.

In this chapter, we focus on employing a simplified hydrological model, grounded in the Richards equation, as an additional level of integrating physics with time-series ML models to accurately predict SWC. Two main approaches, residual learning and physics-informed neural networks (PINNs), are investigated under centralized and distributed settings to deliver accurate multi-step SWC predictions.

4.2 Kansas Mesonet Dataset and Data Preprocessing

Proper data collection and preprocessing are essential steps in developing accurate hybrid and physics-informed SWC models. In this work, the SWC dataset used for training, validation,

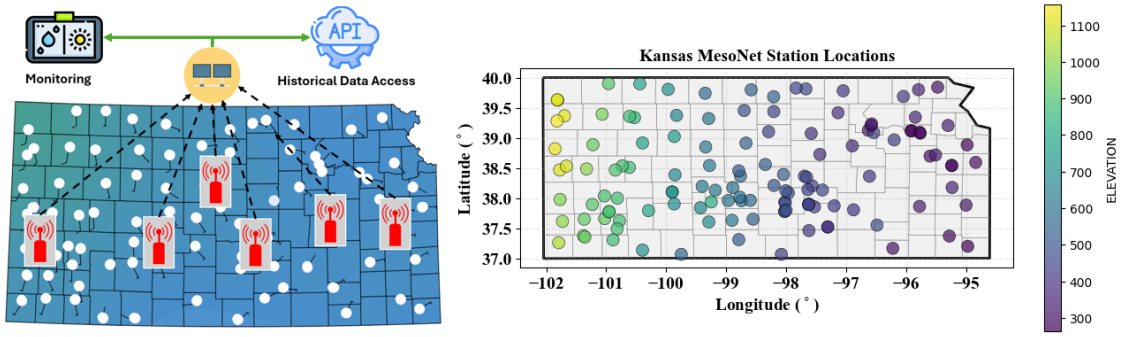

Figure 4.2: *Kansas MesoNet Distributed Stations*

and testing was obtained from the Kansas Mesonet, a multipurpose environmental monitoring network consisting of automated data collection stations distributed across Kansas[106] (Figure 4.2). Each station is equipped with research-grade instrumentation and measures precipitation, air temperature, relative humidity, barometric pressure, wind speed and direction, solar radiation, soil temperature, and volumetric soil water content at depths of 5, 10, 20, and 50 cm. Soil moisture sensors are calibrated and validated according to established protocols with a measurement accuracy of approximately 0.03 cm^3 cm^{-3}[92], and automated quality control checks flag and exclude observations from malfunctioning sensors prior to public reporting. Real-time data are accessible through the Kansas Mesonet REST API, which provides daily averaged measurements of soil water content and associated weather variables.

The selected dataset fetched for this study from Kansas MesoNet spans from January 2021 to October 2025. For each station, the extracted variables include daily precipitation, day of year (DOY), and volumetric water content at 5, 10, 20, and 50 cm. The Kansas Mesonet provides essential geographic metadata for each station, including latitude, longitude, and elevation, which collectively describe each station's position in three-dimensional space. This geographic information serves as the input feature set for K-means spatial clustering, which forms the first step of the distributed hybrid and physics-informed modeling framework developed in this study. In addition, daily precipitation records are reported alongside volumetric SWC measurements at each station. These two quantities are the core inputs to the API model, which provides the physics-based component of both the hybrid uncertainty

quantification and physics-informed neural network frameworks on a per-station basis.

Soil water storage was determined by integrating the volumetric water content at each depth across the corresponding soil layer thickness. Only stations with minimal missing data and at least two years of valid observations were retained, ensuring sufficient temporal coverage for model training and evaluation. Remaining missing values were filled using a seven-day backward moving average, where each missing entry x_t is replaced with the mean of the preceding seven available observations, preserving the local temporal structure while avoiding the introduction of future information during imputation. After removing stations with insufficient time-series data, only 62 of 126 stations in the Kansas MesoNet remain in the time-series SWC dataset, which overall contains $1.01 * 10^5$ data records.

To ensure that all features contribute equally during model training, min-max scaling was applied, transforming all feature values to the range $[0, 1]$:

$$x_{\text{scaled}} = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \tag{4.1}$$

where $x_{\min}$ and $x_{\max}$ are the minimum and maximum values of the feature across the time-series. This normalization improves the stability and convergence of the learning algorithms during training.

4.3 Antecedent Precipitation Index (API) Model

The Antecedent Precipitation Index (API) model provides a physics-based baseline estimate of SWC. Originally developed for runoff prediction[112], the model frames SWC as a first-order autoregressive Markov process in which the next state depends only on the current state and an external forcing term precipitation[113–115]. This structure reduces parameter requirements to three readily available quantities: a time-varying loss coefficient α, and the station-specific minimum and maximum SWC bounds, $S_{\min}$ and $S_{\max}$.

Following the modifications introduced in[112], two adjustments are applied to the standard API formulation. First, the loss coefficient is expressed as a sinusoidal function of the day

of year (DOY) to approximate seasonal variation in atmospheric water demand. Second, the predicted SWC is bounded between $S_{\min}$ and $S_{\max}$, representing the field capacity and wilting point of each soil profile. The model is expressed as:

$$S_t = \min\big(S_{\min} + \alpha\,(S_{t-1} - S_{\min}) + P_t,\ S_{\max}\big) \tag{4.2}$$

where S_t is the SWC at time t, P_t is daily precipitation, and $\alpha \in (0, 1)$ is a lumped loss coefficient that aggregates water losses from evapotranspiration, drainage, and runoff[?] . The seasonal dependence of α is captured through:

$$\alpha = c + (1 - c)\cos\left(2\pi\,\frac{\mathrm{DOY} - \phi}{365}\right) \tag{4.3}$$

where c is an empirical constant controlling the minimum loss rate, and ϕ is a phase offset aligning the seasonal minimum with peak evapotranspiration demand. Despite its parsimony, the API model captures the primary SWC dynamics and seasonal cycles while avoiding the computational overhead of solving Richards' equation and the extensive soil hydraulic property measurements it requires.

The large model capacity of deep neural networks, in this case LSTMs, combined with batch-level parallelism across stations through one-hot station encoding, enables centralized hybrid and physics-informed models to simultaneously learn and predict SWC dynamics across the entire station network. However, while centralized hybrid and physics-informed models consistently outperform the standalone API model and fully data-driven models, they lack any spatial awareness of the Kansas landscape. This blindness to regional variability in soil and climatic conditions limits the applicability of the centralized approach for spatially distributed predictions. In this work, we propose a framework that explicitly accounts for spatial heterogeneity by developing a distributed hybrid and physics-informed modeling strategy. First, K-means clustering is applied to group geographically proximate stations into spatially homogeneous clusters. Then, for each cluster, both hybrid uncertainty quantification and physics-informed modeling strategies are trained independently using either a Seq2Seq LSTM or a decoder-only Transformer architecture. This results in a

distributed network of cluster-level models that retain the predictive strengths of the hybrid and physics-informed frameworks while incorporating an explicit representation of spatial variability across the Kansas Mesonet. In parallel, centralized counterparts of both modeling strategies are trained across the entire station network without spatial clustering, serving as comparison baselines that isolate the effect of spatial clustering from the modeling strategy itself. The following subsections detail each step of this framework.

4.4 Spatial Clustering of Kansas Mesonet Stations

The K-means clustering algorithm partitions observations into spatially coherent groups by minimizing the within-cluster sum of squared Euclidean distances from each observation to its assigned centroid.

In this work, considering the longitude, latitude, and elevation of each station as input features, K-means is used to form geographically homogeneous clusters of Kansas Mesonet stations. These clusters serve as the spatial units over which the distributed hybrid and physics-informed models are independently trained, replacing the single centralized model with a set of cluster-level models that reflect the regional variability of the Kansas landscape.

The number of clusters K is determined using the elbow method, where the inertia $J(K)$ is computed for a range of cluster numbers from $K = 2$ to $K = 15$. Figure 4.3 shows the inertia as a function of the number of clusters, where the optimal K^* corresponds to the point of diminishing marginal reduction in $J(K)$, beyond which additional clusters yield limited improvement in within-cluster compactness.

4.5 Centralized and Distributed Hybrid Uncertainty Quantification

The API model provides a physics-based baseline SWC prediction for each station through Eq. (4.2). However, as reported in our previous work[?] , the API model exhibits systematic

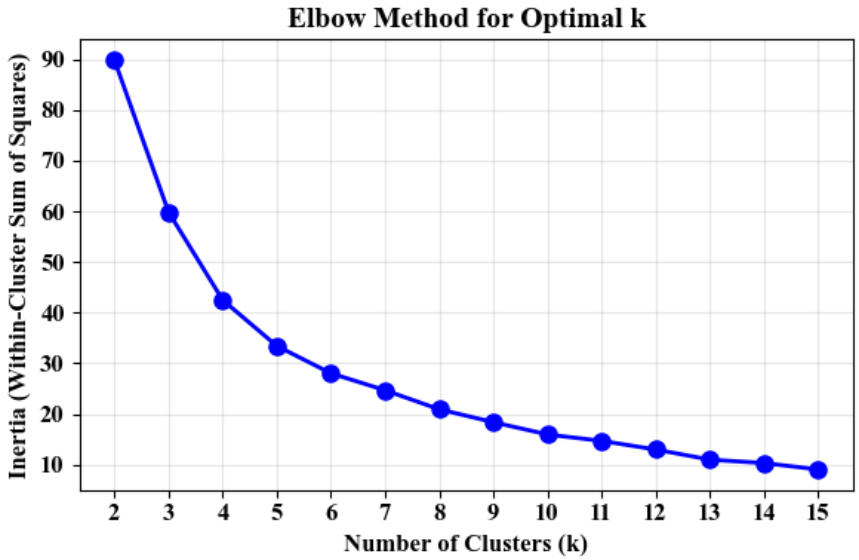

Figure 4.3: *Inertia as a function of the number of clusters K for the Kansas Mesonet stations.*

overestimation and underestimation of SWC under certain seasonal conditions, arising from its simplified representation of evapotranspiration and drainage dynamics. To correct for this, a hybrid uncertainty quantification (UQ) approach is adopted, where a data-driven model is trained to learn the residual error between the API model predictions and the observed SWC values at each station:

$$e_t = S_t^{\text{Obs}} - S_t^{\text{API}} \tag{4.4}$$

where S_t^{Obs} is the measured SWC from the Kansas Mesonet and S_t^{API} is the API model prediction at time t. The UQ model, either the Seq2Seq LSTM or the decoder-only Transformer, is trained to predict the next-step residual e_{t+1} from a set of dynamic and static covariates:

$$\hat{e}_{t+1} = \mathcal{F}_{\text{UQ}}(e_{t-k}, \ldots, e_t,\ P_t,\ \sin(2\pi\,\text{DOY}/365),\ \cos(2\pi\,\text{DOY}/365),\ x,\ y,\ z) \tag{4.5}$$

where $e_{t-k}, \ldots, e_t$ is the sequence of residual errors over the preceding $k = 7$ days, P_t is daily

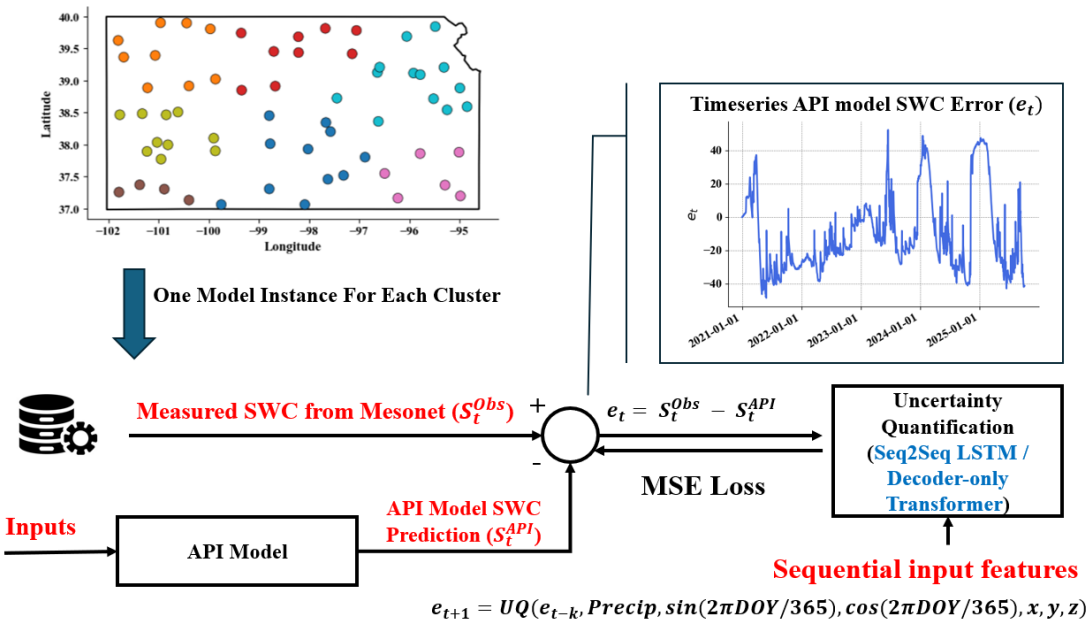

Figure 4.4: *Hybrid UQ Model Training scheme*

precipitation, the seasonal encoding DOY ($\sin(2\pi\,\mathrm{DOY}/365)$, $\cos(2\pi\,\mathrm{DOY}/365)$) serving as a temporal feature, and x, y, z are the longitude, latitude, and elevation of the station respectively. The error sequence and precipitation constitute the dynamic covariates, capturing the temporal evolution of the API model's uncertainty, while the geographic coordinates serve as static covariates that distinguish between stations within the same cluster. The UQ model is trained by minimizing the mean squared error between the predicted and actual residuals:

$$\mathcal{L}_{\mathrm{UQ}} = \frac{1}{N}\sum_{t=1}^{N}(e_t - \hat{e}_t)^2 \tag{4.6}$$

The training mechanism is illustrated in Figure 4.4, where the measured SWC and API model predictions are used to compute the residual time-series, which is then passed, along with the dynamic and static covariates, to the UQ model. At inference, shown in Figure 4.5, the trained UQ model produces the corrected residual estimate $\hat{e}_{t+1}$, which is added to the next-step API model prediction to form the final hybrid SWC forecast:

$$S_{t+1}^{\mathrm{Hybrid}} = S_{t+1}^{\mathrm{API}} + \hat{e}_{t+1} \tag{4.7}$$

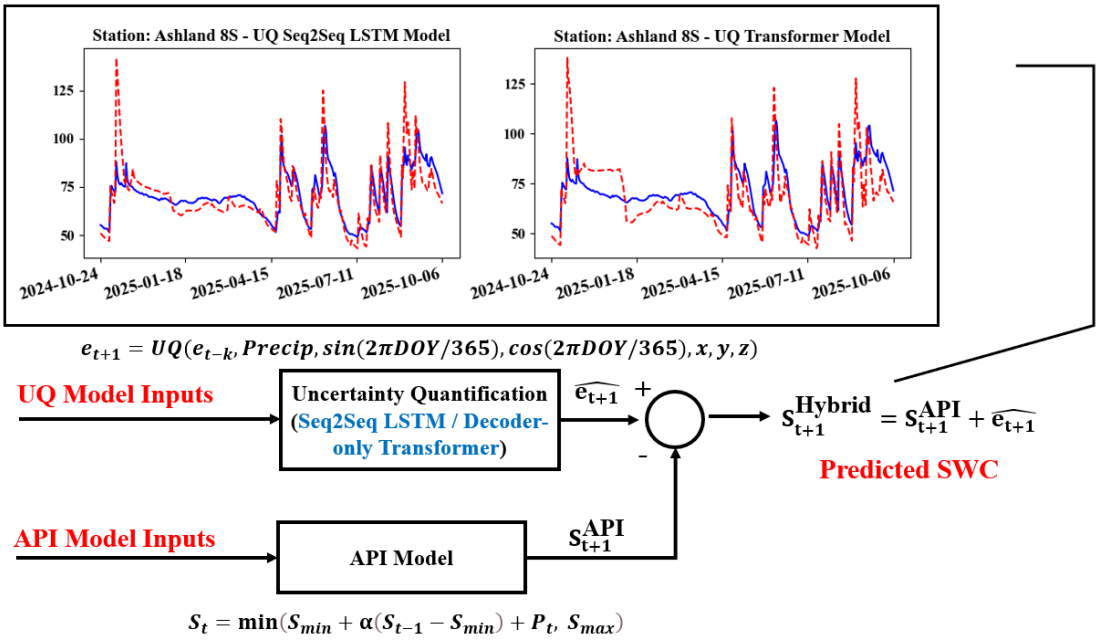

Figure 4.5: *Hybrid UQ Model Inference scheme*

This formulation preserves the physics-based structure of the API model while allowing the UQ model to correct for its systematic deviations. The UQ framework is implemented under two settings. In the centralized setting, a single UQ model instance is trained across the entire station network. In the distributed setting, following the K-means clustering step, one UQ model instance is trained per cluster rather than across the entire station network, introducing spatial awareness into the hybrid framework, with one independent model instance per cluster capturing the regional characteristics of each geographic group.

For the UQ-based Seq2Seq LSTMs, the centralized model uses an encoder containing 3 stacked LSTM layers with 64, 64, and 32 cells, with dropout layers in between, connected to a fully connected decoder. Each of the distributed models has a similar architecture, but with 64, 32, and 16 cells per layer.

For the UQ-based decoder-only Transformers, the centralized model has 2 Transformer blocks with an attention dimension of 32, 4 attention heads, and a feed-forward dimension of 128, while its distributed counterparts have a similar architecture with a feed-forward dimension of 64.

For training the models, the Adam optimizer with adaptive learning rates was used, and

early stopping was employed to prevent overfitting. Each model was initially configured for 25 epochs; however, proper model checkpointing and early stopping halted training at an earlier epoch to ensure adequate model training without overfitting.

4.6 Centralized and Distributed Physics-Informed SWC Models

While the hybrid UQ framework treats the API model as the primary predictor and uses the ML models to correct its residual error, in the physics-informed (PINN) perspective the neural network becomes the primary predictor, and the API model is embedded directly into the training objective to constrain the learned solution toward physically consistent behavior[24? ;25]. Rather than learning an explicit correction term, the PINN model is trained to predict the SWC value $\hat{S}_t$ at each time step such that it simultaneously agrees with the observed Mesonet data and remains consistent with the API model's representation of soil-water dynamics governed by Eq. (4.2).

This dual objective is enforced through a composite loss function that combines a data-driven term and a physics-based term. Unlike a fixed weighting scheme, the relative contribution of each term is adapted per station based on the local fidelity of the API model, as measured on the training data:

$$\mathcal{L}_{\text{PINN}} = \frac{1}{N}\sum_{t=1}^{N}\left[(1-\beta_i)\left(S_t^{\text{Obs}} - \hat{S}_t\right)^2 + \beta_i\left(S_t^{\text{API}} - \hat{S}_t\right)^2\right] \tag{4.8}$$

where S_t^{Obs} is the measured SWC from the Kansas Mesonet, S_t^{API} is the corresponding API model prediction, N is the number of training samples, and $\beta_i \in [0,1]$ is a station-specific physics weighting coefficient. By construction, $(1-\beta_i)$ and β_i are complementary and sum to unity at every sample, ensuring the total loss is always a convex combination of the data and physics terms.

The station-specific weight β_i is defined as:

$$\beta_i = 1 - \mathrm{NRMSE}_i^{\mathrm{API}} \tag{4.9}$$

where $\mathrm{NRMSE}_i^{\mathrm{API}}$ is the normalized root mean squared error of the API model at station i, computed on the training split only to prevent data leakage. This formulation has a direct physical interpretation. At stations where the API model performs well ($\mathrm{NRMSE}_i^{\mathrm{API}} \to 0$), $\beta_i \to 1$ and the physics loss dominates, anchoring the network to the API model's physically consistent seasonal behavior. Conversely, at stations where the API model is a poor local approximation ($\mathrm{NRMSE}_i^{\mathrm{API}} \to 1$), $\beta_i \to 0$ and the loss reduces to a purely data-driven objective, allowing the network to learn directly from the Mesonet observations without being constrained by an unreliable physics reference. This adaptive weighting directly addresses a PINN limitation where applying a uniform physics penalty across spatially heterogeneous stations amplifies the API model's spatially structured biases.

After substituting the API model from Eq. (4.2) into the physics term, the loss function becomes:

$$\mathcal{L}_{\mathrm{PINN}} = \frac{1}{N}\sum_{t=1}^{N}\left[(1-\beta_i)\left(S_t^{\mathrm{Obs}} - \hat{S}_t\right)^2 + \beta_i\left(\min\left(S_{\min} + \alpha\left(S_{t-1} - S_{\min}\right) + P_t,\ S_{\max}\right) - \hat{S}_t\right)^2\right] \tag{4.10}$$

The PINN model takes both dynamic and static covariates as input. The dynamic covariates consist of the historical sequence of observed SWC values over the preceding 7 days, together with daily precipitation P_t and the seasonal encoding $(\sin(2\pi\,\mathrm{DOY}/365),\ \cos(2\pi\,\mathrm{DOY}/365))$. The static covariates are the station's geographic coordinates x, y, and z, which allow the model to distinguish stations within the same cluster. The PINN model is thus formulated as:

$$\hat{S}_t = \mathcal{F}_{\mathrm{PINN}}\left(S_{t-k}^{\mathrm{Obs}}, \ldots, S_{t-1}^{\mathrm{Obs}},\ P_t,\ \sin(2\pi\,\mathrm{DOY}/365)),\ \cos(2\pi\,\mathrm{DOY}/365)),\ x,\ y,\ z\right) \tag{4.11}$$

The principal difference from the UQ formulation in Eq. (4.5) is that the dynamic covariate is the historical SWC sequence itself rather than the API model's residual error sequence, since the PINN model is tasked with predicting SWC directly rather than correcting an existing prediction.

As with the UQ framework, the PINN models are developed under both centralized and distributed settings. Centralized PINN models are trained across the entire station network without spatial clustering, following the adaptive loss formulation in Eq. (4.8). In the distributed setting, the same Seq2Seq LSTM and decoder-only Transformer architectures described for the distributed UQ framework are adopted, following the per-cluster structure established by the K-means spatial grouping. Each model is trained independently per cluster using its own set of station-specific β_i weights computed from that cluster's training data.

4.7 Results and Discussions

In this section, the performance of each proposed framework is evaluated on the held-out test set across all Kansas Mesonet stations. The analysis proceeds in two stages. First, the results of each framework are examined through visual diagnostics, including time-series plots at geographically representative stations, parity plots aggregated across all stations, and error distribution plots, to characterize the qualitative behavior of each model. Second, a quantitative comparison is conducted using NRMSE and mean bias computed per station and aggregated across the full network, enabling a systematic evaluation of the distributed clustered models against the API baseline and the centralized models. The UQ and PINN frameworks are each evaluated for both the Seq2Seq LSTM and decoder-only Transformer architectures, with clustered and centralized training variants compared throughout to isolate the effect of spatial partitioning from the modeling strategy itself.

4.7.1 Spatial Clustering Results

The elbow method (Figure 4.3) identifies $K = 5$, 6, and 7 as candidate values, corresponding to the region of sharpest inertia reduction where additional clusters yield diminishing returns in within-cluster compactness. To determine the optimal K^* among these candidates, the silhouette score is computed for each station under each configuration.

The silhouette score[116] quantifies how well each observation fits within its assigned cluster relative to neighboring clusters. For a station i, it is defined as:

$$s_i = \frac{b_i - a_i}{\max(a_i,\ b_i)} \tag{4.12}$$

where a_i is the mean distance from station i to all other stations within the same cluster, and b_i is the mean distance from station i to all stations in the nearest neighboring cluster. The score ranges from -1 to 1, where values approaching 1 indicate that the station is well matched to its own cluster and clearly separated from others, values near 0 suggest the station sits close to a cluster boundary, and negative values indicate the station may have been assigned to the wrong cluster.

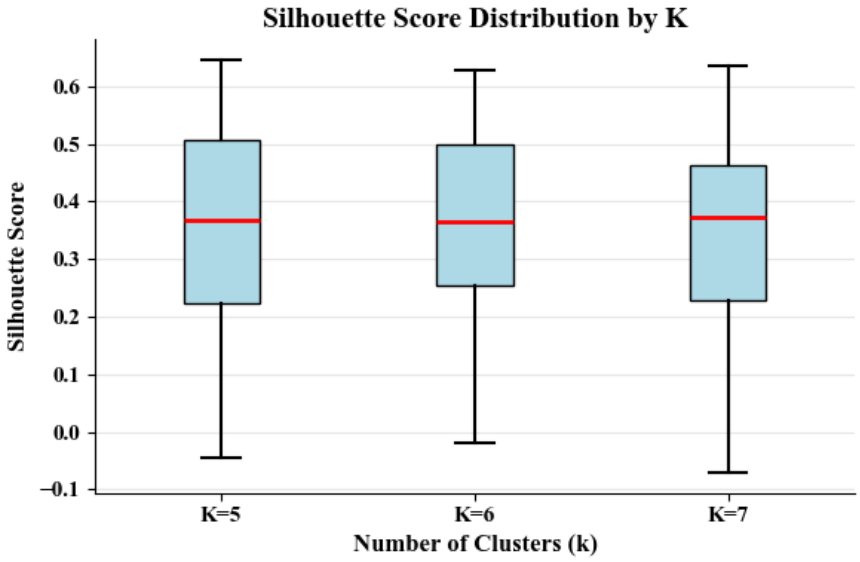

Figure 4.6: *Silhouette score distributions for the candidate cluster numbers.*

Figure 4.6 presents the corresponding silhouette score distributions for all three configu-

rations, computed in the standardized feature space of latitude, longitude, and elevation. All three candidates yield comparable median silhouette scores of approximately 0.37, confirming that none produces geometrically poorly defined clusters. The spatial maps in Figure 4.7 provide the distinguishing basis for the final selection. $K = 5$ merges climatically distinct regions in central and eastern Kansas into overly broad partitions, while $K = 7$ begins to fragment the relatively homogeneous western high plains into geographically arbitrary subgroups. The $K = 6$ configuration produces the most interpretable spatial partition, separating the semi-arid high plains in the west, the central transition zone, the Flint Hills, and the more humid southeastern and northeastern regions into coherent, recognizable agro-climatic divisions. Accordingly, $K^* = 6$ is selected as the optimal number of clusters.

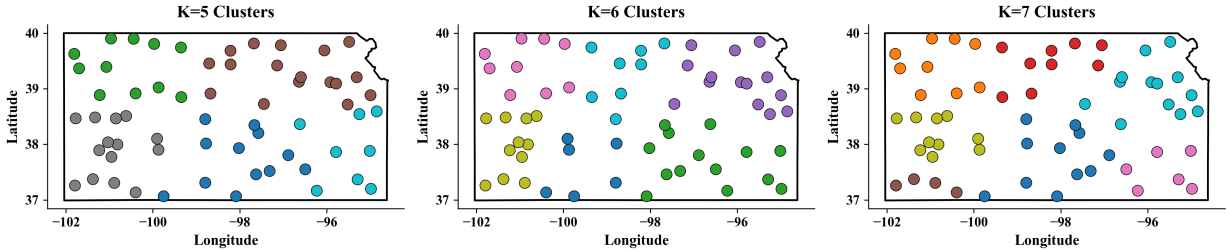

Figure 4.7: *Clustering results under different candidate values of K.*

Figure 4.8 presents the final $K^* = 6$ partition in both 2D and 3D projections, confirming that the clusters align with the west-to-east climatic gradient across Kansas. The 3D visualization further reveals that elevation drives meaningful vertical stratification, with western clusters exceeding 900 m and eastern clusters below 500 m; these gradients correspond to systematic differences in temperature, atmospheric demand, and precipitation patterns influencing SWC dynamics. Together, the 2D and 3D projections validate that the K-means partitioning recovers agro-climatically coherent station groupings rather than arbitrary geographic divisions.

4.7.2 Hybrid Uncertainty Quantification Results

Figures 4.9 and 4.10 present the time-series predictions of the UQ-based Seq2Seq LSTM and Transformer models at six representative stations, one per cluster, selected by maximizing ge-

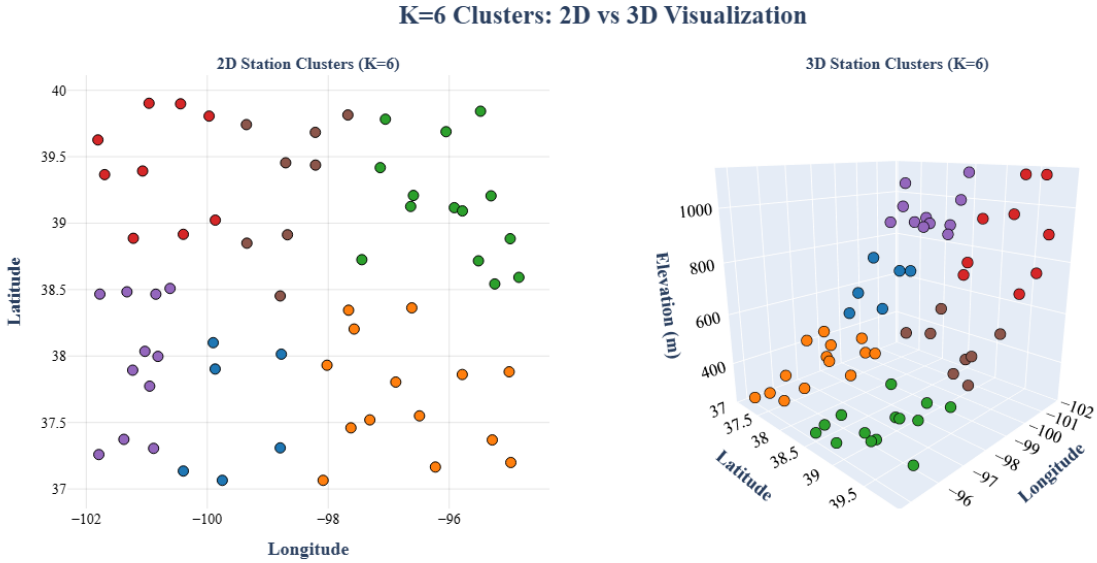

Figure 4.8: *2D and 3D projections of the final clustering with* $K^* = 6$.

ographic separation using a greedy max-min Haversine distance criterion[117]. These stations span the full spatial extent of Kansas, from the semi-arid high plains (Tribune, Cluster 5; Meade, Cluster 1) to the humid eastern regions (Uniontown 3NW, Cluster 2; Washington, Cluster 3) and the central transition zone (Ludell 3N, Cluster 4; Hoisington 4S, Cluster 6). Both the clustered and centralized UQ models track the observed SWC dynamics closely across all six stations, capturing seasonal drydown trends and rapid precipitation-driven responses. The API model, by contrast, shows large systematic deviations across all stations, particularly at Meade and Tribune, where it consistently overpredicts or loses track of the observed dynamics entirely, confirming the limitations of the purely physics-based approach under spatially heterogeneous soil and climate conditions.

Figure 4.11 presents the parity plots aggregated across all stations for the LSTM and Transformer architectures. The API model produces a wide, diffuse scatter, with predictions compressed within a limited dynamic range and failing to follow the 1:1 line, particularly at low observed SWC values. Both the clustered and centralized UQ models produce substantially tighter scatter along the 1:1 diagonal across the full SWC range, representing a significant improvement over the API baseline. The centralized models show marginally

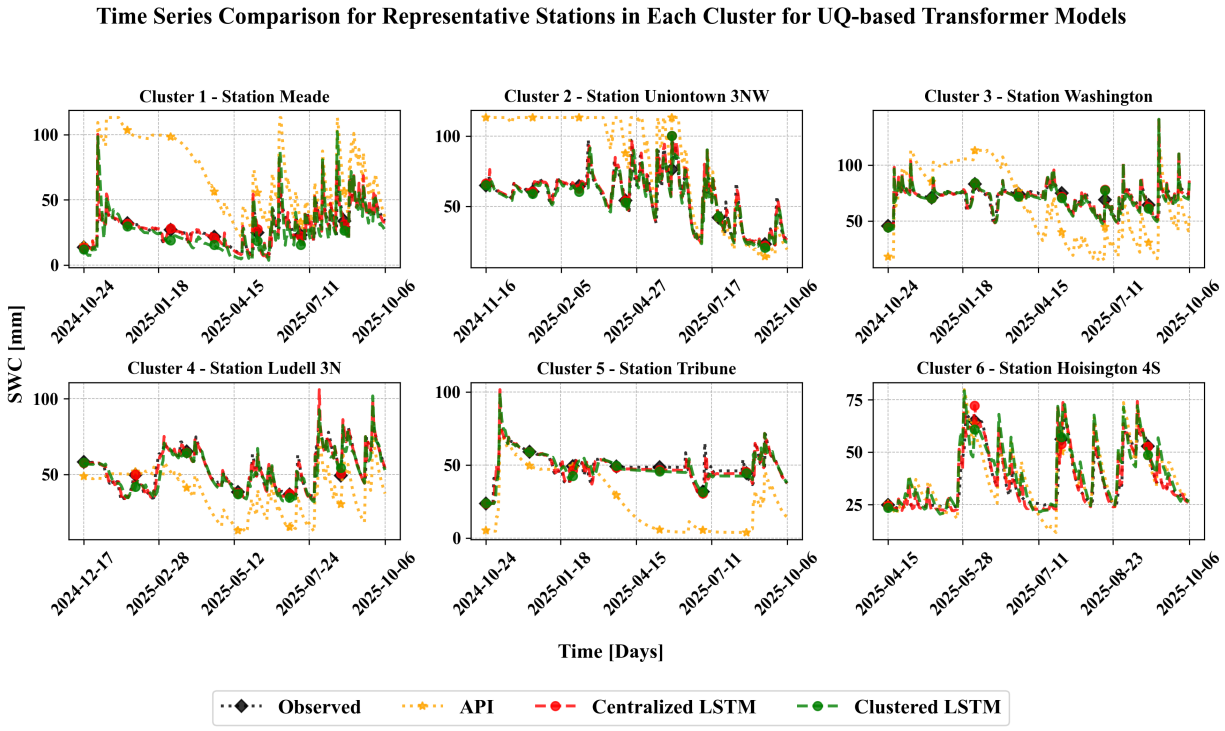

Figure 4.9: *Time-series plots for the UQ-based LSTM models.*

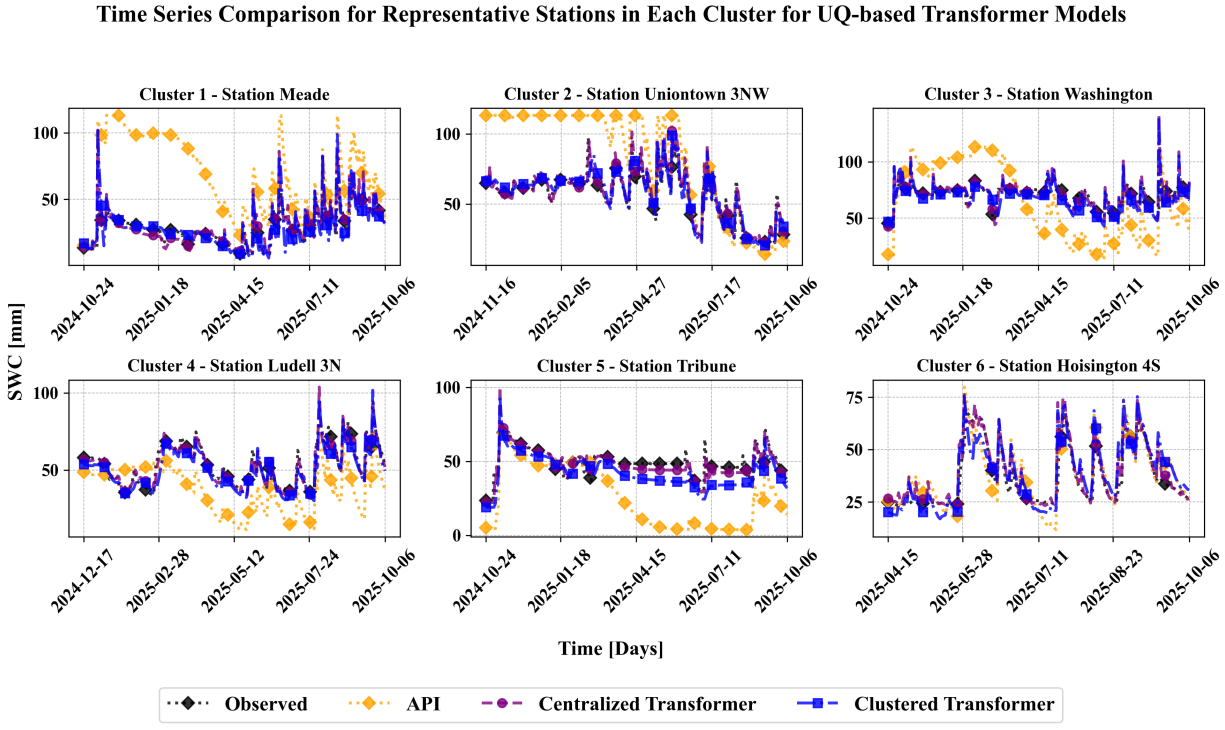

Figure 4.10: *Time-series plots for the UQ-based Transformer models.*

denser point concentration along the diagonal at mid-range SWC values, while the clustered models exhibit slightly wider scatter at the tails. This suggests that the centralized models slightly outperform the clustered models in terms of prediction accuracy; a detailed comparative analysis of which configuration performs better overall is presented later in this section.

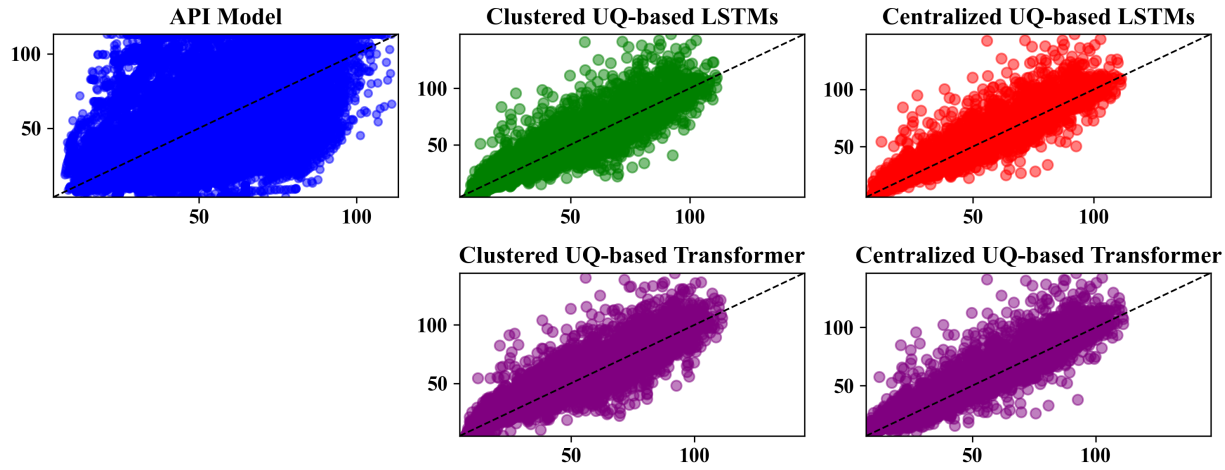

Figure 4.11: *Parity plots for the UQ-based models.*

Furthermore, Figure 4.12 shows the error distributions for all UQ configurations. The API model produces a broad, left-skewed distribution with substantial density extending beyond −50 mm, reflecting systematic underprediction at many stations. Both UQ frameworks, clustered and centralized, produce near-symmetric distributions tightly centered at zero with negligible tails, confirming that the residual correction mechanism effectively eliminates the API model's systematic bias regardless of training strategy.

4.7.3 Physics-Informed Neural Network Results

Figures 4.13 and 4.14 present the time-series predictions of the physics-informed Seq2Seq LSTM and Transformer models at the same six representative stations used in the UQ analysis. Both the clustered and centralized PINN models track the observed SWC dynamics considerably better than the API model across all stations, capturing the seasonal drydown and precipitation responses. However, unlike the UQ framework, where the clustered and

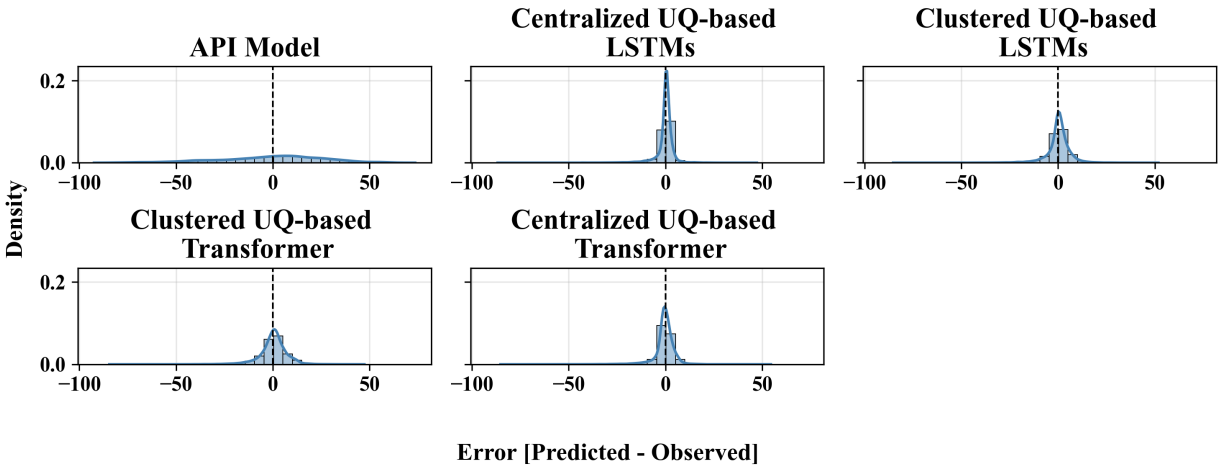

Figure 4.12: *Error distribution plots for the UQ-based models.*

centralized models performed comparably in temporal tracking, the PINN models show more visible station-to-station variability, particularly at Meade (Cluster 1) and Tribune (Cluster 5), where both PINN configurations occasionally diverge from the observed trajectory during prolonged dry periods. This divergence reflects the influence of the adaptive physics constraint: at stations where the API model is a poor local approximation, the β_i weighting reduces the physics penalty. Still, the network must then rely more heavily on limited local training data to learn the correct dynamic behavior of SWC.

Figure 4.15 presents the parity plots for the PINN configurations. Both the clustered and centralized PINN models show clear improvement over the API baseline, with scatter clouds substantially better aligned along the 1:1 diagonal across the full observed SWC range. The clustered and centralized configurations produce visually comparable alignment for both the LSTM and Transformer architectures, with no configuration showing a systematic directional bias relative to the 1:1 line.

Figure 4.16 shows the error distributions for all PINN configurations. All PINN models produce broader, flatter distributions compared to the UQ framework, with heavier tails extending to ± 50 mm and a residual positive skew inherited from the API model's systematic tendencies. While both the clustered and centralized PINN configurations substantially reduce the API model's large negative bias, the UQ residual correction approach remains more effective at producing tight, unbiased error distributions across the station network.

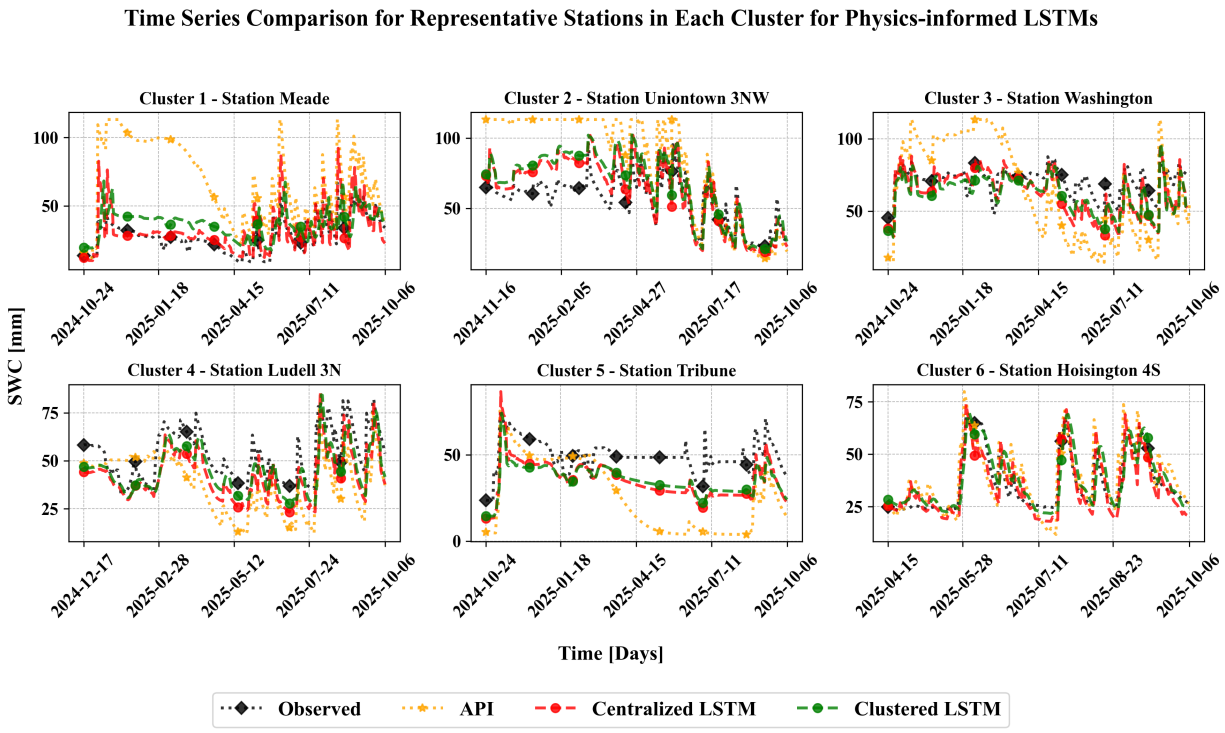

Figure 4.13: *Time-series plots for the physics-informed LSTM models.*

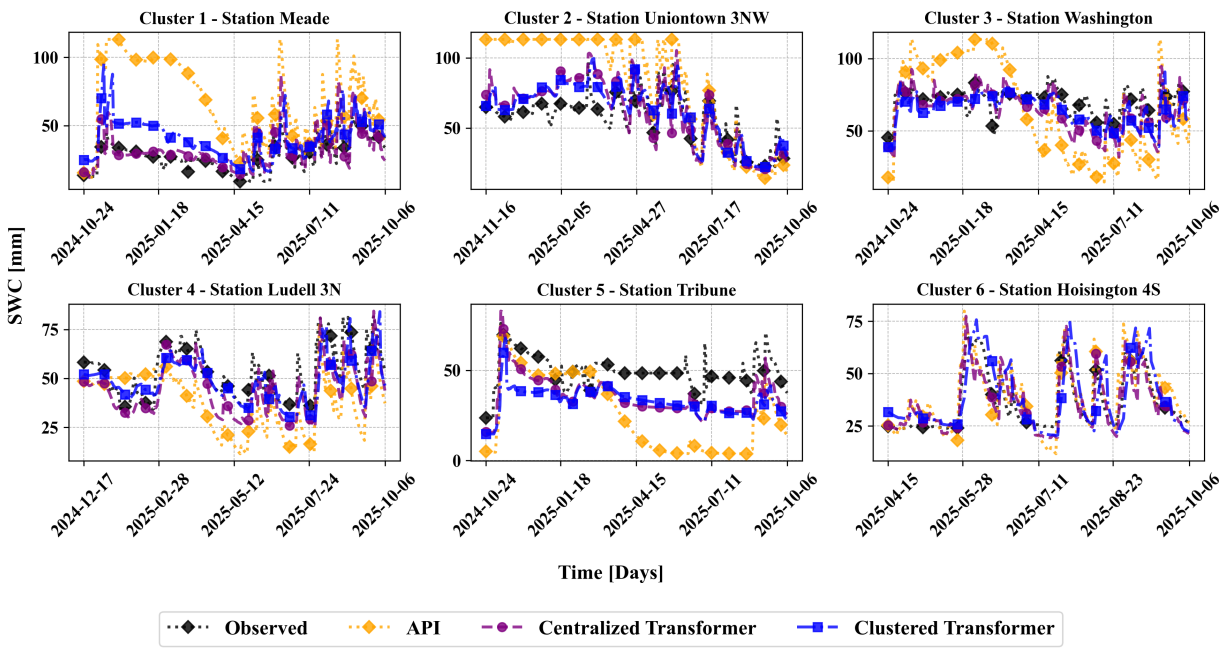

Figure 4.14: *Time-series plots for the physics-informed Transformer models.*

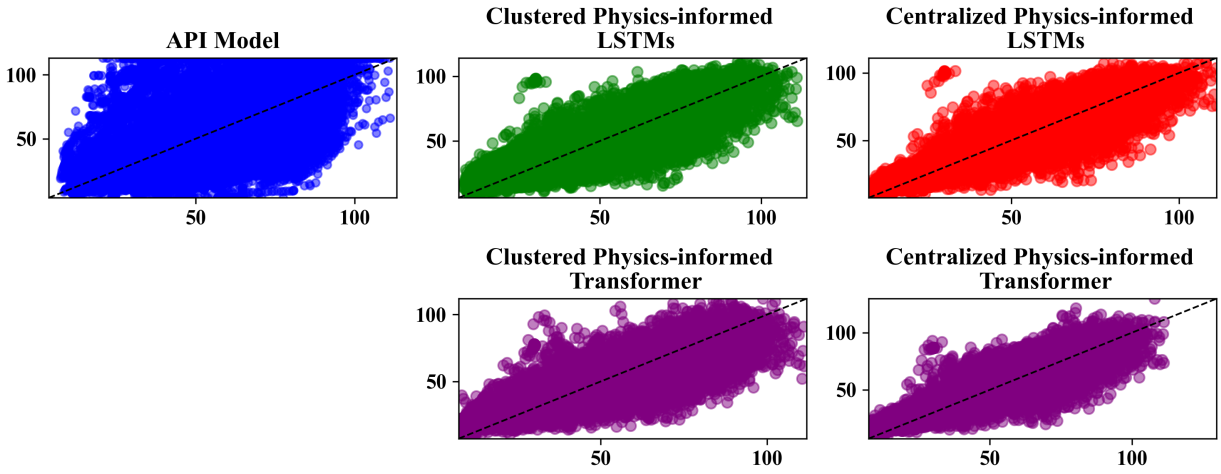

Figure 4.15: *Parity plots for the physics-informed models.*

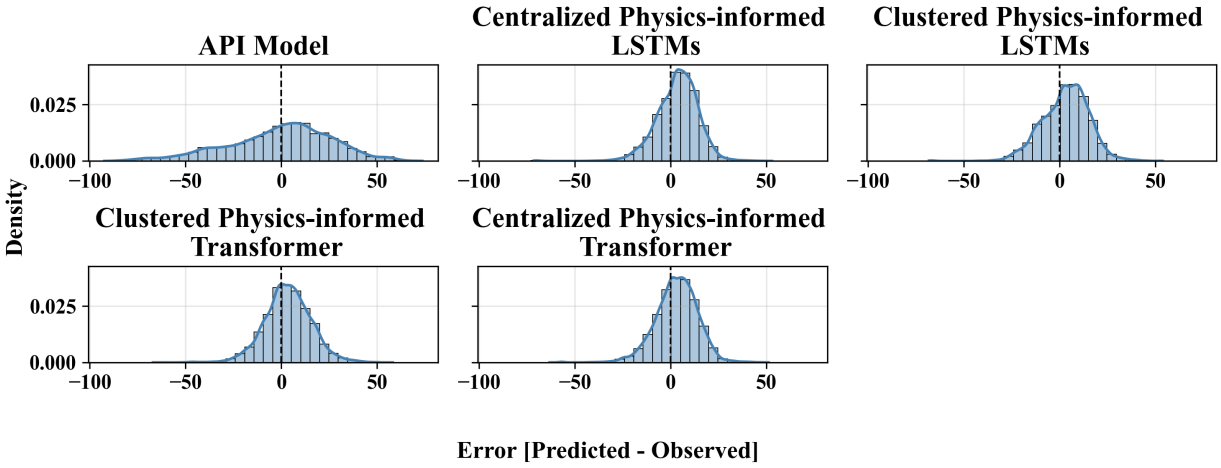

Figure 4.16: *Error distribution plots for the physics-informed models.*

4.7.4 Overall Comparison

While the visual time-series, parity, and error distribution analyses provide insight into each model's prediction behavior, a quantitative evaluation is necessary to systematically compare performance across all 62 stations. Two complementary objectives are considered in this work: (1) temporal prediction accuracy at the station level, and (2) spatial consistency of errors across the network. To address the first objective, NRMSE and mean bias (MB) are computed per station for each model, quantifying the regression accuracy and the systematic over- or under-prediction tendency of each framework. To address the second

objective, Moran's I is computed on station-level mean residuals to assess whether prediction errors exhibit statistically significant spatial autocorrelation across the Kansas Mesonet network. Together, these three metrics provide a comprehensive evaluation of each model's performance from both temporal and spatial perspectives.

Figure 4.17 presents the NRMSE distribution across all 62 stations for each model configuration. The API model shows the highest NRMSE, with a median of approximately 0.45 and extreme outliers exceeding 1.0 at individual stations, confirming its inability to capture the spatially heterogeneous SWC dynamics across Kansas. Both the UQ and PINN frameworks substantially outperform the API baseline across all stations and both architectures, with UQ achieving the lowest median NRMSE (≈ 0.10–0.13) and PINN at an intermediate level (≈ 0.19–0.23). Figure 4.18 shows the mean bias distributions, where the API model exhibits the widest spread, with extreme station-level biases reaching ± 40 mm, while both the UQ and PINN frameworks reduce the bias distribution to a narrow band centered near zero, indicating that the physics-informed and residual correction strategies effectively eliminate the API model's systematic over- and under-prediction tendencies across the network.

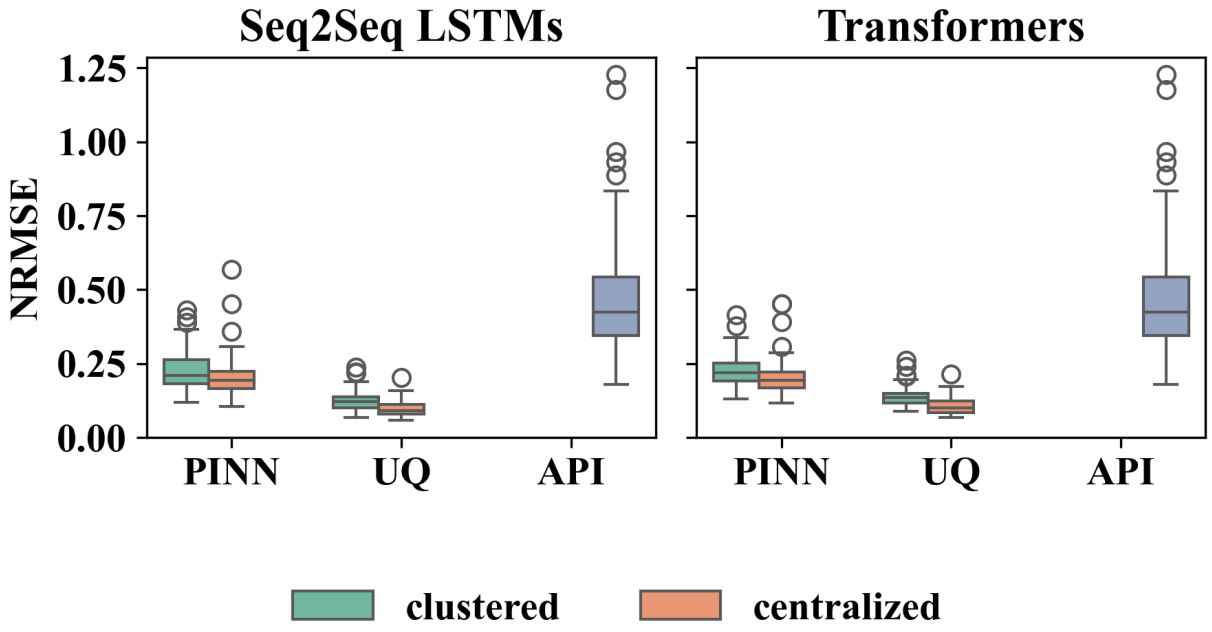

Figure 4.17: *NRMSE box plots of the SWC models across stations.*

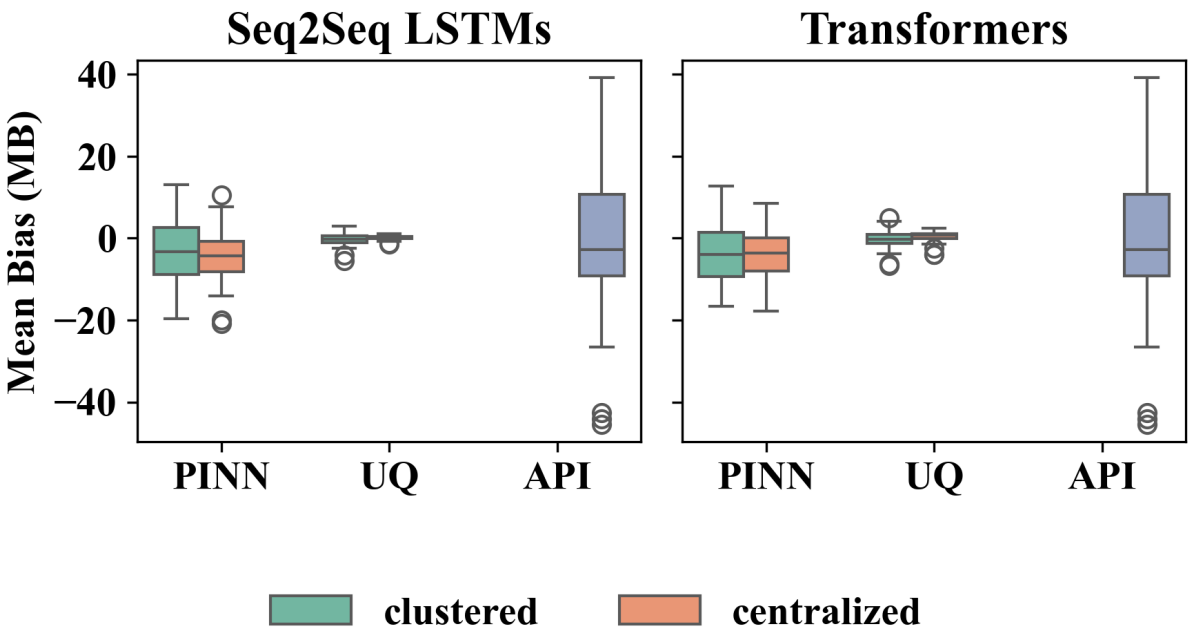

Figure 4.18: *Mean bias box plots of the SWC models across stations.*

Within each framework, the centralized models consistently achieve slightly lower median NRMSE than their clustered counterparts. However, as shown in Table 4.1, the centralized configurations consistently yield higher Moran's I values than their clustered counterparts in both the UQ and PINN frameworks, indicating that centralized training introduces geographically structured biases that the clustering approach mitigates. The mean bias distributions in Figure 4.18 show no meaningful difference between the clustered and centralized configurations, confirming that the systematic bias reduction is attributable to the framework type (UQ or PINN) rather than the spatial training strategy.

Figure 4.19 presents the joint NRMSE–Moran's I trade-off across all configurations. The API model exhibits a high NRMSE despite a low Moran's I, indicating that its spatial uniformity reflects consistently inaccurate performance rather than acceptable spatial generalization. Among the ML models, the UQ-based clustered LSTM achieves the best joint performance, combining a competitive NRMSE (≈ 0.12) with the lowest Moran's I of all configurations (≈ 0.02), placing it closest to the ideal region.

Despite comparable NRMSE to the UQ framework, the PINN configurations exhibit

higher Moran's I values, particularly the centralized physics-informed LSTM ($I \approx 0.33$, $p < 0.05$), confirming that integrating a globally structured physics constraint can make the ML model more prone to regional biases. Finally, it can be inferred that the proposed clustering approach consistently reduces Moran's I compared with centralized training in both frameworks, demonstrating that spatial partitioning is necessary to achieve accurate and geographically unbiased SWC predictions across the Kansas Mesonet network.

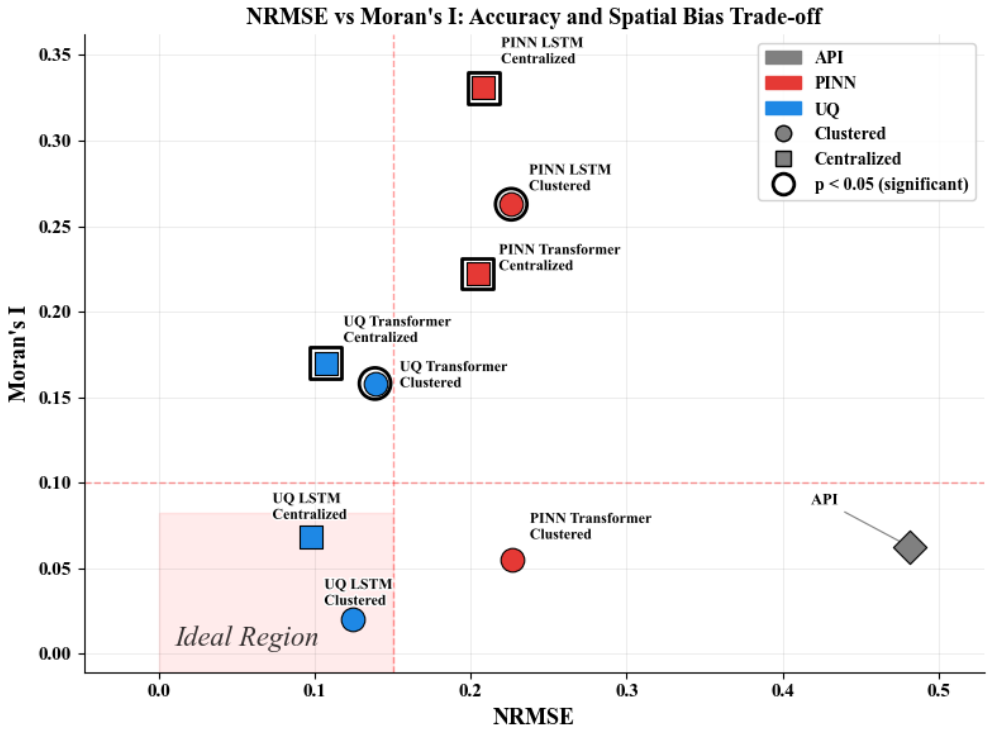

Figure 4.19: *Joint NRMSE–Moran's I trade-off across all model configurations.*

4.8 Conclusion

In this chapter, the inclusion of a simplified mechanistic model as a core physics-based component of time-series ML models was analyzed through an SWC prediction case study. The key findings indicate that the time-series learning component substantially enhances the dynamic model's accuracy, while decomposing the original problem into a set of distributed sub-problems carries certain trade-offs relative to its centralized counterpart. Specifically,

Table 4.1: *Overall quantitative metrics for each model configuration averaged across all 62 Kansas Mesonet stations on the test set.*

Framework	Architecture	Training	NRMSE	Mean Bias	Moran's I
API	–	–	0.481	−0.144	0.063
PINN	LSTM	Centralized	0.208	−3.898	0.331
PINN	LSTM	Clustered	0.226	−3.279	0.263
PINN	Transformer	Centralized	0.204	−3.391	0.222
PINN	Transformer	Clustered	0.227	−3.587	0.055
UQ	LSTM	Centralized	0.097	+0.202	0.069
UQ	LSTM	Clustered	0.124	−0.195	0.020
UQ	Transformer	Centralized	0.107	+0.458	0.170
UQ	Transformer	Clustered	0.139	−0.259	0.158

both the hybrid UQ and physics-informed strategies markedly outperformed the standalone API model, confirming that neither first-principles knowledge nor data-driven learning alone is sufficient for systems with poorly characterized, spatially heterogeneous dynamics. Although centralized training attained slightly lower prediction errors, it inherited geographically structured biases from the globally imposed physics reference; the distributed clustered formulation mitigated this spatial autocorrelation, yielding accurate and spatially unbiased SWC predictions across the Kansas Mesonet network. Within the broader scope of this dissertation, this chapter explores another avenue for integrating physics with time-series ML models by systematically fusing imperfect physical models with data, providing a reliable dynamic modeling foundation for real-time agricultural decision-making.

Chapter 5

Multimodal Time-Series Modeling of Plant-Based Meat Extrusion

5.1 Introduction

In this chapter, we focus on integrating expert feedback and image data modalities alongside sensor measurements to develop a dynamic process model for a food extrusion system. Whereas the preceding chapters addressed processes for which first-principles descriptions exist and surrogate models primarily served to alleviate computational bottlenecks in dynamic optimization and control, the food extrusion process considered here belongs to a class of systems in which the underlying physics itself is only partially characterized.

Extrusion is the primary manufacturing route for plant-based meat analogs, transforming raw plant proteins into structured products through the coordinated application of heat, pressure, and shear. The process is essential for aligning protein molecules into the fibrous structures that replicate the texture of animal protein products[118;119]. Fibrousness of the final product is a key quality attribute, and maintaining it consistently requires continuous monitoring and precise regulation of the extrusion conditions. Achieving this level of control requires an accurate dynamic process model. However, model development remains challenging because the interaction of material composition, moisture content, temperature, and

mechanical shear induces dynamic variations in material properties such as viscosity and phase behavior[120;121]. Traditional physics-based models struggle to capture the nonlinear, highly coupled dynamics inherent in processing plant protein-based complex matrices[122], and in practice such processes are frequently treated as black boxes, with limited insight into the relationships between operational parameters and product quality[123].

Consistent with the time-series machine learning theme developed throughout this dissertation, data-driven modeling offers a powerful alternative for capturing these complex extrusion dynamics. ML models can characterize the nonlinear relationships between key input parameters, such as feed composition, screw speed, feed rate, in-barrel moisture, and barrel temperature, and critical operational variables, including specific mechanical energy (SME) input, pressure, and temperature[124;125]. Such models serve as efficient surrogates for real-time monitoring and predictive control, directly paralleling the role of the surrogate models developed in Chapter 3. In the extrusion context, ML has been applied to predict rheological properties in biopolymer mixtures, providing insight into structuring mechanisms during high-moisture extrusion[126], and to optimize textural attributes of plant-based products by identifying critical input features such as protein, fat, and moisture content[127].

A defining feature of the extrusion problem, however, is that product quality is not directly measurable through conventional process instrumentation. Texture profile analysis (TPA) has been the standard method for quantifying mechanical properties of extruded products, such as hardness, cohesiveness, and chewiness; yet TPA is a contact-based, destructive technique that requires considerable time and resources, limiting its practicality for high-throughput and real-time applications. Computer vision (CV) methods provide a non-contact, rapid alternative by evaluating surface features in captured images, enabling indirect assessment of textural characteristics. Rather than measuring mechanical properties directly, CV extracts visual attributes such as color, surface patterns, and structural features to predict texture-related parameters, and ML techniques can further enhance these evaluations by identifying correlations between structural features and mechanical properties. For instance, back-propagation neural networks have been used to predict hardness and gumminess from image-derived features of puffed products[128], and optimized ML-based models

have linked image features to the hardness, resilience, and chewiness of plant-based meat products with substantial predictive accuracy[129]. Beyond conventional textural attributes, CV-based approaches have also been employed to assess fibrousness and porosity, offering a scalable pathway for structural monitoring of the extrusion process.

Despite these advances, a critical gap remains in dynamically linking operational parameters to structural quality attributes within a single predictive framework. Industrial process data are often recorded at irregular or mismatched sampling rates, which has motivated time-aware modeling architectures that explicitly account for the elapsed time between observations[130]. In parallel, multimodal data fusion methods have been developed in industrial and manufacturing contexts to integrate heterogeneous sensor streams for process monitoring and quality assessment[131–133]. However, most existing approaches focus on sensor-level fusion and do not explicitly incorporate CV-derived structural descriptors within the temporal dynamics of the process time-series. This limitation is particularly consequential for food extrusion, where both the process dynamics and the evolving product structure must be represented within a unified framework to enable real-time monitoring, optimization, and adaptive control.

Motivated by this gap in the food extrusion case study, this chapter presents a multimodal framework that combines computer vision techniques with time-series machine learning models to connect the process dynamics with product structural attributes. In this work, expert feedback is also incorporated to fine-tune the multimodal model, embedding an additional layer of induced physics that is derived purely from expert knowledge and limited image data rather than from a mechanistic model. In the absence of first-principles descriptions of structure formation, the expert-assigned structure scores act as a surrogate for the unmodeled physics of the process. The feedback encodes the accumulated domain understanding of how fibrousness and porosity emerge from thermomechanical history, and they allow the framework to learn a physically meaningful process-to-quality mapping.

The proposed framework consists of two coordinated components. First, unsupervised and semi-supervised computer vision algorithms quantify product structure from cross-sectional images of the extrudate; the semi-supervised approach is fine-tuned on the expert-

provided structure scores, augmenting edge-based porosity and fibrousness features with Gabor filter-based descriptors to substantially improve scoring accuracy. Second, because the image modality is sampled far more sparsely than the process measurements, gradient-boosting models (XGBoost and LightGBM) fuse the CV-derived structure scores with the time-series process data, yielding an integrated predictive model that tracks structure formation alongside die temperature and specific mechanical energy dynamics. Beyond its immediate monitoring value, this process-to-quality mapping constitutes the modeling foundation for future quality-driven dynamic optimization and model predictive control of extrusion systems.

5.2 Extrusion Process Description

This study employed a pilot-scale extrusion system to process plant-based meat analogs with varying structural characteristics and to investigate the influence of key operational parameters on product quality (Figure 5.1). Extrusion experiments were conducted to collect time-series process data using a data acquisition system and high-resolution images for subsequent modeling and analysis. Detailed descriptions of the extrusion equipment, processing conditions, and material formulations are provided in the following sections.

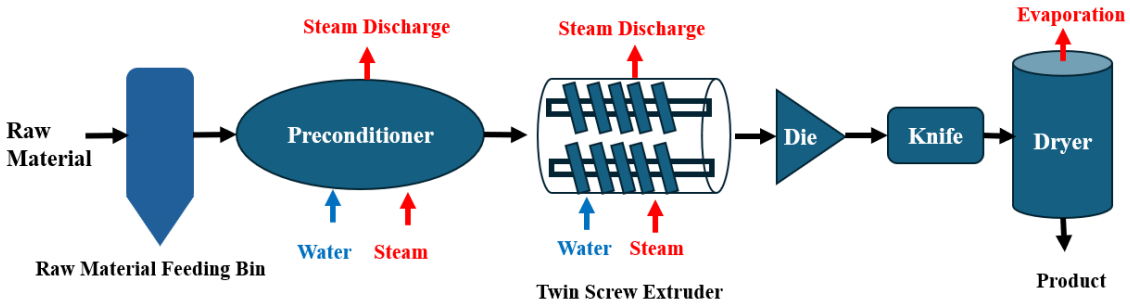

Figure 5.1: *Schematic of the twin-screw extrusion system used in the study.*

5.2.1 Ingredients and formulation

Three distinct plant protein formulations with 70-74% protein content were designed to optimize the structure and texture of plant-based meat analogs, based on previous research[134]. These formulations varied depending on the protein isolate used: pea protein isolate (PPI), soy protein isolate (SPI), or wheat gluten (WG).

5.2.2 Extrusion operational conditions

All formulations were processed using a pilot-scale co-rotating twin-screw extruder (TX-52, Wenger Manufacturing, Sabetha, KS, USA). The extruder barrel had four heating zones, set sequentially to 30, 50, 80, and 110°C from the feed section to the discharge end. A constant feed rate of 50 kg/h was maintained, and the extruder screw speed ranged from 323 to 455 RPM. The screw and die configurations followed those previously studied by Webb et al.[134]. Water was added at rates of 15.2-27.1 kg/h, depending on the formulation, resulting in different in-barrel moisture (IBM) conditions. Different combinations of screw speed and IBM resulted in three process conditions for each formulation. Temperature measurements were collected using thermocouple probes installed along each barrel head and upstream of the die. Post-extrusion, product samples were collected every 2 minutes for time-series texture analysis, corresponding to time points of 0, 2, 4, 6, and 8 minutes from the start of each extrusion run.

SME, a key indicator of the energy imparted to the material during extrusion, was calculated as follows:

$$\text{SME} = \frac{W - W_0}{m} \tag{5.1}$$

where W is the motor power (kW), W_0 is the no-load motor power, and m is the mass flow rate (kg/s). SME values were used to infer the material viscosity and the extent of cross-linking under varying IBM conditions. A data acquisition system continuously monitored and recorded process variables at a one-second sampling rate throughout the extrusion runs. Logged parameters included preconditioner discharge temperature, barrel zone and die tem-

peratures, bin feeder speed (RPM), preconditioner cylinder speed (RPM), extruder motor load, screw and knife speeds (RPM), water injection rates into the extruder and preconditioner, and motor power (kW). This time-series dataset was subsequently used to model the dynamics of the extrusion process and to predict textures using machine learning.

5.2.3 Image acquisition and visual analysis

To examine the internal structure of extruded textured vegetable protein (TVP) products, high-resolution macro images were captured using a Nikon D750 digital camera fitted with a 105 mm macro lens and an SB-R200 wireless remote flash. The imaging setup included a Kaiser Copy Stand RS1, two Dracast Camlux Pro LED light panels, and an 18% grey card as the background to ensure standardized lighting and color balance. Image acquisition was performed using CaptureOne software. Before imaging, dried TVP pieces were rehydrated in tap water for 30 minutes and then drained for 5 minutes. Each piece was then sectioned both longitudinally and transversely to expose the internal structure, facilitating visual inspection of fibrosity (cross-linking and layering density) across different treatment conditions.

For final product structure analysis, 36 images were obtained. Two extrudate pieces were randomly selected from each of the nine treatment conditions and cut along both longitudinal and transverse planes, yielding four images per product. For time-series structure analysis, one piece was randomly sampled at 2-minute intervals (0, 2, 4, 6, and 8 minutes) during the initial 10 minutes of extrusion, resulting in five pieces per run. These pieces were sliced horizontally and imaged, producing 45 images for temporal analysis. In total, 81 images (36 from final products and 45 from the time-series analysis) were collected for visual examination. All product structure images were provided with subjective scores for fibrosity ranging from 1 to 10 (1 for a very porous structure; 10 for a highly fibrous and layered structure), which were used for training and evaluating the semi-supervised CV-based model, as well as for generating the ground-truth structure scores integrated into the time-series modeling framework.

5.3 Integration of Time-Series ML and CV for Extrusion Modeling

The integration of time-series ML and CV techniques enables the capture of dynamic relationships between extrusion process parameters and product structural characteristics. The proposed framework is structured around two key components: (1) structure quantification through CV-based methods, incorporating an unsupervised approach to generate structure scores and a semi-supervised refinement process informed by expert input; and (2) integration of time-series image-derived structure scores with process data using shallow ML models, such as XGBoost and LightGBM, to predict structure formation with die temperature and SME dynamics. Each component addresses specific challenges in the extrusion process, thereby enhancing process monitoring and product quality assessment. The following sections detail the methodologies and their respective objectives.

5.3.1 Data collection

Extrusion experiments were conducted to evaluate the dynamic process conditions influencing structure formation in plant-based meat production. Three feed compositions were investigated: FBC + Gluten + SPC, FBC + PPI + SPC, and FBC + SPI + SPC. These formulations were chosen to represent a range of plant-based protein sources, enabling a comparative analysis of extrusion behavior. Three time-series extrusion runs were performed for each composition under varying in-barrel moisture conditions (low and medium). Each run lasted approximately 20 minutes to capture transient and steady state process characteristics.

Structure characterization samples were collected every two minutes during the first 10 minutes of each run, resulting in 81 image samples across all feed types and trials. Both horizontal and vertical cross-sections of the extrudates were prepared to evaluate structural anisotropy and variability. Beyond the 10-minute mark, the extrudates were shaped into patties; these samples were excluded from structural analysis and ML integration because

the focus was on early-stage structure formation. Process variables, including screw speed (RPM), water feed rates, mechanical load, and temperature, were continuously recorded from both the preconditioner and extruder at one-second intervals, producing a high-resolution time-series. In contrast, structure imaging generated a low-frequency dataset. This mismatch in temporal resolution poses a challenge for model development and data fusion, which is addressed through the integration strategies described in subsequent sections.

5.3.2 Preprocessing and feature engineering

A thorough understanding of the systematic interactions among extrusion variables is critical for developing robust predictive models. As illustrated in Figure 5.2, extrusion dynamics are governed by a network of interdependent relationships. Raw material properties, feed rates, and preconditioner conditions influence the distribution of thermal and mechanical energy, propagating through intermediate parameters such as in-barrel moisture, specific thermal energy (STE), and residence time. These intermediate states, in turn, affect key measurable outputs, including die pressure, SME, and die temperature, variables that directly shape the final product structure. In this framework, the variables in the upper region of Figure 5.2 represent interconnected inputs responsible for process variability and internal state evolution. These inputs include raw material moisture, preconditioner water flow rate, raw material feed rate, screw speeds, and extrusion load. The middle region captures transitional state features, such as in-barrel moisture and residence time, while the lower region reflects critical output variables, specifically die temperature, SME, and die pressure, that directly affect product structure. As highlighted in the figure, the selection of model outputs is guided by this hierarchical structure and informed by extrusion-structure relationships.

Although transitional variables such as residence time and in-barrel moisture provide valuable insights into the internal dynamics of the extrusion process, they are not used as direct model inputs because real-time measurements are typically limited in extrusion systems. However, their effects are implicitly captured through the first-level input features. For example, variations in feed rate, water content, and screw speed are closely correlated

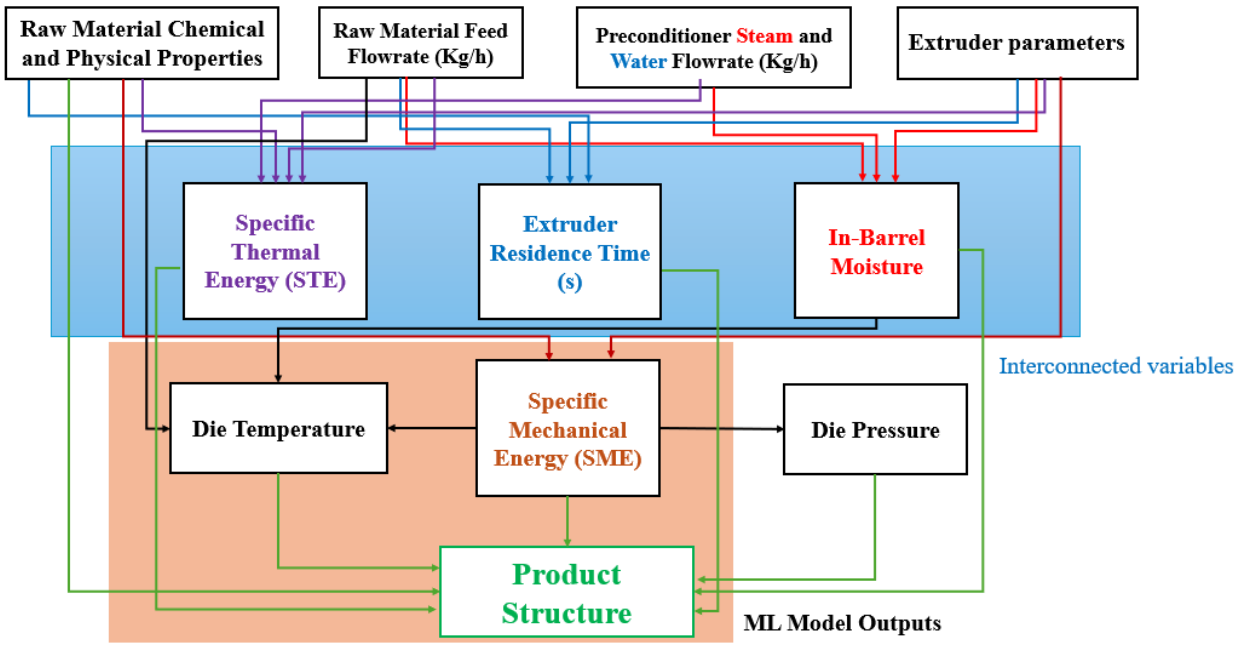

Figure 5.2: *Interconnected network of extrusion process variables.*

with changes in residence time and in-barrel moisture, enabling the ML models to infer their influence indirectly from the observed inputs. Similarly, while die pressure significantly impacts the final product structure, it is often unavailable at high temporal resolution. Instead, its functional effects are represented through SME, which integrates both mechanical loading and energy dissipation within the system. SME thus serves as an accessible proxy for several dynamic variables that influence product formation and is used as a target variable in predictive modeling.

The dataset used for model development consists of high-frequency time-series process data sampled at 1-second intervals across 10 extrusion runs, resulting in approximately 10,000 data points. It encompasses various operational parameters, including screw speeds, loads, water content, and zone temperatures. Among these variables, die temperature and SME were selected as the primary targets for prediction because they critically influence product structure development. In contrast, temperatures in the early zones remain relatively stable near ambient conditions and exhibit minimal influence on extrusion dynamics. The sequential input features comprise historical values of the target outputs, while covariates,

including extruder loads, screw speeds (RPMs), zone temperatures, and water contents, are incorporated to account for their effects on system behavior and the evolution of outputs.

For modeling purposes, it is critical to differentiate between independent and dependent input features. Independent features correspond to external control parameters that are not significantly affected by internal system dynamics, whereas dependent features emerge from intrinsic system-material interactions. In this study, zone temperatures, regulated by PID controllers, are treated as independent inputs, as they remain relatively stable and are minimally affected by downstream process variations. In contrast, die temperature, SME, water content, and extrusion load are dependent variables that reflect the system's dynamic thermal and mechanical responses. Both independent and dependent features are integrated as covariates, along with sequential die temperature and SME measurements, to capture the complex interactions between controlled inputs and system responses. This structure establishes a physically consistent and process-aligned basis for sequence modeling. The vector of independent covariates, $I(t)$, consists of controllable process inputs:

$$\begin{aligned} I(t) = [&\text{Raw Material Moisture},\ \text{Extruder Water Content},\ \text{Extruder Screw Speed},\\ &\text{Preconditioner Water Content},\ \text{Zone 3 Temperature},\ \text{Zone 4 Temperature}] \end{aligned} \tag{5.2}$$

The vector of dependent covariates, $D(t)$, represents the measured process state variables:

$$D(t) = [\text{Bulk Density},\ \text{Extruder Load},\ \text{Specific Mechanical Energy},\ \text{Die Temperature}] \tag{5.3}$$

As the next step, all selected features were standardized using Z-scores. This method transforms the feature by subtracting its mean and dividing by its standard deviation, ensuring zero mean and unit variance[?] . The transformation is expressed as:

$$x' = \frac{x - \mu}{\sigma} \tag{5.4}$$

where x represents the original feature value, μ is the mean of the feature within the dataset,

σ denotes its standard deviation, and x' is the normalized value. This scaling ensures that all features contribute equally to the model, preventing those with larger magnitudes from dominating the learning process. Z-score standardization is particularly beneficial for time-series models, as it accelerates convergence during training and improves overall performance by maintaining consistent gradient behavior across features.

5.4 CV methods for structure quantification

This study introduces two CV-based approaches for structure quantification: an unsupervised method and a semi-supervised method. The unsupervised approach assigns numerical structure scores to extrusion samples by extracting features like edge patterns. This enables automated classification into fibrous or porous categories without requiring labeled data. To improve accuracy, the semi-supervised method integrates expert-provided labels, refining structure scores based on these annotations. Labeled data enhances the model's ability to distinguish structure variations more effectively.

5.4.1 Unsupervised learning CV method

The proposed unsupervised learning-based CV method quantifies structure characteristics by analyzing image samples through edge detection and the Hough transform. As shown in Figure 5.3, the workflow begins with Canny edge detection to extract edges from the sample images, which serves as the basis for subsequent structural analysis. Canny edge detection was chosen because it is robust to noise, computationally efficient, and effective at detecting well-defined boundaries, making it suitable for limited datasets and real-time applications. The resulting edge map is then processed using the Hough transform to identify porous structures as circular features and fibrous structures as linear patterns. This approach leverages the distinct geometric signatures of pores and fibers in extruded products, enabling interpretable feature extraction without requiring large labeled datasets or deep learning-based models.

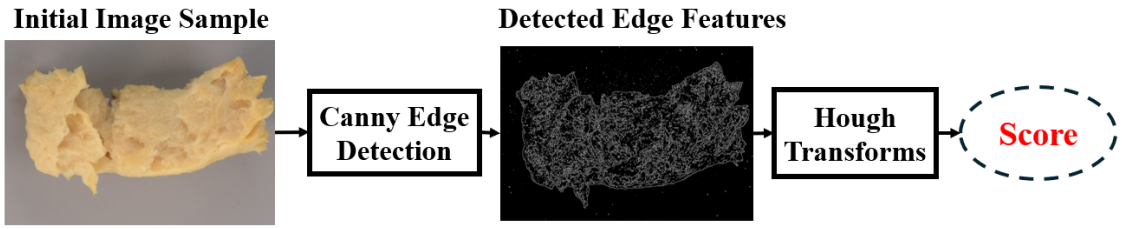

Figure 5.3: *Unsupervised CV algorithm for structure quantification.*

To quantify structure, the algorithm applies the Hough circle transform to identify circular features associated with porosity and the Hough line transform to detect linear features indicative of fibrousness. The counts of these features are normalized to compute their relative contributions, forming the basis of a structure score. This score is assigned on a scale of 1 to 10, with higher values indicating greater fibrousness. A threshold of 5 is used for binary classification to distinguish porous from fibrous samples. This scoring system provides a proportional, interpretable metric for quantifying structure, thereby enabling further integration with time-series ML models.

This method offers several advantages, particularly its ability to operate without labeled data, making it suitable for applications where expert-annotated training sets are unavailable. Leveraging edge detection and geometric transformations to extract essential structural features with minimal preprocessing enables effective performance on small image datasets. Its unsupervised nature reduces reliance on domain-specific knowledge, enhancing its applicability to diverse structure analysis tasks. Moreover, the method is computationally efficient and easy to implement, allowing for rapid deployment without extensive preprocessing or expert intervention.

Despite these advantages, the method has certain limitations. Its performance depends on fine-tuning Canny edge detection parameters, such as edge detection thresholds, which must be carefully adjusted for different datasets to suppress noise and improve feature extraction, as illustrated in Figure 5.4. This requirement complicates standardization across diverse image samples. Additionally, variations in imaging conditions, including shadows, inconsistent lighting, and camera quality, can affect the method's robustness. The reliance on geometric interpretations may also oversimplify the representation of structure, poten-

tially leading to misclassification in cases with complex or irregular structures. Furthermore, the absence of expert knowledge integration limits its ability to assess ambiguous or complex structures.

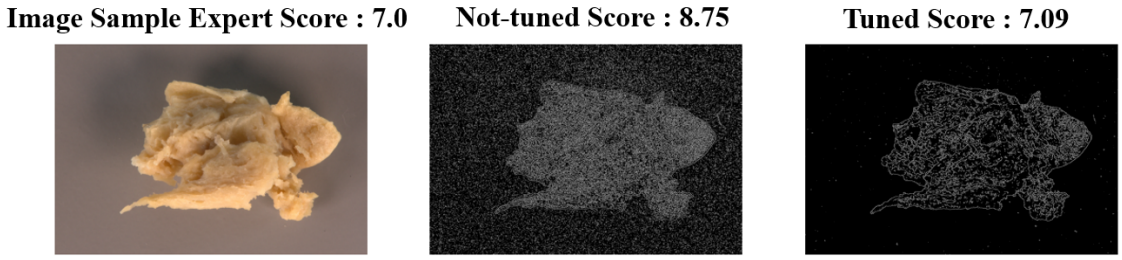

Figure 5.4: *Effect of tuning Canny edge detection parameters on the performance of the unsupervised CV method.*

5.4.2 Semi-supervised learning CV method

The semi-supervised method extends the unsupervised approach by integrating expert-labeled data and additional structure features extracted through Gabor filtering. This enhancement mitigates the limitations of the unsupervised method by incorporating a richer set of features and leveraging supervised learning to improve the accuracy of structure quantification. This approach is semi-supervised because it combines unsupervised feature extraction with a supervised learning model.

As illustrated in Figure 5.5, Gabor filters are used to extract detailed structure features, which are then integrated with edge-based porosity and fibrousness scores computed using Hough transforms. These image-derived features are fed into an XGBoost regressor, with the expert-labeled structure scores (1-10) used solely as regression targets during training. The expert scores are not part of the input features; instead, they serve as ground-truth labels that direct the supervised learning process.

The dataset comprises 81 image samples, including horizontal and vertical cuts of the final product for data augmentation. The data is split into 80% for training and 20% for testing. The regressor is configured with five estimators and optimized using an MSE loss function. This setup enables precise prediction of structure scores on a 0-10 scale. A threshold of 5

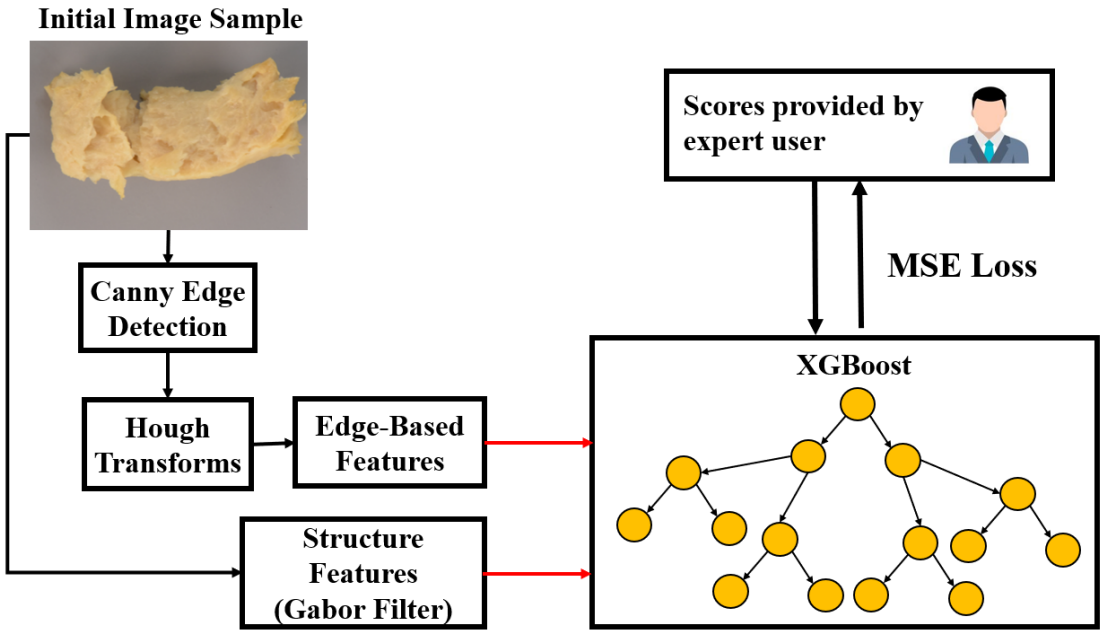

Figure 5.5: *Semi-supervised CV algorithm for structure quantification.*

is applied for binary classification into porous and fibrous categories. The refined structure scores contribute to an integrated time-series dataset, thereby establishing a link between structure characteristics and extrusion process parameters.

5.5 Integrated time-series framework for extrusion dynamics and structure modeling

The integrated dataset combines high-frequency process data (sampled every second) with low-frequency image-based structure scores (sampled every two minutes). To enable joint analysis, the process data are downsampled to one-minute intervals, and the resulting gaps in the structure-score time-series are filled by interpolation. Interpolation is applied only to the low-frequency image-based structure scores and is used exclusively to temporally align these scores with the process data, not to reconstruct high-frequency structural transients. Because the structure scores capture slowly evolving, cumulative characteristics, linear interpolation preserves their intended trends and avoids introducing artificial high-frequency behavior or

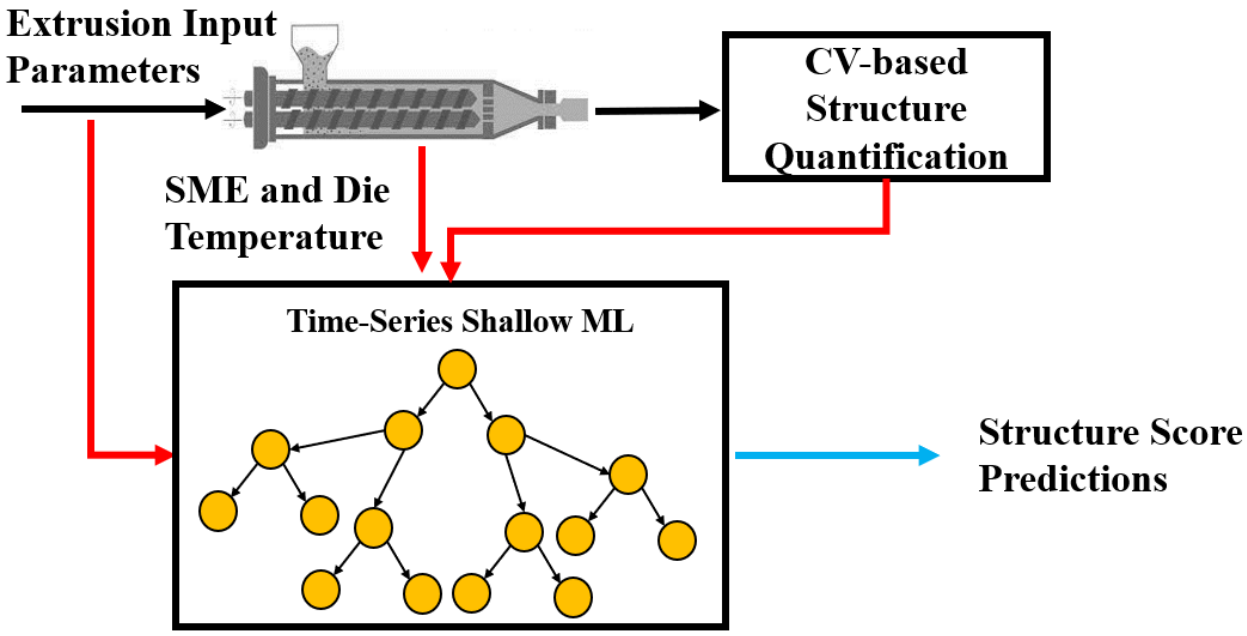

Figure 5.6: *Schematics of the integrated time-series framework for structure score prediction.*

obscuring rapid process dynamics. This preprocessing step ensures continuity while keeping the dynamic characteristics of the extrusion process.

The final dataset includes time-aligned operational parameters, such as zone temperatures and structure scores derived from the semi-supervised method. For each feed composition, 30 minutes of data are collected, resulting in a small yet representative dataset. To maintain temporal consistency, data splitting is performed independently for each time-series, with 80% allocated for training and 20% for testing.

After data partitioning, z-score scaling is applied to normalize the input features, ensuring that all variables contribute equally during model training. XGBoost and LightGBM models are then used to predict the next-minute structure score from lagged input features, including prior values of die temperature, SME, and structure score (Figure 5.6). The integrated time-series framework incorporates observations from the previous 2 minutes, thereby capturing temporal dependencies and dynamic system behavior that are critical for accurate predictions.

Although deep learning models such as transformers and RNNs are well-suited for cap-

turing long-term dependencies, they are not used in this context due to the limited dataset size, which increases the risk of overfitting and reduces generalizability. In contrast, XGBoost and LightGBM are better suited for this task, given their ability to handle small datasets and their built-in regularization, which ensures robust performance while maintaining computational efficiency. Mathematically, the modeling framework is formulated as follows:

$$Y(t) = [T_{\text{die},t}, S_{\text{structure},t}] \tag{5.5}$$

$$Y_{t+1:t+k} = F_{\text{XGB-LightGBM}}(Y_{t-1:t-K}, I_{t-1:t-K}, D_{t-1:t-K}) \tag{5.6}$$

Both models are trained using the MSE loss function to optimize predictive accuracy. Given the small dataset size, hyperparameters are carefully tuned to enhance generalization and mitigate overfitting. The key hyperparameter configurations are as follows:

- **XGBoost:** 30 estimators, maximum depth of 2, learning rate of 0.1, and full dataset usage for subsampling.
- **LightGBM:** 30 estimators, maximum depth of 2, learning rate of 0.1, four leaves per tree, and full data usage for subsampling.

5.6 Results and Discussion

This section presents the simulation results and evaluations of the proposed methodologies for modeling and analyzing twin-screw extrusion of plant-based meat. The results are organized into three parts. First, the effectiveness of CV techniques is examined, with an emphasis on a semi-supervised approach to structure quantification that employs edge detection and image-based analysis to estimate fibrousness and porosity scores. Each modeling component is evaluated independently using task-specific performance metrics to isolate its contribution before integration into the framework. Third, the integration of these methodologies into a multimodal framework is evaluated, demonstrating how ensemble models can effectively link high-frequency process data with low-frequency structural predictions, enabling real-time

monitoring and process optimization.

5.6.1 CV-based structure quantification results

CV-based methods quantify structure by analyzing detected edges and extracting structure features. Using a threshold of five, images are classified into two categories: fibrous or porous. To evaluate the performance of both unsupervised and semi-supervised structure quantification approaches, binary classification metrics, accuracy, confusion matrices, ROC curves, and AUC values are used. The scoring performance is assessed using MAE and mean bias, with predictions compared against expert-labeled references. This dual evaluation framework enables both binary classification analysis and a more granular assessment of structure scoring. The latter is particularly critical in the multimodal framework, where precise structure scores are key training features.

Figure 5.7 presents the confusion matrices for both methods, illustrating their ability to differentiate between structure types in a test set comprising 25 time-sequenced image samples. The components of the confusion matrix are defined as follows:

- **True Positives (TP)**: *Fibrous* images correctly classified as *fibrous*.
- **False Positives (FP)**: *Porous* images incorrectly classified as *fibrous*.
- **True Negatives (TN)**: *Porous* images correctly classified as *porous*.
- **False Negatives (FN)**: *Fibrous* images incorrectly classified as *porous*.

In structure classification, **FP** errors are particularly critical because they mislabel porous structures as fibrous, leading to incorrect assumptions about the product's structural properties. Similarly, **FN** errors misclassify fibrous structures as porous, compromising the classification system's reliability. The unsupervised method exhibits more pronounced **FP** errors, highlighting its limited ability to accurately distinguish porous structures. In contrast, the semi-supervised approach demonstrates significant improvement, eliminating **FP** errors while increasing **TP** detections.

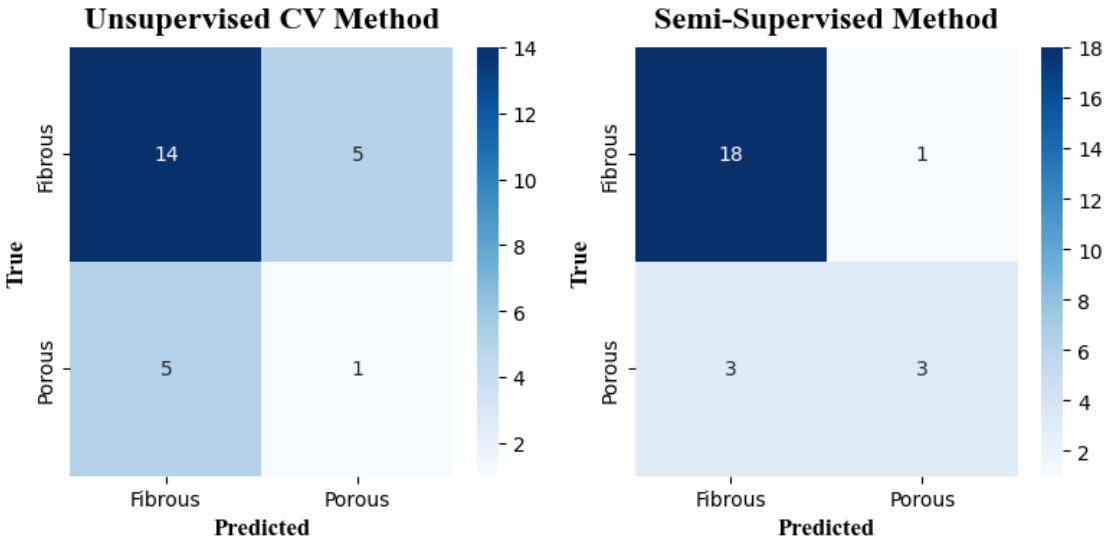

Figure 5.7: *Confusion matrices for the CV-based methods.*

Building on the information provided by the confusion matrices, Figure 5.8 presents the ROC curves for both the unsupervised and semi-supervised CV methods. A preliminary visual assessment suggests that the semi-supervised model better distinguishes between fibrous and porous structures. This improvement stems from integrating edge-based and structure features extracted using Gabor filters and expert-labeled data for fine-tuning. The ROC curve further illustrates that the semi-supervised approach achieves a higher TP rate while effectively reducing FPs, where porous structures are misclassified as fibrous. In contrast, the unsupervised method struggles to suppress false positives, thereby limiting its ability to reliably differentiate between structure types. The higher AUC metric reinforces the superior classification performance of the semi-supervised method.

The evaluation metrics for binary classification and structure score prediction are summarized in Table 5.1. Accuracy is the proportion of correctly classified structures among all samples and is a key indicator of binary classification performance. MAE quantifies the average absolute deviation between predicted structure scores and expert-labeled values, measuring prediction precision. Mean bias indicates the directional error in predictions, where negative values suggest a tendency to underestimate structure scores. The accuracy values align with previous observations from ROC curves and confusion matrices, reaffirming

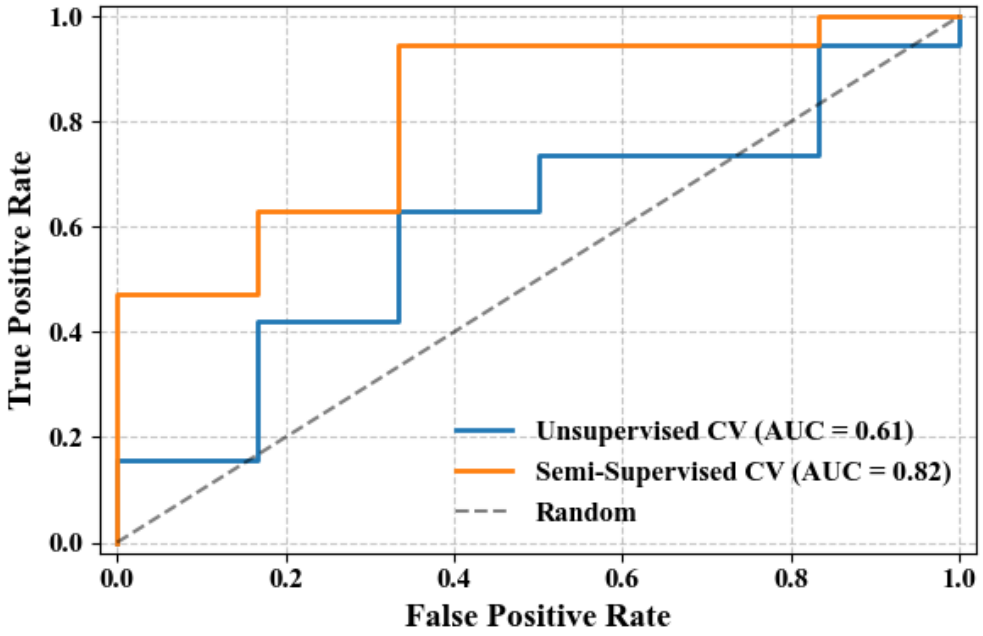

Figure 5.8: *Comparison of ROC curves for CV-based methods.*

the semi-supervised method's superior performance in classifying fibrous and porous structures. For structure score prediction, the lower MAE values indicate that the semi-supervised approach produces predictions closer to expert-labeled scores. Additionally, the mean bias values demonstrate negligible systematic bias in the semi-supervised method, whereas the unsupervised approach consistently underestimates structure scores.

Table 5.1: *Evaluation metrics for CV methods: Binary classification and structure score prediction.*

Method	Accuracy	MAE	Mean Bias
Unsupervised CV Method	0.60	1.29	-0.13
Semi-Supervised CV Method	0.86	1.08	0.14

These metrics further validate the advantages of the semi-supervised approach, which integrates expert-labeled data with Gabor filter-based structure features. This combination enhances classification accuracy and the reliability of structure score predictions, which are essential for the integrated time-series framework to effectively capture the dynamic relationships between operational parameters and structure attributes. Figure 5.9 compares the structure scores generated by the unsupervised and semi-supervised methods with

expert-labeled ground-truth scores across different feed compositions in time-series data. The semi-supervised method exhibits greater consistency with expert annotations, particularly in regions with pronounced structure transitions. In contrast, the unsupervised approach exhibits more pronounced deviations, particularly in complex or ambiguous structures. These results underscore the importance of incorporating domain knowledge via expert-labeled data, thereby enhancing the accuracy of the semi-supervised framework.

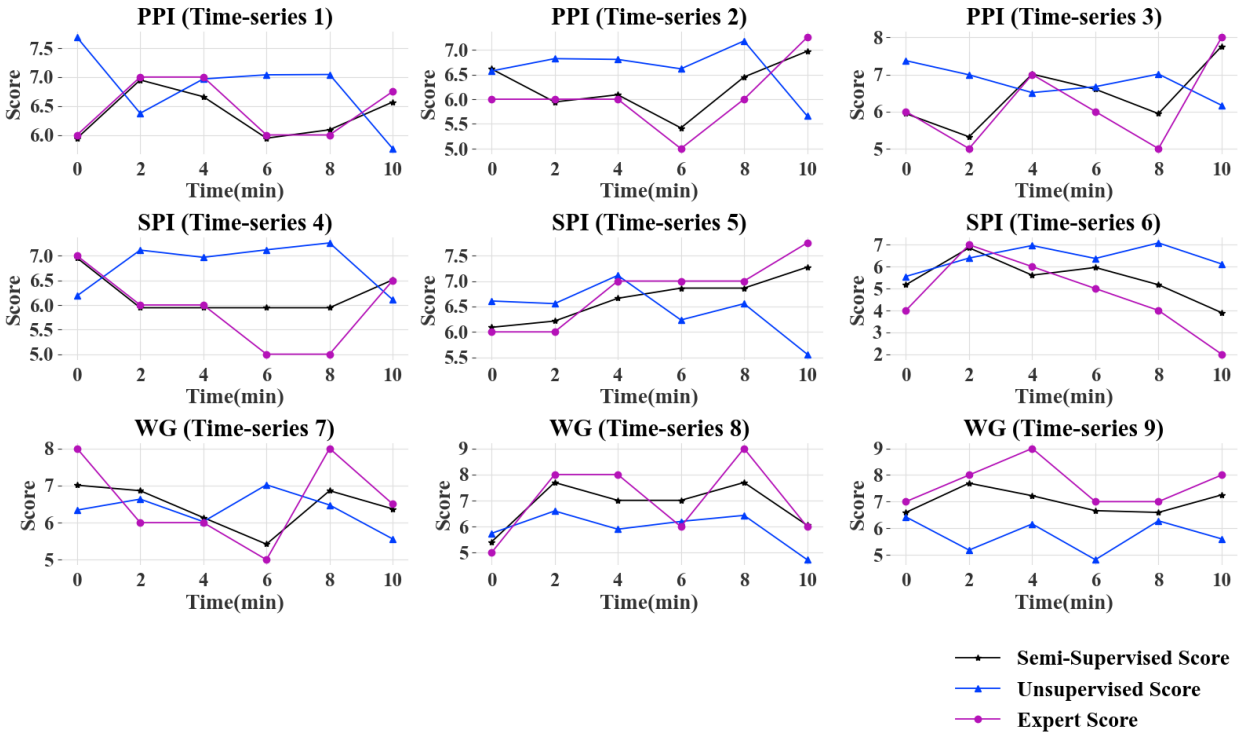

Figure 5.9: *Comparison of structure scores predicted by the unsupervised and semi-supervised methods with expert-labeled scores across feed compositions and time-series data.*

Furthermore, the integration of domain knowledge through expert feedback offers another avenue for developing physics-guided dynamic models, in which deviations from process measurements can be rectified by expert input despite the absence of first-principles models.

5.6.2 Time-series ML integration with structure quantification

The integration of extrusion process dynamics with dynamic structure evolution is achieved by computing structure scores from time-series samples, constructing a dataset at a 1-minute

sampling rate, and aligning temperature records with their corresponding structure scores. This integrated dataset establishes a direct relationship between process parameters and product structure, facilitating the development of a multimodal model that captures the interaction between operational conditions and product quality.

Given the relatively small size of the integrated dataset, the training and test sets are used to evaluate model performance. The limited data increases the risk of poor generalization and extrapolation, necessitating rigorous validation. This study primarily examines the feasibility and effectiveness of the proposed framework for predicting the evolution of dynamic structures, rather than optimizing or deploying the production model. The findings provide insights into the challenges and benefits of integrating time-series process data with structure quantification methods and expert feedback.

Integration with XGBoost and LightGBM

Figures 5.10, 5.11, and 5.12 present the time-series predictions generated by the XGBoost model, while Figures 5.13, 5.14, and 5.15 display the corresponding predictions from the LightGBM model across different feed compositions. Both models effectively capture key trends in die temperature, SME, and structure score dynamics within the training set. However, discrepancies between observed and predicted values in the test set indicate challenges in generalization, primarily due to the limited size and variability of the integrated dataset.

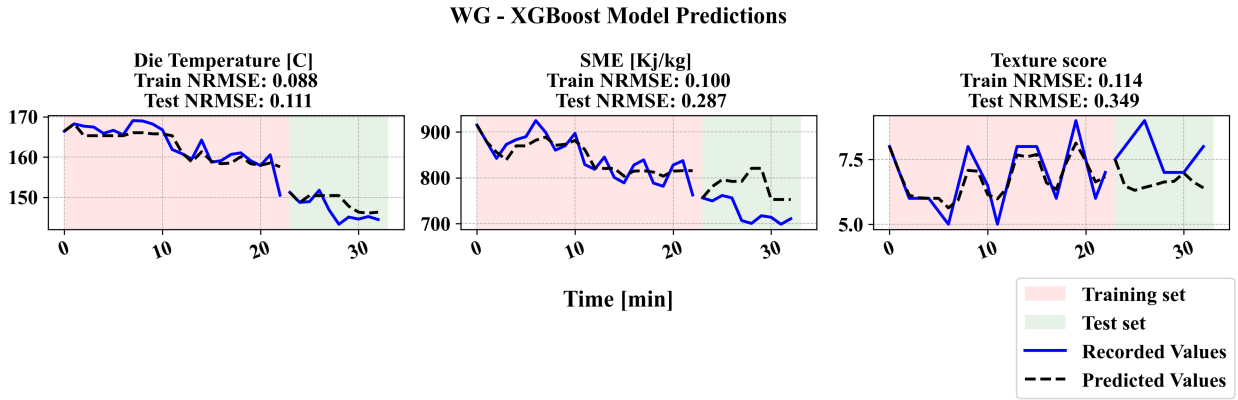

Figure 5.10: *XGBoost model time-series prediction for WG.*

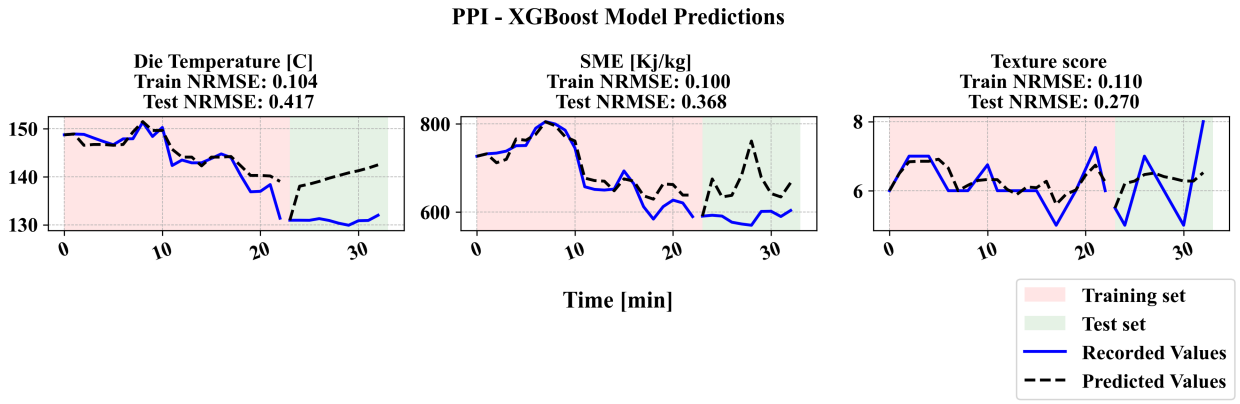

Figure 5.11: *XGBoost model time-series prediction for PPI.*

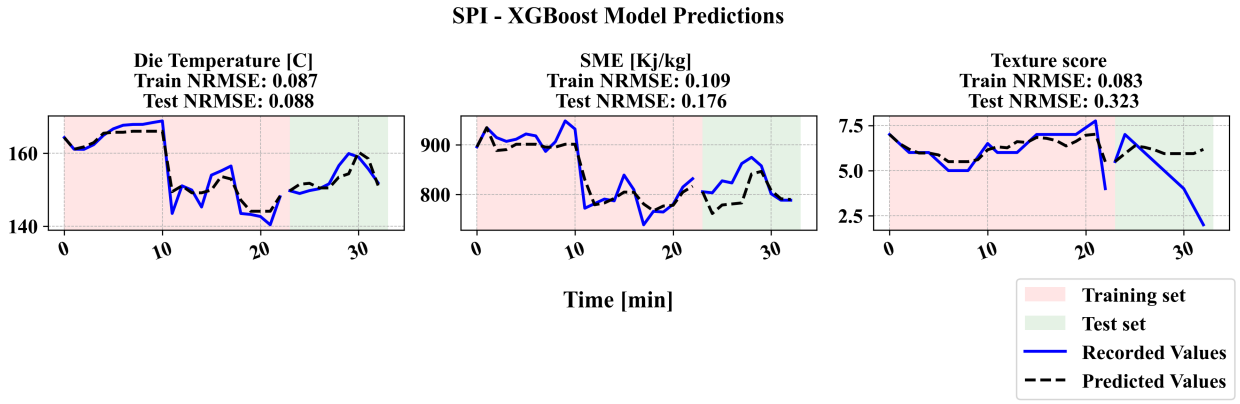

Figure 5.12: *XGBoost model time-series prediction for SPI.*

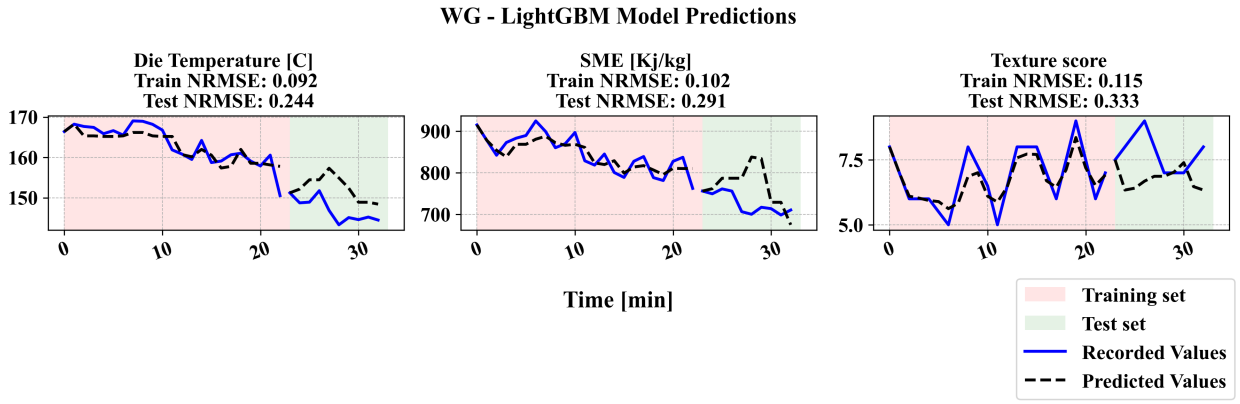

Figure 5.13: *LightGBM model time-series prediction for WG.*

Overall, both XGBoost and LightGBM demonstrate the feasibility of integrating operational parameters and structure scores within a unified ML framework. While XGBoost effectively captures dominant trends in the data, LightGBM offers a computationally effi-

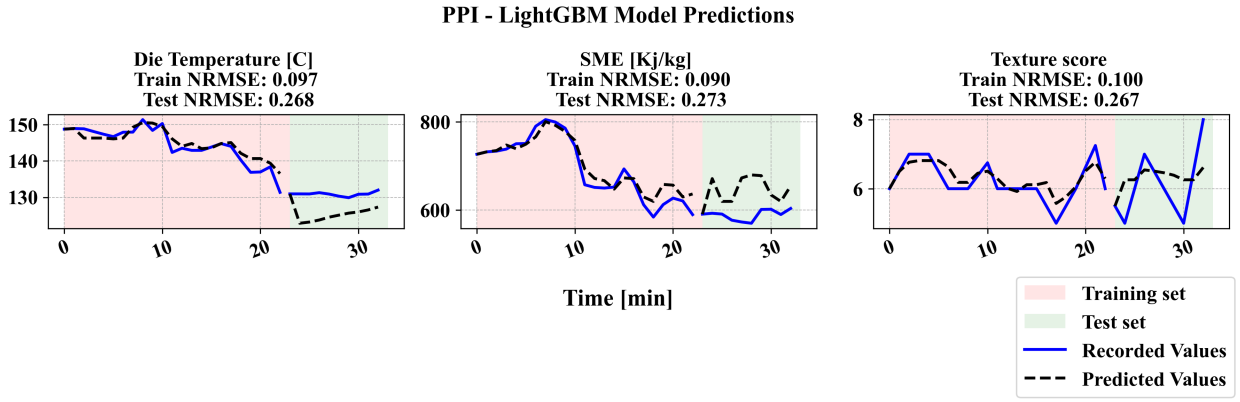

Figure 5.14: *LightGBM model time-series prediction for PPI.*

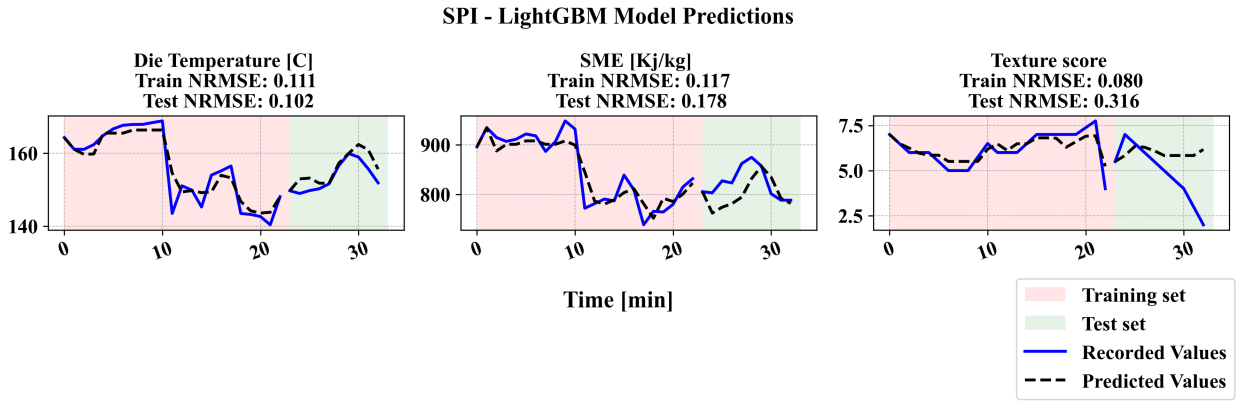

Figure 5.15: *LightGBM model time-series prediction for SPI.*

cient alternative suited for rapid inference. This integration underscores the potential for real-time monitoring and control of extrusion processes by establishing a direct link between process conditions and product structure characteristics. Future advancements, such as data augmentation and model refinement, can further enhance generalization, leading to more accurate and robust predictions in dynamic extrusion environments.

General evaluations of integrated models

The performance metrics for the integrated models, XGBoost and LightGBM, are summarized in Table 5.2. The table presents MAE and NRMSE for the training and testing datasets, enabling a comparative assessment of the predicted outputs. A comparison of performance metrics indicates that XGBoost and LightGBM yield similar results during training, with

consistently low NRMSE and MAE across all process parameters. However, evaluation on the test set highlights limitations in generalization, particularly for SME and die temperatures, where both models exhibit elevated NRMSE and MAE. In predicting structure scores, the models perform comparably, with LightGBM achieving a test set NRMSE similar to XGBoost's. These findings suggest that although both models struggle to generalize predictions of process outputs to unseen data, they achieve higher accuracy in estimating structure scores. This underscores the potential of integrating process data with CV-based features for constrained applications.

Table 5.2: *Evaluation metrics for XGBoost and LightGBM across all feed compositions.*

Model	**Parameter**	**MAE**		**NRMSE**		**Data Range**
		Train	**Test**	**Train**	**Test**	
XGBoost	Die Temperature [C]	1.403	4.207	0.052	0.192	169.04 - 129.91
XGBoost	SME [kJ/kg]	14.695	51.770	0.051	0.224	948.12 - 570.06
XGBoost	Structure Score	0.275	1.026	0.074	0.203	9.0 - 2.0
XGBoost Mean		**-**	**-**	**0.059**	**0.206**	
LightGBM	Die Temperature [C]	1.432	2.787	0.052	0.117	169.04 - 129.91
LightGBM	SME [kJ/kg]	15.282	50.092	0.052	0.210	948.12 - 570.03
LightGBM	Structure Score	0.281	1.028	0.073	0.199	9.0 - 2.0
LightGBM Mean		**-**	**-**	**0.059**	**0.175**	

The limited performance of the integrated models on unseen data is primarily due to the small dataset size, which is constrained by the sparse two-minute sampling interval for image collection. Although hyperparameter tuning was employed to mitigate overfitting, the disparity between high-frequency process data (sampled every second) and low-frequency image acquisition restricts the volume of usable data. This mismatch in sampling rates introduces a bottleneck in model integration, thereby reducing the robustness and generalizability of structure score predictions in the time-series framework. Despite these challenges, the models offer valuable insights into the dynamic relationship between process conditions and product structure, underscoring opportunities for future improvements.

Although the CV-based structure scoring in this study is illustrated with a two-minute image sampling interval, neither the proposed CV method nor its integration into the multimodal time-series framework is inherently limited to a fixed sampling rate. In principle, the

framework can operate at higher temporal resolution, and the main obstacle to real-time deployment is the image acquisition rate during extrusion operations rather than computational delay. As a result, practical real-time implementation would require faster image sampling, which can be enabled by hardware solutions such as automated camera triggering[135] and edge computing architectures[136]. Deploying lightweight image processing pipelines on embedded vision systems or microcontroller-based platforms would further support on-device feature extraction and near–real-time structure inference, enabling the framework to provide real-time or near-real-time monitoring, depending on operational and hardware constraints.

5.7 Conclusions

In this chapter, the integration of expert feedback and image modalities into time-series process modeling was investigated, with an application to food extrusion systems. A semi-supervised CV method, fine-tuned on expert-assigned structure scores, quantified product structure from cross-sectional images and outperformed its unsupervised counterpart in both classification accuracy and scoring error, while gradient-boosting models fused the resulting structure scores with the process measurements into an integrated dynamic model. The findings of this chapter indicate that, even in the absence of first-principles models, collecting domain knowledge and embedding it in dynamic modeling alongside additional data types, such as images, offers a viable path toward developing unconventional dynamic models for processes whose governing physics is poorly characterized. Moreover, the presented time-series modeling framework opens an avenue for real-time decision-making, dynamic optimization, and even model predictive control of food extrusion systems, where the control objective could target product quality directly, since the developed dynamic model links process parameters to product quality under circumstances where soft sensing is limited or challenging.

Chapter 6

Transformer-Based Monitoring for Cyber-Resilient Operation of Highly Integrated Chemical Process Systems

6.1 Introduction

Throughout this dissertation, several case studies with different levels of integration between domain knowledge and time-series ML regression have been presented for the dynamic modeling of process systems, ranging from surrogate models embedded in model predictive control (Chapter 3), to hybrid physics-informed formulations (Chapter 4), to multimodal architectures linking process time-series data with complementary measurement modalities (Chapter 5). A remaining aspect of the overall research objective is the capabilities of time-series ML methods in addressing process resiliency in the event of cyberattacks. In this chapter, the focus is therefore shifted from regression to classification, answering the question of how time-series ML models can be employed as monitoring components for detecting and identifying adversarial cyberattacks, particularly in highly integrated process systems where attack-induced disturbances propagate rapidly across interconnected units.

The relevance of this question follows directly from the structural trends motivating the

earlier chapters. The chemical manufacturing industry is transitioning toward highly integrated process networks, characterized by enhanced energy and material recycling and optimized heat and mass transfer to improve efficiency and sustainability. This evolution is supported by the deployment of advanced sensing and computational technologies that enable real-time monitoring, optimization, and control, leading to the classification of modern manufacturing systems as cyber-physical systems (CPS)[137;138]. While CPS offer substantial advantages, including cost reduction and improved energy utilization[139], their increased connectivity exposes them to heightened cyber vulnerabilities[140–142]. Cyberattacks on industrial control systems can compromise safety, disrupt operations, and lead to significant economic losses[143;144]. Consequently, ensuring the security and resilience of CPS necessitates advanced detection, mitigation, and recovery strategies that operate within the closed-loop estimation and control architecture itself.

Substantial research effort has been directed toward this goal over the past two decades. Early and still widely used approaches rely on model-based residual analysis combined with statistical change detection, most notably cumulative sum (CUSUM) and chi-squared detectors that flag persistent deviations between measured and predicted process behavior[145;146]. Although computationally attractive, such threshold-based statistical schemes are vulnerable to stealthy falsification strategies deliberately shaped to remain below detection thresholds, and their effectiveness degrades as the number of coupled measurement channels grows[147]. In parallel, control-theoretic contributions have addressed secure state estimation under adversarial corruption[148], geometric characterizations of detectability[149], game-theoretic designs that balance robustness and resilience[150], and MPC formulations that mitigate denial-of-service attacks[151]. Within the process systems engineering literature, cyber-secure control-detector architectures[40], active detection strategies based on control mode switching and set-based analysis[152;153], sensor-actuator pairing for resilient interconnection design[154], and Lyapunov-based economic MPC formulations with integrated detection and handling[155? ;156] have been developed to preserve closed-loop stability under attack.

More recently, ML-based approaches have demonstrated strong potential for enhancing detection capabilities across different control architectures. Artificial neural network detec-

tors have been combined with Lyapunov-based MPC to alleviate the impact of cyberattacks on chemical processes[39], and ML-based detection has been integrated with encrypted control for enhanced cybersecurity of nonlinear processes[143]. The effects of various attack scenarios have been systematically examined across centralized, decentralized, and distributed MPC frameworks, providing insight into the vulnerabilities and robustness of each architecture[157], while stealthy sensor and actuator attacks on evolving nonlinear process systems have also been addressed[158]. Physics-informed ML formulations, including graph convolutional recurrent networks that encode process topology, have been proposed for cyberattack detection and resilient control of process networks[159;160], and ML-based state reconstruction methods have been introduced to recover measurements compromised by falsified data[161].

Despite this progress, a critical challenge persists in accurately *detecting and localizing* cyberattacks within highly integrated process systems. The tight thermal and material coupling among units means that a disturbance induced by an attack on a single sensor propagates rapidly through recycle streams and shared estimation layers, obscuring the original point of intrusion. The complexity is propagated when estimation and control are tightly coupled. In closed-loop configurations that integrate MPC with MHE, corrupted measurements distort the reconstructed states, which in turn misguide the control actions, amplifying the attack's effect throughout the network. This cascading amplification, referred to in this work as the snowball effect, makes even a single compromised upstream sensor capable of triggering system-wide instability and makes the distinction between routine process disturbances and malicious interventions increasingly difficult. Moreover, when multiple sensors may be compromised simultaneously, the identification task acquires a combinatorial multi-label structure with inherent class imbalance that single-channel statistical detectors and small-scale ML demonstrations are limited in their ability to address. Therefore, there is a need for methodologies that detect and localize attacks with high precision in such settings and mitigate their impact without compromising the stability and performance of the overall system.

This chapter addresses this gap by leveraging the same class of time-series architectures employed for regression in the preceding chapters, specifically self-attention transformers,

in a multi-label classification problem. The sequential learning paradigm allows the models to learn historical context and system dynamics when identifying anomalies, capturing long-range temporal dependencies and cross-variable interactions that make coordinated multi-sensor attacks distinguishable from benign process variability. Two aspects of time-series ML are examined in this case study. (i) whether time-series classification can jointly perform attack *detection* (determining that an attack is active) and *identification* (localizing the compromised sensors) under closed-loop MHE-MPC operation, and (ii) how the monitoring architecture itself, centralized versus distributed, affects detection, identification, and recovery performance in highly integrated systems.

The proposed frameworks are demonstrated on the benzene alkylation with ethylene process[162], a benchmark petrochemical system for ethylbenzene production comprising four continuous stirred-tank reactors and a flash separator connected through recycle streams. The strong thermal and material interdependencies of this process, regulated by an integrated MHE-MPC scheme in which temperature measurements are used to estimate unmeasured concentration states[163;164], make it a representative testing benchmark for sensor-targeted adversarial attacks. Four categories of sensor falsification are considered: min-max, surge, noise-based, and set-difference attacks, spanning single- and multi-sensor compromise scenarios.

Using the proposed benchmark process, two centralized frameworks are first developed[165]. an end-to-end (ETE) decoder-only transformer that jointly performs attack detection and compromised-sensor identification, and a modular two-tier architecture in which a lightweight feedforward neural network handles binary detection and activates a transformer-based multi-label classifier only when an attack is flagged, decoupling detection from the inherently identification task. To address the scalability limitations of centralized monitoring, distributed counterparts of both frameworks (DETE and D2Tier) are then formulated, in which a modularity-based community detection algorithm decomposes the process network into strongly coupled subsystems whose local models operate in parallel, reducing subsystem dimensionality and the combinatorial complexity of multi-sensor identification while preserving system-wide coverage. Since detection and identification alone do not restore resilient

operation, all frameworks are integrated with an extended Kalman filter (EKF)-based post-attack reconstruction mechanism that excludes the identified corrupted measurements from the estimation update and selectively replaces them with reconstructed estimates, allowing the MHE-MPC loop to maintain estimation fidelity and closed-loop stability under sustained attacks. Together, these layers extend the role of time-series ML in this dissertation from predictive components for dynamic optimization and control to protective components for resilient operation.

6.2 Representative sensor-level adversarial attacks in process Systems

Cyberattacks in chemical process systems can take various forms designed to evade data-driven and model-based detection mechanisms. This study focuses on adversarial attacks targeting measurement sensors, altering the output to falsified values. Since control systems rely on sensor feedback for decision-making, such manipulations can lead to substantial deviations in closed-loop system behavior. Unaware of the compromised measurements, the controller acts based on incorrect information, resulting in suboptimal or destabilizing control actions. This work considers four significant types of sensor-level adversarial incidents: min-max attacks, surge attacks, noise-based attacks, and set difference attacks. These attack scenarios are utilized to generate training and testing datasets for evaluating the proposed detection and mitigation frameworks.

6.2.1 Min-max attacks

Min-max attacks manipulate sensor outputs to the extreme values within their permissible operational range, either the minimum or the maximum, while avoiding the activation of alarm systems. These attacks maintain a constant falsified value throughout their duration and can induce the most significant deviation from the nominal state in the absence of

countermeasures. The min-max attack is formulated as follows[40]:

$$\bar{y}(t_i) = \arg \min_{\hat{y} \in \mathbb{R}^s} / \max \{\hat{y}(t_i) \in \Upsilon\}, \quad \forall i \in [i_0,\ i_0 + L_a], \tag{6.1}$$

where $\bar{y}(t_i)$ is the falsified measured output vector during the attack window $[i_0,\ i_0 + L_a]$, L_a specifies the length of the attack window, and $\Upsilon = \{\hat{y} \in \mathbb{R}^s \mid \hat{y}_l \leq \hat{y} \leq \hat{y}_u\}$ defines the operational bounds, with $\hat{y}_l$ and $\hat{y}_u$ representing the lower and upper limits, respectively.

6.2.2 Noise-based attacks

Random noise is continuously added to the true state values throughout noise-based attacks. The noise follows a zero-mean distribution with a specified maximum amplitude and remains within the allowable operational bounds. The falsified measurement is defined as:

$$\bar{y}(t_i) = x(t_i) + \delta(t_i), \quad \forall i \in [i_o,\ i_o + L_a], \tag{6.2}$$

where $\delta(t_i)$ represents the noise value at each time step, drawn randomly from a redefined distribution.

6.2.3 Surge attacks

Surge attacks combine min-max and noise-based strategies and are executed in two distinct phases. In the first phase, sensor outputs are forced to their respective minimum or maximum values, replicating the behavior of a min-max attack. This external manipulation is maintained for a predefined duration L_s. In the subsequent phase, bounded random noise is added to the manipulated output, creating transient fluctuations while preserving the external baseline. This two-stage structure models an abrupt and persistent deviation followed by stochastic perturbations. The surge attack is mathematically represented as:

$$\hat{y}(t_i) = \arg \min / \max \{y(t_i) \in \mathbb{Y}\}, \quad \forall i \in [i_0, i_0 + L_s], \tag{6.3}$$

$$\bar{y}(t_i) = \hat{y}(t_i) + \eta(t_i), \quad \forall i \in [i_0 + L_s, i_0 + L_a], \tag{6.4}$$

where $\eta(t_i)$ denotes the bounded noise introduced during the second phase.

6.2.4 Set difference attacks

Set difference attacks apply a deterministic or structured deviation, denoted by $\Gamma(t_i)$, directly into the sensor measurement without an initial extremal perturbation. In contrast to surge attacks, which follow a two-phase manipulation involving a transient external shift followed by additive noise, set difference attacks employ a single-phase falsification. This falsification can take the form of constant offsets, periodic signals, or adversarially designed patterns. Such structures enable more targeted and potentially stealthy behavior, capable of emulating both deterministic and stochastic anomalies. The compromised measurement is modeled as:

$$\bar{y}(t_i) = \hat{y}(t_i) + \Gamma(t_i), \tag{6.5}$$

where $\Gamma(t_i)$ represents the injected deviation, which may be crafted to bypass conventional statistical anomaly detection techniques.

6.3 Cyberattack Detection Methods

Protecting highly integrated process systems from cyberattacks hinges on timely detection of malicious activity and accurate identification of the specific components affected, tasks that become increasingly complex in interconnected environments. In this context, detection refers to determining the presence and activity of a cyberattack, whereas identification involves recognizing which specific sensors or equipment have been compromised. Given the potential for simultaneous compromise of multiple sensors in such systems, the problem can be effectively formulated as a multi-label classification task. Unlike single-label classification, which predicts a single output class, multi-label classification enables the model to assign multiple labels, one per sensor, indicating whether each is functioning normally or under

attack. This approach accommodates both single- and multi-sensor attack scenarios. In typical machine learning classification tasks involving class imbalance, techniques such as data resampling and weighted loss functions are commonly applied to mitigate the imbalance and reduce learning bias[166]. In this study, however, only a weighted loss function is employed, as the goal is to preserve the natural frequency of attack scenarios to reflect realistic operational conditions. A comprehensive set of standard multi-label classification metrics is used to evaluate the performance of the proposed frameworks under single- and multi-sensor attack scenarios. These metrics include accuracy, precision, recall, F1-score, area under the ROC curve (AUC), Jaccard index, root RMSE, and explained variance (EV). Although the AUC serves as the primary quantitative metric for comparative assessment, ROC curves are also presented to visually compare model performance across various attack configurations. Collectively, these metrics assess the models' detection effectiveness and robustness by quantifying their ability to minimize false positives and missed detections. A detailed summary of these evaluation criteria is provided in the Supporting Information.

6.3.1 Multi-Label classification and binary cross-entropy loss function

Multi-label classification is a supervised learning paradigm in which multiple labels may be associated with a single instance, unlike traditional single-label classification, where each instance is assigned only one class[167;168]. A single observation may indicate simultaneous attacks on multiple sensors. Given the complexity of highly integrated process systems, handling both single-attack and multi-attack cases is essential for accurate detection and identification. Neural network-based classifiers commonly employ the binary cross-entropy (BCE) loss function for training in multi-label settings[?] . BCE treats each label prediction as an independent binary classification task, allowing the model to concurrently optimize performance across multiple labels. The BCE loss for a given sample is defined as:

$$\mathcal{L}_{\text{BCE}} = -\frac{1}{N}\sum_{i=1}^{N}\left[p_i \log(\hat{p}_i) + (1-p_i)\log(1-\hat{p}_i)\right], \tag{6.6}$$

where p_i denotes the ground-truth value of the i-th label, $\hat{p}_i$ is the corresponding predicted probability, and N is the total number of labels. This formulation enables the model to independently learn label-specific patterns relevant to various attack scenarios.

A key challenge in multi-label classification is class imbalance, particularly when certain attack types or compromised sensors occur less frequently. The standard BCE loss assigns equal weight to all labels, which may lead to biased predictions favoring more prevalent attack patterns. To address this issue, a weighted binary cross-entropy (WBCE) loss is employed, which introduces label-specific weights based on their relative frequencies in the training data[168]. The weighted loss function is expressed as:

$$\mathcal{L}_{\text{WBCE}} = -\frac{1}{N}\sum_{i=1}^{N} w_i \left[p_i \log(\hat{p}_i) + (1 - p_i)\log(1 - \hat{p}_i)\right], \tag{6.7}$$

where w_i is the weight assigned to the i-th label, typically calculated as the inverse of its frequency in the dataset. This reweighting mitigates the influence of dominant classes and enhances the model's ability to detect rare but critical attack patterns.

6.4 Time-series classification frameworks for attack detection and identification

This section presents the time-series classification frameworks developed for cyberattack detection and compromised-sensor identification under closed-loop MHE-MPC operation. Two centralized architectures are first introduced, in which a single model monitors the entire process network. an end-to-end transformer that performs both tasks jointly, and a two-tier architecture that decouples them. Their distributed counterparts, built on the community decomposition later discussed, are then formulated to address scalability in highly integrated process systems.

6.4.1 Centralized end-to-end (ETE) transformer

Self-attention transformers have demonstrated strong capabilities in capturing temporal patterns in time series data and have been widely used in process modeling, control, and forecasting[169–171], as also demonstrated in the preceding chapters of this dissertation. This chapter repurposes transformers from their conventional use in regression to a multi-label classification setting, enabling joint detection and identification of cyberattacks within a unified framework (Figure 6.1).

Figure 6.1: *ETE centralized transformer for cyberattack detection and sensor identification.*

The transformer model operates on sensor data sequences. Training and inference use a time series windowing strategy in which the model receives a fixed-length sequence of historical sensor measurements; this sliding window of past observations incorporates temporal memory into the model input, allowing the network to account for short-term process dynamics rather than relying on instantaneous measurements. This sequential learning paradigm allows the model to incorporate historical context and system dynamics when identifying anomalies, improving robustness to transient disturbances and enhancing the ability to detect attacks that emerge gradually or intermittently. Transformers are preferred over recurrent architectures, such as RNNs and LSTMs, because of their non-recurrent structure and improved training efficiency. They process the input sequence in parallel, avoiding sequential dependencies during training and reducing computational overhead[172;173]. In addition, the self-attention mechanism enables the model to capture long-range temporal dependencies and complex cross-sensor interactions, making it particularly effective in distinguishing subtle attack signatures from nominal process variations and in identifying coordinated attacks in highly integrated process systems.

The ETE centralized decoder-only transformer integrates detection and identification tasks into a single architecture. Given a sequence of multivariate sensor measurements, the model simultaneously determines whether an attack has occurred and identifies the affected sensors. One key advantage of this approach is its reduced inference time, bypassing the need for separate detection and classification stages, thereby facilitating real-time implementation. However, the joint formulation also introduces potential limitations. Misclassification in the detection stage may propagate to the identification task, increasing the likelihood of compounded errors. Specifically, false positives and negatives are not decoupled, leading to reduced accuracy under ambiguous or noisy conditions.

6.4.2 Centralized two-tier framework

While the ETE centralized transformer model enables fast inference by jointly performing attack detection and sensor identification, it becomes less scalable as system complexity grows. Errors in the ETE setup directly impact both tasks, reducing overall robustness and complicating retraining or fine-tuning as new data is introduced. Moreover, performing both tasks on all samples simultaneously increases computational load and limits adaptability to new attack patterns. To address these limitations, a two-tier detection and identification framework is developed, in which detection and identification are decoupled and executed sequentially. As shown in Figure 6.2, the first tier employs a FNN for binary classification to determine the presence of an attack. Upon detection, the second tier is activated, comprising a transformer network designed for multi-label classification to identify which specific sensors are compromised. This hierarchical approach isolates detection errors, preventing them from directly affecting sensor identification.

Unlike the ETE model, where all samples undergo complete processing regardless of attack presence, the two-tier framework activates the computationally intensive transformer only when an attack is detected. This significantly reduces training and inference costs, particularly in large-scale systems. Furthermore, the transformer is trained solely on attack data, enabling it to focus on attack-specific patterns without interference from normal pro-

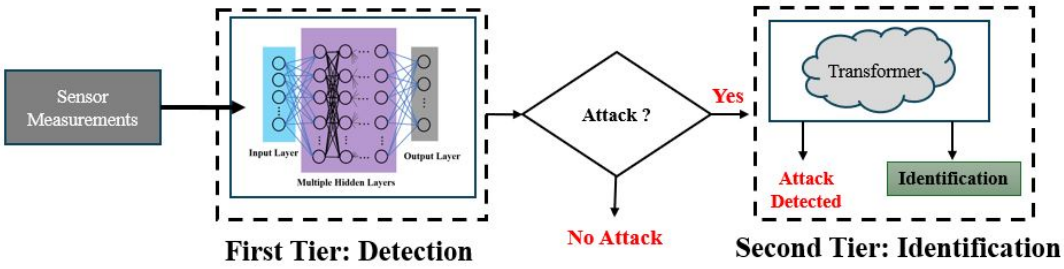

Figure 6.2: *Two-tier framework schematics.*

cess variations. The FNN processes sequential sensor data and is trained using BCE loss. When an attack is detected, the transformer is invoked and trained using weighted BCE loss to correct for class imbalance and prevent bias toward frequently compromised sensors. By separating detection and identification, the framework enhances both accuracy and robustness while reducing computational demands. The FNN ensures rapid detection, and the transformer specializes in targeted sensor identification, thereby improving adaptability and resilience in highly integrated process systems.

6.4.3 Community-based distributed frameworks: DETE and D2Tier

Although the centralized frameworks monitor the process network as a single entity, their computational and learning burden grows with system dimension and with the combinatorial structure of the multi-label identification task. To address these scalability limitations, distributed counterparts of both centralized architectures are formulated on top of graph-based community decomposition, in which the integrated process network is partitioned into strongly coupled subsystems and each community is monitored by its own local model.

Distributed Weighted Graph-based Community Detection

To establish a distributed framework, the highly integrated process system is first represented as a weighted interaction network derived from the underlying process model. In this network, nodes represent process variables, including states, inputs, and outputs, while

weighted edges quantify the strength of dynamic interactions among them. The decomposition of the process network into local communities is guided by a modularity-based criterion that assesses whether interactions within each community are stronger than those across communities. The modularity index is defined as:

$$Q = \frac{1}{m} \sum_{i,j} \left(A_{ij} - \frac{d_i d_j}{m} \right) \delta_{s_i, s_j}, \tag{6.8}$$

where A_{ij} denotes the weighted adjacency between nodes i and j, d_i is the weighted degree of node i, m is the total edge weight of the network, and δ_{s_i,s_j} equals one if nodes i and j are assigned to the same community and zero otherwise. Larger modularity values correspond to decompositions with dense intra-community interactions and comparatively weaker inter-community interactions.

Rather than evaluating modularity via a computationally expensive combinatorial search, a spectral formulation is used to obtain an efficient, scalable network decomposition. Each node i is mapped to an energy-based spectral vector r_i, constructed from the leading eigenvectors of the modularity matrix. These spectral vectors characterize the structural role of each node within the weighted interaction network. For a given community s, the aggregate community representation is defined as:

$$R_s = \sum_{i \in s} r_i, \tag{6.9}$$

which captures the collective interaction pattern of the nodes assigned to that community. Based on this representation, the modularity index can be equivalently written as:

$$Q = \frac{1}{m} \sum_{s=1}^{L} \|R_s\|^2 = \frac{1}{m} \sum_{s=1}^{L} \sum_{i \in s} r_i^\top R_s, \tag{6.10}$$

where L denotes the total number of communities. This compact formulation shows that modularity increases when the spectral vectors of individual nodes are strongly aligned with the aggregate vector of their assigned community.

Community assignments are updated by moving each node to the community whose aggregate vector is most closely aligned with the node's corresponding vector. The modularity change associated with moving node i from community B to community A is given by

$$\begin{aligned}\Delta Q &= \frac{1}{m}\left[(R_A + r_i)^2 + (R_B - r_i)^2 - \|R_A\|^2 - \|R_B\|^2\right] \\ &= \frac{1}{m} r_i^\top (R_A - R_B),\end{aligned} \tag{6.11}$$

which provides a simple and interpretable criterion for modularity maximization. Iteratively applying this update yields a decomposition that preserves strongly coupled groups of variables while suppressing unnecessary cross-community connections.

After community identification, the original process model is reorganized into L interacting subsystems, each corresponding to a detected community. The resulting distributed state-space representation is expressed as:

$$\begin{aligned}\dot{x}_{(i)}(t) &= f_{(i)}\big(x(t), u_{(i)}(t)\big) + \omega_{(i)}(t), \quad i = 1, \ldots, L, \\ y_{(i)}(t) &= g_{(i)}\big(x(t), u_{(i)}(t)\big),\end{aligned} \tag{6.12}$$

where the global state, input, disturbance, and output vectors are defined as $x = [x_{(1)}^\top \; x_{(2)}^\top \; \ldots \; x_{(L)}^\top]^\top$, $u = [u_{(1)}^\top \; u_{(2)}^\top \; \ldots \; u_{(L)}^\top]^\top$, $\omega = [\omega_{(1)}^\top \; \omega_{(2)}^\top \; \ldots \; \omega_{(L)}^\top]^\top$, and $y = [y_{(1)}^\top \; y_{(2)}^\top \; \ldots \; y_{(L)}^\top]^\top$, respectively. The corresponding dimensions satisfy $n = \sum_{i=1}^{L} n_i$, $m = \sum_{i=1}^{L} m_i$, and $r = \sum_{i=1}^{L} r_i$. This modularity-driven decomposition yields physically meaningful and computationally efficient subsystems, providing a natural foundation for distributed monitoring, attack detection, and attack identification in large-scale process networks[? ?] .

Distributed end-to-end (DETE) transformer framework

Building on the community decomposition, a DETE transformer framework is developed to simultaneously detect attacks and identify compromised sensors. As shown in Figure 6.3, each detected community is assigned an independent transformer model that operates only on the local sensor measurements for that community. For each community, the learning

task is formulated as a multi-label classification problem, in which the output vector jointly encodes the attack-occurrence indicator and the identities of the compromised sensors in that community. This formulation enables a single model to perform both detection and identification without separate post-processing or sequential decision-making stages.

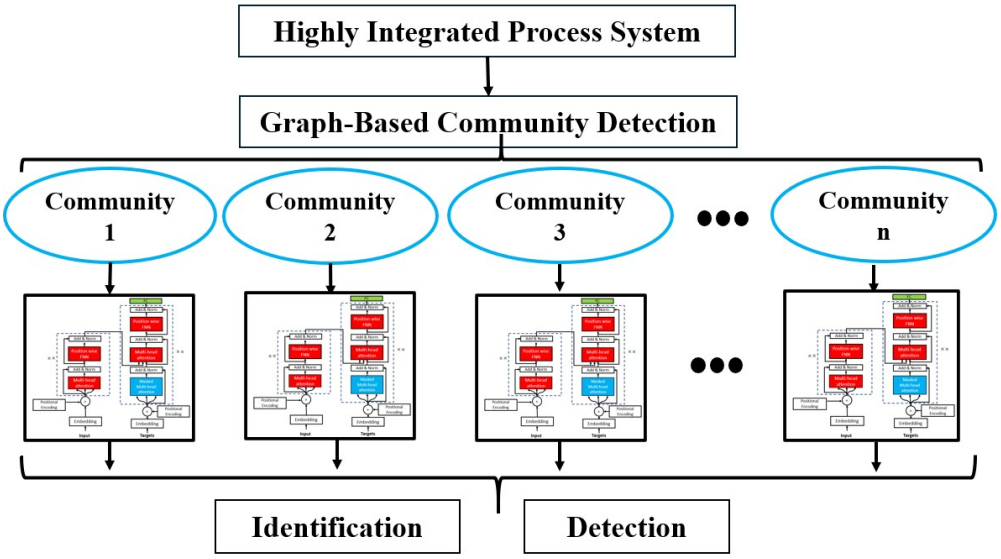

Figure 6.3: *DETE transformer framework for community-level attack detection and identification in highly integrated process systems.*

Formally, for the i-th community, let $\mathbf{X}_{(i)}^{k} = \left[\mathbf{y}_{(i)}(k-T+1), \ldots, \mathbf{y}_{(i)}(k)\right]$ denote the input sequence constructed from the most recent T samples of local measurements for that community. Here, $\mathbf{y}_{(i)}(k) \in \mathbb{R}^{r_i}$ represents the vector of measured sensor outputs for community i at discrete time step k, where r_i is the number of sensors monitored in that community. The window length T defines the model's temporal memory, consistent with the windowing strategy of the centralized frameworks[165], embedding short-term process dynamics into the model input so that the learning framework can exploit the temporal correlations that distinguish normal operation from adversarial behavior.

Each community is equipped with an independent transformer model, $\mathcal{T}_{(i)}(\cdot)$, which maps the local time series input to an attack-status output, $\hat{\mathbf{z}}_{(i)}^{k} = \mathcal{T}_{(i)}\left(\mathbf{X}_{(i)}^{k}\right)$. For communities with multiple monitored sensors, i.e., $r_i > 1$, the model produces a multi-label output $\hat{\mathbf{z}}_{(i)}^{k} \in \{0,1\}^{r_i+1}$, where the first r_i components correspond to sensor-level attack indicators. A value of 1 in these components indicates that the corresponding sensor measurement is

compromised. The final component, $\hat{z}^k_{(i),\text{attack}}$, represents the community-level attack detection decision and indicates whether any attack is inferred within that community. This joint formulation enables simultaneous attack detection and sensor identification in a single inference step. For communities where only a single state variable is monitored, i.e., $r_i = 1$, detection and identification coincide: the output space reduces to a binary decision, and the corresponding transformer model is trained solely for attack detection, since the identity of the compromised sensor is known by construction once an attack is detected.

At the system level, the distributed outputs from all communities are aggregated to obtain a global attack assessment. Let $\hat{\mathbf{Z}}^k = \left[\hat{\mathbf{z}}^{k\top}_{(1)}, \hat{\mathbf{z}}^{k\top}_{(2)}, \ldots, \hat{\mathbf{z}}^{k\top}_{(L)}\right]^\top$ denote the concatenated output vector across all L communities, where L is the total number of detected communities in the process network. A network-wide attack indicator is then computed by aggregating the community-level detection decisions $\hat{a}^k = \bigvee_{i=1}^{L} \hat{z}^k_{(i),\text{attack}}$, where $\vee$ denotes the logical OR operator. This aggregation ensures that an attack detected in any community triggers a global alert while preserving the distributed structure of the inference process. Because the community-level transformer models operate independently and in parallel, the resulting architecture supports scalable distributed inference, with computational complexity increasing approximately linearly with the number of communities.

Distributed two-tier (D2Tier) framework

The distributed two-tier (D2Tier) framework extends the centralized two-tier architecture of Section 6.4.2 to a fully distributed implementation by integrating the same graph-based community decomposition. Unlike the DETE framework, which jointly performs detection and identification in a single model, the two-tier architecture explicitly separates these tasks, decoupling binary attack detection from sensor-level localization, which is inherently a multi-label classification problem. In addition to the complexity reduction discussed for the centralized case, this separation avoids logically inconsistent inference outcomes.

As illustrated in Figure 6.4, each detected community is assigned an independent two-stage learning pipeline. In the first tier, an FNN is trained to detect community-level attacks

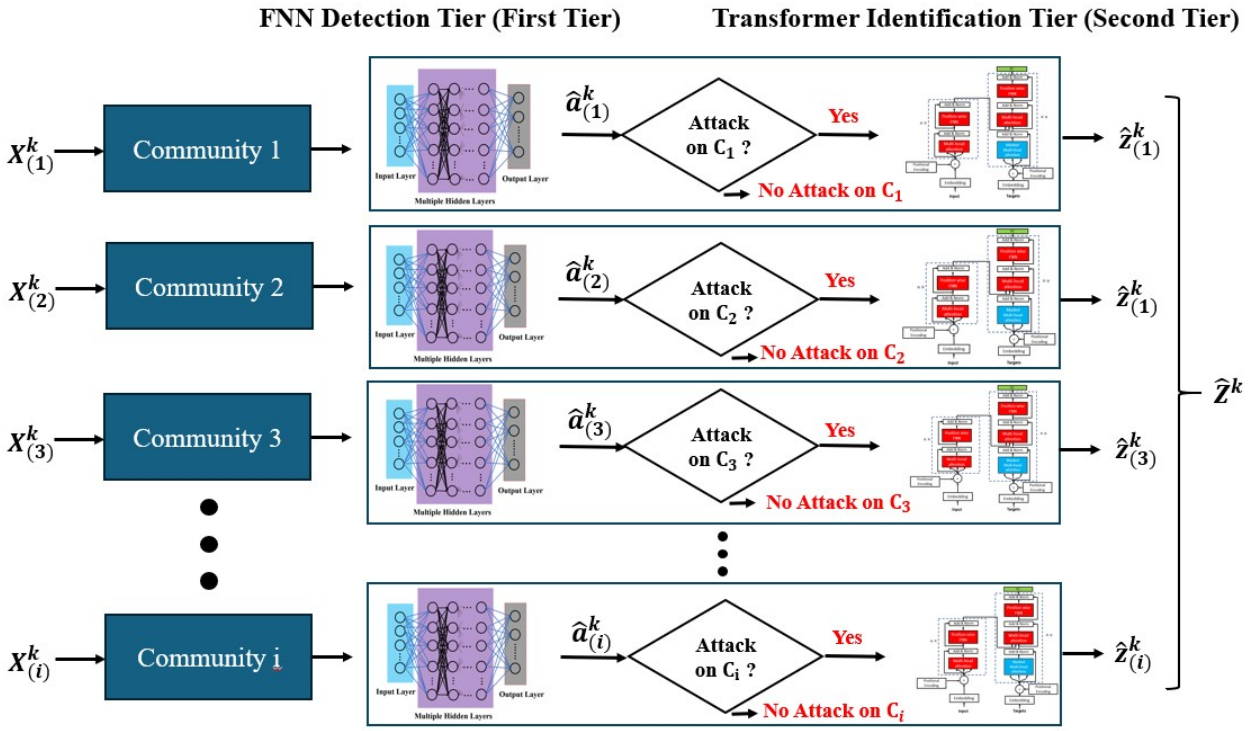

Figure 6.4: *D2Tier framework for community-level attack detection and identification in highly integrated process systems.*

using only local measurements. This detector operates independently within each community and determines whether any sensor in that community is under attack, without localizing the compromised measurement. Once an attack is detected, the second tier is activated, consisting of a transformer-based model trained to identify the compromised sensor(s) within the corresponding community. Consistent with the centralized two-tier training strategy, the second-tier transformer is trained exclusively on samples from attack scenarios, with nominal operating conditions intentionally excluded, which reduces the amount of data required for sensor-level identification and lowers the computational cost of training high-capacity transformer models at the community level.

During inference, the two tiers operate sequentially and conditionally. For the i-th community, the same time series input representation $\mathbf{X}_{(i)}^k$ as in the DETE framework is used. The first-tier detector computes a community-level binary decision $\hat{a}_{(i)}^k = \mathcal{D}_{(i)}(\mathbf{X}_{(i)}^k)$, indicating whether an attack is absent or present in community i at time k, where $\mathcal{D}_{(i)}(\cdot)$ denotes the community-specific FNN detector. The second-tier transformer is activated only

when an attack is detected. Accordingly, the sensor-level identification output is defined conditionally:

$$\hat{\mathbf{z}}_{(i)}^{k} = \begin{cases} \mathbf{0}, & \hat{a}_{(i)}^{k} = 0 \\ \mathcal{T}_{(i)}\big(\mathbf{X}_{(i)}^{k}\big), & \hat{a}_{(i)}^{k} = 1 \end{cases} \tag{6.13}$$

where $\mathcal{T}_{(i)}(\cdot)$ is the transformer-based identification model and $\hat{\mathbf{z}}_{(i)}^{k} \in \{0,1\}^{r_i}$ denotes the predicted sensor-level attack labels for community i. This conditional formulation enforces logical consistency between detection and identification by preventing scenarios in which an attack is detected without identifying any compromised sensor, or in which a sensor is flagged as compromised without a corresponding attack detection. An exception occurs for communities with only one monitored sensor, where, as in the DETE framework, detection and identification are inseparable; for these communities, the first-tier FNN is bypassed, and the transformer model directly generates a decision output that simultaneously represents both attack detection and sensor identification.

At the network level, distributed outputs from all communities are aggregated to provide a global assessment, consistent with the DETE framework: the network-wide attack indicator is computed as $\hat{a}^k = \bigvee_{i=1}^{L} \hat{a}_{(i)}^{k}$, and the community-level identification outputs are concatenated to construct a network-wide attack localization map. Because all communities operate independently and concurrently, the resulting architecture preserves the scalability and computational efficiency of the DETE framework while extending the two-tier strategy to a distributed setting suitable for high-dimensional, highly integrated process systems.

6.5 Post-attack state reconstruction using extended Kalman filters

Following attack detection and sensor identification, it is essential to reconstruct compromised measurements to preserve closed-loop stability. An extended Kalman filter (EKF) is employed for state reconstruction, a widely adopted technique for nonlinear systems subject

to sensor compromise[160;174]. EKF extends the classical Kalman filter to nonlinear systems and is governed by the following process and measurement models:

$$\begin{aligned} x_{k+1} &= \tilde{f}(x_k, u_k) + w_k, \quad w_k \sim \mathcal{N}(0, Q_k), \\ \hat{y}_k &= h(x_k) + v_k, \quad v_k \sim \mathcal{N}(0, R_k), \end{aligned} \tag{6.14}$$

where $\tilde{f}$ denotes the discrete-time nonlinear model function, x_k is the system state, $\hat{y}_k$ the sensor measurement, and u_k the control input. The process and measurement noise terms, w_k and v_k, are modeled as zero-mean Gaussian variables with covariances Q_k and R_k, respectively. The EKF operates in two sequential steps: prediction and update.

Prediction:

$$\begin{aligned} \hat{x}_{k+1|k} &= \tilde{f}(\hat{x}_{k|k}, u_k), \\ P_{k+1|k} &= F_k P_{k|k} F_k^T + Q_k, \quad F_k = \frac{\partial \tilde{f}}{\partial x}\Big|_{\hat{x}_{k|k}}. \end{aligned} \tag{6.15}$$

Update:

$$\begin{aligned} \tilde{y}_k &= y_k - h(\hat{x}_{k|k-1}), \\ K_k &= P_{k|k-1} H_k^T (H_k P_{k|k-1} H_k^T + R_k)^{-1}, \quad H_k = \frac{\partial h}{\partial x}\Big|_{\hat{x}_{k|k-1}}, \\ \hat{x}_{k|k} &= \hat{x}_{k|k-1} + K_k \tilde{y}_k, \\ P_{k|k} &= (I - K_k H_k) P_{k|k-1}. \end{aligned} \tag{6.16}$$

Here, F_k and H_k are the Jacobians of the system and measurement models, respectively, and K_k is the Kalman gain that weights the influence of new observations.

In the centralized frameworks, the EKF operates on the full process model. When a sensor is identified as compromised, its measurements are excluded from the measurement update, and the EKF estimates the true states based on the process model, control inputs, and measurements from uncompromised sensors. These estimates are continuously refined through new observations, and the reconstructed values are substituted for the corrupted measurements before being passed to the MHE-MPC module. This selective reconstruction enables the system to maintain stable performance despite adversarial interference.

In the distributed frameworks, the same reconstruction principle is implemented at the

community level. An independent EKF-based reconstruction agent is assigned to each detected community, and reconstruction is activated only for communities identified as under attack by either the DETE or the D2Tier architecture. This localized activation confines reconstruction to the affected portion of the process network, while the measured outputs from communities operating under nominal conditions remain unchanged. For the i-th community, the EKF uses the local nonlinear process model and measurement equations:

$$\begin{aligned} x_{(i)}(k+1) &= \tilde{f}_{(i)}\big(x(k), u_{(i)}(k)\big) + w_{(i)}(k), \\ \hat{y}_{(i)}(k) &= h_{(i)}\big(x(k), u_{(i)}(k)\big) + v_{(i)}(k), \end{aligned} \tag{6.17}$$

where $u_{(i)}(k)$ denotes the local control input vector, and $w_{(i)}(k)$ and $v_{(i)}(k)$ represent zero-mean Gaussian process and measurement noise with covariance matrices $Q_{(i)}$ and $R_{(i)}$, respectively. The prediction and update steps follow the formulation in Equations (6.15) and (6.16), applied to the local model. Compromised measurements identified within community i are excluded from the local measurement update, and the corrupted signals are reconstructed as $\tilde{y}_{(i)}(k) = h_{(i)}\big(\hat{x}(k|k)\big)$ before being passed to the MHE-MPC module. By integrating EKF-based reconstruction at the community level, the distributed frameworks avoid unnecessary global state estimation and limit the propagation of corrupted information throughout the process network.

6.6 Case Study: Integrated Process of Benzene Alkylation with Ethylene

This section applies the proposed frameworks to detect, identify, and reconstruct cyberattacks in a representative integrated chemical process, benzene alkylation with ethylene. This reaction is a cornerstone of the petrochemical industry, primarily used to produce ethylbenzene, a key intermediate in the synthesis of styrene. The process involves the catalytic alkylation of benzene with ethylene in the presence of an acid catalyst, conducted in either

liquid- or gas-phase reactors[162]. The case study considered here consists of an ethylene-based benzene alkylation process comprising four continuous stirred-tank reactors (CSTRs) followed by a flash separator. The system operates under elevated pressure to ensure liquid-phase conditions with nearly constant molar volumes. The process involves three principal reactions:

$$\text{Benzene} + \text{Ethylene} \rightarrow \text{Ethylbenzene} \qquad \text{(Reaction 1)}$$

$$\text{Ethylbenzene} + \text{Ethylene} \rightarrow \text{Diethylbenzene} \qquad \text{(Reaction 2)}$$

$$\text{Benzene} + \text{Diethylbenzene} \rightarrow \text{Ethylbenzene} \qquad \text{(Reaction 3)}$$

In the first reaction, benzene (A) reacts with ethylene (B) to form ethylbenzene (C), the primary and desired product. In the second reaction, additional ethylene (B) reacts with the previously formed ethylbenzene (C) to produce diethylbenzene (D), an undesired byproduct. While diethylbenzene is less valuable and is typically removed in downstream separation units, its formation is thermodynamically favored under excess ethylene and high-conversion conditions. The third reaction is a transalkylation step, in which diethylbenzene (D) reacts with benzene (A) to regenerate two moles of ethylbenzene (C). This reaction enhances product selectivity and reduces waste by converting a low-value byproduct back into the main product.

The configuration of the reactors, feed injection points, recycle streams, and the flash separator is illustrated in Figure 6.8. The process initiates with the co-feeding of pure benzene (A) and ethylene (B) into the first reactor to carry out Reaction 1. Additional ethylene (B) is introduced into the second and third reactors to enhance conversion through further alkylation. The effluent from the third reactor is routed to a flash separator, which recovers unreacted benzene (A) for recycling to the first and fourth reactors. Ethylbenzene (C), the target product, is withdrawn from the bottom stream of the separator. The fourth reactor is dedicated to the transalkylation reaction (Reaction 3). In this unit, controlled quantities of diethylbenzene (D), sourced from an external separation unit, are combined with recycled

benzene to improve product selectivity by converting the byproduct into the desired product. Each reactor and the separator are equipped with a jacketed heat exchanger. Due to the strong thermal and material coupling, the process is managed via multiple control agents to maintain optimal conversion and selectivity[163;164]. A first-principles dynamic model is used to simulate the process. This model comprises a set of interactive ordinary differential equations (ODEs) representing reaction kinetics, mass and energy balances, and phase behavior. The system is characterized by 25 state variables, 17 manipulated inputs, and 15 measured outputs. A detailed formulation of the process model, including the complete set of parameters, is presented by Ebrahimi and Pourkargar[163].

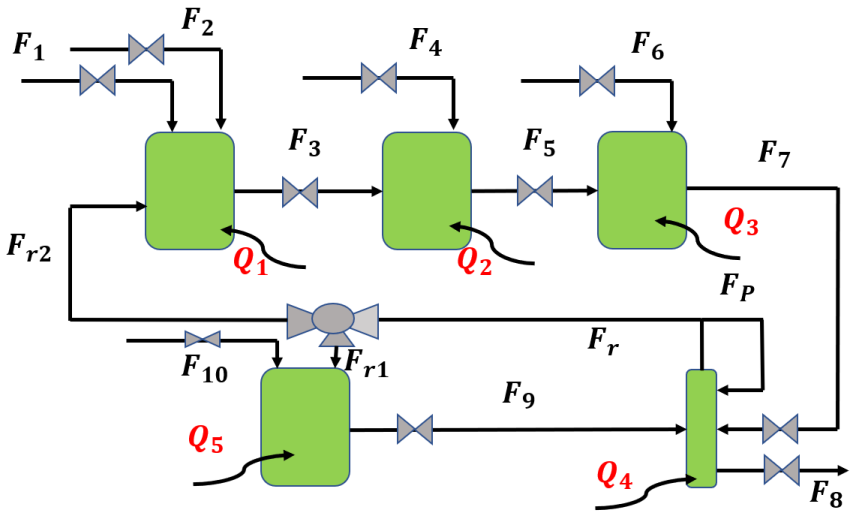

Figure 6.5: *Benzene alkylation process schematic.*

6.6.1 Data generation and preprocessing

Effective training of the proposed frameworks requires a large-scale, well-balanced dataset derived from closed-loop simulations of the process model. Because classification performance is highly sensitive to data imbalance, especially in rare-attack scenarios, dataset biases can impair the model's ability to generalize. Therefore, data generation should include both normal and compromised operating conditions across various attack types. To address the large data requirements of transformer architectures, simulations were performed under 200 ran-

domized initial conditions. Each simulation corresponds to a closed-loop system integrating MHE and MPC, implemented using the *do-mpc* framework[175] (Figure). At each time step, the MPC computes optimal control inputs, which are applied to the process, while MHE uses temperature sensor measurements to estimate concentration states. Under normal operation, this setup maintains system stability. However, when cyberattacks compromise sensor measurements, the resulting estimation errors can destabilize the system. Even a single compromised sensor may trigger cascading failures, underscoring the importance of accurate detection and localization.

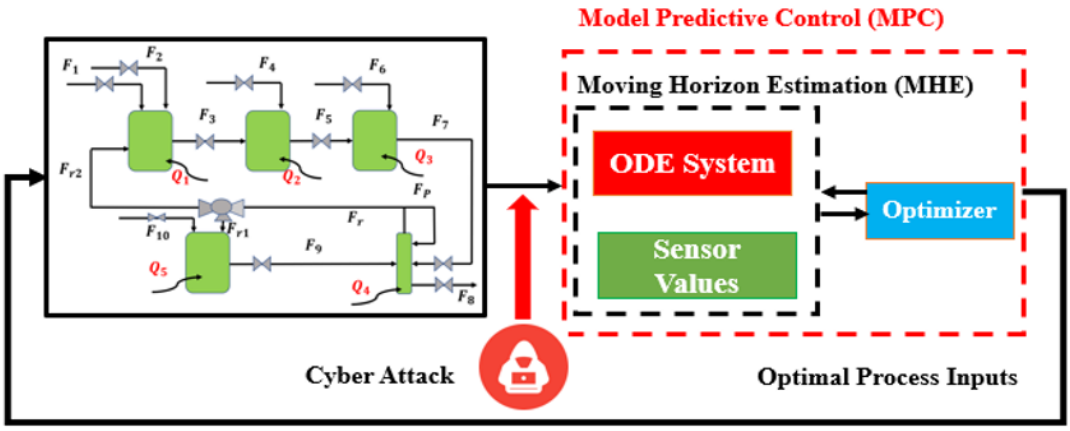

Figure 6.6: *MPC-MHE loop under cyber attack*

The dataset includes 32 distinct attack scenarios involving single and multi-sensor compromises. Simulations were conducted under 120 randomized initial conditions for each scenario, including both normal and attacked cases (Table 6.1), resulting in approximately 5×10^6 data points. Each trajectory spans 36 time steps with a sampling interval of 60 units.

Table 6.1: *Attack configurations used for data generation.*

Attack Type	η	δ	Γ
Min-Max	-	-	-
Surge	0.1	-	-
Noise	-	0.35	-
Set Difference	-	-	0.095

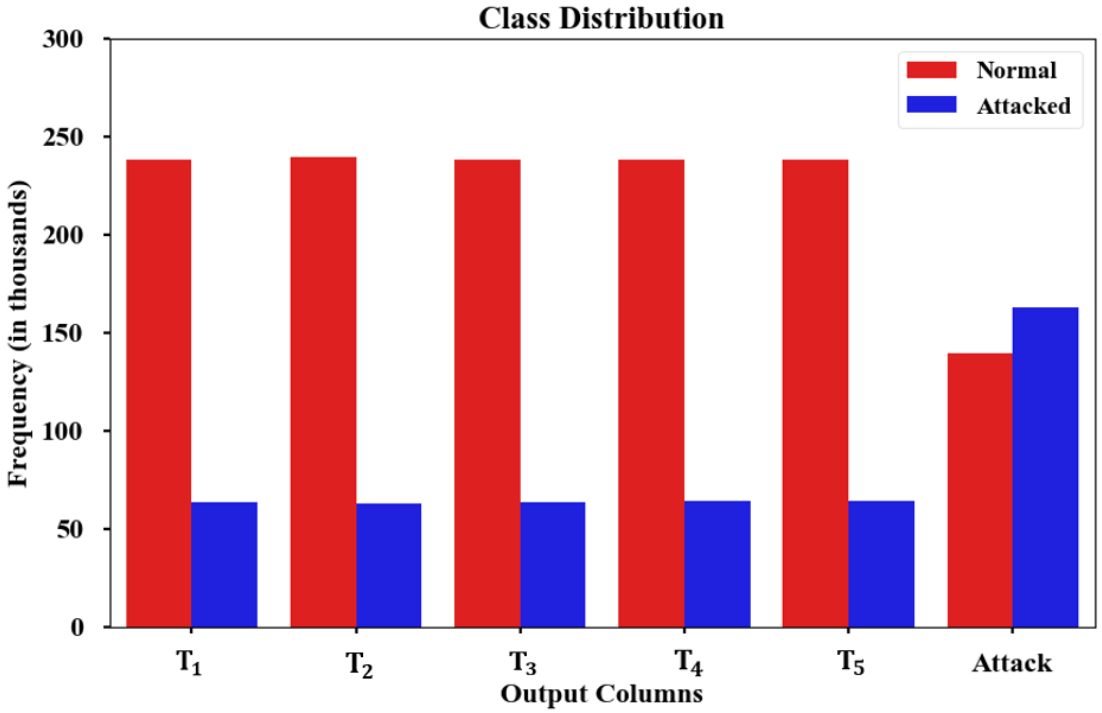

Figure 6.7: *Dataset distribution across different labels.*

While normal and attack conditions are generally balanced, an inherent imbalance exists among sensor-specific labels because multi-sensor attack scenarios are naturally less frequent. The number of possible attack combinations is given by the binomial coefficient:

$$S_k = \binom{n}{k} = \frac{n!}{k!(n-k)!}. \tag{6.18}$$

For five sensors ($n = 5$), this yields:

$$S_1 = \binom{5}{1} = 5, \quad S_2 = \binom{5}{2} = 10, \quad S_3 = \binom{5}{3} = 10, \quad S_4 = \binom{5}{4} = 5, \quad S_5 = \binom{5}{5} = 1. \tag{6.19}$$

As this distribution shows, double- and triple-sensor attacks are most frequent, whereas full-sensor compromises are rare. Figure 6.7 depicts the distribution of normal and attacked samples for each temperature sensor (Sensor T_1–T_5), as well as for the aggregated binary label ("Any Sensor Attacked"). The red bars denote normal operation, whereas the blue bars represent attack instances. The binary label indicates if any of the five sensors experienced

an attack within a given sequence. The figure demonstrates the presence of class imbalance in the dataset.

Because this imbalance is structurally inherent, standard balancing techniques such as oversampling or undersampling[176] are avoided, as they would introduce artificial distributions and reduce model robustness. Instead, the model is trained directly on the naturally imbalanced dataset to accurately capture realistic operational conditions. A weighted BCE loss is employed to adequately emphasize minority attack scenarios during learning without artificially modifying their natural occurrence frequency. Finally, one-hot encoding is applied for multi-label classification, with each sensor assigned a binary indicator, 1 for compromised and 0 for normal. The dataset is partitioned into training (70%), validation (10%), and test (20%) sets, preserving the proportion of normal and compromised conditions across all subsets.

6.6.2 Model implementation and training

The four proposed frameworks are trained and evaluated on the closed-loop dataset described in the previous section. all models share a common training configuration. Stochastic gradient descent (SGD) is used as the optimizer[177], early stopping based on validation loss is applied to prevent overfitting[178], and model checkpointing is used to retain the parameters corresponding to the minimum validation loss. To mitigate the inherent class imbalance in the attack labels, the multi-label identification models employ a weighted BCE loss in which the weight for each class i is computed as:

$$w_i = \frac{1}{\sqrt{p_i + \epsilon}}, \tag{6.20}$$

where p_i is the proportion of positive samples for class i, and ϵ is a small constant added for numerical stability. The square root operation mitigates extreme weighting, promoting stable gradient updates across both frequent and infrequent classes.

Centralized frameworks

For the centralized ETE framework, a decoder-only transformer performs multi-label classification over the full sensor set, learning the mapping:

$$\hat{\mathbf{y}} = f_{\text{ETE}}(X) \in \{0,1\}^{d+1}, \tag{6.21}$$

where $X \in \mathbb{R}^{T\times d}$ denotes the input sequence of temperature measurements over T time steps from d sensors, and $\hat{\mathbf{y}} \in \{0,1\}^{d+1}$ represents the predicted binary labels. The first d elements indicate which sensors are compromised, while the final element denotes whether any cyberattack has occurred. The model architecture comprises three transformer layers, each with four self-attention heads, a model dimension of 128, and a feedforward network dimension of 256 to capture complex temporal dependencies. The output layer utilizes a sigmoid activation function to generate independent probabilities for each label, consistent with the multi-label classification objective. Training is performed with a batch size of 256, and an adaptive learning rate scheduler adjusts the learning rate based on validation loss trends to support convergence and generalization.

In the centralized two-tier framework, each tier is implemented with a specialized architecture tailored to its objective. The first tier addresses the binary detection task:

$$\hat{y}_{\text{attack}} = f^{(1)}(X) \in \{0,1\}, \tag{6.22}$$

where the binary output $\hat{y}_{\text{attack}}$ indicates the presence (1) or absence (0) of an attack. The function $f^{(1)}$ is implemented as an FNN with four hidden layers using ReLU activation functions, trained on the full dataset with BCE loss. Upon detection ($\hat{y}_{\text{attack}} = 1$), the second tier localizes the compromised sensors:

$$\hat{\mathbf{y}}_{\text{sensors}} = f^{(2)}(X) \in \{0,1\}^{d}, \tag{6.23}$$

where $f^{(2)}$ is a decoder-only transformer trained exclusively on samples corresponding to

attack scenarios, using the weighted BCE loss of Equation (6.20). Both tiers are trained with a batch size of 256, and inference follows the conditional structure described in Section 6.4.2, the network-level analogue of Equation (6.13).

Distributed frameworks

Based on the community detection results[163;164], the five temperature sensors are grouped into three communities, as summarized in Table 6.2. This decomposition reflects the modular interaction structure of the benzene alkylation process and forms the basis for the distributed learning architecture. For each community, an independent self-attention-based transformer model is trained on a time series of prior temperature measurements from sensors within that community, using a fixed sequence length of $T = 10$ time steps to learn short-term temporal dynamics relevant to attack detection and identification.

Table 6.2: *Community detection for temperature sensors.*

Community	Assigned Sensors
C_1	T_1
C_2	T_2, T_3
C_3	T_4, T_5

In the DETE framework, each community-level transformer performs joint detection and identification with a multi-label output encoded as binary one-hot representations, in which each sensor within a community is assigned a compromise-status indicator and a community-level detection label. For single-sensor communities, such as C_1, the output reduces to a single identification label, which also serves as the detection decision for that community. Each community-level transformer consists of three encoder layers, four self-attention heads per layer, a model dimension of 64, and a feedforward dimension of 256, trained with the weighted BCE loss of Equation (6.20). Because sequentially training multiple transformer models is computationally demanding, a parallel training strategy is adopted using multi-threading and concurrent programming with early stopping and checkpointing applied per community.

In the D2Tier framework, the first-tier FNN detects community-level binary attacks.

The network comprises four fully connected layers with ReLU activations, and dropout regularization is applied between hidden layers to reduce overfitting[179]. A sigmoid activation function is used in the output layer to estimate the probability of an attack for each community, and the model is trained with BCE loss. The second tier employs a transformer-based architecture similar to the DETE models, configured specifically for multi-label classification of attacked sensors, with sensor-wise sigmoid outputs. Consistent with the centralized two-tier training strategy, nominal no-attack samples are excluded from the second-tier training set, and the identification transformer is activated only when the first tier flags a community as compromised.

6.6.3 Centralized detection and identification performance

The ROC curve in Figure 6.8 illustrates the overall classification performance of the centralized ETE (CETE) transformer framework for both attack detection and sensor identification tasks. The high AUC values across attack scenarios confirm the model's strong discriminative ability. Notably, the model maintains robust performance even as the number of compromised sensors increases, indicating its effectiveness in handling multi-sensor attack conditions. The increasing AUC trend suggests enhanced sensitivity and reliability in detecting and identifying complex cyberattacks involving multiple compromised sensors. The evaluation is further divided into separate analyses of the detection and identification tasks to provide a more granular assessment. Standard classification metrics, accuracy, precision, recall, and F1-score, are computed for each task to offer detailed insights into the framework's effectiveness.

Table 6.3 summarizes the detection performance of the CETE transformer across different attack types. The "Test Set Size" column denotes the number of closed-loop simulations (i.e., time series from MHE/MPC simulations) used for each scenario. The results demonstrate that the model achieves high precision and recall, indicating a balanced trade-off between false positives and false negatives, and robust detection performance across all attack types.

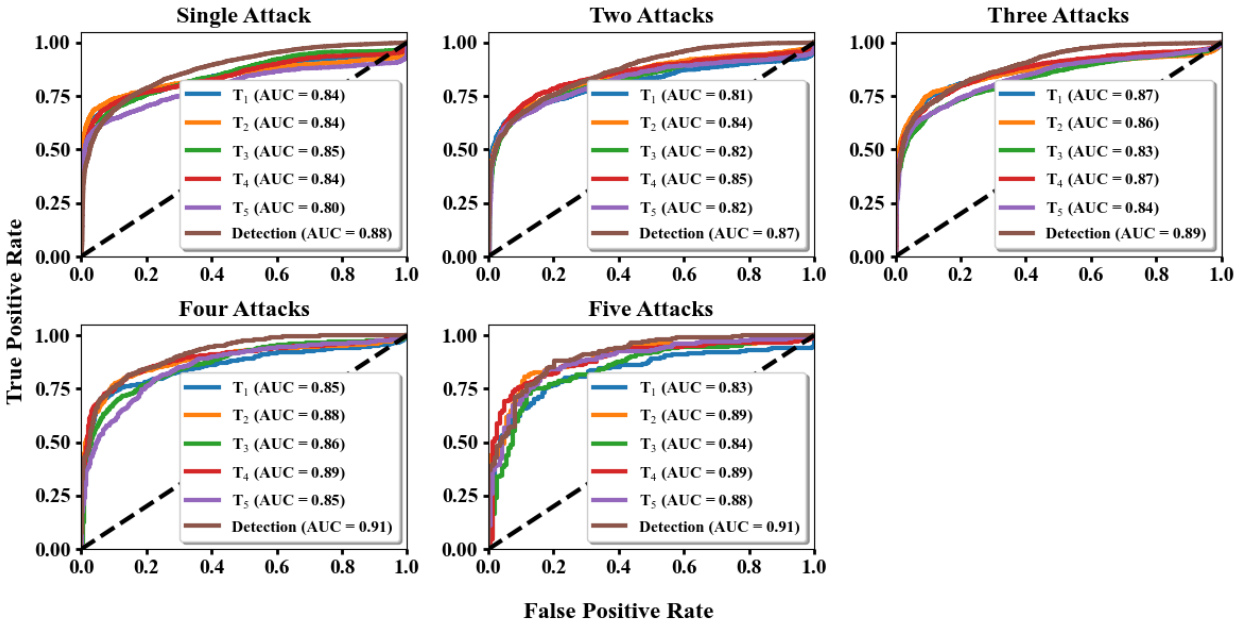

Figure 6.8: *ROC curve and AUC values for the performance of the CETE transformer framework.*

Table 6.3: *CETE transformer detection performance across different attack types.*

Attack Type	Precision	Recall	F1-Score	Accuracy	Test Set Size
Surge	0.8692	0.7903	0.8279	0.8200	203
Min-Max	0.8718	0.6645	0.7542	0.7565	203
Noise	0.8092	0.6521	0.7222	0.7324	203
Set Difference	0.8747	0.8690	0.8719	0.8552	203
Overall	0.8576	0.8942	0.8755	0.8612	812

To further assess the model's detection robustness under varying levels of attack complexity, Table 6.4 presents performance metrics corresponding to different numbers of simultaneously compromised sensors. The results show that the model maintains high precision across all cases, effectively minimizing false alarms. As concurrent attacks increase, recall improves, indicating that the model becomes increasingly sensitive to deviations caused by multiple compromised sensors. This results in higher F1-scores, as more substantial perturbations in sensor data enhance the distinguishability of attack patterns.

Accuracy remains relatively stable, reflecting the model's robustness in handling diverse cyberattack complexities. However, the decreasing test set size with increasing attack counts reveals a natural class imbalance in multi-attack scenarios. Although the model performs

Table 6.4: *CETE transformer detection metrics across multi-attack scenarios.*

Attack Count	Precision	Recall	F1-Score	Accuracy	Test Set Size
1	0.8733	0.8438	0.8583	0.8447	402
2	0.8743	0.8904	0.8823	0.8659	171
3	0.8645	0.9235	0.8930	0.8790	161
4	0.8556	0.9385	0.8951	0.8756	68
5	0.8708	0.8584	0.8640	0.8398	9

reasonably across all cases, further improvements are needed, specifically, enhancing recall for single-attack scenarios and refining precision under multi-attack conditions to ensure reliable detection across varying intensities. As with detection, Table 6.5 summarizes the CETE transformer's overall identification performance across different attack types. Table 6.6 further shows these results by the number of simultaneous sensor attacks, illustrating performance trends relative to attack complexity.

Table 6.5: *CETE transformer identification performance across different attack types.*

Attack Type	Precision	Recall	F1-Score	Accuracy	Jaccard	Test Set Size
Surge	0.8555	0.8082	0.8301	0.7890	0.8372	203
Min-Max	0.8156	0.5619	0.5230	0.7123	0.6542	203
Noise	0.5471	0.6058	0.6234	0.7079	0.6265	203
Set Difference	0.8908	0.9418	0.9155	0.8793	0.8962	203
Overall	0.7607	0.6406	0.6941	0.6791	0.7387	812

Table 6.6: *CETE transformer identification performance across different attack scenarios.*

Attack Count	Precision	Recall	F1-Score	Accuracy	Jaccard	Test Set Size
1	0.6334	0.6401	0.6324	0.7018	0.7255	402
2	0.7354	0.6207	0.6710	0.6530	0.7342	171
3	0.8423	0.6486	0.7328	0.6229	0.7337	161
4	0.8887	0.6370	0.7413	0.6399	0.7502	68
5	0.9270	0.6566	0.7684	0.6152	0.7337	9

The identification results demonstrate the CETE transformer's capability to localize compromised sensors. The Jaccard index, used to quantify similarity between predicted and actual sets of compromised sensors, indicates that the model typically predicts subsets closely aligned with the ground truth. This suggests a reliable identification of the presence of attacks and their specific locations within the sensor network. A performance trend is evident

across varying levels of attack complexity. The model shows improved identification as compromised sensors increase, likely due to more pronounced deviations in the input sequences under multi-sensor attacks. Conversely, single-sensor attacks yield subtler anomalies, making them more difficult to distinguish from background noise and leading to reduced recall. Despite maintaining relatively high precision across all cases, the lower recall in single-attack conditions reflects a conservative model bias, favoring precision over sensitivity. Although overall accuracy slightly declines with higher attack counts, partly due to the reduced representation of these cases, the Jaccard index remains stable. This consistency indicates that even under challenging multi-attack scenarios, the model effectively captures key aspects of the actual attack pattern, reinforcing its robustness in localizing sensor-level intrusions.

The two-tier framework is evaluated using the same performance metrics as the CETE framework. As shown in Figure 6.9, the ROC curve for the two-tier model's performance on a test dataset comprising 840 closed-loop simulations, including both attack and non-attack scenarios across all attack types, demonstrates improved results. The corresponding AUC values and sensor-level ROC curves for detection and identification highlight consistent performance gains over the ETE framework. These improvements can be attributed to the decoupling of binary detection from the inherently imbalanced multi-label identification task. This separation enhances overall performance and improves model interpretability and computational efficiency. The framework facilitates more targeted learning and improved generalization by isolating each subtask and training specialized models. Additionally, the conditional activation of the identification module during inference avoids unnecessary computation in the absence of detected attacks.

Table 6.7 presents the overall detection performance across different attack types, while Table 6.8 details results by the number of simultaneous attacks. The framework maintains high detection accuracy and robustness across both single- and multiple-sensor attack scenarios. Notably, the performance remains stable even as the number of compromised sensors increases, underscoring the advantage of handling detection as a focused binary classification task.

Tables 6.9 and 6.10 summarize the identification performance across various attack types

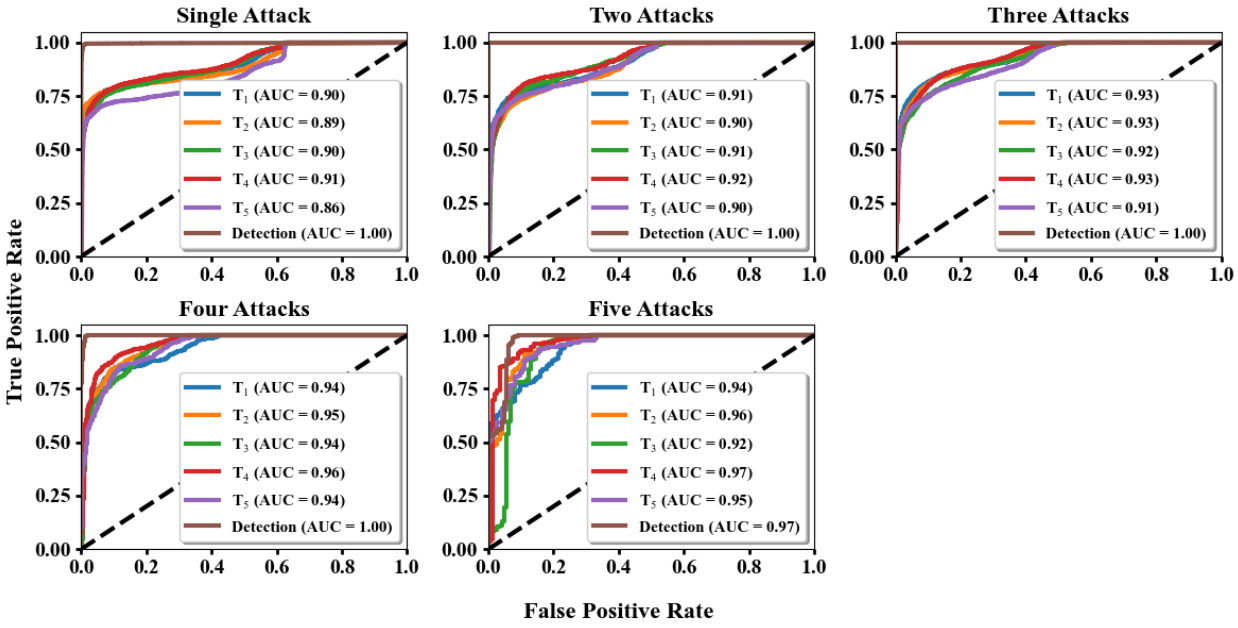

Figure 6.9: *ROC curve and AUC values for the two-tier framework.*

Table 6.7: *Overall detection performance of the two-tier framework across attack types.*

Attack Type	Precision	Recall	F1-Score	Accuracy	Test Set Size
Surge	0.8263	0.9975	0.9039	0.8838	203
Min-Max	0.8326	0.9998	0.9086	0.8869	203
Noise	0.8315	0.9992	0.9077	0.8916	203
Set Difference	0.8411	1.0000	0.9137	0.8929	203
Overall	0.8324	0.9991	0.9082	0.8917	812

and numbers of concurrent attacks. The framework demonstrates strong capabilities in localizing compromised sensors, with consistent improvement as the number of attacked sensors increases. This trend suggests that higher attack severity yields more distinguishable patterns in the data, facilitating accurate identification. Although lower attack counts present more subtle deviations, the model maintains competitive precision and recall. The Jaccard index remains stable, indicating substantial overlap between predicted and actual attacked sensor sets, further confirming the effectiveness of the two-tier framework in handling complex multi-label identification tasks.

A comprehensive comparison of the CETE and two-tier frameworks, based on the results presented in Tables 6.3 and 6.10, highlights the trade-offs between unified and modular architectures. In terms of detection performance, both models demonstrate high accuracy.

Table 6.8: *Detection metrics by number of attacked sensors for the two-tier framework.*

Attack Count	Precision	Recall	F1-Score	Accuracy	Test Set Size
1	0.8327	0.9984	0.9080	0.8873	402
2	0.8283	0.9997	0.9060	0.8924	171
3	0.8466	0.9900	0.9169	0.8943	161
4	0.8163	0.9900	0.8989	0.8802	68
5	0.8154	0.9900	0.8983	0.8652	9

Table 6.9: *Overall identification performance of the two-tier framework across attack types.*

Attack Type	Precision	Recall	F1-Score	Accuracy	Jaccard	Test Set Size
Surge	0.8426	0.9081	0.8720	0.8361	0.8866	203
Min-Max	0.8896	0.5410	0.5827	0.5993	0.6746	203
Noise	0.4499	0.6968	0.5400	0.4800	0.6360	203
Set Difference	0.8582	0.9850	0.9167	0.8563	0.8976	203
Overall	0.7164	0.7614	0.7350	0.7016	0.7798	812

However, the two-tier framework exhibits a more favorable balance between precision and recall (Table 6.7), particularly under increasingly complex attack scenarios (Table 6.8). This advantage stems from the first-tier binary classifier, which isolates the detection task and mitigates the effects of class imbalance inherent in multi-label settings.

The two-tier framework again outperforms the CETE approach for sensor identification, especially in cases involving multiple compromised sensors (Table 6.10). This improvement is attributed to the second-tier transformer model, which is trained exclusively on attack-present data, enhancing its capability to accurately localize affected sensors. The resulting gains in F1-score and Jaccard index (Tables 6.5 and 6.6) underscore the benefits of task-specific model training, which supports better generalization and robustness under complex cyberattack conditions.

The two-tier framework does, however, exhibit a characteristic trade-off in identification performance: it achieves higher recall but slightly lower precision compared to the ETE model. This occurs because the identification module is triggered only when an attack is detected and operates under the assumption that at least one sensor is compromised. Consequently, it is more likely to flag additional sensors as attacked, increasing the number of true positives and introducing more false positives. This trade-off reflects a deliberate design

Table 6.10: *Identification metrics by number of attacked sensors for the two-tier framework.*

Attack Count	Precision	Recall	F1-Score	Accuracy	Jaccard	Test Set Size
1	0.5576	0.7580	0.6343	0.7074	0.7514	402
2	0.6922	0.7562	0.7188	0.6892	0.7837	171
3	0.8174	0.7803	0.7977	0.6704	0.7982	161
4	0.9044	0.7498	0.8185	0.6891	0.8182	68
5	0.9413	0.7679	0.8452	0.6629	0.8112	9

choice that favors sensitivity to ensure potentially compromised components are not overlooked once an intrusion is detected. Additional insights into each framework's classification behavior are provided by the confusion matrices included in Appendix A.

6.6.4 Distributed detection and identification performance

Using the same classification metrics, the performance of the proposed distributed frameworks is evaluated on the benzene alkylation process. Both the DETE framework and the D2Tier architecture are assessed using a test set of single- and multi-sensor attack scenarios generated from closed-loop MHE–MPC simulations. The evaluation is conducted from two complementary perspectives. First, community-level performance is analyzed to quantify the effectiveness of localized detection and identification within each decomposed subsystem. This analysis highlights the value of integrating graph-based community detection with the proposed learning frameworks. It shows how modular decomposition affects classification accuracy, false alarm rates, and sensor-level attribution. Second, network-level performance is evaluated by aggregating community-level outputs to assess overall system-wide detection and identification capabilities across the integrated process network. Finally, a network-level comparative analysis is performed against the centralized frameworks evaluated in Section 6.6.3. This comparison provides insight into the trade-offs between centralized and distributed architectures with respect to scalability, robustness, and classification performance in highly integrated chemical process systems. The test set composition and sizes for each attack type and attack cardinality are identical to those of the centralized evaluation reported in Section 6.6.3.

Community-level analysis

Figure 6.10 presents community-level receiver operating characteristic (ROC) curves for the distributed ETE and two-tier architectures. In Community 1, which includes only a single sensor, both approaches use the same transformer-based detector, yielding nearly identical ROC curves and area under the curve (AUC) values. In Communities 2 and 3, however, show clear performance differences. The distributed ETE model consistently achieves higher AUC values and steeper ROC curves than the two-tier architecture. Specifically, the ETE model shows a sharper rise toward high true positive rates at low false positive rates, indicating stronger sensitivity in the low-FPR region. In contrast, the two-tier model shows more gradual ROC transitions, particularly in Community 3, suggesting reduced discriminative capability under the current training configuration.

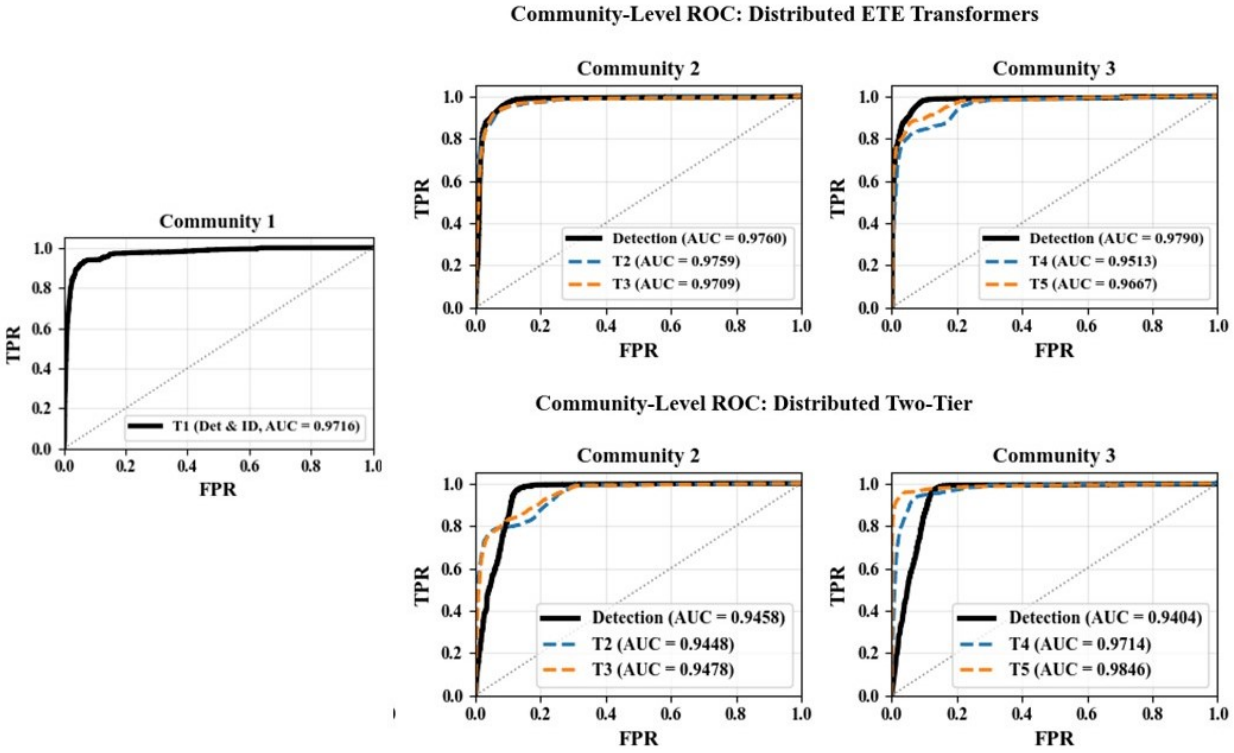

Figure 6.10: *Community-level ROC curves for the distributed ETE and two-tier frameworks.*

Table 6.11 compares community-level detection performance between the distributed two-tier and distributed DETE architectures. For the multi-sensor communities (C2 and C3),

the two-tier architecture achieves very high recall, indicating an aggressive detection strategy that captures nearly all attack instances. However, this increased sensitivity comes at the cost of reduced precision, suggesting greater susceptibility to false alarms. In contrast, the DETE framework offers a more balanced precision–recall trade-off, yielding more consistent overall performance, as reflected in its F1-score. This trend is particularly evident in C3, where DETE improves precision while maintaining competitive recall, thereby enabling more reliable detection decisions. These results indicate that joint learning within the DETE architecture stabilizes community-level detection behavior in multi-sensor subsystems.

Table 6.11: *Community-level comparison of detection performance.*

Community	Method	Accuracy	Precision	Recall	F1-score
C1	Two-tier / DETE	0.9411	0.9023	0.8089	0.8531
C2	Two-tier	0.8795	0.7422	0.9916	0.8490
	DETE	0.9328	0.8889	0.9181	0.9033
C3	Two-tier	0.8854	0.7552	0.9895	0.8566
	DETE	0.9299	0.9368	0.8550	0.8940

Building on the detection results, Table 6.12 presents the corresponding community-level identification performance. In Community 1, with a single sensor and the same transformer-based detector, both methods exhibit identical accuracy and localization behavior. In Community 2, the DETE framework achieves superior overall identification performance, with higher accuracy and a more balanced precision–recall trade-off than the two-tier model. Notably, DETE also yields a higher Jaccard index in this subsystem, indicating stronger agreement between the predicted and true sets of attacked sensors. Because the Jaccard index directly quantifies set-level overlap, this result demonstrates that DETE provides more consistent and reliable sensor localization in C2.

Table 6.12: *Comparison of community-level identification performance across methods.*

Community	Method	Accuracy	Precision	Recall	F1-score	Jaccard
C1	Two-tier / DETE	0.9411	0.9263	0.8927	0.9081	0.7438
C2	Two-tier	0.9063	0.7895	0.7777	0.7816	0.8696
	DETE	0.9383	0.8417	0.8790	0.8595	0.9126
C3	Two-tier	0.9525	0.8995	0.8787	0.8889	0.9307
	DETE	0.9302	0.8872	0.7753	0.8266	0.8977

In Community 3, a different trend emerges. Although the two-tier approach achieves slightly higher accuracy and recall, DETE maintains competitive precision. The Jaccard index remains high for both methods, indicating that despite differences in sensitivity, both architectures achieve substantial overlap with the true sets of sensors that were attacked. These results indicate that localization performance is subsystem-dependent: DETE provides more consistent set-level agreement in C2, whereas in C3, the hierarchical separation of detection and identification in the two-tier architecture appears to improve sensitivity.

Network-level analysis

To gain further insight into the performance of the distributed frameworks from a network-level perspective, test-set evaluations are aggregated across all communities and sensors. The resulting ROC curves, shown in Figure 6.11, clearly distinguish the two architectures. At the detection level, the DETE framework exhibits a stronger ROC curve, with a steeper rise in the low false positive rate region and a higher overall AUC than the D2Tier approach. In contrast, the two-tier model shows a more gradual transition from low to high true positive rates, indicating reduced sensitivity when detection decisions are aggregated across the entire network. At the sensor level, both architectures demonstrate strong separability; however, the DETE framework provides more consistent discrimination across all temperature sensors, whereas the two-tier model exhibits slightly greater variability among sensors.

Moreover, Table 6.13 summarizes the aggregated network-level detection performance. The two-tier model achieves higher recall, indicating a strong ability to identify nearly all attack instances at the network level. However, this increased sensitivity comes with lower precision and reduced overall consistency. In contrast, the DETE framework exhibits more balanced detection behavior, achieving higher accuracy, precision, and F1-score. These results indicate that, when decisions are aggregated across the entire network, joint learning within DETE provides more reliable and stable detection performance, whereas the two-tier strategy prioritizes sensitivity at the expense of false-alarm control.

Following the detection results, Table 6.14 summarizes the aggregated network-level iden-

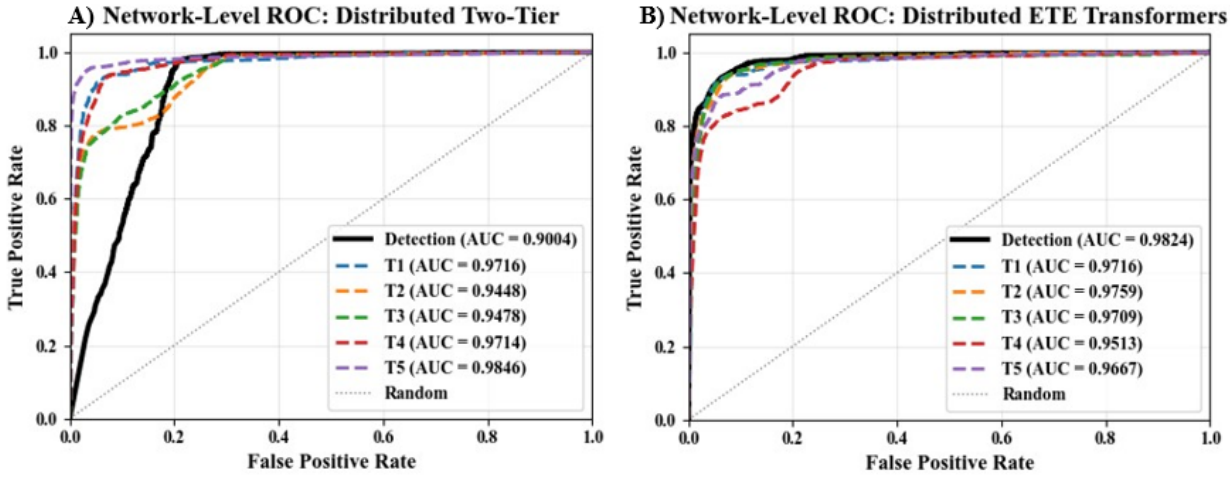

Figure 6.11: *ROC curves at the network level for the DETE and D2Tier frameworks.*

Table 6.13: *Network-level detection performance comparison.*

Method	Accuracy	Precision	Recall	F1-score
DETE	0.9294	0.9523	0.9147	0.9331
D2Tier	0.8913	0.8455	0.9764	0.9063

tification performance. Both architectures exhibit comparable localization capabilities at the network scale, with closely aligned recall values. However, the distributed DETE framework achieves slightly higher accuracy, precision, F1-score, and Jaccard index. The improvement in the Jaccard index is particularly important because it indicates stronger set-level agreement between the predicted and actual combinations of attacked sensors. Overall, these results suggest that although both methods effectively localize attacks when performance is aggregated across communities, the joint learning strategy in DETE enables more consistent and reliable sensor attribution at the network level. The distributed DETE framework demonstrates superior network-level performance, yielding stronger and more consistent results across detection and identification tasks. This finding underscores the value of community-based decomposition for multi-label classification in cyberattack detection and identification. By reducing subsystem dimensionality while preserving joint representation learning, the proposed framework improves prediction stability, supports a more balanced precision-recall trade-off, and enhances set-level consistency in attack localization.

Tables 6.15 and 6.16 offer a more detailed assessment of detection and identification

Table 6.14: *Network-level identification performance comparison.*

Method	Accuracy	Precision	Recall	F1-score	Jaccard
DETE	0.9356	0.8720	0.8235	0.8450	0.8574
D2Tier	0.9318	0.8561	0.8244	0.8388	0.8504

performance by attack type. At the detection stage, the two-tier architecture consistently achieves high recall for min–max and noise attacks, indicating strong sensitivity to subtle perturbations. However, this sensitivity often comes with lower precision, suggesting a higher rate of false positives. In contrast, the distributed DETE framework provides a more balanced precision–recall trade-off, particularly for structured attacks such as surge and set-difference attacks. For these cases, DETE achieves more consistent overall performance because the coherent deviations introduced by structured attacks are effectively captured through joint learning.

Table 6.15: *Network-level detection performance by attack type.*

Attack Type	Method	Accuracy	Precision	Recall	F1-score
Min–max	D2Tier	0.8657	0.8464	0.9313	0.8868
Min–max	DETE	0.8485	0.9941	0.7362	0.8459
Noise	D2Tier	0.8898	0.8366	0.9894	0.9066
Noise	DETE	0.9151	0.8903	0.9615	0.9245
Set-difference	D2Tier	0.8997	0.8481	1.0000	0.9178
Set-difference	DETE	0.9684	0.9493	0.9968	0.9725
Surge	D2Tier	0.8986	0.8539	0.9860	0.9152
Surge	DETE	0.9772	0.9909	0.9678	0.9793

At the identification stage, both architectures perform well against surge and set-difference attacks, achieving high F1-scores and strong set-level agreement. The Jaccard metric further confirms substantial overlap between the predicted and true sets of attacked sensors under these structured perturbations. Min–max attacks remain the most challenging for both models, lowering localization recall because their bounded deviations are less distinguishable from normal process variability. Noise attacks yield moderate localization performance across both architectures, reflecting the inherent difficulty of attributing stochastic disturbances to specific sensors. Overall, DETE provides more stable performance across attack types, whereas the two-tier framework prioritizes detection sensitivity, particularly for subtle

perturbations.

Table 6.16: *Network-level identification performance by attack type.*

Attack Type	Method	Accuracy	Precision	Recall	F1-score	Jaccard
Min–max	D2Tier	0.8988	0.9250	0.6138	0.7221	0.7603
Min–max	DETE	0.9051	0.9826	0.5841	0.7233	0.7613
Noise	D2Tier	0.8704	0.6892	0.8023	0.7364	0.7577
Noise	DETE	0.8735	0.6949	0.8219	0.7432	0.7585
Set-difference	D2Tier	0.9790	0.9340	0.9745	0.9533	0.9369
Set-difference	DETE	0.9829	0.9492	0.9742	0.9613	0.9513
Surge	D2Tier	0.9706	0.9417	0.9239	0.9320	0.9288
Surge	DETE	0.9732	0.9496	0.9248	0.9363	0.9413

Tables 6.17 and 6.18 further evaluate performance as a function of the number of sensors attacked simultaneously. At the detection level, the two-tier architecture maintains extremely high recall as attack cardinality increases, indicating strong sensitivity to coordinated multi-sensor deviations. However, this heightened sensitivity is associated with comparatively lower precision. In contrast, DETE preserves a more balanced precision–recall trade-off as the number of attacked sensors increases, yielding higher overall F1-scores at moderate and high attack cardinalities. This trend suggests that joint detection–identification learning improves robustness when attacks propagate across multiple sensors in the network.

Table 6.17: *Network-level attack detection performance as a function of attack cardinality.*

Attack Count	Method	Accuracy	Precision	Recall	F1-score
1	D2Tier	0.8833	0.8525	0.9522	0.8996
1	DETE	0.8996	0.9508	0.8618	0.9041
2	D2Tier	0.8944	0.8490	0.9975	0.9173
2	DETE	0.9477	0.9540	0.9569	0.9555
3	D2Tier	0.8928	0.8324	0.9997	0.9084
3	DETE	0.9531	0.9541	0.9579	0.9560
4	D2Tier	0.8920	0.8413	1.0000	0.9138
4	DETE	0.9638	0.9530	0.9853	0.9689
5	D2Tier	0.8994	0.8497	1.0000	0.9187
5	DETE	0.9651	0.9614	0.9779	0.9695

At the identification stage, both architectures exhibit gradual performance degradation as the number of attacked sensors increases, reflecting the growing combinatorial complexity of multi-label attack localization. Nevertheless, DETE maintains slightly stronger set-level

consistency for moderate attack cardinalities, as indicated by the Jaccard index. At extreme cardinalities, the performance gap between the two methods narrows, suggesting that highly coordinated attacks yield stronger aggregate signatures that are more distinguishable in both frameworks. The cardinality analysis indicates that DETE scales more consistently under multi-sensor attack conditions, whereas the two-tier strategy increasingly prioritizes detection sensitivity as attack cardinality increases. The corresponding network-level confusion matrices for both distributed frameworks are provided in Appendix A, offering additional insight into the balance between false alarms and missed detections at the sensor and network levels.

Table 6.18: *Network-level attack identification performance as a function of attack cardinality.*

Attack Count	Method	Accuracy	Precision	Recall	F1-score	Jaccard
1	D2Tier	0.9529	0.7644	0.8487	0.7997	0.8485
1	DETE	0.9559	0.7820	0.8461	0.8079	0.8519
2	D2Tier	0.9301	0.8444	0.8673	0.8551	0.8518
2	DETE	0.9334	0.8562	0.8659	0.8589	0.8602
3	D2Tier	0.9015	0.8861	0.7975	0.8386	0.8355
3	DETE	0.9081	0.9033	0.7956	0.8443	0.8498
4	D2Tier	0.8885	0.9393	0.8105	0.8692	0.8412
4	DETE	0.8970	0.9526	0.8162	0.8783	0.8579
5	D2Tier	0.8525	0.9724	0.7617	0.8532	0.8366
5	DETE	0.8458	0.9769	0.7464	0.8452	0.8335

6.6.5 Post-attack state reconstruction performance

The proposed detection and identification frameworks are further integrated into the closed-loop control system architecture. At each sampling interval, the frameworks analyze incoming temperature measurements to determine whether an attack is present and identify any compromised sensors. This information is relayed to the EKF-based state reconstructor of Section 6.5, which selectively replaces only the corrupted measurements with reconstructed estimates while retaining uncompromised sensor data; in the distributed setting, localized EKF estimators are activated only for the communities identified as under attack. This

selective reconstruction enhances state estimation accuracy and preserves system stability. Given that the primary control objective of the benzene alkylation process is the production of ethylbenzene (C), reconstruction performance is assessed using the temperature trajectories throughout the network and the outlet ethylbenzene concentration from the final reactor, C_{C5}. Both time-domain and state-space visualizations are employed, and performance is quantified using RMSE and EV. To ensure a fair comparison across all four frameworks, identical compromised-sensor locations are used across all attack types. In the single-attack case, the temperature sensor of the final reactor, T_5, is compromised, representing a critical downstream disturbance that directly affects the primary control objective. In the multi-attack scenario, simultaneous attacks are applied to T_1, T_2, and T_4, thereby affecting the upstream reactor dynamics and the separation unit.

Centralized frameworks

For the centralized frameworks, the EKF utilizes reconstructed values to mitigate the impact of sensor corruption, enabling the MHE-MPC system to maintain closed-loop stability. Figure 6.12 illustrates detection and reconstruction results using the CETE framework, while Figure 6.13 presents corresponding results using the two-tier framework under a surge-type attack on T_5. Both frameworks successfully detect, localize, and reconstruct the compromised signal in persistent-attack scenarios. However, comparative analysis of temperature and concentration trajectories, particularly for T_1, T_2, T_4, and C_{C5}, indicates that the two-tier architecture achieves earlier detection. This leads to faster activation of the reconstruction module and quicker stabilization of the closed-loop system.

In the multi-attack scenario, Figure 6.14 depicts the detection and state reconstruction results obtained using the CETE transformer-based framework. Corresponding outcomes using the two-tier framework are presented in Figure 6.15 . Both frameworks can accurately detect and reconstruct process states under concurrent multi-sensor attack conditions. However, the cascading impact of sensor corruption, the snowball effect, becomes particularly evident in the closed-loop behavior of downstream units, such as reactors T_3 and T_5. This

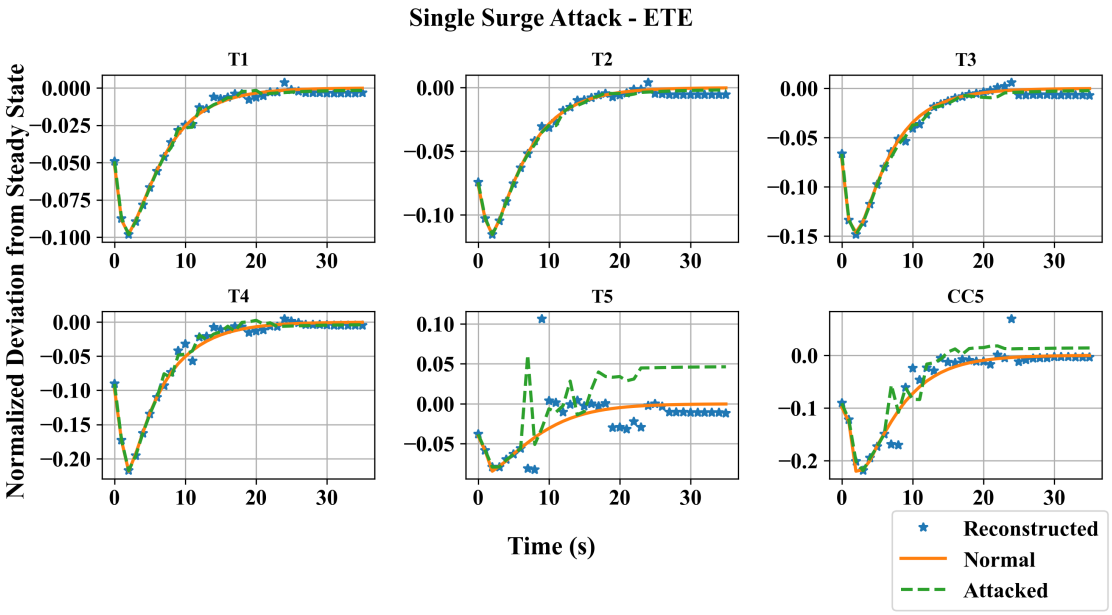

Figure 6.12: *Closed-loop detection and reconstruction under a single surge attack using the ETE framework.*

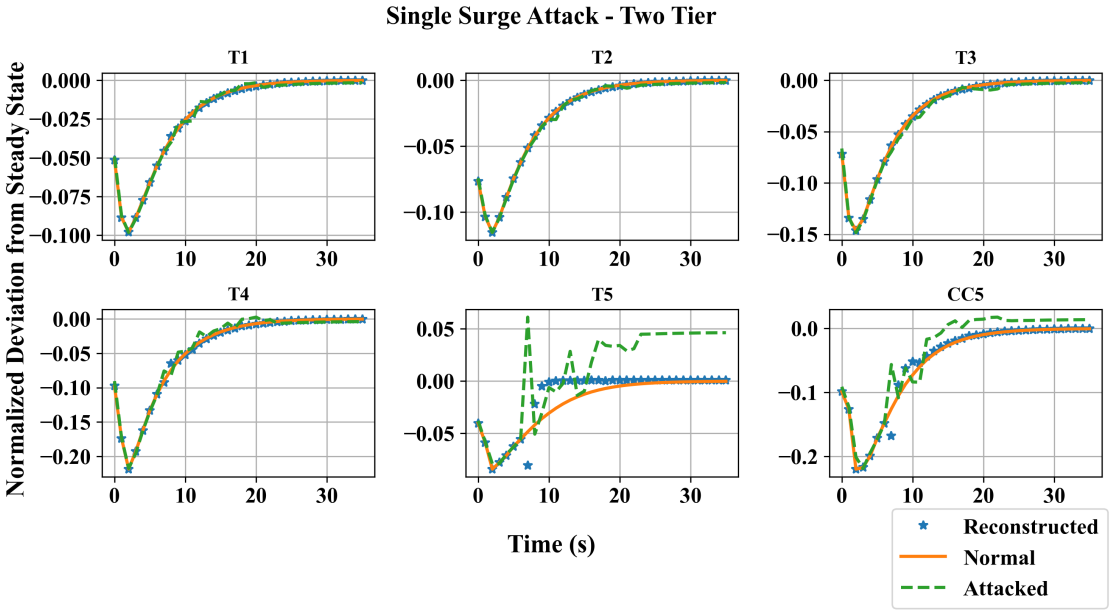

Figure 6.13: *Closed-loop detection and reconstruction under a single surge attack using a two-tier framework.*

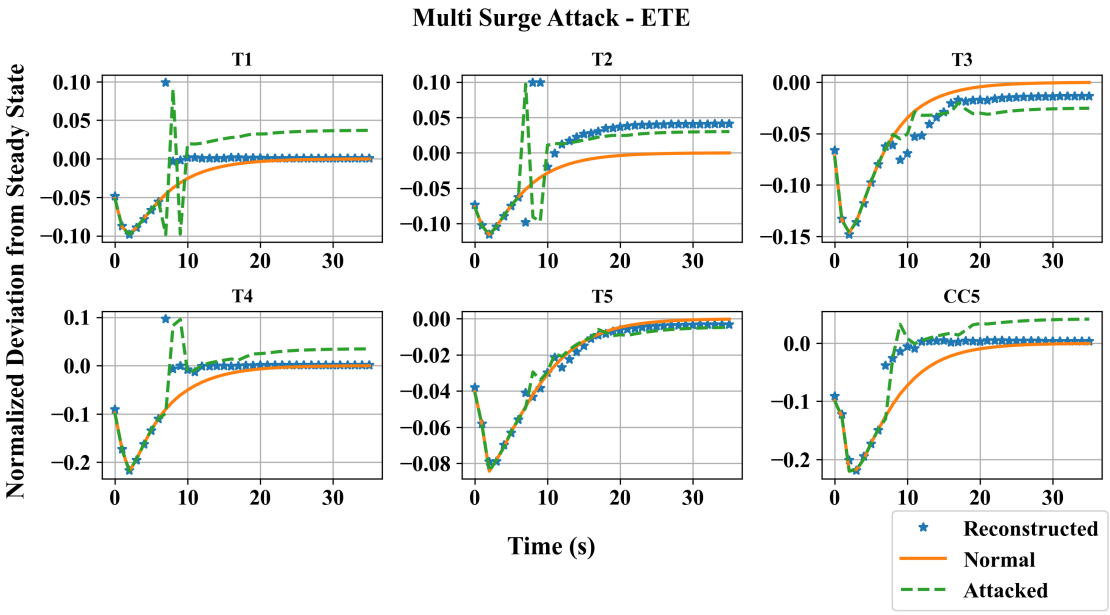

Figure 6.14: *Closed-loop detection and reconstruction under multi-sensor surge attacks using the ETE transformer framework.*

phenomenon underscores how compromised upstream measurements can propagate throughout the system, amplifying disturbances and complicating recovery. As such, the results reinforce the importance of explicitly addressing multi-point attack scenarios in designing cyber-resilient monitoring and control architectures.

As summarized in Tables 6.19 and 6.20, the two-tier framework generally achieves lower RMSE and higher EV in most single-attack cases, indicating more accurate reconstruction and better fidelity in tracking the true system states. Under multi-sensor attacks, while both frameworks perform comparably in scenarios dominated by surge or additive noise, the two-tier model exhibits more consistent reconstruction performance across various disturbances. In particular, it maintains robustness in the presence of set-point shifts and high-noise conditions, where model stability and sensitivity to input corruption are especially critical. Supplementary figures illustrating the reconstruction results for additional attack types are provided in Appendix A.

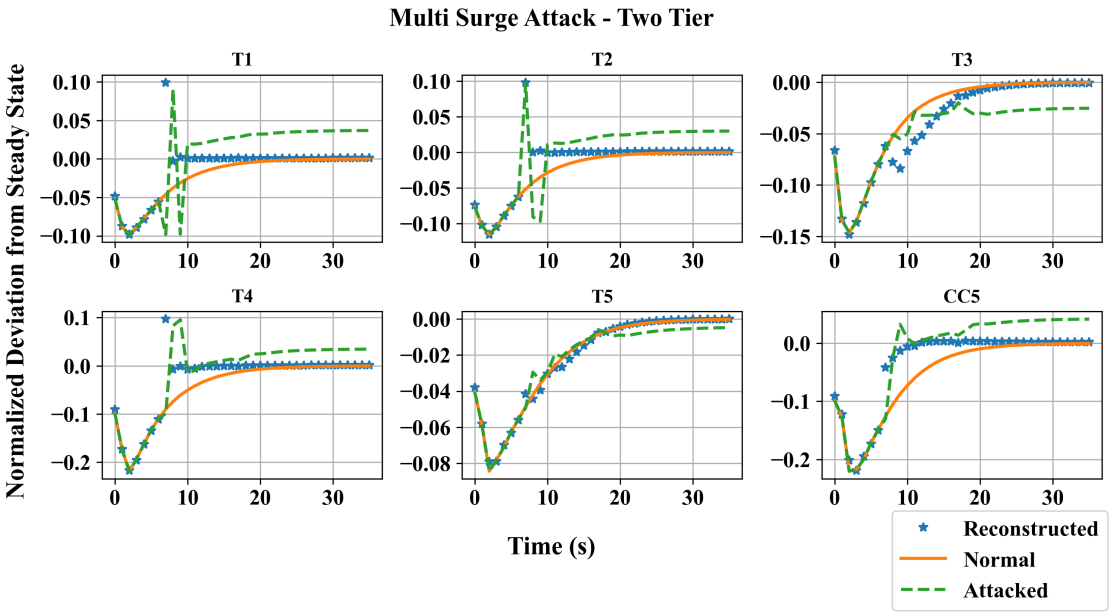

Figure 6.15: *Closed-loop detection and reconstruction under multi-sensor surge attacks using the two-tier framework.*

Table 6.19: *EKF-based reconstruction metrics under single-sensor attacks for the centralized frameworks.*

Attack Type	Detection Framework	RMSE	EV
Min-Max	CETE	0.0092	0.9046
Min-Max	Two-Tier	0.0069	0.9631
Surge	CETE	0.0177	0.7834
Surge	Two-Tier	0.0123	0.8693
Noise	CETE	0.0106	0.9279
Noise	Two-Tier	0.0100	0.9402
Set Difference	CETE	0.0147	0.7955
Set Difference	Two-Tier	0.0078	0.9535

Distributed frameworks

For the distributed frameworks, once compromised sensors are identified at the community level, localized EKF estimators are activated to estimate the affected states, reconstruct corrupted measurements, and restore consistent state trajectories for continued closed-loop operation. For each attack type, results are obtained for both the single- and multi-sensor cases; the min–max attack is examined in detail here, and the corresponding time-domain visualizations for the remaining attack types are provided in Appendix A.

Table 6.20: *EKF-based reconstruction metrics under multi-sensor attacks for the centralized frameworks.*

Attack Type	Detection Framework	RMSE	EV
Min-Max	CETE	0.0193	0.8349
Min-Max	Two-Tier	0.0170	0.8938
Surge	CETE	0.0301	0.6580
Surge	Two-Tier	0.0259	0.7048
Noise	CETE	0.0159	0.8540
Noise	Two-Tier	0.0078	0.9506
Set Difference	CETE	0.0226	0.7979
Set Difference	Two-Tier	0.0191	0.8756

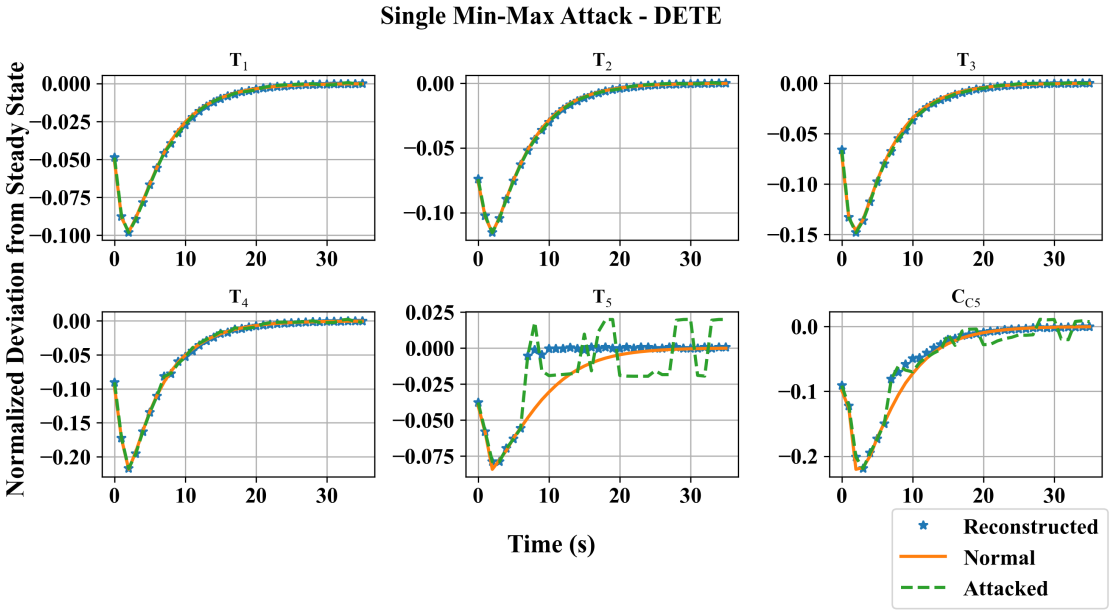

Figure 6.16: *Closed-loop detection and reconstruction under a single min–max attack using the DETE framework.*

Under the min–max attack, both distributed frameworks exhibit stable post-attack recovery in the single- and multi-sensor scenarios, as shown in Figure 6.16. In the single-sensor case involving T_5, the time series plots in Figures 6.16 and 6.17 show that the min–max attack induces a clear deviation in T_5 that propagates to the outlet concentration C_{C5}.

In the multi-sensor scenario with attacks on T_1, T_2, and T_4, the attack impact is more pronounced and propagates across multiple units, as shown in the time series plots in Figures 6.18, 6.19. Larger deviations from steady state operation are observed in the upstream reactors and the separator, reflecting the system-wide effects of coordinated min–max manip-

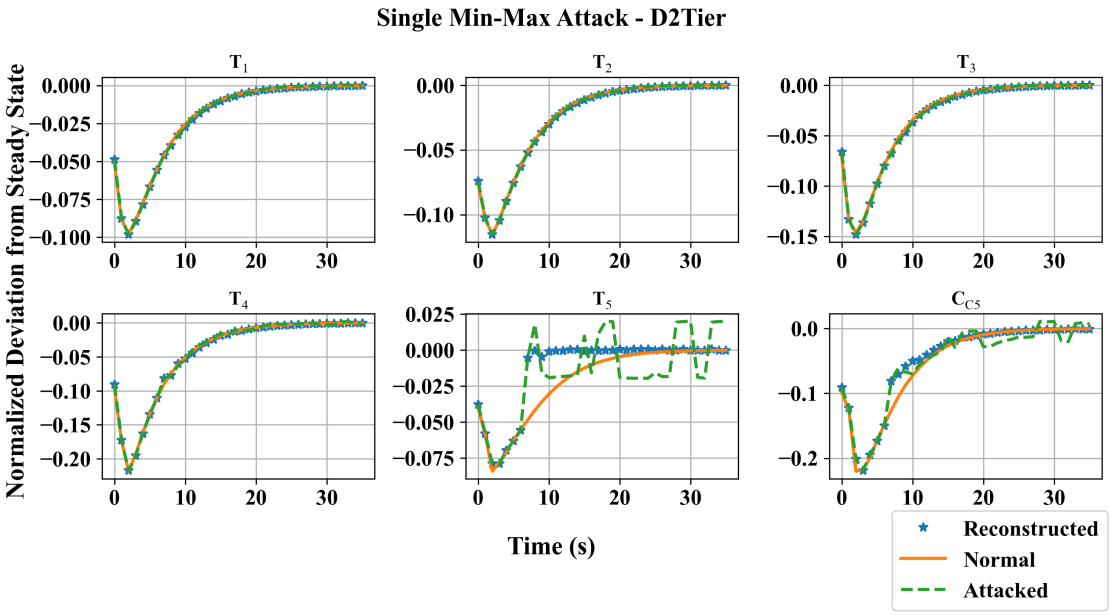

Figure 6.17: *Closed-loop detection and reconstruction under a single min–max attack using the D2Tier framework.*

ulation. Detection and localization performance directly affect the subsequent reconstruction stage. The distributed DETE framework correctly identifies attacks on T_2 and T_4, enabling partial reconstruction and reducing deviations in the affected trajectories. By contrast, the distributed two-tier architecture localizes the attack primarily to T_4, while attacks on T_1 and T_2 remain undetected and, therefore, unreconstructed. Consequently, residual deviations persist in these channels despite the overall stabilization of the system.

To provide an overall assessment of the effectiveness of the distributed reconstruction strategy, Tables 6.21 and 6.22 summarize the RMSE and EV metrics for single- and multi-sensor attacks across all attack types. As shown in Table 6.21, reconstruction performance under single-sensor attacks varies by attack type. Min–max, noise, and set-difference attacks are reconstructed with consistently low RMSE and high EV values for both frameworks, indicating limited performance degradation when the corrupted measurements are localized. In contrast, surge attacks are the most challenging single-attack scenario, resulting in higher RMSE and lower EV values than the other attack types. For these attacks, the D2Tier outperforms DETE, suggesting greater robustness in reconstructing abrupt transient distortions. Overall, among single attacks, surge perturbations produce the largest reconstruction

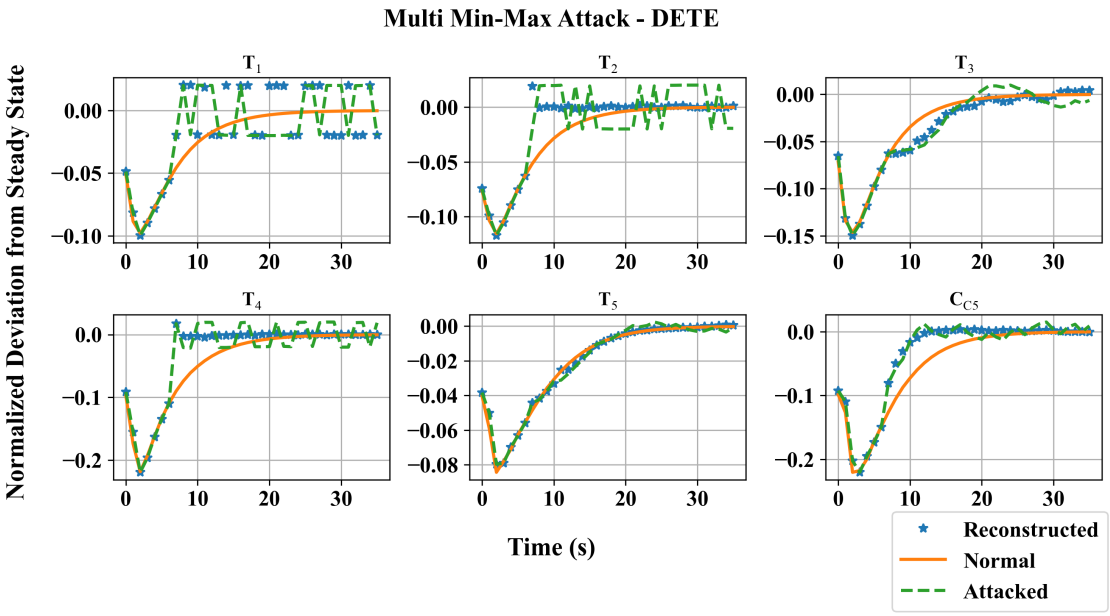

Figure 6.18: *Closed-loop detection and reconstruction under a multi-sensor min–max attack using the DETE framework.*

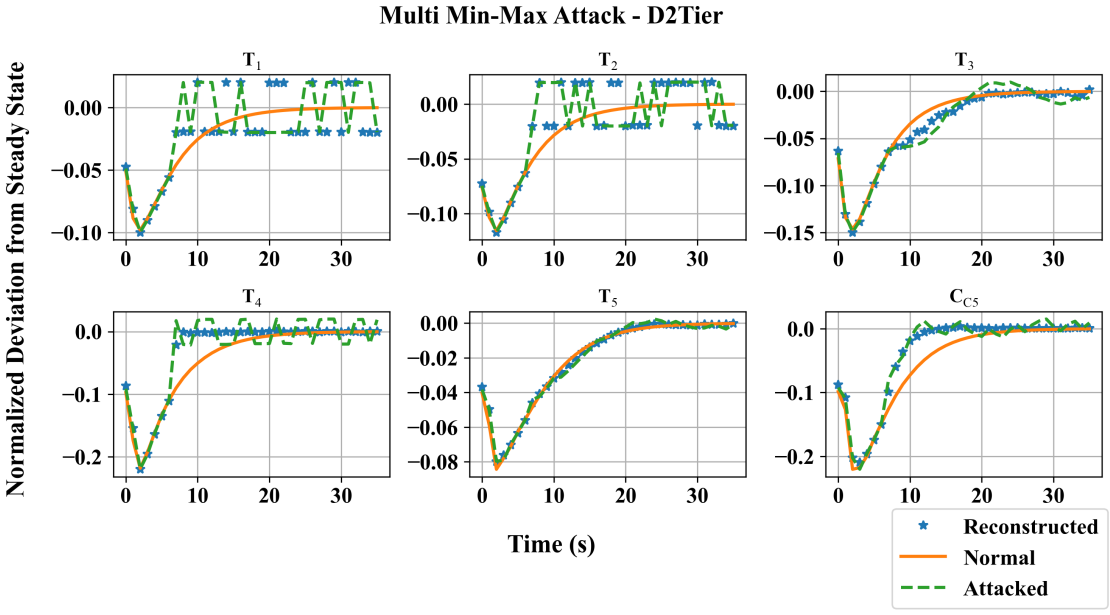

Figure 6.19: *Closed-loop detection and reconstruction under a multi-sensor min–max attack using the D2Tier framework*

errors, whereas min–max and noise attacks are comparatively easier to mitigate. Under coordinated multi-sensor attacks, as shown in Table 6.22, reconstruction becomes more challenging across all attack types. Surge attacks exhibit the most severe degradation, with the highest RMSE and lowest EV values, indicating that simultaneous abrupt manipulations introduce sustained deviations that are more difficult to correct. Noise attacks also lead to a notable reduction in reconstruction performance, particularly for DETE, reflecting its sensitivity to distributed stochastic perturbations. In contrast, min–max and set-difference attacks remain comparatively manageable, with only moderate increases in RMSE and EV, and EV remains above 0.83 in most cases. Overall, multi-sensor surge attacks represent the most demanding reconstruction scenario, whereas coordinated min–max and set-difference attacks cause moderate but controlled performance degradation. A detailed comparison of the centralized and distributed methods, along with an extended analysis of their computational efficiency, is presented in the following section.

Table 6.21: *Post-attack reconstruction performance under single-sensor attacks for the distributed frameworks.*

Attack Type	Framework	RMSE	EV
Min–max	DETE	0.007806	0.958749
Min–max	D2Tier	0.007817	0.958383
Surge	DETE	0.014230	0.853314
Surge	D2Tier	0.012322	0.878848
Noise	DETE	0.007149	0.951127
Noise	D2Tier	0.007135	0.959289
Set Difference	DETE	0.007940	0.956311
Set Difference	D2Tier	0.008169	0.955707

6.7 Comparative evaluation of centralized and distributed frameworks

To conclude the case study, the four proposed frameworks are compared at the network level across the detection, identification, and post-attack reconstruction tasks. For clarity, the centralized ETE and two-tier frameworks are hereafter denoted CETE and C2Tier,

Table 6.22: *Post-attack reconstruction performance under coordinated multi-sensor attacks for the distributed frameworks.*

Attack Type	Framework	RMSE	EV
Min–max	DETE	0.019394	0.837951
Min–max	D2Tier	0.018149	0.844878
Surge	DETE	0.028919	0.688838
Surge	D2Tier	0.029529	0.637622
Noise	DETE	0.023005	0.707770
Noise	D2Tier	0.020226	0.866647
Set Difference	DETE	0.016915	0.885949
Set Difference	D2Tier	0.016914	0.885989

respectively, to distinguish them from their distributed counterparts, DETE and D2Tier. For this comparison, all metrics are aggregated over the complete set of attack types and scenarios under identical closed-loop operating conditions and attack definitions; the reported values may therefore differ from the per-attack-type and per-cardinality overall values of Section 6.6.3, which reflect different aggregation bases.

Table 6.23 compares the network-level detection performance of the centralized frameworks with that of the proposed distributed frameworks. In the centralized implementation, the two-tier architecture improved detection performance relative to the single ETE model. This improvement was primarily due to separating binary detection from multi-label identification, which helped stabilize training in the high-dimensional joint modeling setting. When all sensors were modeled together, the optimization problem became more complex; therefore, task separation reduced learning complexity and improved model performance. In contrast, after introducing the community-based decomposition strategy, the DETE framework achieved the strongest overall detection performance. Because each community operates within a lower-dimensional subspace, the detection task becomes inherently more tractable. Under this reduced-complexity setting, the additional task separation imposed by the two-tier strategy no longer provides the same advantage. Instead, joint learning within DETE enables shared features to support both detection and identification more effectively, thereby improving overall consistency at the network level.

A similar pattern is observed in Table 6.24 for the identification task. In the centralized setting, the two-tier model outperforms the single ETE architecture because separating

Table 6.23: *Centralized vs. distributed network-level detection performance.*

Method	Accuracy	Precision	Recall	F1-score
CETE	0.7856	0.8351	0.7474	0.7888
C2Tier	0.8917	0.8324	0.9991	0.9082
DETE	0.9294	0.9523	0.9147	0.9331
D2Tier	0.8913	0.8455	0.9764	0.9063

detection and localization mitigates the combinatorial complexity of identifying multiple sensors under attack simultaneously. However, after decomposing the system into smaller communities, the label space within each subsystem is significantly reduced. Consequently, the identification task is less affected by combinatorial complexity. In the distributed setting, the DETE framework achieves stronger overall localization performance, including improved set-level agreement. These results suggest that, when subsystem dimensionality is reduced, maintaining a shared representation for detection and identification becomes beneficial.

Table 6.24: *Centralized vs. distributed network-level identification performance.*

Method	Accuracy	Precision	Recall	F1-score	Jaccard
CETE	0.6780	0.7606	0.6383	0.6928	0.7378
C2Tier	0.7022	0.7147	0.7616	0.7342	0.7796
DETE	0.9356	0.8720	0.8235	0.8450	0.8574
D2Tier	0.9318	0.8561	0.8244	0.8388	0.8504

The hierarchical separation in the two-tier approach inherently introduces a trade-off between information preservation and classification complexity. In the centralized setting, complexity reduction was the dominant consideration because the joint model had to process a high-dimensional input space and a large combinatorial label structure. In this case, separating detection and identification simplified the optimization problem, and the resulting loss of shared task information was outweighed by the reduction in model complexity. However, after community-based decomposition, the classification complexity within each subsystem is already substantially reduced. Consequently, the relative benefit of task separation diminishes, while the information loss caused by decoupling detection from identification becomes more consequential. In the distributed setting, preserving shared representations across tasks is therefore more advantageous, leading to the observed reversal in performance.

A similar comparison can be drawn from the post-attack reconstruction results in Tables 6.25 and 6.26, which report the RMSE and EV metrics averaged across all attack types. Under single-sensor attacks, the C2Tier architecture achieves the lowest average RMSE and the highest EV, indicating the most accurate steady state recovery. The distributed frameworks, DETE and D2Tier, exhibit slightly higher RMSE and moderately lower EV, yet they still maintain strong reconstruction fidelity. By contrast, the CETE architecture shows noticeably lower EV, suggesting a reduced ability to preserve dynamic trends after reconstruction. These results indicate that when corruption is localized to a single sensor, C2Tier separation remains effective due to its stable coordination between detection and localization.

Table 6.25: *Average reconstruction performance across attack types under single-sensor attacks.*

Architecture	Framework	RMSE	EV
2*Distributed	D2Tier	0.008861	0.938057
	DETE	0.009281	0.929875
2*Centralized	CETE	0.013685	0.809388
	C2Tier	0.007316	0.960062

Table 6.26: *Average reconstruction performance across attack types under coordinated multi-sensor attacks.*

Architecture	Framework	RMSE	EV
2*Distributed	D2Tier	0.021205	0.808784
	DETE	0.022058	0.780127
2*Centralized	CETE	0.022481	0.770155
	C2Tier	0.021349	0.746020

Under coordinated multi-sensor attacks, reconstruction performance declines across all architectures, as evidenced by higher RMSE and lower EV. In this setting, the performance gap between centralized and distributed methods narrows. Among the distributed approaches, the D2Tier framework achieves the highest EV and exhibits RMSE comparable to that of the centralized models. However, no single architecture consistently outperforms the others under multi-point corruption. Instead, reconstruction accuracy depends strongly on multi-sensor identification performance. When attacks are only partially localized or remain un-

detected, residual deviations persist and prevent full recovery. This behavior indicates that reconstruction effectiveness is directly governed by detection and identification accuracy, particularly as attack cardinality increases.

The combined evaluation of detection, identification, and post-attack reconstruction reveals a consistent trend. In the centralized setting, separating detection and identification mitigates the complexity of high-dimensional classification and improves performance. After community-based decomposition, however, the classification task within each subsystem becomes substantially simpler. In this distributed setting, joint learning within DETE becomes more advantageous, yielding stronger network-level detection and identification performance. For reconstruction, differences among architectures become less pronounced, especially under multi-sensor attacks, where performance is governed primarily by localization accuracy rather than model structure. These results demonstrate that community-based distributed modeling not only simplifies detection and identification but also supports stable and scalable post-attack recovery in highly integrated process systems. The computational efficiency of the four frameworks is examined next to complete the comparison.

6.8 Conclusions

This chapter extended the role of time-series machine learning in this dissertation from regression-based dynamic modeling to multi-label classification for resilient operation. Four transformer-based frameworks for cyberattack detection and compromised-sensor identification, the centralized CETE and C2Tier architectures and their community-based distributed counterparts, DETE and D2Tier, were developed and integrated with EKF-based post-attack state reconstruction under closed-loop MHE-MPC operation. Evaluated on the benzene alkylation benchmark across four categories of single- and multi-sensor adversarial attacks, all frameworks successfully detected, localized, and recovered from measurement corruption while preserving closed-loop stability.

The comparative analysis revealed that the preferred learning architecture depends on problem dimensionality. In the centralized setting, separating detection from identification

improved performance by mitigating the combinatorial complexity of the multi-label task, whereas after community-based decomposition, joint learning within DETE became more advantageous, yielding the strongest network-level detection and identification performance. The distributed frameworks substantially improved attack localization relative to their centralized counterparts, and post-attack reconstruction quality was found to be governed primarily by identification accuracy rather than model structure, with unidentified sensors remaining unreconstructed under coordinated multi-sensor attacks. These results demonstrate that community-based distributed monitoring offers a scalable and reliable foundation for cyber-resilient operation of highly integrated process systems.

Although the proposed frameworks generalize well within the considered attack space, their performance under novel or adaptive attack strategies remains an open challenge, motivating future work on adversarial robustness and domain generalization. Together with the preceding chapters, this chapter completes the objectives of the dissertation, where similar time-series architectures that served as predictive components for dynamic modeling, optimization, and control also function as protective components for resilient operation in the presence of adversarial cyberattacks.

Chapter 7

Conclusions and Future Directions

7.1 Concluding Remarks

The principal focus of this dissertation is that reliable dynamic modeling of complex process systems is, at its core, a problem of integrating knowledge of process systems. No single source of information, mechanistic simulation, first-principle models, sensory measurement, or structural insight, is sufficient on its own to capture the behavior of modern chemical, agricultural, and food processing systems at the fidelity and speed required for real-time decision-making. High-fidelity mechanistic models such as CFD simulations encode deep physical understanding but are computationally expensive in real time; purely data-driven models are fast but blind to the principles governing the processes they represent; sensory data streams are rich but heterogeneous and often disconnected from the quality attributes that ultimately matter; and the topology of an integrated plant carries essential information that centralized formulations discard. The frameworks developed throughout this dissertation were designed to bring these complementary layers of domain knowledge together, with time-series machine learning serving as the common representational core through which they are combined.

Several layers of domain knowledge were leveraged across this work. Simulation-embedded knowledge treats high-fidelity mechanistic models as data generators, distilling the accuracy

of multiscale digital twins into learned surrogates that are computationally compatible with real-time optimization. Physical laws enter through hybrid and physics-informed formulations, in which governing equations constrain and inform the learning process rather than being replaced by it. Multimodal sensory knowledge integrates product images with time-series process measurements, connecting operating conditions to quality attributes that were previously accessible only through slow offline experimentation. Structural knowledge exploits process topology and decomposition, recognizing that an integrated plant is not an undifferentiated whole but a network of functionally coherent subsystems whose structure can be mirrored in the architecture of the learning framework itself.

Passing these knowledge layers through a common time-series learning core yields a family of dynamic process models, surrogate, hybrid, multimodal, and distributed representations of process dynamics that constitute the core of this dissertation. These models were evaluated by the decisions they enable rather than by open-loop prediction accuracy alone. In dynamic optimization and predictive control, learned surrogates resolve the computational bottleneck of model-based control. In real-time monitoring, state forecasting and product quality assessment become feasible at the pace of the process. In resilient operation, learned models detect, identify, and reconstruct the effects of cyberattacks on integrated systems. Demonstrating these capabilities across chemical, agricultural, and food-processing applications supports the generality of the approach: the same knowledge-integration principles hold whether the system is a catalytic reactor network, a soil profile responding to weather, or a food extrusion line.

Beyond the individual frameworks, this work supports a broader conclusion about the practice of machine learning in process systems engineering. Across every application, the main challenge was not selecting a network architecture but confronting the realities of process knowledge and process data, such as irregular and biased sampling from computationally constrained simulators, missing and heterogeneous field observations, scarce multimodal samples, adversarially corrupted measurements, and the challenge of embedding physical and structural insight into an ML model. In this sense, knowledge integration is not a preprocessing step that precedes modeling; it is the modeling problem itself. The contribution

of this dissertation is a set of concrete, open-loop and closed-loop-validated strategies for carrying out that integration across different application domains.

7.2 Future Motivations: Toward Automated Reasoning for Knowledge Integration

The knowledge-integration framework developed in this dissertation also exposes its own principal limitation. The integration is still performed by the engineer. Choosing which knowledge layers to invoke for a given problem, structuring a hybrid model, decomposing an integrated plant, fusing sensory modalities, and responding when a model degrades or an anomaly appears all remain acts of expert judgment. These decisions are made offline and are revisited only when performance deteriorates. Process systems continue to grow in scale, and the available process knowledge, ranging from historical data and simulation archives to operating procedures, maintenance records, alarm logs, and operator experience, keeps expanding in both volume and heterogeneity. Under these conditions, manual integration becomes the bottleneck.

A compelling direction for future research is therefore to employ new-generation AI systems, particularly large language models (LLMs) and agentic AI architectures, as an automated reasoning layer above the time-series modeling core developed here. The frameworks in this dissertation integrate knowledge through carefully designed and mathematically supported pipelines, whereas LLM-based agents offer the prospect of performing that integration adaptively and online. With this perspective, several opportunities stand out.

The first opportunity is the automated orchestration of the knowledge layers themselves. An agentic system with access to the modeling tools developed in this dissertation, including the surrogate training pipelines, hybrid model structures, decomposition algorithms, and detection frameworks, could select the combination of knowledge sources and model forms suited to a given process, configure and execute the corresponding workflow, evaluate the outcome, and iterate. Much of the current modeling effort goes into tasks such as diagnosing

why a surrogate fails in a particular operating regime or selecting a suitable plant decomposition. These are, at heart, reasoning tasks over process knowledge, and they are natural candidates for agentic automation.

The second opportunity is the online integration of unstructured process knowledge. Much of the knowledge relevant to process operation exists in forms that time-series models cannot incorporate, such as standard operating procedures, piping and instrumentation documentation, shift logs, alarm rationalization studies, and incident reports. LLMs make these sources machine-readable for the first time. Coupling language-based reasoning over such documents with the state estimates produced by time-series models would allow a decision-support system to place a prediction or an anomaly in its operational context, for instance by linking a detected deviation to a maintenance action recorded only in a shift log, and to propose responses grounded in both process data and documented practice.

The third opportunity is agentic decision-making for resilient operation. The detection, identification, and reconstruction capabilities developed in this dissertation establish situational awareness under abnormal conditions, while the complementary challenge of response remains open. An agentic architecture could close this loop by reasoning over the identified fault or attack, the plant topology, and the available control degrees of freedom to propose mitigation actions, reconfigure distributed controllers, or escalate to human operators with an interpretable account of the situation. In this vision, the time-series models of this dissertation act as the perception layer of a larger autonomous system, while LLM-based agents supply the deliberative reasoning that currently resides only with the operator.

These opportunities also carry clear obligations. Language models are fallible reasoners that can assert incorrect facts with confidence, and their outputs carry none of the stability, feasibility, or safety guarantees that model-based control theory provides. Their role in process operation must therefore be structured so that every consequential action is verified against physics-based models, constrained by the same rigorous formulations that underpin predictive control, and validated in closed loop. These are the same disciplines this dissertation has applied to time-series learning. Developing verification layers, uncertainty quantification, and human-oversight protocols for LLM-driven reasoning in safety-critical

process environments is itself a rich research agenda, and a necessary complement to the capabilities described above.

In summary, this dissertation demonstrates that integrating layered domain knowledge through a time-series machine learning core yields dynamic process models that support optimization, control, monitoring, and resilient operations across chemical, agricultural, and food-processing systems. The natural continuation of this work is to automate the act of integration itself, pairing the quantitative modeling frameworks developed here with emerging reasoning systems so that process knowledge in every form, whether simulated, physical, sensory, structural, or textual, can inform process decisions continuously and online, at the scale that modern integrated systems demand. We hope this work serves as a foundation for the next generation of intelligent and resilient process systems.

Bibliography

[1] Adil Moghar and Mhamed Hamiche. Stock market prediction using lstm recurrent neural network. *Procedia computer science*, 170:1168–1173, 2020.

[2] Sidra Mehtab and Jaydip Sen. A time series analysis-based stock price prediction using machine learning and deep learning models. *International journal of business forecasting and marketing intelligence*, 6(4):272–335, 2020.

[3] Tingting Yuan, Wilson da Rocha Neto, Christian Esteve Rothenberg, Katia Obraczka, Chadi Barakat, and Thierry Turletti. Machine learning for next-generation intelligent transportation systems: A survey. *Transactions on emerging telecommunications technologies*, 33(4):e4427, 2022.

[4] Changxi Ma, Guowen Dai, and Jibiao Zhou. Short-term traffic flow prediction for urban road sections based on time series analysis and lstm_bilstm method. *IEEE Transactions on Intelligent Transportation Systems*, 23(6):5615–5624, 2021.

[5] Firuz Ahamed Nahid, Hussain Mahmud Chowdhury, and Mohammad Nayeem Jahangir. Machine learning techniques for demand forecasting in the electricity sector. *Forecasting Methods for Renewable Power Generation*, pages 131–172, 2025.

[6] Prosper O Ugbehe, Ogheneruona E Diemuodeke, and Daniel O Aikhuele. Electricity demand forecasting methodologies and applications: a review. *Sustainable Energy Research*, 12(1):19, 2025.

[7] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553): 436–444, 2015.

[8] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N

Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[9] Norman P Jouppi, Cliff Young, Nishant Patil, David Patterson, Gaurav Agrawal, Raminder Bajwa, Sarah Bates, Suresh Bhatia, Nan Boden, Al Borchers, et al. In-datacenter performance analysis of a tensor processing unit. In *Proceedings of the 44th annual international symposium on computer architecture*, pages 1–12, 2017.

[10] Constantinos C Pantelides and JG Renfro. The online use of first-principles models in process operations: Review, current status and future needs. *Computers & Chemical Engineering*, 51:136–148, 2013.

[11] Laurens Van Hoecke, Dieter Boeye, Arturo Gonzalez-Quiroga, Gregory S Patience, and Patrice Perreault. Experimental methods in chemical engineering: Computational fluid dynamics/finite volume method—cfd/fvm. *The Canadian Journal of Chemical Engineering*, 101(2):545–561, 2023.

[12] Joel L Plawsky. *Transport phenomena fundamentals*. CRC press, 2020.

[13] Luisa Peterson, Ion Victor Gosea, Peter Benner, and Kai Sundmacher. Digital twins in process engineering: An overview on computational and numerical methods. *Computers & Chemical Engineering*, 193:108917, 2025.

[14] Atharv Bhosekar and Marianthi Ierapetritou. Advances in surrogate based modeling, feasibility analysis, and optimization: A review. *Computers & Chemical Engineering*, 108:250–267, 2018.

[15] Kevin McBride and Kai Sundmacher. Overview of surrogate modeling in chemical process engineering. *Chemie Ingenieur Technik*, 91(3):228–239, 2019.

[16] Ruth Misener and Lorenz Biegler. Formulating data-driven surrogate models for process optimization. *Computers & Chemical Engineering*, 179:108411, 2023.

[17] William Bradley, Jinhyeun Kim, Zachary Kilwein, Logan Blakely, Michael Eydenberg, Jordan Jalvin, Carl Laird, and Fani Boukouvala. Perspectives on the integration between first-principles and data-driven modeling. *Computers & Chemical Engineering*, 166:107898, 2022.

[18] Bianca Williams and Selen Cremaschi. Selection of surrogate modeling techniques for surface approximation and surrogate-based optimization. *Chemical Engineering Research and Design*, 170:76–89, 2021.

[19] Peter Benner, Serkan Gugercin, and Karen Willcox. A survey of projection-based model reduction methods for parametric dynamical systems. *SIAM review*, 57(4): 483–531, 2015.

[20] Bart Besselink, Umut Tabak, Agnieszka Lutowska, Nathan Van de Wouw, Henk Nijmeijer, Daniel J Rixen, ME Hochstenbach, and WHA Schilders. A comparison of model reduction techniques from structural dynamics, numerical mathematics and systems and control. *Journal of Sound and Vibration*, 332(19):4403–4422, 2013.

[21] Jan C Schulze and Alexander Mitsos. Nonlinear model order reduction of dynamical systems in process engineering: review and comparison. *Nonlinear Dynamics*, 114(9): 630, 2026.

[22] Moritz Von Stosch, Rui Oliveira, Joana Peres, and Sebastião Feyo De Azevedo. Hybrid semi-parametric modeling in process systems engineering: Past, present and future. *Computers & Chemical Engineering*, 60:86–101, 2014.

[23] Niket Sharma and YA Liu. A hybrid science-guided machine learning approach for modeling and optimizing chemical processes. *arXiv preprint arXiv:2112.01475*, 2021.

[24] Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics*, 378: 686–707, 2019.

[25] George Em Karniadakis, Ioannis G Kevrekidis, Lu Lu, Paris Perdikaris, Sifan Wang, and Liu Yang. Physics-informed machine learning. *Nature Reviews Physics*, 3(6): 422–440, 2021.

[26] Guang Hu, Nikolaos Prasianakis, Sergey V Churakov, and Wilfried Pfingsten. Performance analysis of data-driven and physics-informed machine learning methods for thermal-hydraulic processes in full-scale emplacement experiment. *Applied Thermal Engineering*, 245:122836, 2024.

[27] Mohammed S Alhajeri, Fahim Abdullah, Zhe Wu, and Panagiotis D Christofides. Physics-informed machine learning modeling for predictive control using noisy data. *Chemical Engineering Research and Design*, 186:34–49, 2022.

[28] Jiewu Leng, Kaiwen Zuo, Caiyu Xu, Xueliang Zhou, Shuai Zheng, Jiawen Kang, Qiang Liu, Xin Chen, Weiming Shen, Lihui Wang, et al. Physics-informed machine learning in intelligent manufacturing: a review. *Journal of Intelligent Manufacturing*, 37(6): 2215–2257, 2026.

[29] Guofei Pang and George Em Karniadakis. Physics-informed learning machines for partial differential equations: Gaussian processes versus neural networks. In *Emerging frontiers in nonlinear science*, pages 323–343. Springer, 2020.

[30] Zhiyong Wu, Huan Wang, Chang He, Bingjian Zhang, Tao Xu, and Qinglin Chen. The application of physics-informed machine learning in multiphysics modeling in chemical engineering. *Industrial & Engineering Chemistry Research*, 62(44):18178–18204, 2023.

[31] Octavio Loyola-Gonzalez. Black-box vs. white-box: Understanding their advantages and weaknesses from a practical point of view. *IEEE access*, 7:154096–154113, 2019.

[32] Vikas Hassija, Vinay Chamola, Atmesh Mahapatra, Abhinandan Singal, Divyansh Goel, Kaizhu Huang, Simone Scardapane, Indro Spinelli, Mufti Mahmud, and Amir Hussain. Interpreting black-box models: a review on explainable artificial intelligence. *Cognitive Computation*, 16(1):45–74, 2024.

[33] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. A survey of methods for explaining black box models. *ACM computing surveys (CSUR)*, 51(5):1–42, 2018.

[34] Emrullah ŞAHiN, Naciye Nur Arslan, and Durmuş Özdemir. Unlocking the black box: an in-depth review on interpretability, explainability, and reliability in deep learning. *Neural computing and applications*, 37(2):859–965, 2025.

[35] Aswin Chandrasekar, Tyler Wortley, Euan Bohm, and Prashant Mhaskar. Implementing recurrent neural networks in process systems engineering applications, the right way! *Computers & Chemical Engineering*, 195:109027, 2025.

[36] Zhe Wu, Panagiotis D Christofides, Wanlu Wu, Yujia Wang, Fahim Abdullah, Aisha Alnajdi, and Yash Kadakia. A tutorial review of machine learning-based model predictive control methods. *Reviews in Chemical Engineering*, 41(4):359–400, 2025.

[37] Bernardo B Schwedersky, Rodolfo CC Flesch, and Hiago AS Dangui. Practical nonlinear model predictive control algorithm for long short-term memory networks. *IFAC-PapersOnLine*, 52(1):468–473, 2019.

[38] Mohammed S Alhajeri, Zhe Wu, David Rincon, Fahad Albalawi, and Panagiotis D Christofides. Machine-learning-based state estimation and predictive control of nonlinear processes. *Chemical Engineering Research and Design*, 167:268–280, 2021.

[39] Zhe Wu, Fahad Albalawi, Junfeng Zhang, Zhihao Zhang, Helen Durand, and Panagiotis D Christofides. Detecting and handling cyber-attacks in model predictive control of chemical processes. *Mathematics*, 6(10):173, 2018.

[40] Scarlett Chen, Zhe Wu, and Panagiotis D Christofides. Cyber-attack detection and resilient operation of nonlinear processes under economic model predictive control. *Computers & Chemical Engineering*, 136:106806, 2020.

[41] Tianqi Chen and Carlos Guestrin. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794, 2016.

[42] Yongming Han, Fan Yu, Jiaxin Liu, Zhen Zhang, Bo Ma, Ling Wang, and Zhiqiang Geng. An improved tuna swarm optimization algorithm based xgboost classification method for food risk evaluation. *Swarm Evol. Comput.*, page 102249, 2025.

[43] Yongming Han, Jiaxin Liu, Feng Pan, Qingxu Ni, Bo Ma, and Zhiqiang Geng. Synthesized minority oversampling technique-reverse k-nearest neighbors-k-dimensional tree for dairy food safety risk evaluation. *Expert Syst. Appl.*, 275:127064, 2025.

[44] Yi Zhang, Xiupeng Shi, Sheng Zhang, and Anuj Abraham. A xgboost-based lane change prediction on time series data using feature engineering for autopilot vehicles. *IEEE Trans. Intell. Transp. Syst.*, 23(10):19187–19200, 2022.

[45] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. *NeurIPS*, 30, 2017.

[46] Ahmad Alsharef, Karan Aggarwal, Sonia, Manoj Kumar, and Ashutosh Mishra. Review of ml and automl solutions to forecast time-series data. *Arch. Comput. Method Eng.*, 29(7):5297–5311, 2022.

[47] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. URL https://www.deeplearningbook.org/.

[48] Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural networks*, 61:85–117, 2015.

[49] Sebastian Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016.

[50] Yi Ming Ren, Mohammed S Alhajeri, Junwei Luo, Scarlett Chen, Fahim Abdullah, Zhe Wu, and Panagiotis D Christofides. A tutorial review of neural network modeling approaches for model predictive control. *Computers & Chemical Engineering*, page 107956, 2022.

[51] Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[52] Zelda Mariet and Vitaly Kuznetsov. Foundations of sequence-to-sequence modeling for time series. In *The 22nd international conference on artificial intelligence and statistics*, pages 408–417. PMLR, 2019.

[53] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, page 3104–3112. MIT Press, 2014.

[54] Zonglei Chen, Minbo Ma, Tianrui Li, Hongjun Wang, and Chongshou Li. Long sequence time-series forecasting with deep learning: A survey. *Information Fusion*, 97: 101819, 2023.

[55] Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pages 11121–11128, 2023.

[56] Mohammed Saad Faizan Bangi and Joseph Sang-Il Kwon. Deep hybrid modeling of chemical process: Application to hydraulic fracturing. *Computers & Chemical Engineering*, 134:106696, 2020.

[57] Poonam Dhankhar and Neha Sahu. A review and research of edge detection techniques for image segmentation. *IJCSMC*, 2(7):86–92, 2013.

[58] Priyanka Mukhopadhyay and Bidyut B Chaudhuri. A survey of hough transform. *Pattern Recognit.*, 48(3):993–1010, 2015.

[59] Yanuar Wicaksono, RomiSatrial Wahono, and Vincent Suhartono. Color and texture feature extraction using gabor filter-local binary patterns for image segmentation with fuzzy c-means. *J. Intell. Syst.*, 1(1):15–21, 2015.

[60] Mohiuddin Ahmed, Raihan Seraj, and Syed Mohammed Shamsul Islam. The k-means algorithm: A comprehensive survey and performance evaluation. *Electronics*, 9(8): 1295, 2020.

[61] Mengyao Cui et al. Introduction to the k-means clustering algorithm based on the elbow method. *Accounting, Auditing and Finance*, 1(1):5–8, 2020.

[62] Purnima Bholowalia and Arvind Kumar. Ebk-means: A clustering technique based on elbow method and k-means in wsn. *International Journal of Computer Applications*, 105(9), 2014.

[63] Cort J Willmott and Kenji Matsuura. Advantages of the mean absolute error (mae) over the root mean square error (rmse) in assessing average model performance. *Climate research*, 30(1):79–82, 2005.

[64] RJ Hyndman. *Forecasting: principles and practice*. OTexts, 2018.

[65] Marina Sokolova and Guy Lapalme. A systematic analysis of performance measures for classification tasks. *Information processing & management*, 45(4):427–437, 2009.

[66] Tom Fawcett. An introduction to roc analysis. *Pattern recognition letters*, 27(8): 861–874, 2006.

[67] J.B. Rawlings and D.Q. Mayne. *Model Predictive Control: Theory and Design*. SIAM, 2017.

[68] Amirsalar Bagheri, Thiago Oliveira Cabral, and Davood B Pourkargar. Integrated learning-based estimation and nonlinear predictive control of an ammonia synthesis reactor. *AIChE Journal*, 71(5):e18732, 2025.

[69] Jan Willem Erisman, Mark A Sutton, James Galloway, Zbigniew Klimont, and Wilfried Winiwarter. How a century of ammonia synthesis changed the world. *Nature Geoscience*, 1(10):636–639, 2008.

[70] Douglas R MacFarlane, Pavel V Cherepanov, Jaecheol Choi, Bryan HR Suryanto, Rebecca Y Hodgetts, Jacinta M Bakker, Federico M Ferrero Vallana, and Alexandr N Simonov. A roadmap to the ammonia economy. *Joule*, 4(6):1186–1205, 2020.

[71] BV Babu and Rakesh Angira. Optimal design of an auto-thermal ammonia synthesis reactor. *Computers & Chemical Engineering*, 29(5):1041–1045, 2005.

[72] Natalia Morlanés, Sai P. Katikaneni, Stephen N. Paglieri, Aadesh Harale, Bandar Solami, S. Mani Sarathy, and Jorge Gascon. A technological roadmap to the ammonia energy economy: Current state and missing technologies. *Chemical Engineering Journal*, 408:127310, 2021. ISSN 1385-8947.

[73] D. Saygin, Herib Blanco, F. Boshell, Joseph Cordonnier, Kevin H. R. Rouwenhorst, P. Lathwal, and Dolf Gielen. Ammonia production from clean hydrogen and the implications for global natural gas demand. *Sustainability*, 15(2):1623, 2023. doi:10.3390/su15021623.

[74] Kevin H. R. Rouwenhorst, A. Travis, and L. Lefferts. 1921–2021: A century of renewable ammonia synthesis. *Sustainable Chemistry*, 3(2):11, 2022. doi:10.3390/suschem3020011.

[75] Katie Smart. Review of recent progress in green ammonia synthesis. *Platinum Metals Review*, 66(3):230–244, 2022.

[76] Matthew J Palys, Hanchu Wang, Qi Zhang, and Prodromos Daoutidis. Renewable ammonia for sustainable energy and agriculture: vision and systems engineering opportunities. *Current Opinion in Chemical Engineering*, 31:100667, 2021.

[77] Hanchu Wang, Prodromos Daoutidis, and Qi Zhang. Green ammonia supply chain

design for maritime transportation. *Computer Aided Chemical Engineering*, 49:589–594, 2022.

[78] Kyuha Lee, Xinyu Liu, Pradeep Vyawahare, Pingping Sun, A. Elgowainy, and Michael Wang. Techno-economic performances and life cycle greenhouse gas emissions of various ammonia production pathways including conventional, carbon-capturing, nuclear-powered, and renewable production. *Green Chemistry*, 24:4830–4844, 2022.

[79] Chun Yi Liu et al. Ammonia adsorption on ion exchanged y-zeolites as ammonia storage material. *Journal of the Japan Petroleum Institute*, 46(5):301–307, 2003.

[80] Ji Hye Park, Min Woo Hong, Hyung Chul Yoon, and Kwang Bok Yi. Effects of mgcl2 loading on ammonia capacity of activated carbon for application in temperature swing adsorption, pressure swing adsorption, and pressure-temperature swing adsorption process. *Korean Journal of Chemical Engineering*, 39(10):2775–2782, 2022.

[81] Thiago Oliveira Cabral, Amirsalar Bagheri, and Davood B Pourkargar. Learning-based model reduction and predictive control of an ammonia synthesis process. *Industrial & Engineering Chemistry Research*, 63(23):10325–10342, 2024.

[82] Thiago Oliveira Cabral and Davood B Pourkargar. Nonlinear model predictive control of a modular hydrogen-ammonia and renewable energy generation system. In *2025 American Control Conference (ACC)*, pages 2402–2407. IEEE, 2025.

[83] O. C. Zienkiewicz, R. L. Taylor, and J. Z. Zhu. *The Finite Element Method: Its Basis and Fundamentals*. Elsevier, 2005.

[84] M. P. Bendsøe and O. Sigmund. *Topology Optimization: Theory, Methods, and Applications*. Springer, 2003.

[85] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[86] William E Hart, Carl D Laird, Jean-Paul Watson, David L Woodruff, Gabriel A Hackebeil, Bethany L Nicholson, John D Siirola, et al. *Pyomo-optimization modeling in python*, volume 67. Springer, 2017.

[87] Kurt B Waldman, Peter M Todd, Shahera Omar, Jordan P Blekking, Stacey A Giroux, Shahzeen Z Attari, Kathy Baylis, and Tom P Evans. Agricultural decision making and climate uncertainty in developing countries. *Environmental Research Letters*, 15(11): 113004, 2020.

[88] Tayeb Jamali, Behzad Ghanbarian, and Jürgen Kurths. Spatiotemporal analysis of extreme precipitation events in the united states at mesoscale: Complex network theory. *Journal of Hydrology*, 627:130440, 2023.

[89] Sercan Yalçın, Musa Eşit, and Önder Çoban. A new deep learning method for meteorological drought estimation based-on standard precipitation evapotranspiration index. *Engineering Applications of Artificial Intelligence*, 124:106550, 2023.

[90] Marco Bittelli. Measuring soil water content: A review. *HortTechnology*, 21(3):293–300, 2011.

[91] Erion Bwambale, Felix K Abagale, and Geophrey K Anornu. Smart irrigation monitoring and control strategies for improving water use efficiency in precision agriculture: A review. *Agricultural Water Management*, 260:107324, 2022.

[92] Andres Patrignani, Tyson E Ochsner, Lei Feng, Don Dyer, and Pedro R Rossini. Calibration and validation of soil water reflectometers. *Vadose Zone Journal*, 21(3): e20190, 2022.

[93] Hendrik Rujner, Günther Leonhardt, Jiri Marsalek, Anna-Maria Perttu, and Maria Viklander. The effects of initial soil moisture conditions on swale flow hydrographs. *Hydrological Processes*, 32(5):644–654, 2018.

[94] Luca Brocca, Luca Ciabatta, Christian Massari, Stefania Camici, and Angelica Tarpanelli. Soil moisture for hydrological applications: Open questions and new opportunities. *Water*, 9(2):140, 2017.

[95] Andrea Benedetto, Fabio Tosti, Bianca Ortuani, Mauro Giudici, and Mauro Mele. Soil moisture mapping using gpr for pavement applications. In *2013 7th International Workshop on Advanced Ground Penetrating Radar*, pages 1–5. IEEE, 2013.

[96] Auckpath Sawangsuriya, Tuncer B Edil, and Peter J Bosscher. Modulus-suction-moisture relationship for compacted soils in postcompaction state. *Journal of Geotechnical and geoenvironmental engineering*, 135(10):1390–1403, 2009.

[97] Carol J Miller, Nazli Yesiller, K Yaldo, and S Merayyan. Impact of soil type and compaction conditions on soil water characteristic. *Journal of Geotechnical and Geoenvironmental Engineering*, 128(9):733–742, 2002.

[98] Ganquan Mao and Junguo Liu. Ways v1: a hydrological model for root zone water storage simulation on a global scale. *Geoscientific Model Development*, 12(12):5267–5289, 2019.

[99] Andrew J Guswa, Michael Anthony Celia, and I Rodriguez-Iturbe. Models of soil moisture dynamics in ecohydrology: A comparative study. *Water Resources Research*, 38(9):5–1, 2002.

[100] L. A. Richards. Capillary conduction of liquids through porous mediums. *Physics*, 1 (5):318–333, 1931.

[101] Honglin Zhu, Tingxi Liu, Baolin Xue, and Guoqiang Wang. Modified Richards' equation to improve estimates of soil moisture in two-layered soils after infiltration. *Water*, 10(9):1174, 2018.

[102] Daniel Hillel. *Introduction to Environmental Soil Physics*. Elsevier Academic Press, 2004.

[103] Harry Vereecken, Andrea Schnepf, Jan W Hopmans, Mathieu Javaux, Dani Or, Tiina Roose, Jan Vanderborght, Michael H Young, Wulf Amelung, Matt Aitkenhead, and et al. Modeling soil processes: Review, key challenges, and new perspectives. *Vadose Zone Journal*, 15(5):vzj2015.09.0131, 2016.

[104] Andre Peters, Tobias L Hohenbrink, Sascha C Iden, Martinus Th Van Genuchten, and Wolfgang Durner. Prediction of the absolute hydraulic conductivity function from soil water retention data. *Hydrology and Earth System Sciences Discussions*, 2023:1–32, 2023.

[105] Shima Amani and Hossein Shafizadeh-Moghadam. A review of machine learning models and influential factors for estimating evapotranspiration using remote sensing and ground-based data. *Agricultural Water Management*, 284:108324, 2023.

[106] Andres Patrignani, Mary Knapp, Christopher Redmond, and Eduardo Santos. Technical overview of the kansas mesonet. *Journal of Atmospheric and Oceanic Technology*, 37(12):2167–2183, 2020.

[107] Yann H Kerr, Amen Al-Yaari, Nemesio Rodriguez-Fernandez, Marie Parrens, Beatriz Molero, Delphine Leroux, Simone Bircher, Ali Mahmoodi, Arnaud Mialon, Philippe Richaume, et al. Overview of smos performance in terms of global soil moisture monitoring after six years in operation. *Remote Sensing of Environment*, 180:40–63, 2016.

[108] Fred V Brock, Kenneth C Crawford, Ronald L Elliott, Gerrit W Cuperus, Steven J Stadler, Howard L Johnson, and Michael D Eilts. The oklahoma mesonet: A technical overview. *Journal of Atmospheric and Oceanic Technology*, 12(1):5–19, 1995.

[109] Jerald A Brotzge, J Wang, CD Thorncroft, E Joseph, N Bain, N Bassill, N Farruggio, JM Freedman, K Hemker Jr, D Johnston, et al. A technical overview of the new york state mesonet standard network. *Journal of Atmospheric and Oceanic Technology*, 37 (10):1827–1845, 2020.

[110] John L Schroeder, WS Burgett, KB Haynie, I Sonmez, GD Skwira, AL Doggett, and JW Lipe. The west texas mesonet: A technical overview. *Journal of Atmospheric and Oceanic Technology*, 22(2):211–222, 2005.

[111] Iftikhar Ali, Felix Greifeneder, Jelena Stamenkovic, Maxim Neumann, and Claudia Notarnicola. Review of machine learning approaches for biomass and soil moisture retrievals from remote sensing data. *Remote Sensing*, 7(12):16398–16421, 2015.

[112] Binru Zhao, Qiang Dai, Dawei Han, Huichao Dai, Jingqiao Mao, Lu Zhuo, and Guiwen Rong. Estimation of soil moisture using modified antecedent precipitation index with application in landslide predictions. *Landslides*, 16(12):2381–2393, 2019.

[113] Gerhard Schoener and Mark C Stone. Monitoring soil moisture at the catchment scale–a novel approach combining antecedent precipitation index and radar-derived rainfall data. *Journal of Hydrology*, 589:125155, 2020.

[114] Marcio Augusto Ernesto De Moraes, Walter Manoel Mendes Filho, Rodolfo Moreda Mendes, Cassiano Antonio Bortolozo, Daniel Metodiev, Marcio Roberto Magalhães De Andrade, LA Pampuch, Tatiana Sussel Gonçalves Mendes, and Luana Albertani Pampuch. Antecedent precipitation index to estimate soil moisture and correlate as a triggering process in the occurrence of landslides. *International Journal of Geosciences*, 15(1):70–86, 2024.

[115] Thomas Ramsauer and Philip Marzahn. Global soil moisture estimation based on gpm imerg data using a site specific adjusted antecedent precipitation index. *International Journal of Remote Sensing*, 44(2):542–566, 2023.

[116] Meshal Shutaywi and Nezamoddin N Kachouie. Silhouette analysis for performance evaluation in machine learning with applications to clustering. *Entropy*, 23(6):759, 2021.

[117] Feiyang Ren, Yi Han, Shaohan Wang, and He Jiang. A novel high-dimensional trajec-

tories construction network based on multi-clustering algorithm. *EURASIP Journal on Wireless Communications and Networking*, 2022(1):18, 2022.

[118] Jinchuang Zhang, Qiongling Chen, David L Kaplan, and Qiang Wang. High-moisture extruded protein fiber formation toward plant-based meat substitutes applications: Science, technology, and prospect. *Trends Food Sci. Technol.*, 128:202–216, 2022.

[119] Blake J Plattner, Shan Hong, Yonghui Li, Martin J Talavera, Hulya Dogan, Brian S Plattner, and Sajid Alavi. Use of pea proteins in high-moisture meat analogs: Physicochemical properties of raw formulations and their texturization using extrusion. *Foods*, 13(8):1195, 2024.

[120] Cameron McGuire, Kaliramesh Siliveru, Snehasis Chakraborty, Kingsly Ambrose, and Sajid Alavi. Flow properties of coarse powders used in food extrusion as a function of moisture content. *Processes*, 12(6):1246, 2024.

[121] António Gaspar-Cunha, Francisco Monaco, Janusz Sikora, and Alexandre Delbem. Artificial intelligence in single screw polymer extrusion: Learning from computational data. *Eng. Appl. Artif. Intell.*, 116:105397, 2022.

[122] Adrian Lewandowski and Krzysztof Wilczyński. Modeling of twin screw extrusion of polymeric materials. *Polymers*, 14(2):274, 2022.

[123] Jenna Flory and Sajid Alavi. Use of hydration properties of proteins to understand their functionality and tailor texture of extruded plant-based meat analogues. *J. Food Sci.*, 89(1):245–258, 2024.

[124] Md Imran H Khan, Shyam S Sablani, Richi Nayak, and Yuantong Gu. Machine learning-based modeling in food processing applications: State of the art. *Compr. Rev. Food Sci. Food Saf.*, 21(2):1409–1438, 2022.

[125] GVS Bhagya Raj and Kshirod K Dash. Comprehensive study on applications of artificial neural network in food process modeling. *Crit. Rev. Food Sci. Nutr.*, 62(10): 2756–2783, 2022.

[126] Julie Frost Dahl, Miek Schlangen, Atze Jan van der Goot, and Milena Corredig. Predicting rheological parameters of food biopolymer mixtures using machine learning. *Food Hydrocoll.*, page 110786, 2024.

[127] Sezin Kircali Ata, Jing K Shi, Xuesi Yao, Xin Yi Hua, Sumanto Haldar, Jie Hong Chiang, and Min Wu. Predicting the textural properties of plant-based meat analogs with machine learning. *Foods*, 12(2):344, 2023.

[128] FH Fan, Q Ma, J Ge, QY Peng, William W Riley, and SZ Tang. Prediction of texture characteristics from extrusion food surface images using a computer vision system and artificial neural networks. *J. Food Eng.*, 118(4):426–433, 2013.

[129] Qi Wu, Xun Zhang, Fei Gao, and Min Wu. Study on the residence time and texture prediction of pea protein extrusion based on image analysis. *Foods*, 12(24):4408, 2023.

[130] Ziyi Yang, Kai Wang, Xiaofeng Yuan, Yalin Wang, Chunhua Yang, Weihua Gui, Lingjian Ye, and Feifan Shen. Gaussian-based interval-aware transformer with interval embedding for data sequence modeling with irregular sampling frequency in industrial processes. *IEEE Trans. Ind. Informat.*, 2025.

[131] Nooshin Bahador, Denzil Ferreira, Satu Tamminen, Jukka Kortelainen, et al. Deep learning–based multimodal data fusion: Case study in food intake episodes detection using wearable sensors. *JMIR*, 9(1):e21926, 2021.

[132] WANG Zongxian, JIA Chaoyang, HE Weizhong, FENG Huanhuan, et al. Integrated multi-modal flexible sensors and ai-driven fusion modeling for internal and external quality detection of agricultural products. *Trends Food Sci. Technol.*, page 105401, 2025.

[133] Dana Lahat, Tülay Adali, and Christian Jutten. Multimodal data fusion: an overview of methods, challenges, and prospects. *Proceedings of the IEEE*, 103(9):1449–1477, 2015.

[134] Delaney Webb, Blake J Plattner, Emily Donald, Danielle Funk, Brian S Plattner, and Sajid Alavi. Role of chickpea flour in texturization of extruded pea protein. *J. Food Sci.*, 85(12):4180–4187, 2020.

[135] Hasitha Eranda Bandara, Susantha Wijesinghe, and Manjula Wickramasinghe. Camera self-triggering mechanism for optimal image capturing. *ECS Trans.*, 107(1):1467, 2022.

[136] Bojana Bajic, Nikola Suzic, Slobodan Moraca, Miladin Stefanović, Milos Jovicic, and Aleksandar Rikalovic. Edge computing data optimization for smart quality management: Industry 5.0 perspective. *Sustain.*, 15(7):6032, 2023.

[137] Luis MC Oliveira, Rafael Dias, Carine M Rebello, Márcio AF Martins, Alírio E Rodrigues, Ana M Ribeiro, and Idelfonso BR Nogueira. Artificial intelligence and cyber-physical systems: A review and perspectives for the future in the chemical industry. *AI*, 2(3):27, 2021.

[138] A. Fisher, C. A. Jacobson, E. A. Lee, R. M. Murray, A. Sangiovanni-Vincentelli, and E. Scholte. Industrial cyber-physical systems – iCyPhy. In *Proceedings of the Fourth International Conference on Complex Systems Design & Management (CSD&M)*, pages 21–37. Springer, 2014.

[139] Jiří Jaromír Klemeš, Petar Sabev Varbanov, Sharifah Rafidah Wan Alwi, Zainuddin Abdul Manan, Yee Van Fan, and Hon Huin Chin. *Sustainable process integration and intensification: saving energy, water and resources*. De Gruyter, 2023.

[140] Kaixing Huang, Chunjie Zhou, Yu-Chu Tian, Shuanghua Yang, and Yuanqing Qin. Assessing the physical impact of cyberattacks on industrial cyber-physical systems. *IEEE Transactions on Industrial Electronics*, 65(10):8153–8162, 2018.

[141] W. Duo, M. Zhou, and A. Abusorrah. A survey of cyberattacks on cyber physical systems: Recent advances and challenges. *IEEE/CAA Journal of Automatica Sinica*, 9(5):784–800, 2022.

[142] N. Mtukushe, A. K. Onaolapo, A. Aluko, and D. G. Dorrell. Review of cyberattack implementation, detection, and mitigation methods in cyber-physical systems. *Energies*, 16(13):5206, 2023.

[143] Y. A. Kadakia, A. Suryavanshi, A. Alnajdi, F. Abdullah, and P. D. Christofides. Integrating machine learning detection and encrypted control for enhanced cybersecurity of nonlinear processes. *Computers and Chemical Engineering*, 180:108498, 2024.

[144] Z. Wu and P. D. Christofides. *Process Operational Safety and Cybersecurity*. Springer, 2021.

[145] A. A. Cárdenas, S. Amin, Z.-S. Lin, Y.-L. Huang, C.-Y. Huang, and S. Sastry. Attacks against process control systems: Risk assessment, detection, and response. In *Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security (ASIACCS)*, pages 355–366, 2011.

[146] C. Murguia and J. Ruths. CUSUM and chi-squared attack detection of compromised sensors. In *Proceedings of the 2016 IEEE Conference on Control Applications (CCA)*, pages 474–480, 2016.

[147] D. I. Urbina, J. A. Giraldo, A. A. Cárdenas, N. O. Tippenhauer, J. Valente, M. Faisal, J. Ruths, R. Candell, and H. Sandberg. Limiting the impact of stealthy attacks on industrial control systems. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pages 1092–1105, 2016.

[148] H. Fawzi, P. Tabuada, and S. Diggavi. Secure estimation and control for cyber-physical systems under adversarial attacks. *IEEE Transactions on Automatic Control*, 59(6): 1454–1467, 2014.

[149] F. Pasqualetti, F. Dörfler, and F. Bullo. Control-theoretic methods for cyberphysical security: Geometric principles for optimal cross-layer resilient control systems. *IEEE Control Systems Magazine*, 35(1):110–127, 2015.

[150] Q. Zhu and T. Başar. Game-theoretic methods for robustness, security, and resilience of cyberphysical control systems: Games-in-games principle for optimal cross-layer resilient control systems. *IEEE Control Systems Magazine*, 35(1):46–65, 2015.

[151] Y.-C. Sun and G.-H. Yang. Robust event-triggered model predictive control for cyber-physical systems under denial-of-service attacks. *International Journal of Robust and Nonlinear Control*, 29(14):4797–4811, 2019.

[152] S. Narasimhan, N. H. El-Farra, and M. J. Ellis. A set-based control mode selection approach for active detection of false data injection cyberattacks. In *Proceedings of the American Control Conference (ACC)*, pages 1726–1731. IEEE, 2024.

[153] S. Narasimhan, M. J. Ellis, and N. H. El-Farra. Detection of multiplicative false data injection cyberattacks on process control systems via randomized control mode switching. *Processes*, 12(2):327, 2024.

[154] S. Pequito, F. Khorrami, P. Krishnamurthy, and G. J. Pappas. Analysis and design of actuation–sensing–communication interconnection structures toward secured/resilient LTI closed-loop systems. *IEEE Transactions on Control of Network Systems*, 6(2): 667–678, 2018.

[155] H. Oyama and H. Durand. Integrated cyberattack detection and resilient control strategies using Lyapunov-based economic model predictive control. *AIChE Journal*, 66(12):e17084, 2020.

[156] K. K. Rangan, H. Oyama, and H. Durand. Integrated cyberattack detection and handling for nonlinear systems with evolving process dynamics under Lyapunov-based economic model predictive control. *Chemical Engineering Research and Design*, 170: 147–179, 2021.

[157] Scarlett Chen, Zhe Wu, and Panagiotis D Christofides. Cyber-security of centralized, decentralized, and distributed control-detector architectures for nonlinear processes. *Chemical Engineering Research and Design*, 165:25–39, 2021.

[158] H. Oyama, K. K. Rangan, and H. Durand. Handling of stealthy sensor and actuator cyberattacks on evolving nonlinear process systems. *Journal of Advanced Manufacturing and Processing*, 3(3):e10099, 2021.

[159] G. Wu, H. Zhang, W. Wu, Y. Wang, and Z. Wu. Physics-informed graph convolutional recurrent network for cyberattack detection in chemical process networks. *Industrial and Engineering Chemistry Research*, 64(6):3370–3382, 2025.

[160] G. Wu, Y. Wang, and Z. Wu. Physics-informed machine learning in cyberattack detection and resilient control of chemical processes. *Chemical Engineering Research and Design*, 204:544–555, 2024.

[161] Z. Wu, S. Chen, D. Rincon, and P. D. Christofides. Post cyberattack state reconstruction for nonlinear processes using machine learning. *Chemical Engineering Research and Design*, 159:248–261, 2020.

[162] E. Khlebnikova, A. Bekker, E. Ivashkina, I. Dolganova, and E. Yurev. Thermodynamic analysis of benzene alkylation with ethylene. *Procedia Chemistry*, 15:42–48, 2015.

[163] A. Ebrahimi and D. B. Pourkargar. An adaptive distributed architecture for multi-agent state estimation and control of complex process systems. *Chemical Engineering Research and Design*, 210:594–604, 2024.

[164] Davood B Pourkargar, Manjiri Moharir, Ali Almansoori, and Prodromos Daoutidis. Distributed estimation and nonlinear model predictive control using community detection. *Industrial & Engineering Chemistry Research*, 58(30):13495–13507, 2019.

[165] A. Bagheri, A. Ebrahimi, and D. B. Pourkargar. Self-attention transformer architectures for cyberattack detection and secure state reconstruction of integrated chemical process systems. *Chemical Engineering Research and Design*, 220:162–179, 2025.

[166] Aida Ali, Siti Mariyam Shamsuddin, and Anca L Ralescu. Classification with class imbalance problem. *Int. J. Advance Soft Compu. Appl*, 5(3):176–204, 2013.

[167] André CPLF de Carvalho and Alex A Freitas. A tutorial on multi-label classification techniques. *Foundations of Computational Intelligence Volume 5: Function Approximation and Classification*, pages 177–195, 2009.

[168] Adane Nega Tarekegn, Mario Giacobini, and Krzysztof Michalak. A review of methods for imbalanced multi-label classification. *Pattern Recognition*, 118:107965, 2021.

[169] Niranjan Sitapure and Joseph Sang-Il Kwon. Exploring the potential of time-series transformers for process modeling and control in chemical systems: An inevitable paradigm shift? *Chemical Engineering Research and Design*, 194:461–475, 2023. doi:10.1016/j.cherd.2023.04.028.

[170] George Zerveas, Srideepika Jayaraman, Dhaval Patel, Anuradha Bhamidipaty, and Carsten Eickhoff. A transformer-based framework for multivariate time series representation learning. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 2114–2124, 2021.

[171] Junho Park, Mohammad Reza Babaei, Samuel Arce Munoz, Ashwin N Venkat, and John D Hedengren. Simultaneous multistep transformer architecture for model predictive control. *Computers and Chemical Engineering*, 178:108396, 2023.

[172] Ilvico Sonata and Yaya Heryadi. Comparison of LSTM and transformer for time series data forecasting. In *7th International Conference on Informatics and Computational Sciences (ICICoS)*, pages 491–495. IEEE, 2024.

[173] Sabeen Ahmed, Ian E Nielsen, Aakash Tripathi, Shamoon Siddiqui, Ravi P Ramachandran, and Ghulam Rasool. Transformers in time-series analysis: A tutorial. *Circuits, Systems, and Signal Processing*, 42(12):7433–7466, 2023.

[174] Maria Isabel Ribeiro. Kalman and extended kalman filters: Concept, derivation and properties. *Institute for Systems and Robotics*, 43(46):3736–3741, 2004.

[175] Felix Fiedler, Benjamin Karg, Lukas Lüken, Dean Brandner, Moritz Heinlein, Felix

Brabender, and Sergio Lucia. do-mpc: Towards FAIR nonlinear and robust model predictive control. *Control Engineering Practice*, 140:105676, 2023.

[176] Bin Liu, Konstantinos Blekas, and Grigorios Tsoumakas. Multi-label sampling based on local label imbalance. *Pattern Recognition*, 122:108294, 2022.

[177] Nikhil Ketkar. Stochastic gradient descent. In *Deep learning with Python: A hands-on introduction*, pages 113–132. Springer, 2017.

[178] Yuan Yao, Lorenzo Rosasco, and Andrea Caponnetto. On early stopping in gradient descent learning. *Constructive Approximation*, 26(2):289–315, 2007.

[179] Nitish Srivastava. Improving neural networks with dropout. *University of Toronto*, 182(566):7, 2013.

Appendix A

Appendix A: Supplementary Figures for Chapter 6

A.1 Confusion Matrices for Proposed Frameworks

Figures A.1 and A.2 present the confusion matrices corresponding to the detection and identification performance of the centralized ETE decoder-only transformer and the modular Two-Tier framework, respectively, across all five temperature sensors (T_1 to T_5). In the ETE framework (Figure A.1), both detection and sensor-level identification exhibit generally balanced classification performance. However, a relatively high false-positive rate is observed, suggesting that the model tends to overpredict compromised states, particularly in complex multi-sensor attack scenarios.

Conversely, the two-tier framework (Figure A.2) demonstrates superior detection accuracy, as indicated by the near absence of false negatives in the detection matrix. This suggests that the binary classifier in the first tier effectively captures almost all attack events, thereby allowing the second-tier model to focus solely on sensor-level fault localization. Although this framework appears more conservative in specific sensor identifications, it achieves improved precision and robustness under highly dynamic operating conditions. The modular architecture of the two-tier approach enhances interpretability and reliability in real-time

monitoring and decision-making.

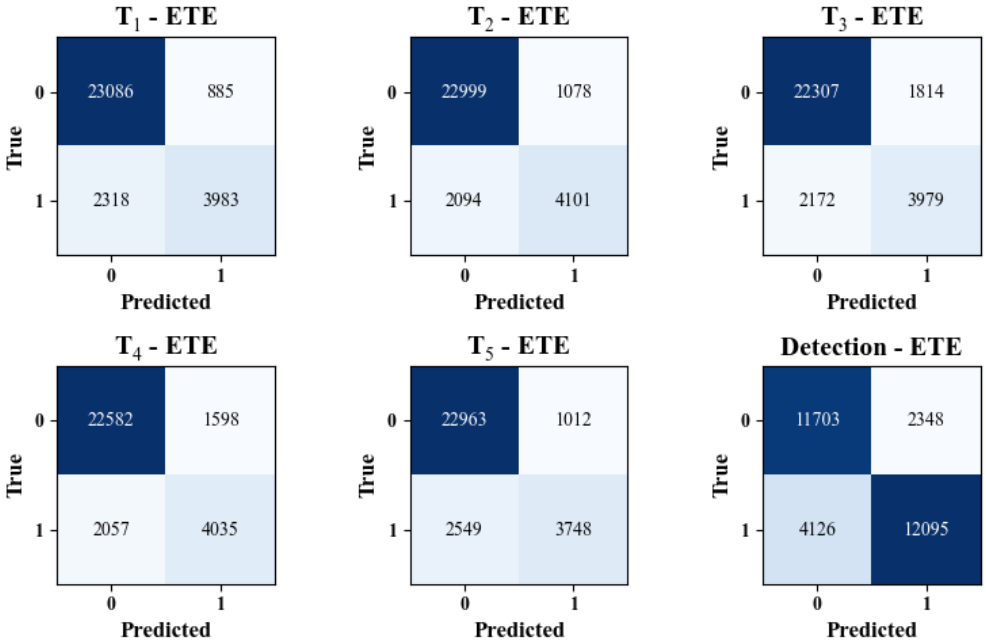

Figure A.1: *Confusion matrix using CETE transformer.*

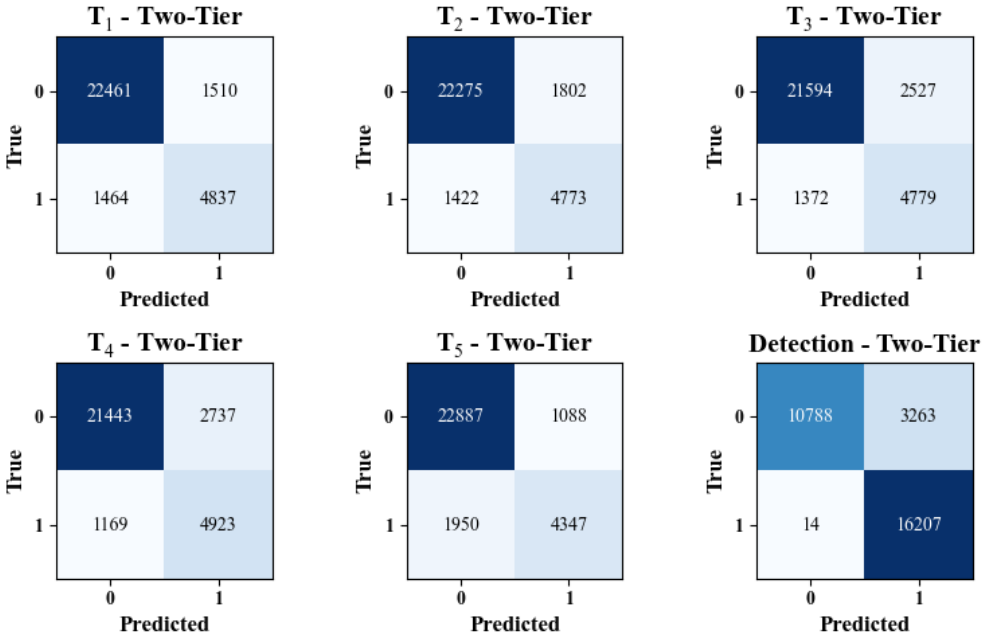

Figure A.2: *Confusion matrix using C2tier framework.*

Figures A.3 and A.4 present the sensor-level and network-level confusion matrices for the distributed DETE and two-tier architectures, respectively. At the network detection level,

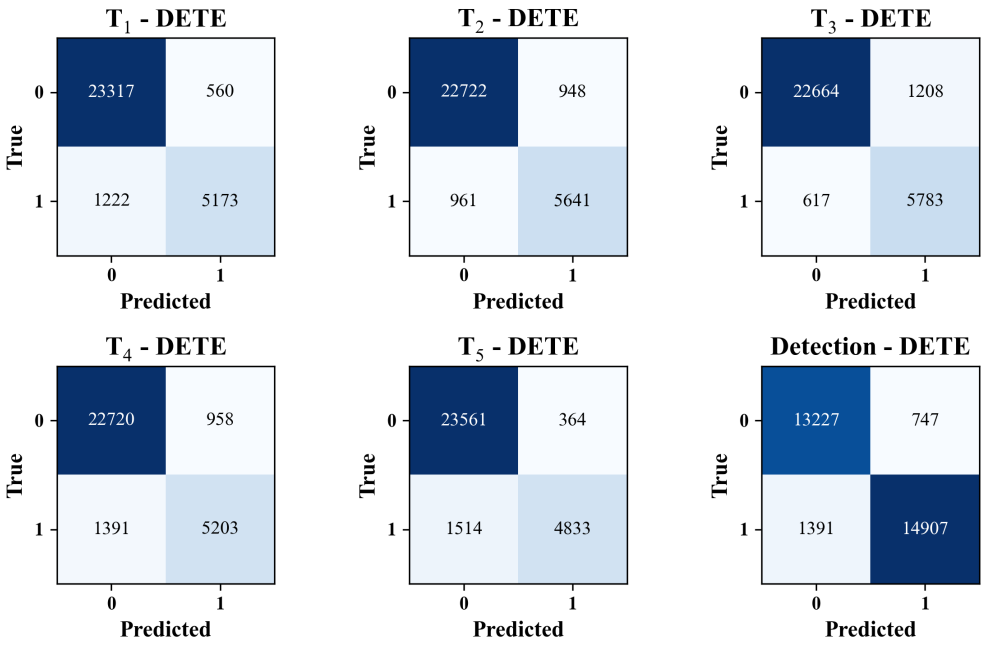

Figure A.3: *Network-level confusion matrix for the DETE framework.*

the two-tier model exhibits noticeably fewer false negatives than DETE, consistent with the strong recall reported in the metric tables. However, this increased sensitivity comes with substantially more false positives, indicating a more aggressive detection strategy. In contrast, DETE produces significantly fewer false positives while maintaining strong true positive counts, resulting in more balanced detection decisions at the network scale.

At the sensor-identification level, both architectures exhibit strong true negative dominance, primarily due to the inherent class imbalance under nominal operating conditions. For sensors T_2, T_3, and T_4, DETE generally reduces false positives compared with the two-tier model, indicating greater selectivity in assigning attacks to specific sensors. Conversely, the two-tier approach produces fewer false negatives for some sensors, suggesting higher sensitivity but also contributing to over-identification. Sensor T_5 exhibits similar behavior across both methods, indicating that attack patterns involving this sensor are consistently distinguishable. Overall, the confusion matrices align with the metric-based findings: the two-tier architecture prioritizes sensitivity, particularly at the detection stage, whereas DETE provides more balanced decision boundaries and limits false-alarm propagation. This distinction becomes more pronounced when sensor-level decisions are aggregated at the network level,

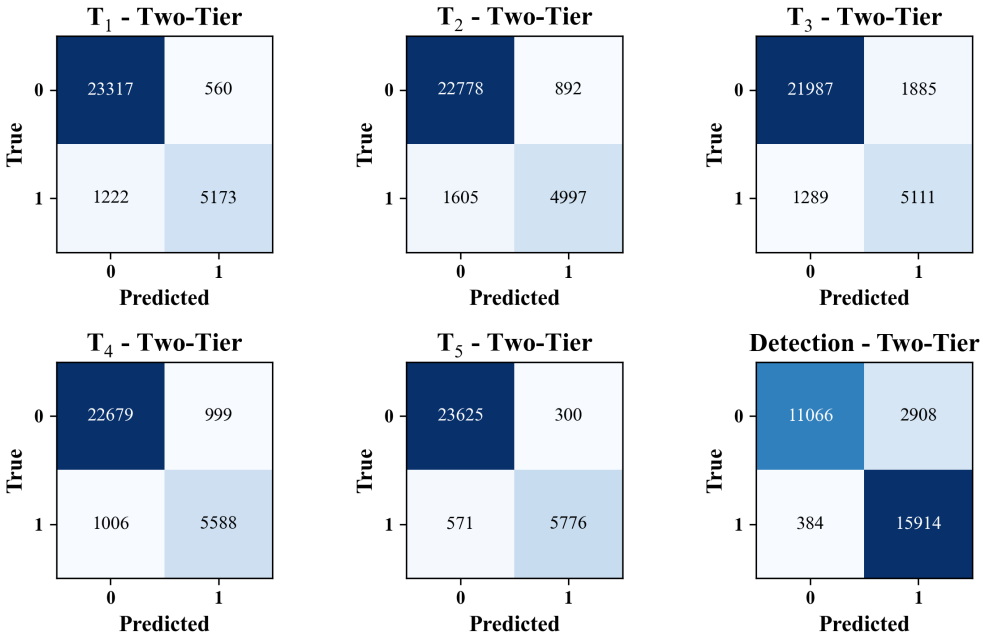

Figure A.4: *Network-level confusion matrix for the D2tier framework.*

where DETE maintains stronger overall stability.

A.2 Post-attack EKF-based state reconstruction

A.2.1 Centralized Methods

Following the discussions in Chapter 6, Figures A.5 - A.16 illustrate the performance of the extended Kalman filter (EKF)-based state reconstruction across various attack types—min-max, noise, and set difference—under both single- and multi-sensor compromise scenarios for the centralized methods. Each figure presents the normalized deviation from the system's steady state for the five temperature sensors (T_1-T_5) and the final product concentration (C_{C5}), comparing the baseline (normal), compromised (attacked), and EKF-reconstructed states. The reconstructed trajectories provide insight into the effectiveness of the proposed frameworks in restoring nominal system behavior following adversarial disruptions.

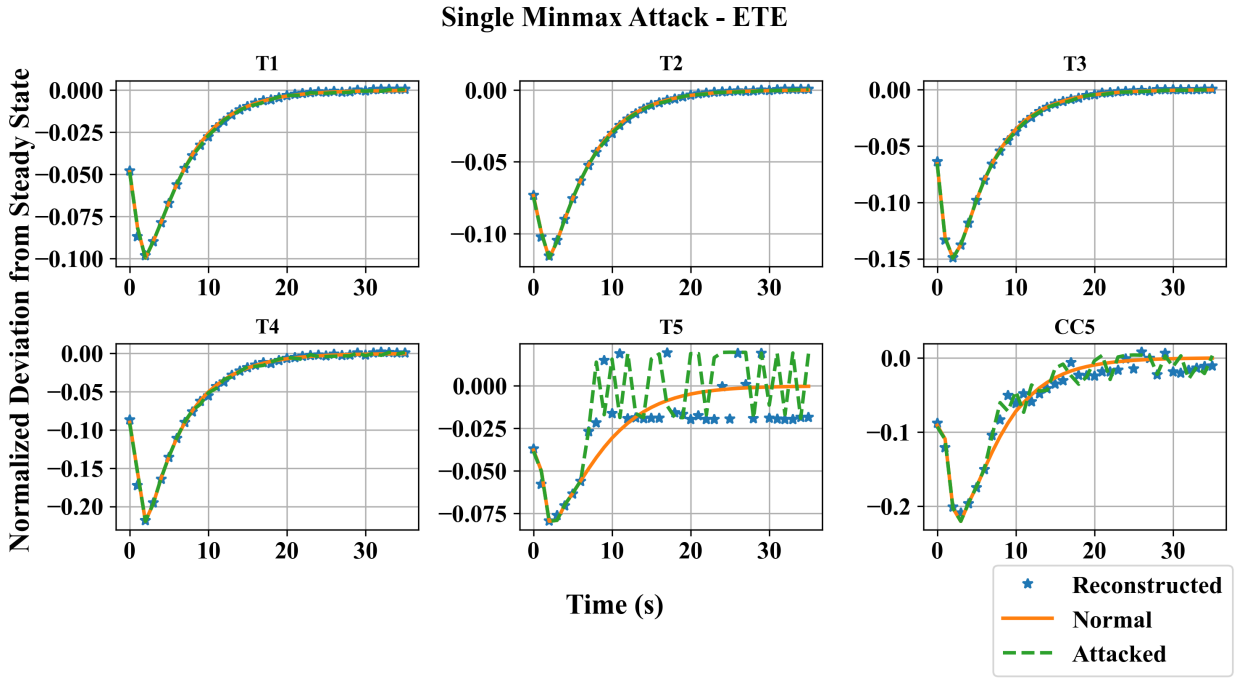

Figure A.5: *Closed-loop simulation results for detection and state reconstruction under single min-max attacks using the CETE transformer.*

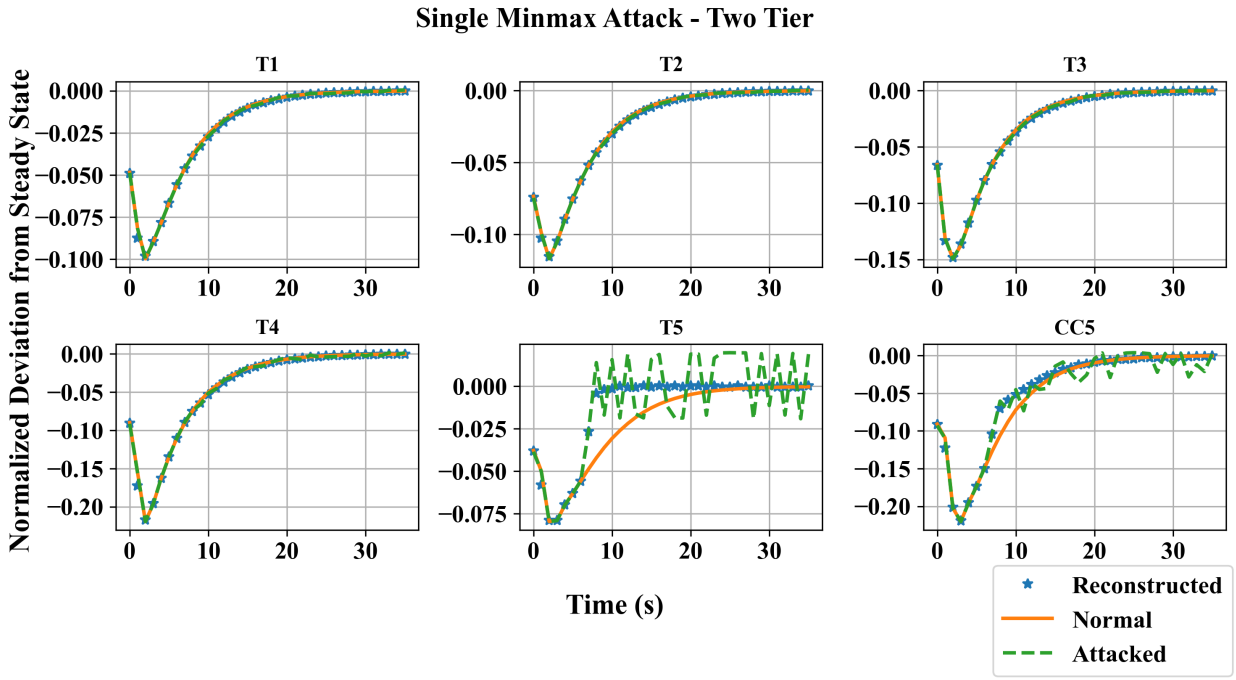

Figure A.6: *Closed-loop simulation results for detection and state reconstruction under single min-max attacks using the C2tier framework.*

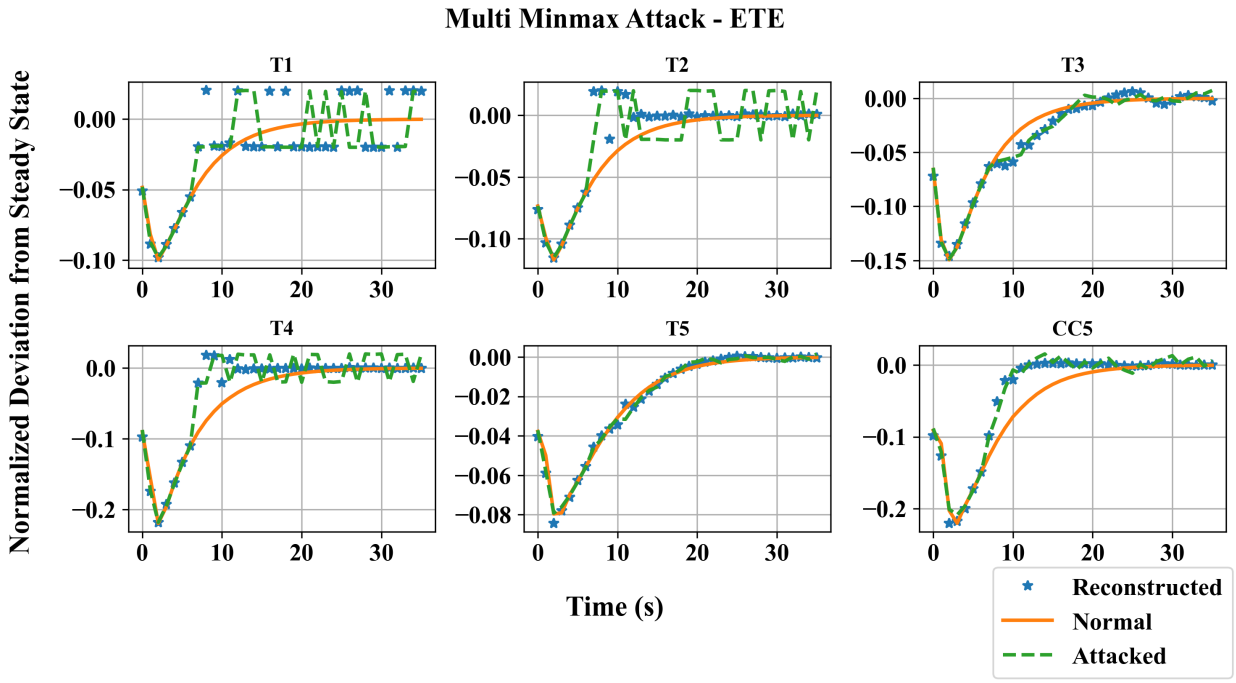

Figure A.7: *Closed-loop simulation results for detection and state reconstruction under multi min-max attacks using the CETE transformer.*

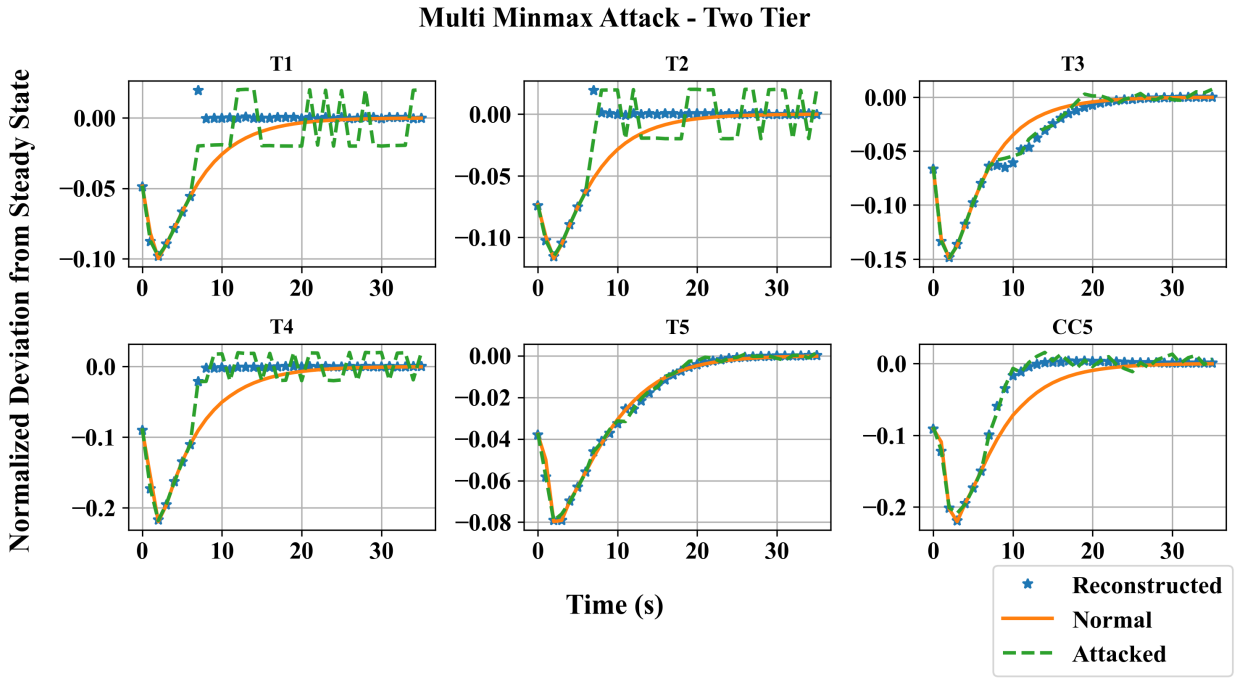

Figure A.8: *Closed-loop simulation results for detection and state reconstruction under multi min-max attacks using the C2tier framework.*

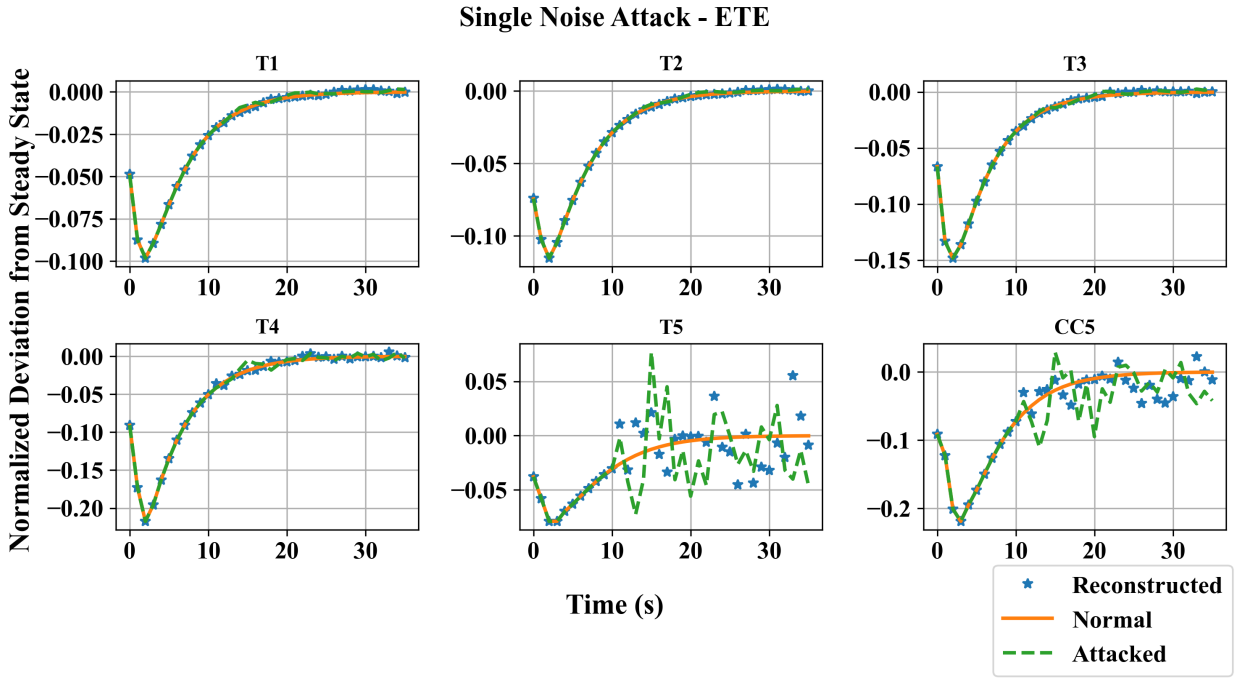

Figure A.9: *Closed-loop simulation results under single noise attacks using the CETE transformer.*

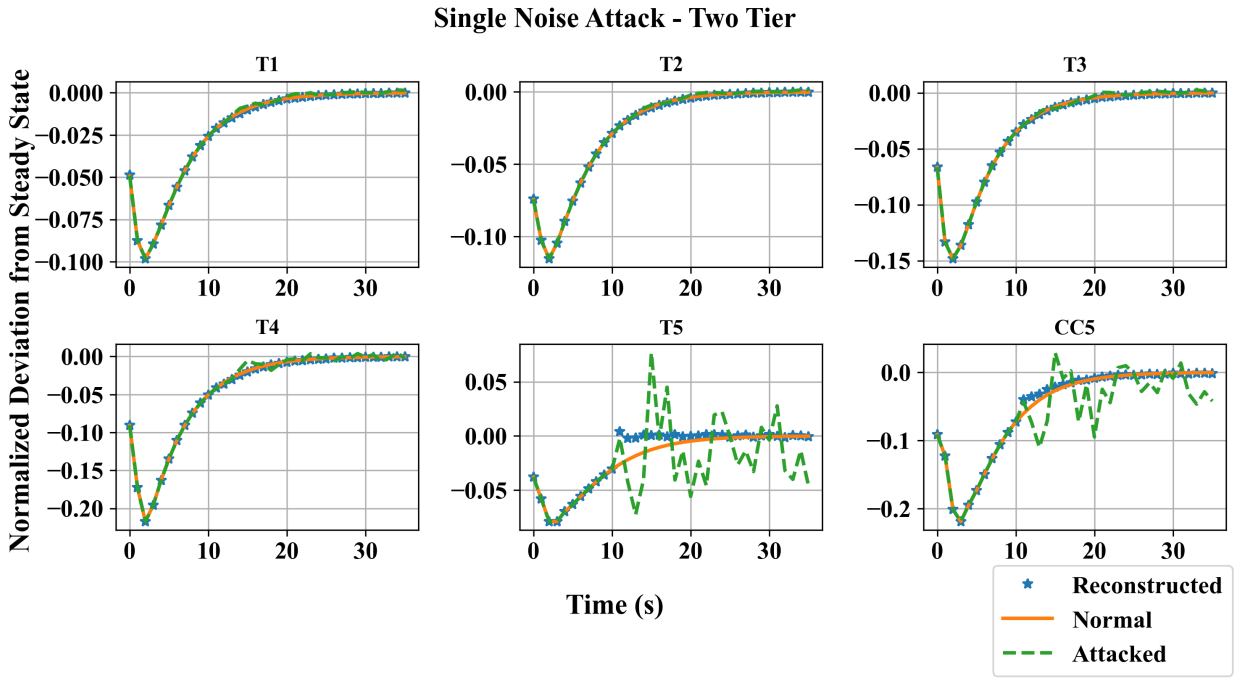

Figure A.10: *Closed-loop simulation results under single noise attacks using the C2tier framework.*

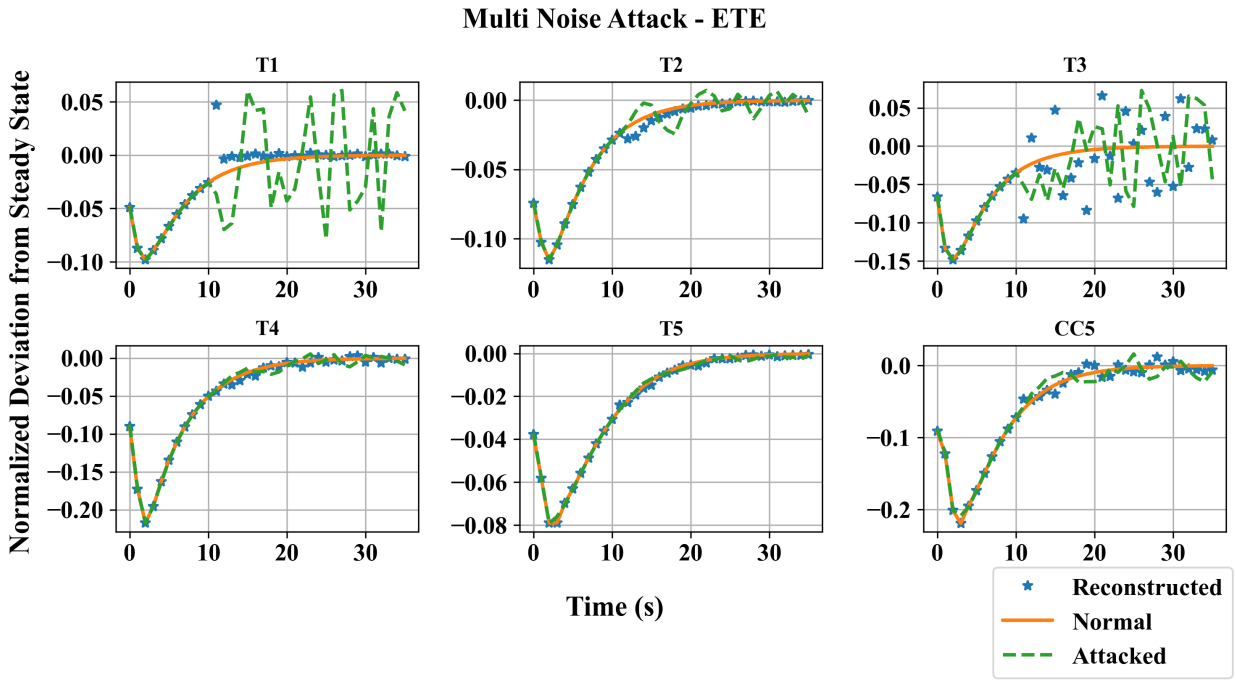

Figure A.11: *Closed-loop simulation results under multi-noise attacks using the CETE transformer.*

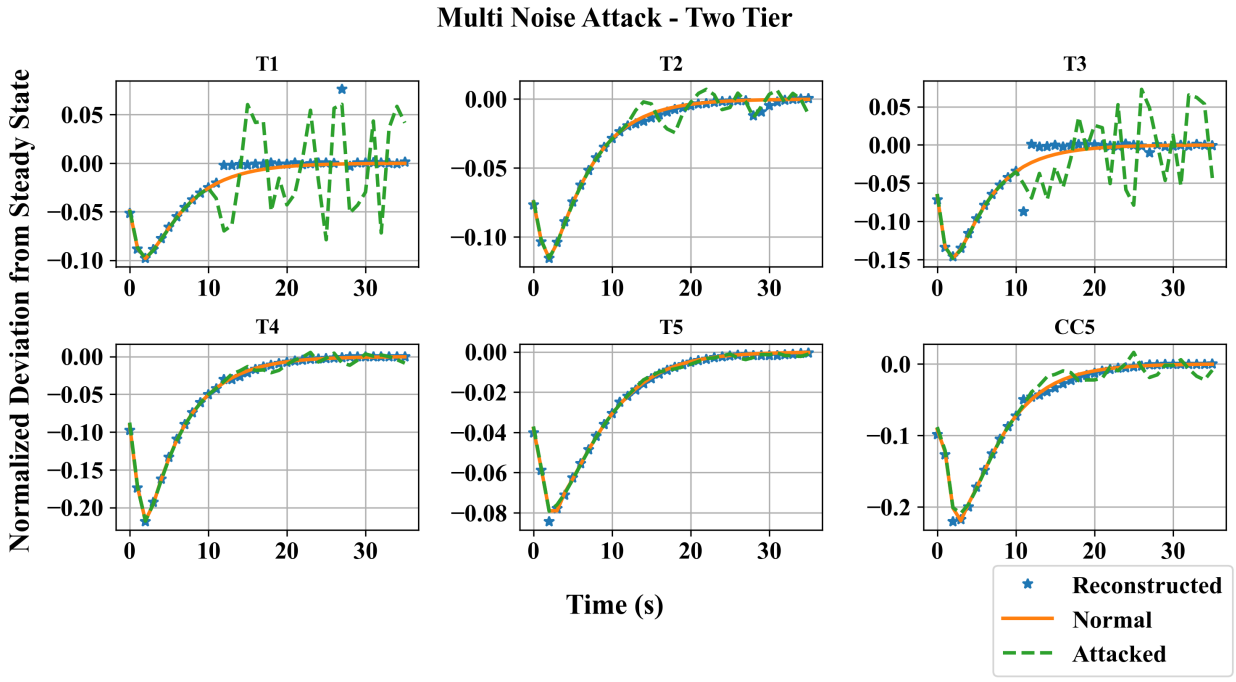

Figure A.12: *Closed-loop simulation results under multi-noise attacks using the C2tier framework.*

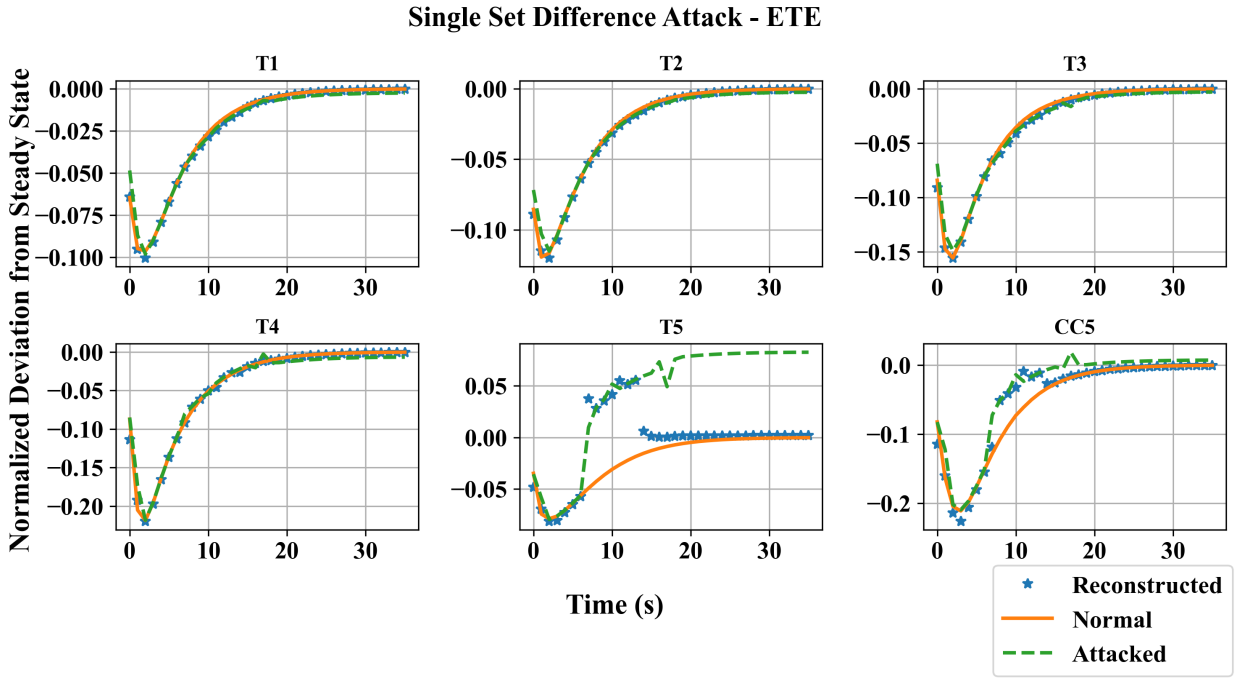

Figure A.13: *Closed-loop simulation results under single set difference attacks using the CETE transformer.*

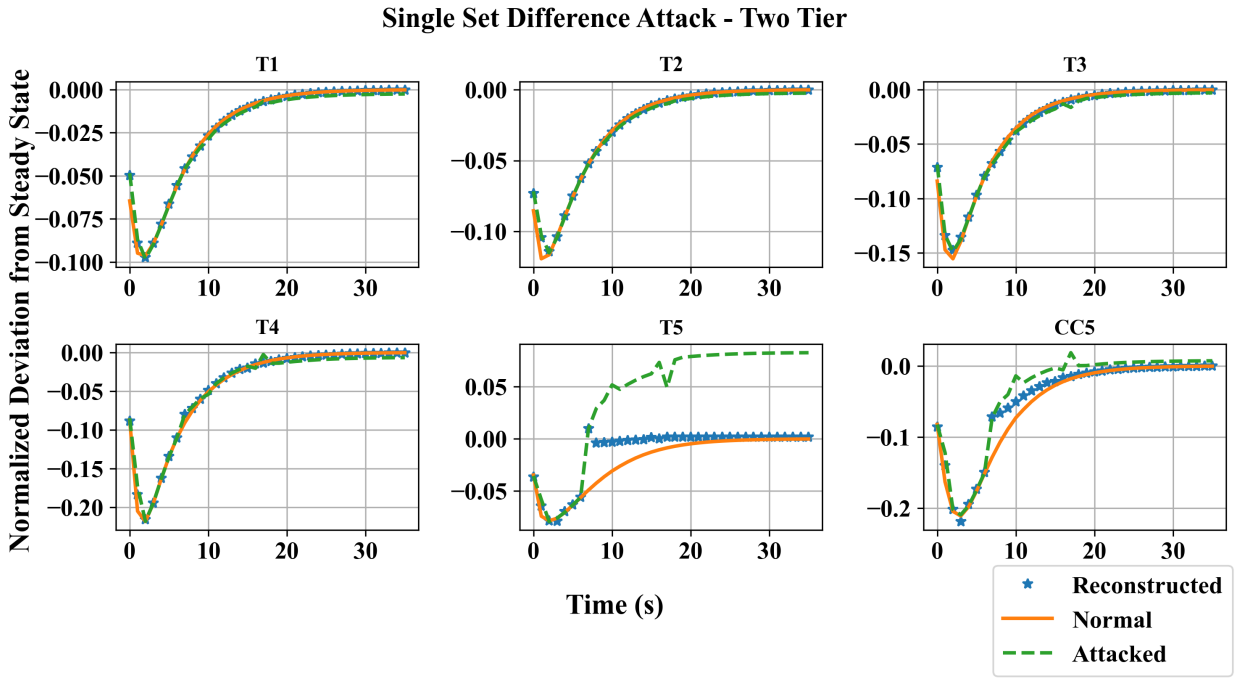

Figure A.14: *Closed-loop simulation results under single set difference attacks using the C2tier framework.*

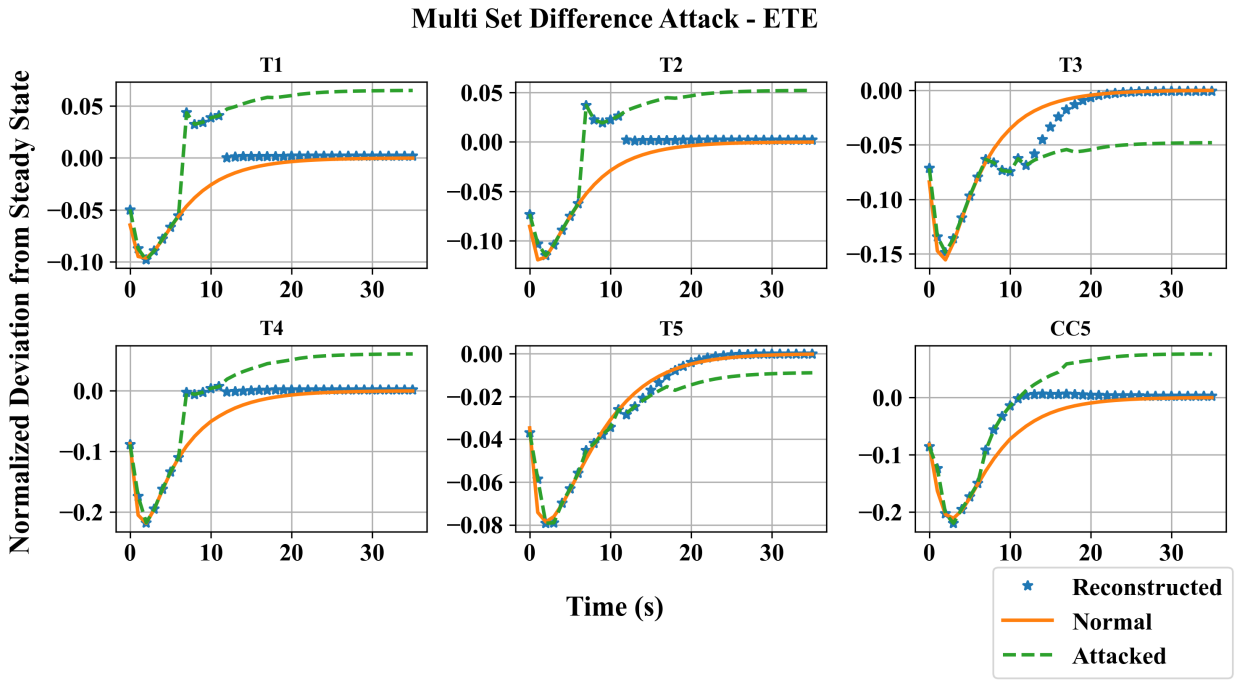

Figure A.15: *Closed-loop simulation results under multi-set difference attacks using the CETE transformer.*

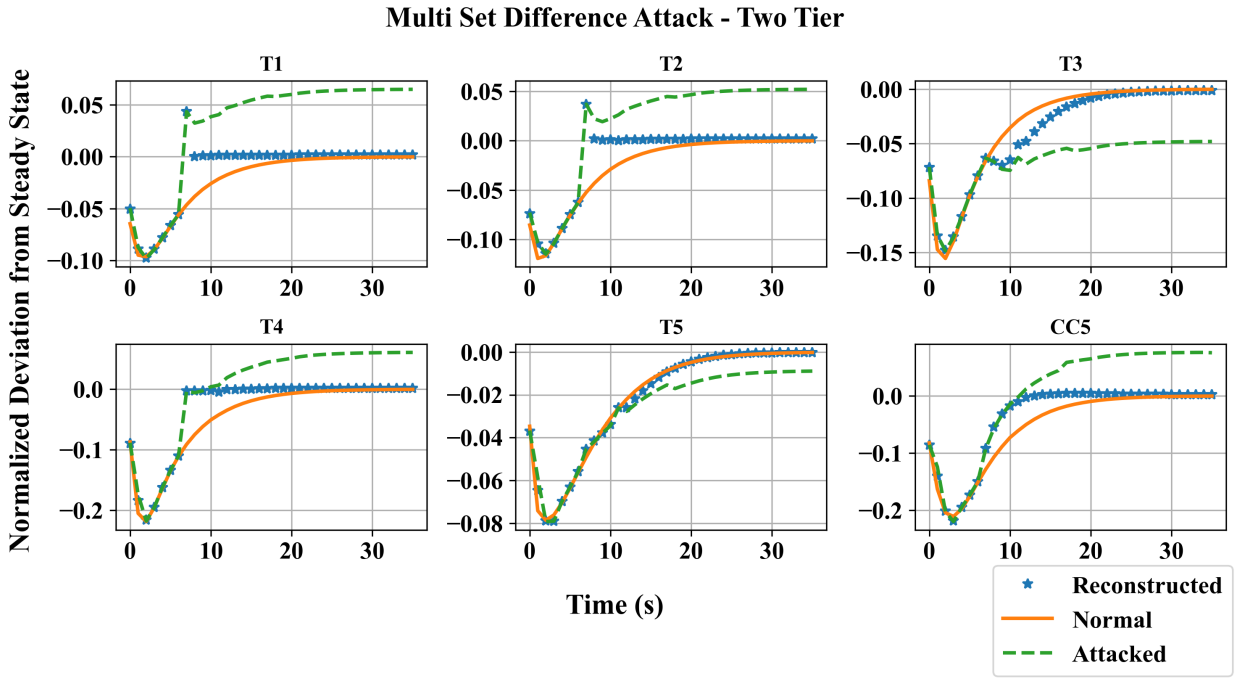

Figure A.16: *Closed-loop simulation results under multi-set difference attacks using the C2tier framework.*

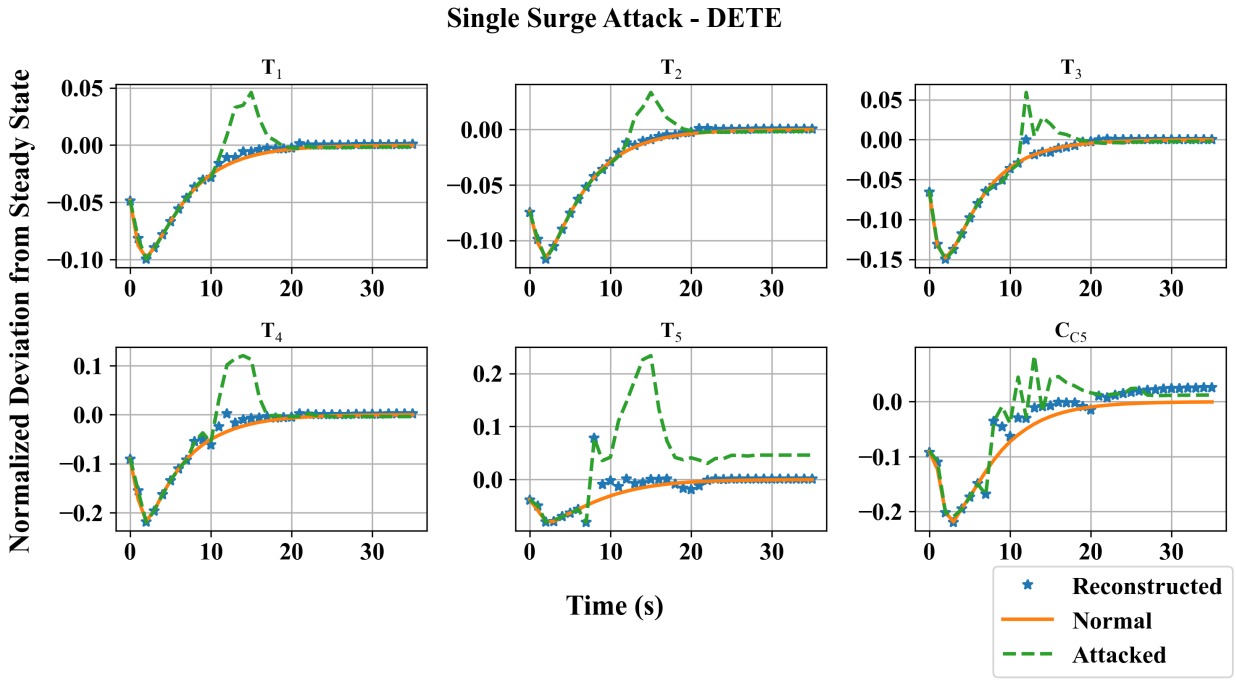

Figure A.17: *Closed-loop detection and reconstruction under a single surge attack using the DETE framework.*

A.2.2 Distributed Methods

Following the centralized methods, Figures A.17 - ?? present reconstruction results for the distributed methods discussed.

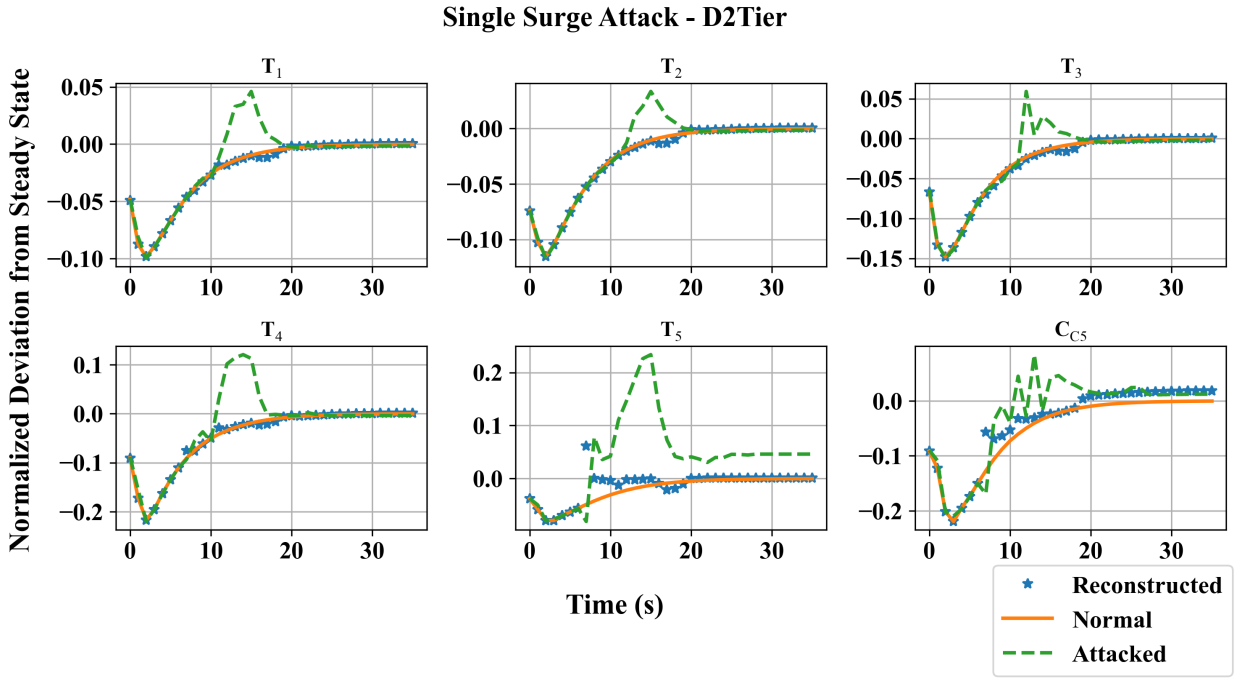

Figure A.18: *Closed-loop detection and reconstruction under a single surge attack using the D2tier framework.*

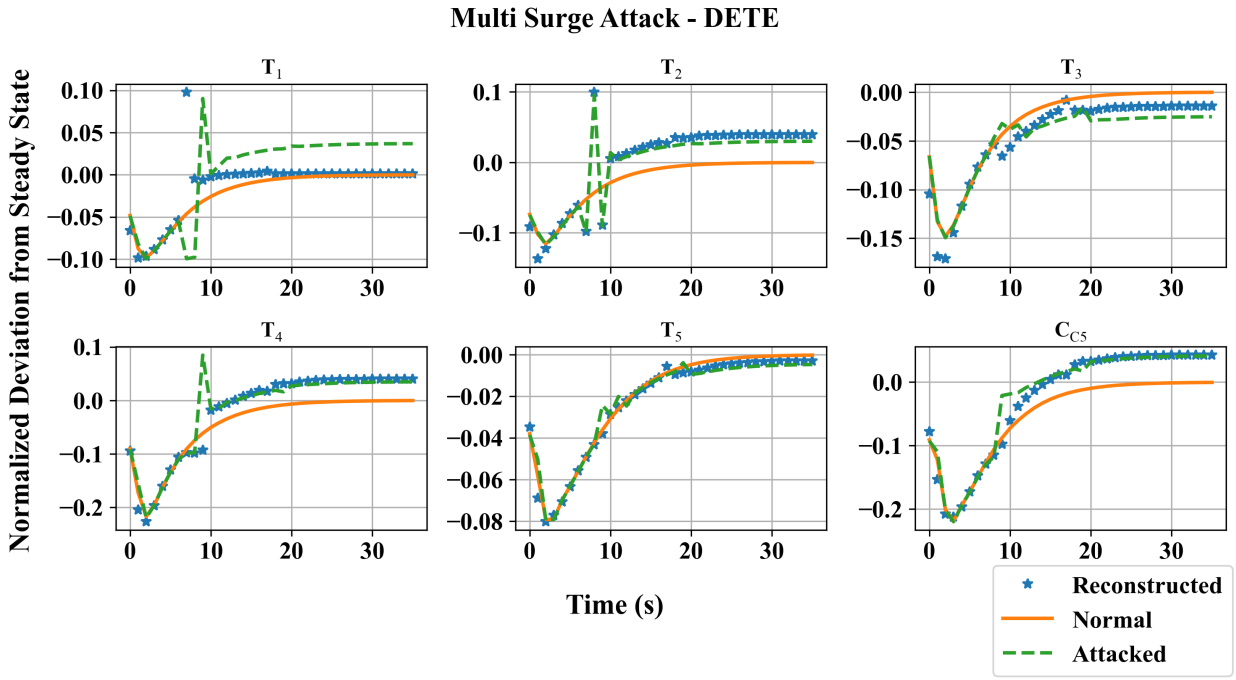

Figure A.19: *Closed-loop detection and reconstruction under a multi-surge attack using the DETE framework.*

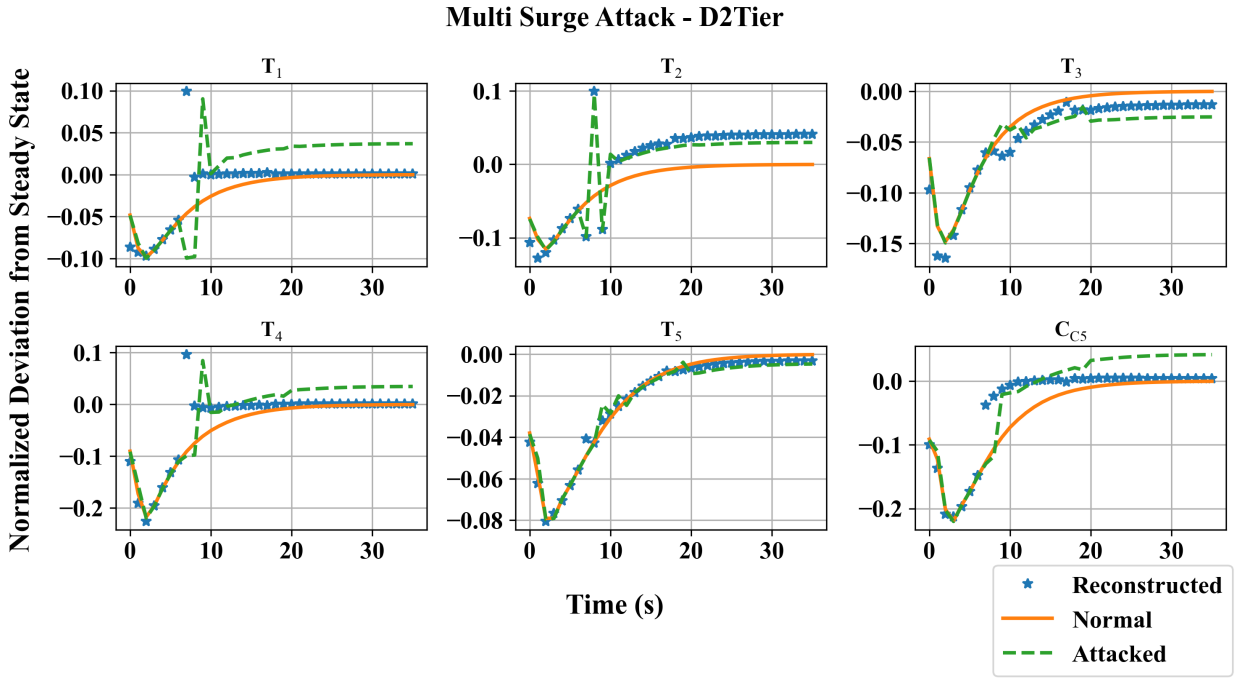

Figure A.20: *Closed-loop detection and reconstruction under a multi-surge attack using the D2tier framework.*

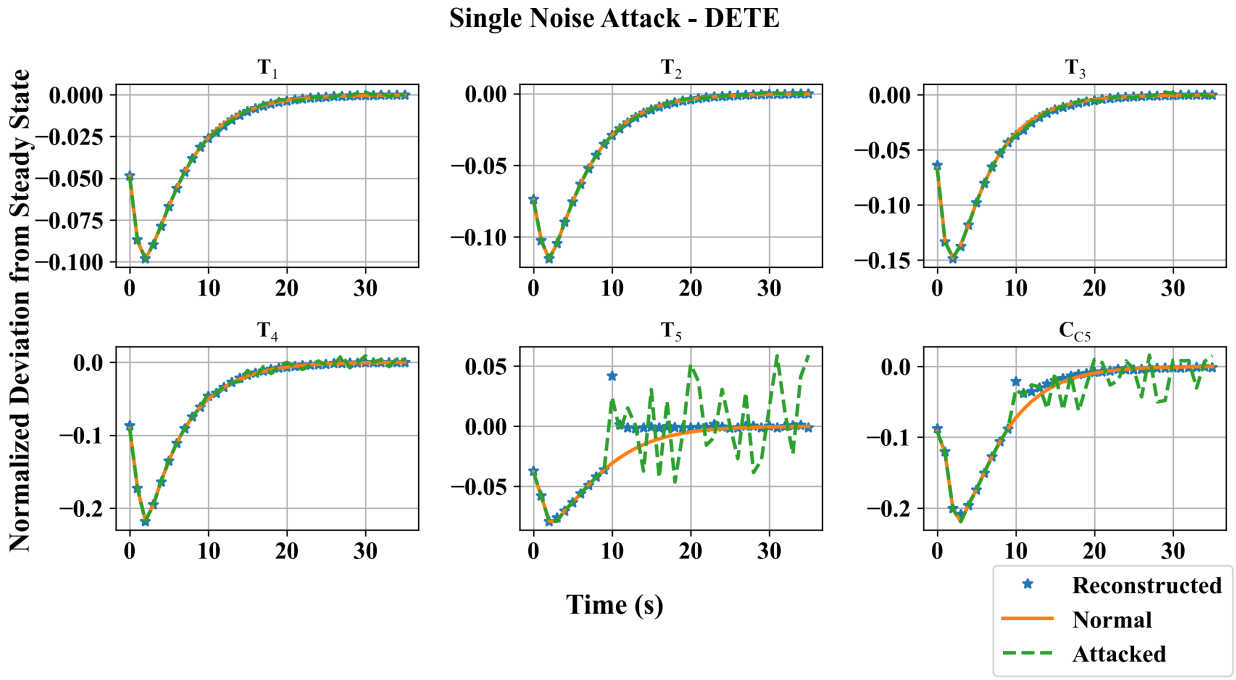

Figure A.21: *Closed-loop detection and reconstruction under a single noise attack using the DETE framework.*

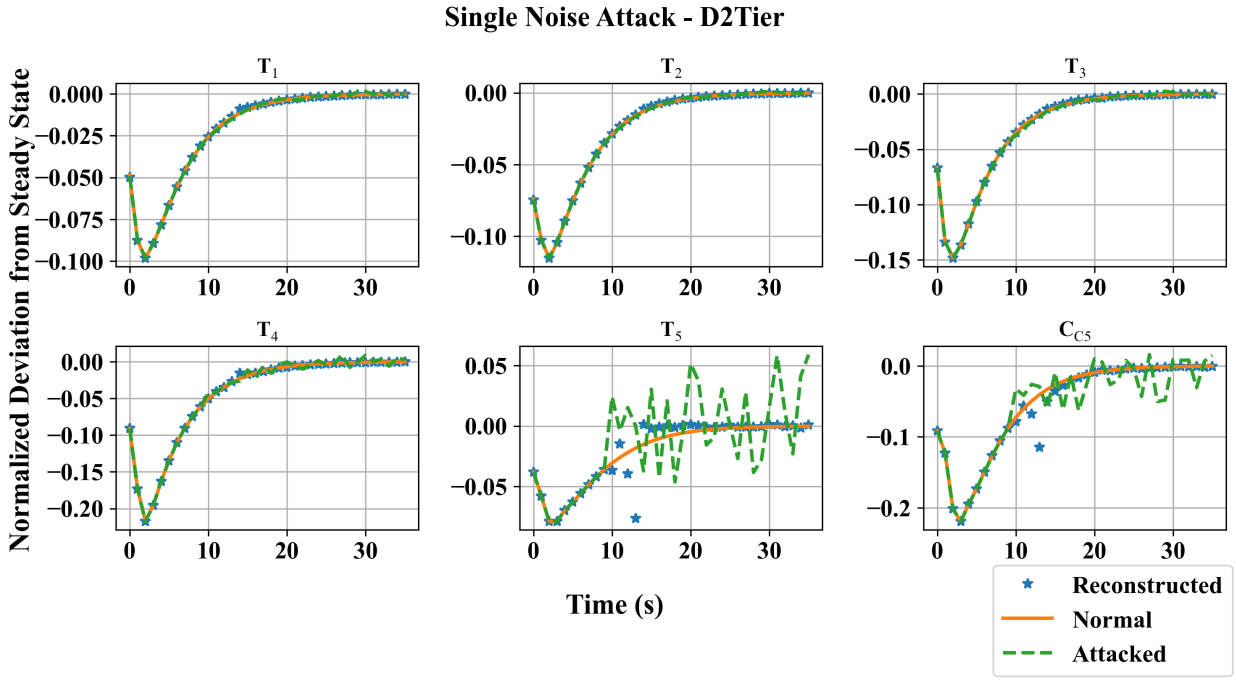

Figure A.22: *Closed-loop detection and reconstruction under a single noise attack using the D2tier framework.*

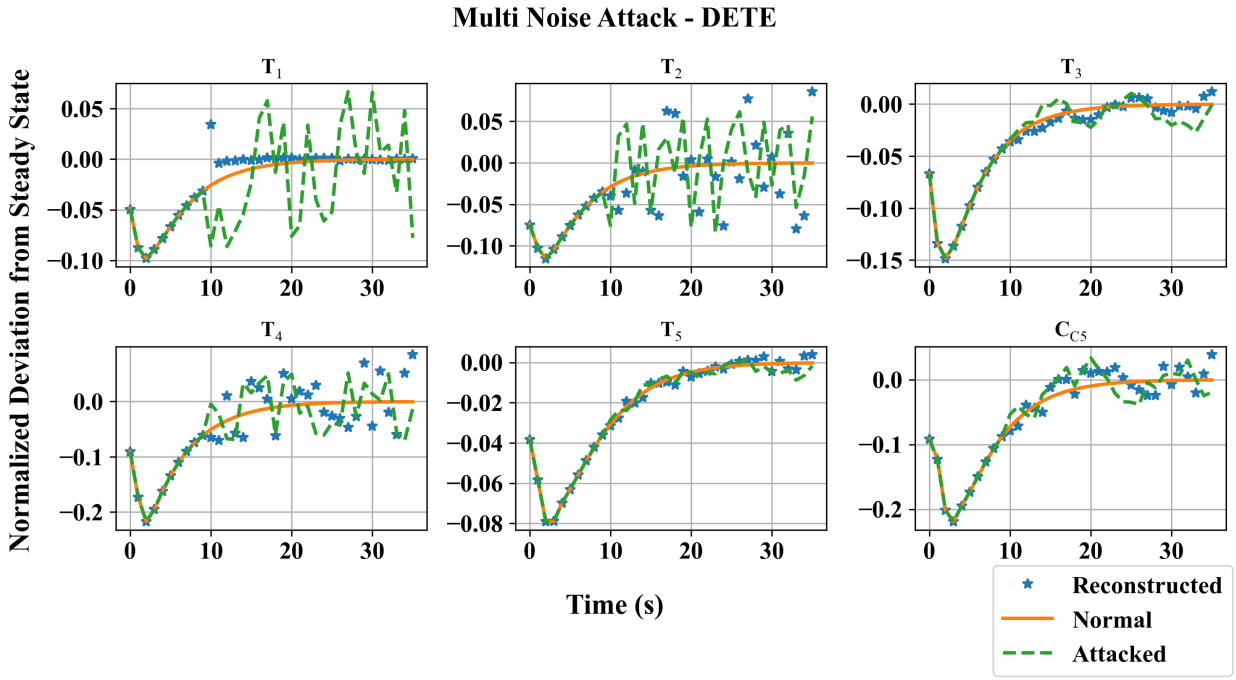

Figure A.23: *Closed-loop detection and reconstruction under a multi-noise attack using the DETE framework.*

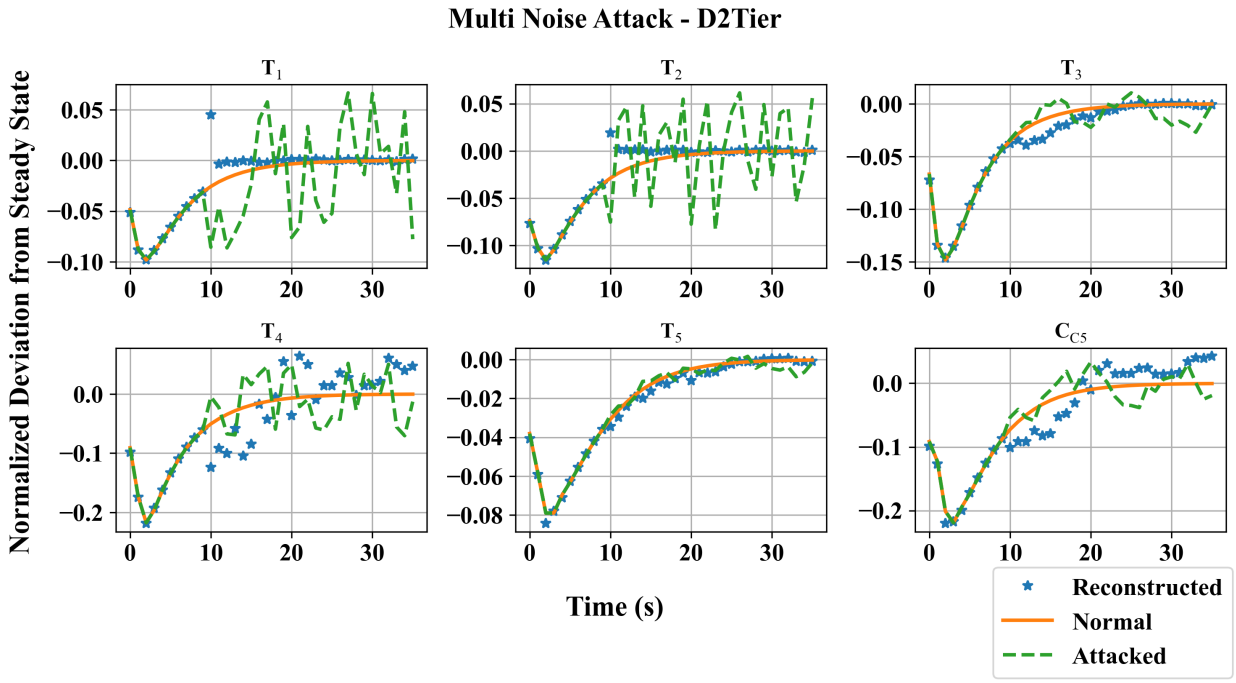

Figure A.24: *Closed-loop detection and reconstruction under a multi-noise attack using the D2tier framework.*

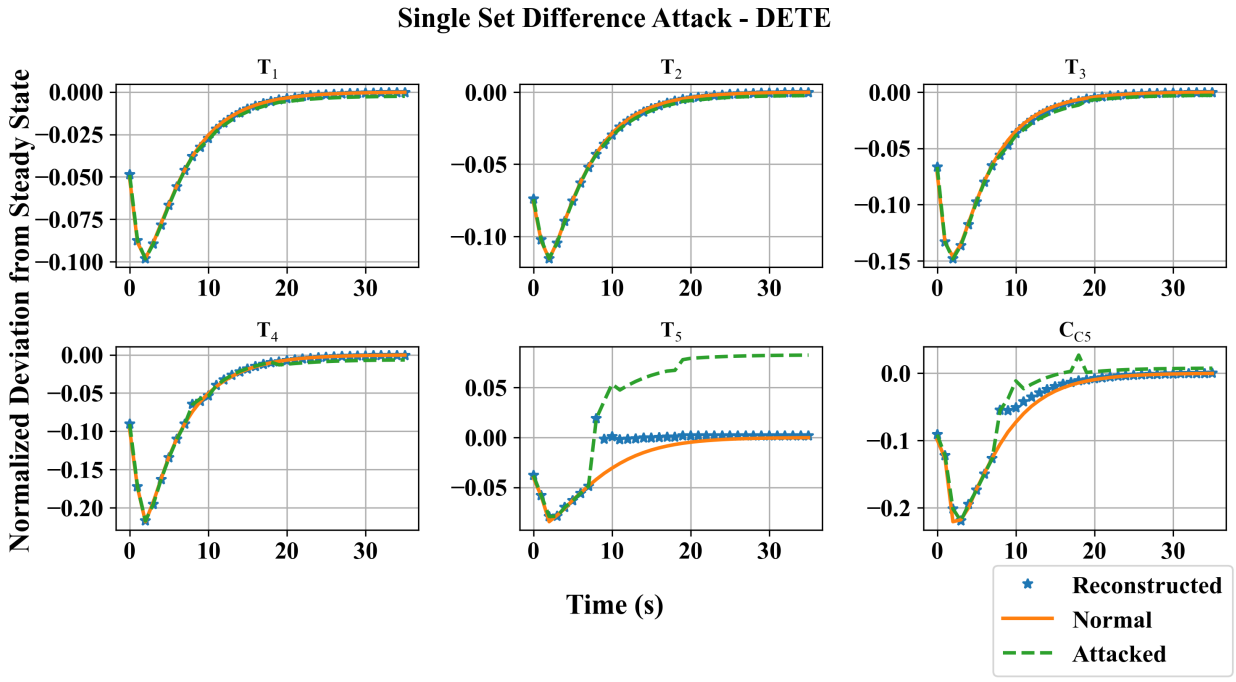

Figure A.25: *Closed-loop detection and reconstruction under a single set-difference attack using the DETE framework.*

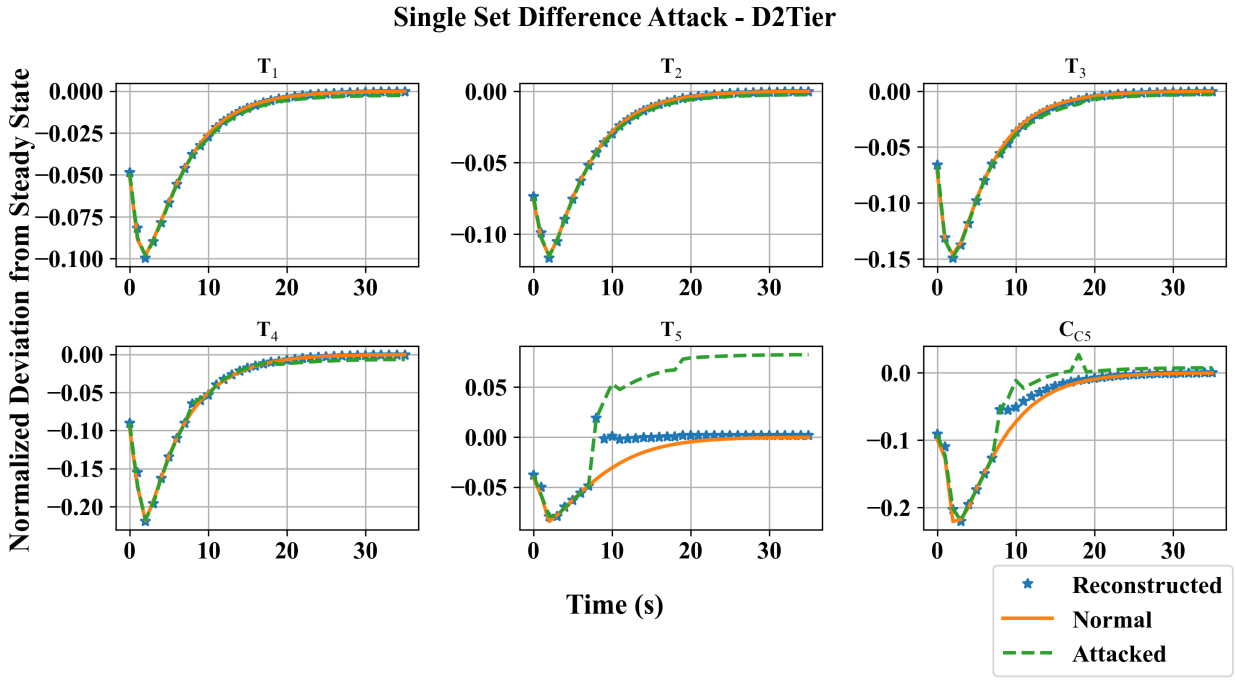

Figure A.26: *Closed-loop detection and reconstruction under a single set-difference attack using the D2tier framework.*

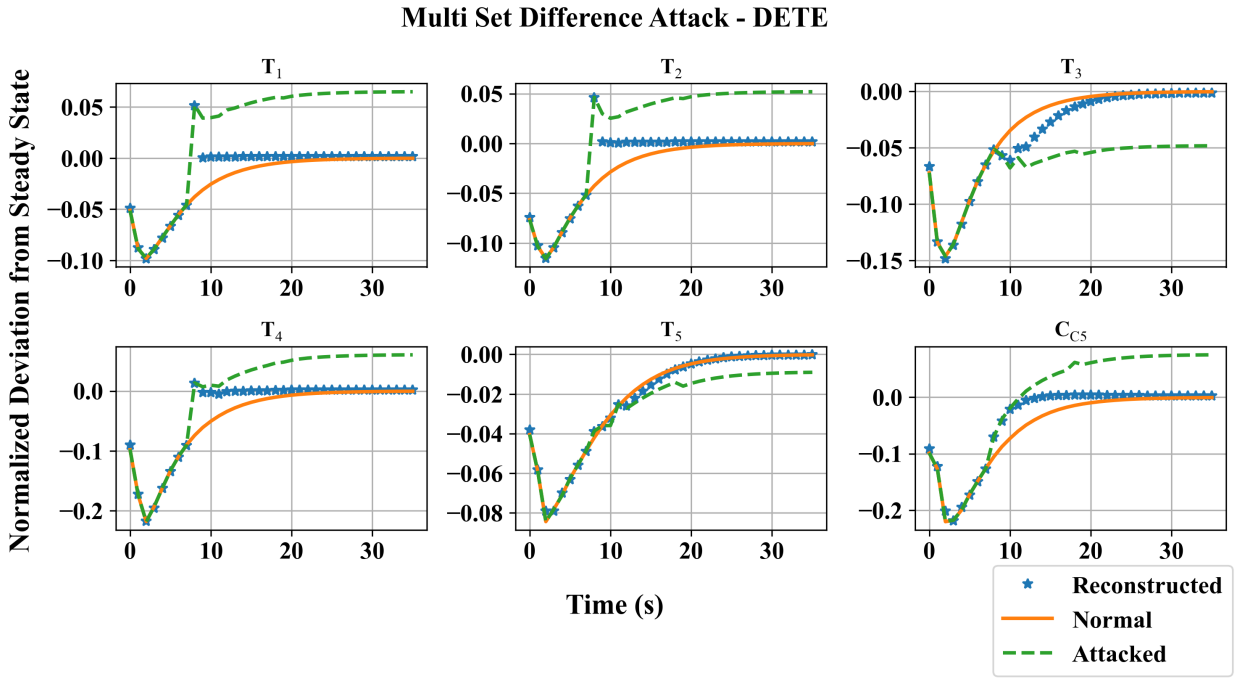

Figure A.27: *Closed-loop detection and reconstruction under a multi-set-difference attack using the DETE framework.*

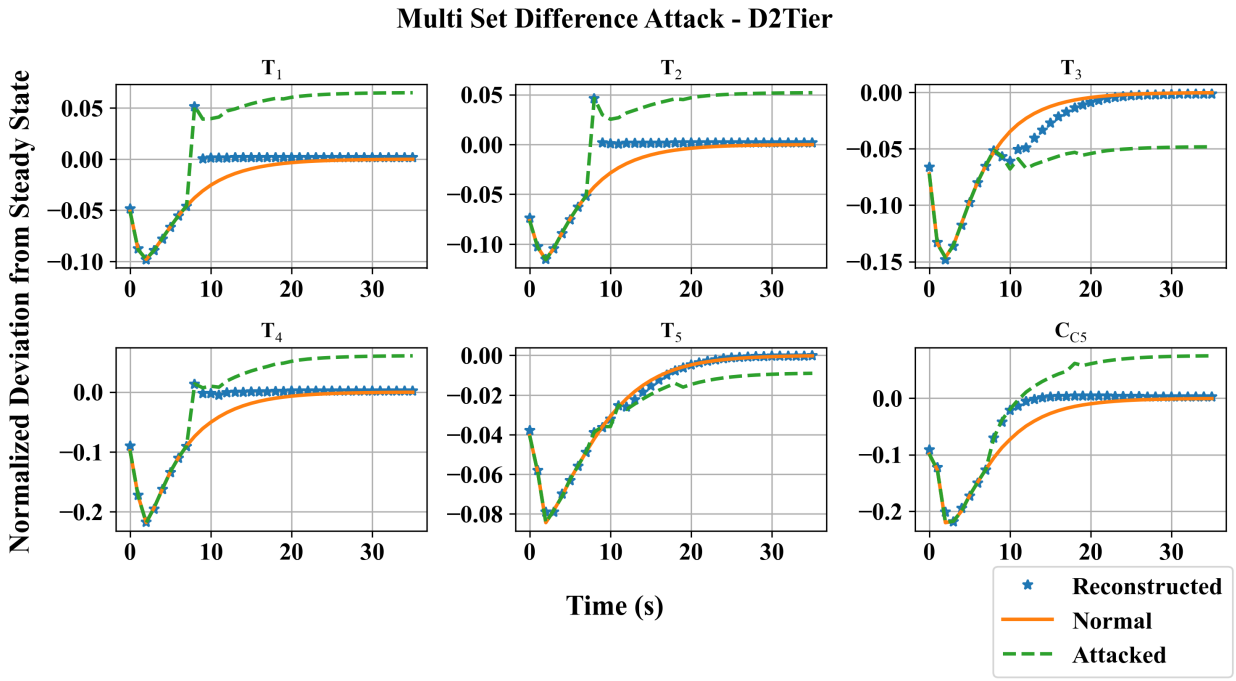

Figure A.28: *Closed-loop detection and reconstruction under a multi-set-difference attack using the D2tier framework.*