algorithms

MDPI

Article

A Hybrid ARX and Deep Sequence Learning Architecture for Multivariate Industrial Time-Series Forecasting: An Application in Fluid Catalytic Cracking

Gulzhan Uskenbayeva [1], Guldana Taganova [2,3,*], Aliya Shukirova [1,*], Ardak Mukhamedrakhimova [1], Salimzhan Tassanbayev [1], Korlan Kulniyazova [1], Gulnara Abitova [4] and Saltanat Turgyn [2]

[1] Department of System Analysis and Control, L. N. Gumilyov Eurasian National University, Astana 010000, Kazakhstan
[2] Higher School of Information Technology and Engineering, Astana International University, Astana 010000, Kazakhstan
[3] Department of Software Engineering, International Engineering and Technological University, Almaty 050060, Kazakhstan
[4] School of Intelligent Systems, Astana IT University, Astana 010000, Kazakhstan
* Correspondence: guldana.kileuzhanova@gmail.com (G.T.); shukirova_ak_1@enu.kz (A.S.)

Abstract

Multi-parametric industrial time-series forecasting is important for predictive monitoring, soft sensing, and decision support in complex process industries. This study proposes a process-informed hybrid ARX-residual deep sequence architecture for short-horizon multi-output forecasting of fluid catalytic cracking unit (FCCU) product yields. The model combines a frozen linear ARX branch with causal temporal convolution, a three-layer GRU, causal self-attention, additive attention pooling, residual gating, and a consistency-regularized multi-output objective. Experiments were performed on an open simulated FCCU benchmark containing 20,160 one-minute observations from seven normal and disturbed operating scenarios. The leakage-controlled pipeline uses scenario-wise chronological splitting, train-only median imputation and standardization, exclusion of fault-timing metadata and direct algebraic target components, and 30 min causal windows for five-minute-ahead forecasting. Across ten prespecified random seeds, the proposed model achieved a historical-test RMSE of 0.022809 $\pm$ 0.000116, compared with 0.022751 for ARX-like Ridge. Dependence-aware statistical analyses did not establish a consistent statistical advantage for either model across the evaluated runs. Rolling-origin evaluation likewise showed similar point accuracy, while leave-one-scenario-out evaluation revealed severe extrapolation failure for the unseen pressure-drop scenario. The results support the reproducibility of the proposed architecture and the importance of the ARX-based dynamic structure, but they do not establish the statistical superiority of the full deep residual architecture. Independent industrial data or newly prespecified simulation trajectories are required for confirmatory external validation.

Keywords: autoregressive–exogenous modeling; deep sequence learning; fluid catalytic cracking; residual forecasting; process-informed modeling; consistency regularization; multivariate time series; scenario-based validation; soft sensors; industrial forecasting

Check for updates

Academic Editor: José Manuel García-Nieto

Received: 6 July 2026
Revised: 17 August 2026
Accepted: 20 August 2026
Published: 27 August 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

Industrial process systems generate large volumes of multi-parametric time-series data that reflect the dynamic interaction of operating conditions, material flows, temperatures,

pressures, actuator states, catalyst circulation, and product-quality indicators. Accurate forecasting of such data is important for predictive monitoring, soft sensing, early detection of process deviations, and decision support in complex production environments. In many process industries, key output variables cannot be measured continuously or directly at the same frequency as routine process measurements. Therefore, data-driven forecasting models can serve as soft sensors that estimate future product behavior from available process measurements [1,2].

Fluid catalytic cracking (FCC) is one of the most important conversion processes in petroleum refining. It converts heavier hydrocarbon fractions into higher-value products, including gasoline-range components, liquefied petroleum gas (LPG), light cycle oil (LCO), and heavier residual fractions. The process is characterized by strong nonlinearities, intensive coupling between process variables, catalyst circulation dynamics, reaction–regeneration balance, fractionation effects, and sensitivity to feed and operating-condition changes [3]. These characteristics make FCC product-yield forecasting a challenging multi-output industrial time-series problem rather than a simple static regression task.

The empirical basis of this study is an open FCC benchmark dataset associated with the FCC-fractionator framework [4–7]. The dataset contains normal and disturbed operating scenarios and provides a reproducible environment for evaluating process monitoring, machine learning, and operation-oriented modeling strategies. Although the benchmark was originally introduced for process monitoring evaluation, its multivariate technological signals and scenario-based temporal structure make it suitable for dynamic forecasting when target variables are defined properly and temporal causality is preserved.

Existing studies on FCC modeling and prediction have shown that both process knowledge and data-driven modeling are important for reliable performance estimation. Classical FCC modeling studies emphasize the complexity of reactor–regenerator interactions, fractionation behavior, and control-oriented process dynamics [5]. Data-driven soft-sensor studies have demonstrated the usefulness of machine learning for estimating difficult-to-measure industrial variables under noisy, collinear, and nonstationary process conditions [6,7]. Recent FCC-focused studies have explored neural-network-based modeling, hybrid deep learning and mechanistic schemes, multiple-level long short-term memory networks, data- and mechanism-driven soft sensors, and attention-based recurrent architectures [8–12]. Broader reviews also confirm that artificial intelligence has become an important direction for catalytic cracking, particularly in product-yield analysis, process monitoring, optimization, and abnormal-condition assessment [13–15].

Research gap: existing approaches often rely either on linear dynamic structures or on purely deep learning architectures. Linear ARX-type models are computationally efficient and can capture strong autoregressive memory, but they are limited in representing nonlinear transitions and complex subsystem interactions. Deep recurrent, convolutional, and attention-based models can learn nonlinear temporal dependencies, but they often treat the process as a black-box mapping and may overlook stable linear dynamic structure that is highly informative in industrial processes. This limitation is especially relevant for FCC product-yield forecasting because the process contains both strong inertia and nonlinear multivariable interactions. Therefore, a hybrid architecture is needed to preserve interpretable ARX-like dynamic behavior while learning nonlinear residual corrections through deep sequence modeling.

To address this gap, this paper develops a process-informed hybrid ARX-residual deep sequence architecture for multi-output FCCU forecasting. The linear ARX branch is used as an explicit dynamic prior, whereas the nonlinear branch is restricted to residual sequence modeling through causal convolution, recurrent memory, self-attention, and temporal pooling. The auxiliary total-yield head is treated as a consistency-regularization device

rather than as independent physical supervision. This distinction is important because the benchmark targets are algebraically related and the study does not employ a complete mechanistic or kinetic FCC model.

The scientific contribution of the study is the formulation and evaluation of a process-informed architecture that couples a frozen ARX dynamic prior with causal deep residual sequence learning for four related FCCU outputs. Generic linear–nonlinear hybridization is not claimed as novel. Additional contributions are (1) an explicitly documented, leakage-controlled data preparation pipeline; (2) target- and scenario-resolved evaluation with scenario-balanced aggregation; (3) ten-seed reproducibility analysis and dependence-aware statistical comparison with a strong ARX-like Ridge reference; (4) development-only rolling-origin and leave-one-scenario-out evaluations; (5) forecast-horizon and component-ablation analyses; (6) comparisons with simpler ARX+MLP, ARX+GRU, and ARX+TCN residual models; and (7) direct grouped permutation-Shapley analysis of the frozen hybrid predictor.

2. Related Works

Because this study uses the open FCCU Dataset for Process Monitoring Evaluation and the associated open FCC-fractionator framework, the literature review is organized around the directions directly relevant to the empirical material: FCC modeling and control, data-driven monitoring and soft sensing, product-yield forecasting from multivariate process data, and modern deep time-series forecasting architectures [1–4].

In the classical FCC literature, the unit is treated as one of the most complex petroleum-refining objects, with pronounced nonlinearity, strong interactions among input and control variables, and substantial sensitivity to feed and operating-mode changes. The review by Pinheiro et al. systematized modeling, monitoring, control, and optimization approaches for FCC and emphasized that adequate process description requires models capable of capturing nonlinear effects and intensive variable interactions observed in practice [5]. This classical process-engineering perspective is important for the present study because FCC product-yield forecasting cannot be reduced to a static regression problem; instead, it requires a dynamic representation of coupled reactor–regenerator, fractionator, catalyst-circulation, and gas-handling behavior.

Data-driven soft sensing and process monitoring form the methodological basis for moving from process interpretation to predictive modeling using historical data. Kadlec et al. showed that data-driven soft sensors are widely used for online prediction, condition monitoring, and deviation detection, while emphasizing typical properties of industrial data, such as noise, missing values, outliers, collinearity, heterogeneous sampling, and dynamic production environments [6]. Qin similarly treated data-driven monitoring as a key direction in statistical and machine learning support for industrial operations [7]. More recent reviews further confirm the importance of soft sensors for industrial monitoring, control, and optimization, particularly when key quality variables are difficult or expensive to measure directly [16]. AI-driven soft sensors have also been discussed as an important direction for sustainable process industries, where predictive models can support product-quality estimation, process efficiency, and decision support under limited online measurements [17].

For FCC applications, recent research has focused on product-yield prediction and key-output estimation. Long et al. proposed a hybrid strategy integrating LASSO-based variable selection and an output-focused back-propagation neural network for FCC modeling [8]. Yang et al. developed a hybrid deep learning and mechanistic kinetics model for predicting FCC performance [9]. Another work by Yang et al. used multiple-level LSTM to predict gasoline yield and emphasized temporal continuity and heterogeneous measurement frequencies in large-scale processing units [10]. Wei et al. proposed a hybrid soft sensor for

key FCC product yield driven by data and process mechanisms [11]. He et al. developed a differential evolutionary dual-stage attention-based LSTM for FCC yield prediction [12]. These studies confirm that modern FCC forecasting increasingly moves from static input–output mappings toward dynamic models that extract temporal information and integrate process-informed structure.

Recent review studies also place this work in a broader scientific context. Khaldi et al. analyzed artificial intelligence perspectives for FCC modeling, control, and optimization [13]. Yang et al. reviewed artificial intelligence methods applied to catalytic cracking processes and identified product-yield analysis, abnormal-condition warning, and optimization as important directions [14]. Acosta-López and de Lasa demonstrated further development of hybrid modeling by combining CPFD and AI for predicting critical FCC reactor variables [15]. The present work extends this line of research by combining a reproducible open FCCU benchmark, leakage-controlled data-centric preparation, and a hybrid ARX-residual deep sequence architecture for dynamic product-yield forecasting.

In parallel with FCC-specific studies, modern time-series forecasting research has rapidly advanced through convolutional, recurrent, linear, and attention-based sequence models. Temporal convolutional networks have shown that causal dilated convolutions can provide long effective memory and strong sequence-modeling performance, challenging the assumption that recurrent networks are always the default choice for temporal modeling [18]. Transformer-based approaches have also been adapted for long-sequence forecasting. Informer introduced ProbSparse self-attention to reduce the computational burden of long-sequence prediction [19]. Autoformer incorporated decomposition and an auto-correlation mechanism to better handle long-term dependencies and periodic patterns [20], while FEDformer combined frequency-domain representation with decomposition to improve efficiency and capture the global time-series structure [21]. Comprehensive surveys confirm that while these Transformer variants successfully address long-range dependencies, their application to multi-parametric industrial series requires careful adaptation [22]. These developments are relevant to industrial forecasting because process data often include inertia, delayed effects, multi-scale temporal patterns, and nonlinear transitions.

At the same time, recent studies have shown that architectural complexity alone does not guarantee superior forecasting accuracy. Zeng et al. demonstrated that simple linear models such as DLinear and NLinear can outperform more complex Transformer-based models on several long-term forecasting benchmarks, emphasizing the importance of preserving temporal structure and choosing baselines carefully [23]. PatchTST later showed that patch-based tokenization and channel-independent Transformer design can improve long-term multivariate forecasting by retaining local temporal semantics and reducing attention complexity [24]. These findings are directly relevant to the present work: they justify the inclusion of strong linear, autoregressive, recurrent, convolutional, attention-based, and modern sequence-learning baselines rather than comparing the proposed model only against weak neural references.

Interpretability is another important requirement for industrial forecasting and soft-sensor applications. SHAP provides a general framework for feature attribution [25]. In this study, target-specific RandomForestRegressor TreeSHAP plots are retained only as exploratory feature-target association summaries; they are not presented as explanations of the hybrid network. Direct interpretation of the proposed model is instead obtained through grouped permutation-Shapley attribution applied to the frozen hybrid predictor, with each input channel treated as one group across its complete 30 min history.

The proposed formulation is related to nonlinear autoregressive models with exogenous inputs (NARX), which jointly map lagged outputs and exogenous variables to future

responses [26]. It is also related to LSTNet, which combines convolutional, recurrent, and autoregressive components for multivariate forecasting [27]. Earlier hybrid ARIMA-ANN forecasting likewise established the broader principle of combining linear and nonlinear predictors [28]. The present work therefore does not claim novelty for linear–nonlinear decomposition itself. Its narrower contribution is the use of a frozen ARX forecast as an explicit additive prior for a causal deep residual branch together with multi-output consistency regularization and an extended evaluation protocol focused on leakage control, scenario balance, temporal dependence, and unseen-scenario stress testing in the FCCU setting.

Overall, the reviewed literature shows three important tendencies. First, FCC product-yield forecasting requires dynamic and process-aware modeling because the process is nonlinear, coupled, and temporally dependent. Second, industrial soft-sensor studies increasingly combine data-driven learning with process knowledge, leakage-controlled validation, and interpretability. Third, modern time-series forecasting research shows that both simple linear dynamic models and deep sequence architectures can be strong, depending on the structure of the data. The present study addresses the gap between these directions by proposing a hybrid architecture in which an ARX-like linear branch captures stable autoregressive–exogenous dynamics, while a nonlinear causal sequence-learning branch models residual temporal behavior through convolutional, recurrent, attention-based, and auxiliary consistency mechanisms.

3. Materials and Methods

3.1. Dataset and Forecasting Task

The study used an open simulated dynamic FCCU benchmark dataset associated with the FCC-fractionator framework [1–4]. The dataset is not an archive of measurements from an operating refinery. Accordingly, the reported results characterize forecasting on reproducible simulated trajectories representing normal and disturbed operating regimes and should not be interpreted as direct industrial validation. The integrated dataset contains 20,160 time-ordered observations from seven scenarios: two normal scenarios and five disturbed scenarios.

Table 1 presents the scenario composition and the chronological split used in the experiments. The data were divided into training, validation, and test subsets within each scenario rather than by random sampling. This choice was important because random splitting would mix past and future observations and could lead to overly optimistic forecasting results. By preserving the temporal order, the evaluation more closely reflects the intended short-term forecasting setting.

Table 1. Summary of the FCCU dataset and chronological scenario-wise split.

Scenario	State	Total Observations	Train	Validation	Test
CAB valve leak	Faulty	1440	1008	216	216
Condenser efficiency decrease	Faulty	1440	1008	216	216
FHN sensor drift	Faulty	1440	1008	216	216
Heat exchanger fouling	Faulty	1440	1008	216	216
NOC stable feed	Normal	2880	2016	432	432
NOC varying feed	Normal	10,080	7056	1512	1512
Pressure drop increase	Faulty	1440	1008	216	216
Total	—	20,160	14,112	3024	3024

Each raw scenario file contained the time variable time_{min} and 46 original process signals (47 columns in total). Integration of the seven scenarios added 16 service and

scenario-control fields for identification, chronological splitting, regime labeling, and fault auditing, together with 12 deterministic engineering variables for unit transformations, aggregate flows, and calculated product yields. The resulting analytical table therefore contains 75 columns without using external information. Of these, only 37 exogenous process variables and four lagged product-yield histories are supplied to the forecasting models. The 47-to-75 transformation is summarized in Supplementary Table S7, and the 37 exogenous channels are listed in Supplementary Table S8.

Table 2 summarizes the model groups considered in the benchmark. The comparison includes both simple and more advanced forecasting approaches, ranging from classical and autoregressive models to recurrent, convolutional, attention-based, and recent sequence-learning architectures. All models were evaluated using the same preprocessing steps, data split, and performance metrics.

Table 2. Model families used in the benchmark comparison.

Model Family	Models	Input Type
Persistence	Persistence	Target history
Linear/regularized	Linear Regression, Ridge, ElasticNet	Exogenous only
ARX reference	ARX-like Ridge	AR + exogenous
Tree/feedforward	Random Forest, MLP	Exogenous only
Recurrent	GRU, Vanilla LSTM, Stacked LSTM, Attention GRU	AR + exogenous
Convolutional/attention	TCN, Causal Transformer	AR + exogenous
Modern sequence models	NLinear, PatchTST-Lite	AR + exogenous
Proposed model	Hybrid ARX–Residual	AR + exogenous

Additional descriptive statistics, data-quality checks, target-distribution comparisons, and covariate-shift summaries are provided in Supplementary Tables S1–S5.

3.2. Target Construction and Data Provenance

The deterministic engineering variables were generated only from quantities contained in each simulated scenario file. Feed flow and reactor-riser temperature were converted using the following relations:

$$F_{\text{feed,lb/min}} = 60F_{3,lb/s} \tag{1}$$

$$F_{\text{feed,t/h}} = F_{\text{feed,lb/min}} \frac{0.45359237 \times 60}{1000} \tag{2}$$

$$T_{\text{r,°C}} = (T_{\text{r,°F}} - 32)\frac{5}{9} \tag{3}$$

The four forecast product-yield targets were constructed as

$$y_{\text{gasoline}} = 100\frac{F_{\text{LN}} + F_{\text{HN}}}{60F_3} \tag{4}$$

$$y_{\text{LCO}} = 100\frac{F_{\text{LCO}}}{60F_3}, y_{\text{LPG}} = 100\frac{F_{\text{LPG}}}{60F_3} \tag{5}$$

$$y_{\text{slurry}} = 100\frac{F_{\text{slurry}}}{60F_3} \tag{6}$$

The auxiliary measured-product total was defined as the algebraic sum of the four primary yields:

$$y_{\Sigma} = y_{\text{gasoline}} + y_{\text{LCO}} + y_{\text{LPG}} + y_{\text{slurry}} \tag{7}$$

Numerical reconstruction of the targets from these equations produced maximum absolute discrepancies between 8.88×10^{-16} and 2.84×10^{-14}, consistent with machine precision. Direct algebraic components of the target equations were therefore excluded from the exogenous predictor set. Supplementary Table S9 gives the complete exclusion and availability audit, including identifiers, split labels, fault descriptors, fault-timing fields, current/future targets, and direct target components. A one-dimensional affine audit found no exact duplicate of any target among the 37 exogenous channels; for the strongest single-variable association (LPG versus fractionator overhead temperature), the maximum residual was 0.472564, ruling out exact algebraic reconstruction.

All 75 columns of the released analytical dataset were audited and assigned to mutually exclusive functional categories. The resulting accounting comprises 37 exogenous process inputs, four primary target columns whose historical values may enter causally, one auxiliary total-yield column, 17 service/temporal/scenario metadata fields, nine direct algebraic target components, and seven additional source or alternative process representations excluded from the fixed 37-channel exogenous protocol. The complete one-to-one accounting and inclusion/exclusion rationale are provided in Supplementary Table S9.

3.3. Leakage-Controlled Data Preparation

The scenario-wise time series were transformed into a causally consistent forecasting representation through variable-role mapping, chronological ordering, leakage control, train-only preprocessing, and causal window construction. Service identifiers, scenario/fault labels, fault-timing metadata, current and future target values, the auxiliary total yield, and direct algebraic target components were excluded from the predictive feature space as detailed in Supplementary Table S9. Figure 1 summarizes the leakage-controlled data-preparation, preprocessing, and standardization workflow used in this study.

Let the original dataset be represented as a collection of observations indexed by scenario and time. For each scenario s in S and time t, the record contains a vector of process variables, target outputs, and service metadata:

$$\mathcal{D} = \left\{ \left(s, t, \boldsymbol{u}_{\mathrm{s,t}}, \boldsymbol{y}_{s,t}, \boldsymbol{m}_{s,t} \right) \right\}, \boldsymbol{u}_{\mathrm{s,t}} \in \mathbb{R}^{37}, \boldsymbol{y}_{\mathrm{s,t}} \in \mathbb{R}^{4}, \boldsymbol{m}_{\mathrm{s,t}} \in \mathbb{R}^{q} \tag{8}$$

Here, $\mathbf{u}_{\mathrm{s,t}}$ is the vector of exogenous process features, $\mathbf{y}_{\mathrm{s,t}}$ is the vector of primary target variables, and $\mathbf{m}_{\mathrm{s,t}}$ is the set of auxiliary and service fields used only for audit, labeling, or scenario control. The target vector contains four product-yield indicators:

$$\boldsymbol{y}_{s,t} = \left[y_{s,t}^{(1)}, y_{s,t}^{(2)}, y_{s,t}^{(3)}, y_{s,t}^{(4)} \right]^{\top} \tag{9}$$

where $\mathrm{y}^{(1)}$ is gasoline proxy yield, $\mathrm{y}^{(2)}$ is light cycle oil yield, $\mathrm{y}^{(3)}$ is LPG yield, and $\mathrm{y}^{(4)}$ is slurry yield. The auxiliary aggregate output is defined as

$$y_{s,t}^{aux} = y_{s,t}^{(1)} + y_{s,t}^{(2)} + y_{s,t}^{(3)} + y_{s,t}^{(4)} \tag{10}$$

Here, the predictive representation contains 37 exogenous process channels and four historical target channels. Scenario identifiers and timestamps are retained only for sorting, segmentation, and integrity checks. Median-imputation parameters and StandardScaler means and standard deviations were fitted on the training subset only and then applied unchanged to validation and historical-test data. No actual missing values were present in the final 41 selected channels, but the train-fitted imputer was retained as part of the reproducible pipeline.

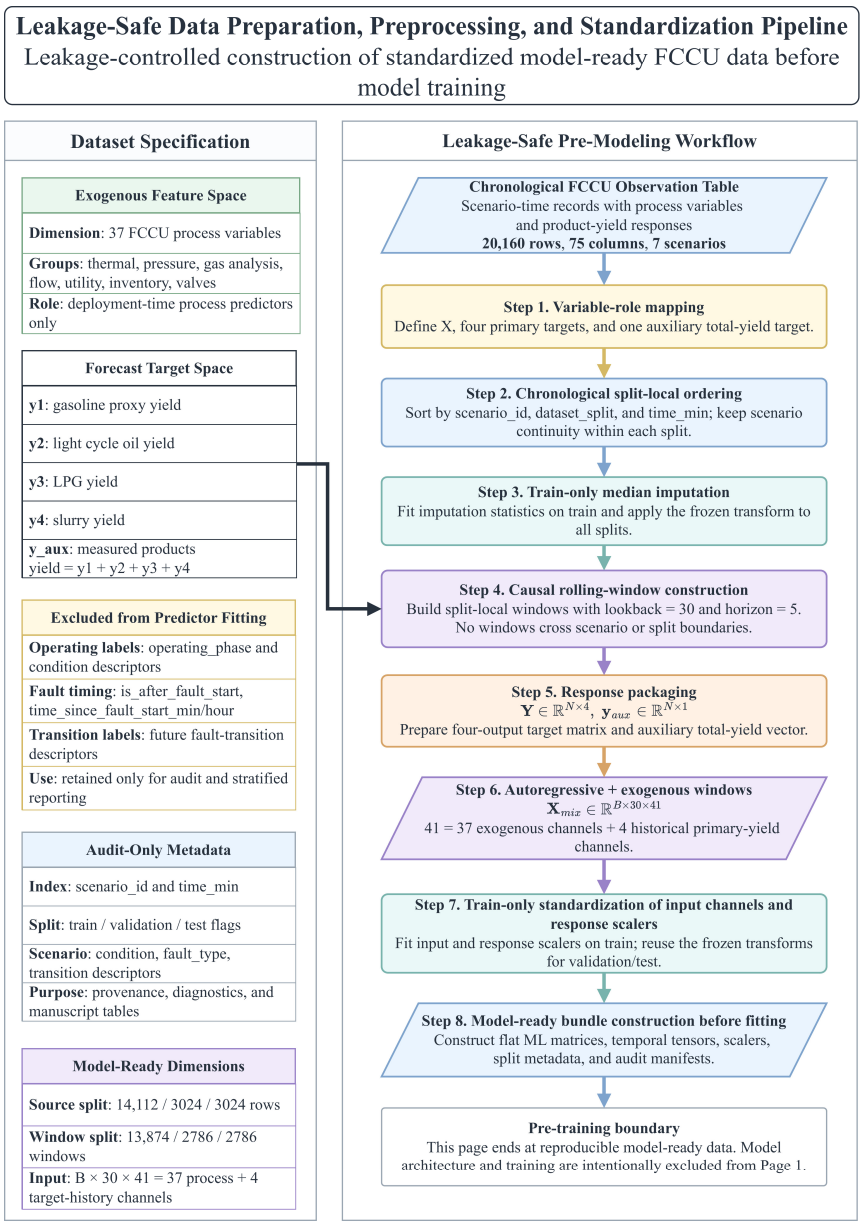


Figure 1. FCCU variable composition and leakage-controlled preprocessing workflow.

Causal Window Construction

After train-fitted preprocessing, each scenario was converted into causal windows separately within its train, validation, and historical-test partition. No window was allowed to cross a scenario boundary or a subset boundary. Let the lookback length be 30 min and the primary forecasting horizon be 5 min. At prediction time t, the model uses only process observations and target histories available through the end of the causal window at t, and predicts the four-dimensional target vector at t + 5:

$$\boldsymbol{X}_{s,t}^{exo} = \left[\boldsymbol{u}_{s,t-L+1}^{*}, \boldsymbol{u}_{s,t-L+2}^{*}, \ldots, \boldsymbol{u}_{s,t}^{*}\right] \in \mathbb{R}^{L\times 37} \tag{11}$$

$$\boldsymbol{Y}_{s,t} = \boldsymbol{y}_{s,t+H} \in \mathbb{R}^{4} \tag{12}$$

To construct the autoregressive–exogenous representation, the history of the target variables is added:

$$\boldsymbol{Y}_{s,t}^{hist} = \left[\boldsymbol{y}_{s,t-L+1}, \boldsymbol{y}_{s,t-L+2}, \ldots, \boldsymbol{y}_{s,t}\right] \in \mathbb{R}^{L\times 4} \tag{13}$$

$$\boldsymbol{X}_{s,t}^{mix} = \left[\boldsymbol{X}_{s,t}^{exo} / \parallel / \boldsymbol{Y}_{s,t}^{hist}\right] \in \mathbb{R}^{L\times 41} \tag{14}$$

The final training set can be written as

$$\mathcal{T} = \left\{\left(\boldsymbol{X}_{s,t}^{mix}, \boldsymbol{Y}_{s,t}, Y_{s,t}^{aux}\right)\right\}_{(s,t)\in\Omega} \tag{15}$$

For the five-minute horizon, causal windowing produced 13,874 training examples, 2786 validation examples, and 2786 historical-test examples. All 41 input channels passed the availability audit: no future targets, split labels, fault labels, fault-timing variables, or direct algebraic target components were supplied to the models.

Unified Forecasting and Baseline Model Formulations

For clarity, the general forecasting task and baseline model families are formulated before the proposed architecture. Let $\mathbf{z}_{s,t}$ denote either an exogenous vectorized window or a mixed autoregressive–exogenous vectorized window, depending on the model family.

$$\hat{\boldsymbol{y}}_{s,t+H} = f_\theta(\boldsymbol{z}_{s,t}), \hat{\boldsymbol{y}}_{s,t+H} \in \mathbb{R}^4 \tag{16}$$

The generic hybrid formulation decomposes the prediction into a linear ARX component and a nonlinear residual component:

$$\hat{\boldsymbol{y}}_{s,t+H}^{lin} = \boldsymbol{W}_{lin} vec\left(\boldsymbol{X}_{s,t}^{mix}\right) + \boldsymbol{b}_{lin} \tag{17}$$

$$\hat{\boldsymbol{y}}_{s,t+H} = \hat{\boldsymbol{y}}_{s,t+H}^{lin} + \boldsymbol{\alpha} \odot \hat{\boldsymbol{y}}_{s,t+H}^{nl} \tag{18}$$

3.4. Proposed Process-Informed ARX-Residual Architecture

The proposed architecture combines a frozen ARX-like linear dynamic branch with a nonlinear deep sequence residual branch. The ARX component serves as an explicit process-informed dynamic prior. The deep branch is not asked to reconstruct the complete process mapping; instead, it learns a target-wise residual correction from the same causal 30×41 input tensor. The architectural and training configuration was selected in an initial 10-trial Optuna search using validation macro-RMSE as the optimization objective and was subsequently frozen before the additional evaluation experiments. The frozen configuration remained unchanged across the ten-seed reproducibility analysis, component ablations, rolling-origin evaluation, leave-one-scenario-out evaluation, and forecast-horizon experiments (Supplementary Tables S10 and S13).

The ARX-like branch receives a vectorized representation of the causal input window and generates a baseline dynamic forecast from lagged target values and exogenous process variables. This branch acts as a process-informed dynamic reference. It also prevents the neural component from learning the entire process mapping from scratch, allowing the nonlinear branch to focus on residual behavior not captured by the linear structure.

The nonlinear branch processes the same input window in sequential tensor form. After input normalization and per-time-step projection, a causal temporal convolutional encoder extracts local temporal patterns without using future information. A gated recurrent unit memory core then captures sequential dependencies and process inertia. Causal self-attention refinement and additive temporal attention pooling are used to identify and aggregate the most informative time states within the input window.

The final forecast is obtained by additively combining the frozen ARX prediction with a target-wise scaled nonlinear residual. The learned residual-scaling vector remains close to zero across the ten random seeds (Supplementary Table S11), indicating that the five-minute forecast is dominated by the ARX prior while the deep branch makes relatively small corrections. An auxiliary head predicts the total measured product yield and is used only to regularize consistency among the multi-output predictions.

Executed Architecture and Input Representation

The executed input tensor is a scaled mixed window containing 30 time steps and 41 channels.

The additive forecast can be written as

$$\hat{y}_{t+h} = \hat{y}_{t+h}^{ARX} + \alpha_{res} \odot \hat{r}_{t+h}^{deep} \tag{19}$$

where the ARX Ridge regularization parameter is $\lambda_{ARX} = 1.0$ and α_{res} is a learned four-dimensional scaling coefficient for the nonlinear residual correction.

The temporal context vector is obtained using additive attention pooling:

$$e_t = \boldsymbol{v}^\top tanh(\boldsymbol{W}_a \boldsymbol{h}_t^{(att)} + \boldsymbol{b}_a), \alpha_t = \frac{exp(e_t)}{\sum_{\tau=1}^{30} exp(e_\tau)}, \boldsymbol{c} = \sum_{t=1}^{30} \alpha_t \boldsymbol{h}_t^{(att)} \tag{20}$$

The nonlinear residual-correction head is applied to the fused context:

$$\boldsymbol{y}_{NL} = \boldsymbol{W}_2 \, Dropout(\phi(\boldsymbol{W}_1 LN(\boldsymbol{c}_{fused}) + \boldsymbol{b}_1)) + \boldsymbol{b}_2 \tag{21}$$

Auxiliary Total-Yield Head and Consistency-Regularized Objective

The auxiliary total-yield head is defined as

$$\hat{y}_{total} = \boldsymbol{w}_a^\top \, Dropout(\phi(\boldsymbol{W}_a LN(\boldsymbol{c}_{fused}) + \boldsymbol{b}_a)) + b_a \tag{22}$$

The final consistency-regularized multi-output objective follows the fixed training specification:

$$\mathcal{L} = \mathcal{L}_{primary} + 0.142631 \, \mathcal{L}_{aux} + 0.066263 \, \mathcal{L}_{consistency} + 0.044448 \, \mathcal{L}_{bounds} \tag{23}$$

The auxiliary-, consistency-, and bounds-loss weights were included in the initial 10-trial Optuna hyperparameter search and were selected using validation macro-RMSE as the optimization objective. The selected values were 0.142631, 0.066263, and 0.044448, respectively. After this initial validation-based selection, these coefficients were frozen and retained unchanged in all subsequent reviewer-closure experiments. Given the limited 10-trial tuning budget, global optimality of these values is not claimed. Because the auxiliary total is exactly the algebraic sum of the primary targets, it is not independent physical supervision. Accordingly, the architecture is described as process-informed and consistency-regularized rather than as a mechanistically constrained model claiming independent physical supervision.

Figure 2 illustrates the proposed hybrid ARX and deep sequence learning architecture. The diagram shows how the causal input window is processed through the ARX-like linear branch and the nonlinear sequence-learning branch, and how the final product-yield forecast is obtained through residual correction and auxiliary total-yield supervision.

3.5. Baseline Models and Evaluation Protocol

The primary benchmark includes 16 forecasting models: persistence; exogenous-only Linear Regression, Ridge Regression, ElasticNet, Random Forest, and MLP; ARX-like Ridge; recurrent and attention-augmented recurrent networks; TCN; Causal Transformer;

NLinear; PatchTST-Lite; and the proposed model. Three additional ARX-residual comparators (ARX+MLP, ARX+GRU, and ARX+TCN) were subsequently evaluated as diagnostic comparisons. Thus, 19 configurations were actually executed. LightGBM is documented as an excluded candidate because the required Apple Silicon libomp runtime was incompatible with the execution environment; it is not counted as an evaluated model. Full executed configurations and actual tuning budgets are reported in Supplementary Tables S12 and S13.

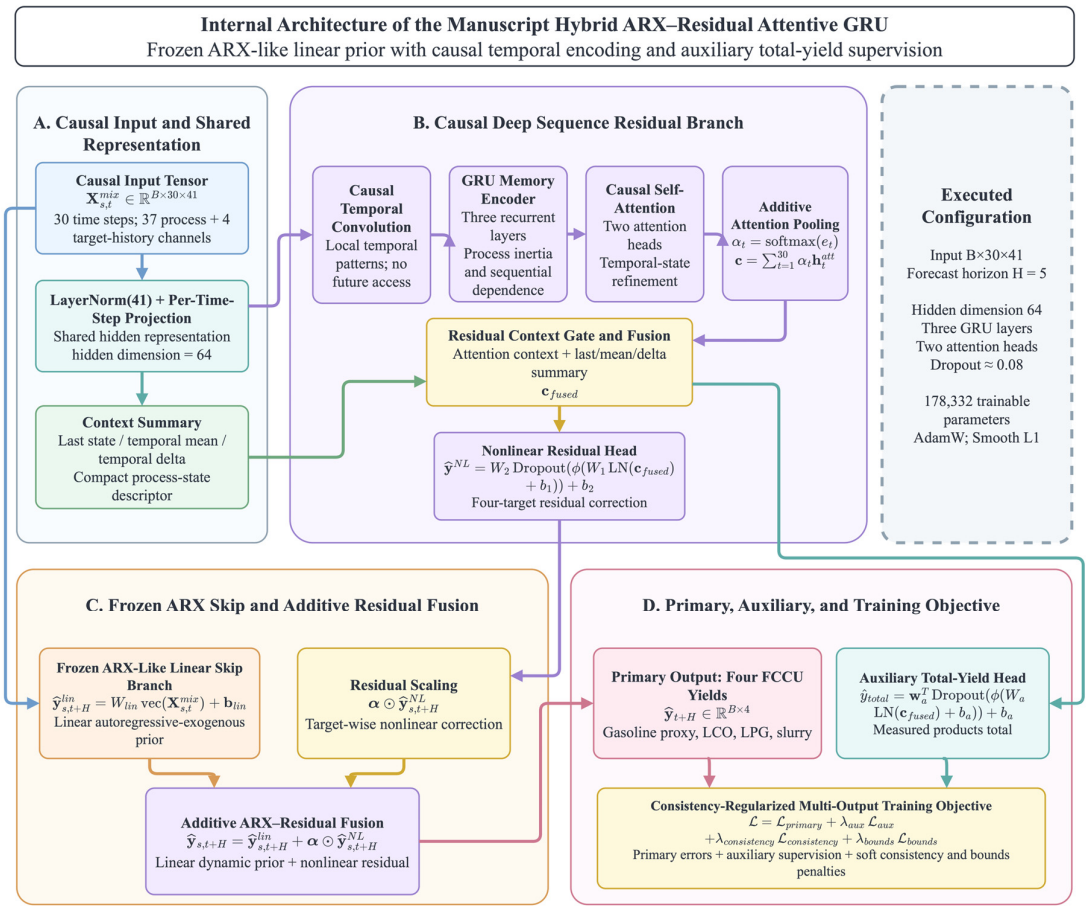


Figure 2. Proposed process-informed ARX-Residual Attentive GRU architecture for forecasting four FCCU product yields. The auxiliary total-yield head supports consistency regularization and is not treated as independent physical supervision.

The forecasting task is multi-output regression using a 30 min causal history to predict four product-yield targets five minutes ahead. Model development, checkpoint selection, and any residual-shrinkage selection used training and validation data only. For every neural run, the checkpoint was chosen by the minimum validation macro-RMSE. The historical-test subset was not used for architecture selection, hyperparameter tuning, early stopping, epoch selection, residual-coefficient selection, or random-seed selection; it was accessed only after the experimental protocol had been fixed.

Classical models use vectorized causal windows, whereas neural sequence models operate on batch x time x feature tensors. Median imputation and StandardScaler parameters are fitted on training data only. Predictions are inverse-transformed to the original product-yield scale before evaluation. RMSE and MAE are therefore reported in weight-percentage points of feed, R2 is dimensionless, and sMAPE is reported in percent using

the multiplication-by-100 convention. For each metric, the macro value is the unweighted mean of the four target-wise metrics:

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}, \tag{24}$$

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N} |y_i - \hat{y}_i|, \tag{25}$$

$$\mathrm{R}^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \overline{y})^2}, \tag{26}$$

$$\mathrm{sMAPE} = \frac{100}{N}\sum_{i=1}^{N}\frac{2\,|\,y_i - \hat{y}_i\,|}{|\,y_i\,| + |\,\hat{y}_i\,| + \varepsilon}. \tag{27}$$

where y_i and $\hat{y}_i$ denote the observed and predicted target values, respectively, $\overline{y}$ is the mean observed value, N is the number of evaluated samples, and ε is a small constant used to avoid division by zero. Metrics are reported on validation and test subsets, with the test set reserved for final model comparison.

The proposed model is evaluated against 15 baseline and reference models under a unified experimental protocol. The benchmark includes naive, linear, regularized, ensemble, feedforward neural, autoregressive–exogenous, recurrent, convolutional, attention-based, and modern sequence-learning models.

PatchTST-Lite tokenizes the time window into temporal patches before applying attention-based encoding. NLinear applies a channel-wise linear transformation to centered temporal trajectories. All models are trained and evaluated on the same leakage-controlled data representation. Additional optimization and training-procedure details are provided in Supplementary Table S6.

Hyperparameter selection for the proposed model was performed using an initial 10-trial Optuna study. The optimization objective was validation macro-RMSE; the historical-test subset was not used for architecture, optimization-parameter, or loss-weight selection. The study used TPESampler with seed 42 and MedianPruner with three startup trials. The search space comprised hidden dimension {64, 96, 128}, number of GRU layers {1, 2, 3}, number of attention heads {2, 4, 8} subject to divisibility by the hidden dimension, dropout [0.05, 0.20], learning rate [0.001, 0.004] on a logarithmic scale, weight decay $[1 \times 10^{-6}, 5 \times 10^{-4}]$ on a logarithmic scale, batch size {64, 128}, auxiliary-loss weight [0.10, 0.35], consistency-loss weight [0.05, 0.30], and bounds-loss weight [0.005, 0.05] on a logarithmic scale. Trial 1 achieved the lowest validation macro-RMSE (0.020483) and selected hidden dimension 64, three GRU layers, two attention heads, dropout 0.079951 (reported as approximately 0.08), learning rate 0.002040, weight decay 0.000040, batch size 128, auxiliary-loss weight 0.142631, consistency-loss weight 0.066263, and bounds-loss weight 0.044448. This configuration was subsequently frozen without further architecture-level tuning for the multi-seed, ablation, rolling-origin, leave-one-scenario-out, and multi-horizon experiments.

The resulting frozen configuration uses hidden dimension 64, three GRU layers, two attention heads, dropout approximately 0.08, AdamW with initial learning rate 0.002040 and weight decay 0.000040, batch size 128, a maximum of 36 epochs, and early stopping after six epochs without validation improvement. The ARX branch is fitted in float64 using multi-output Ridge Regression with LSQR (lambda_ARX = 1.0, tolerance 10^{-8}, maximum 10,000 iterations) and is then frozen. The computational experiments were performed in Python 3.11.15 using NumPy 1.26.4, scikit-learn 1.8.0, PyTorch 2.10.0, Optuna 4.8.0, and SHAP 0.48.0 on Apple Silicon MPS. The full fixed configuration and residual-scaling statistics are given in Supplementary Tables S10 and S11.

The complete model registry and experimental configuration details are provided in Supplementary Tables S10–S13.

For direct model interpretation, grouped permutation-Shapley attribution was computed by calling the frozen proposed model with seed 42. Each of the 41 input channels was treated as one group containing its full 30 min history. The calculation used 28 scenario-stratified windows, 64 permutations, and 32 antithetically ordered permutation pairs. Monte Carlo error and local-additivity diagnostics were recorded. A separate ExtraTrees model was fitted only to assess surrogate fidelity against the hybrid predictions; it was not used to obtain the principal Shapley attributions.

4. Results

4.1. Forecasting Performance

Table 3 reports the primary benchmark results. Across ten prespecified seeds, the proposed model achieved validation RMSE 0.020590 and historical-test RMSE 0.022809, MAE 0.017378, R2 0.967491, and sMAPE 0.134715%. ARX-like Ridge achieved a slightly lower historical-test RMSE of 0.022751 and a slightly higher R2 of 0.967819, whereas the proposed model had marginally lower MAE and sMAPE. The dependence-aware statistical analyses reported in Section 4.5 did not establish a consistent statistical advantage for either model.

Table 3. Primary forecasting performance comparison.

Model	Inputs	Validation RMSE	Historical-Test RMSE	MAE	R2	sMAPE (%)
Proposed Hybrid (10-seed mean)	AR + exog.	0.020590	0.022809	0.017378	0.967491	0.134715
ARX-like Ridge	AR + exog.	0.020699	0.022751	0.017394	0.967819	0.134932
Persistence	Target history	0.032977	0.032705	0.026215	0.905538	0.230038
Ridge Regression	Exog. only	0.030800	0.038231	0.024985	0.964187	0.153101
Linear Regression	Exog. only	0.0323	0.0391	0.0255	0.9596	0.1634
ElasticNet	Exog. only	0.0325	0.0407	0.0263	0.9611	0.1637
Causal Transformer	AR + exog.	0.041743	0.050105	0.036245	0.955105	0.198149
GRU	AR + exog.	0.038632	0.050425	0.034408	0.956235	0.188396
PatchTST-Lite	AR + exog.	0.0412	0.0538	0.0403	0.9534	0.2121
NLinear	AR + exog.	0.0424	0.0605	0.0395	0.9557	0.2069
Vanilla LSTM	AR + exog.	0.0444	0.0614	0.0414	0.9406	0.2208
Stacked LSTM	AR + exog.	0.0679	0.0654	0.0487	0.9227	0.2887
Attention GRU	AR + exog.	0.0420	0.0688	0.0457	0.9429	0.2289
TCN	AR + exog.	0.0586	0.0749	0.0549	0.9358	0.2685
Random Forest	Exog. only	0.1456	0.0928	0.0670	0.7769	0.4678
MLP	Exog. only	0.0686	0.1059	0.0709	0.8210	0.3870

The values in Table 3 deliberately distinguish the ten-seed primary estimate from seed-specific diagnostics. For seed 42, the validation RMSE was 0.020643 and historical-test RMSE was 0.022796. The multi-seed ablation analysis in Table 4 uses a separate fixed three-seed set (11, 42, and 101), for which the full-model mean RMSE is 0.022823 ± 0.000036. These values are not interchangeable.

The comparison shows that the strongest reference is the compact ARX-like Ridge model rather than a weak baseline. Standalone recurrent and Transformer-inspired baselines have substantially higher errors, but the full hybrid architecture does not show a stable directional advantage over ARX-like Ridge itself. The result therefore supports the importance of a correctly specified autoregressive–exogenous formulation and motivates a more cautious interpretation of the incremental deep residual branch.

Table 4. Multi-seed ablation analysis of the proposed architecture.

Configuration	Mean Test RMSE	SD	Difference vs. Full
Without causal temporal convolution	0.022729	0.000043	−0.000094
Without causal self-attention	0.022760	0.000047	−0.000063
Without attention pooling	0.022795	0.000192	−0.000028
Without auxiliary/consistency/bounds regularization	0.022819	0.000160	−0.000003
Full model	0.022823	0.000036	0
Without residual context gate	0.022840	0.000098	+0.000018
Without frozen ARX branch	0.087400	0.004842	+0.064577

The primary benchmark is summarized below; all RMSE and MAE values are in wt%-points of feed and sMAPE is expressed as a percentage.

Comparison with Classical and Deep Learning Models

The classical exogenous-only linear models, including Ridge Regression, Linear Regression, and ElasticNet, also achieved competitive performance. This confirms that the exogenous process variables contain substantial predictive information. However, these models do not explicitly use the autoregressive history of the target variables in the same way as the ARX-like and hybrid models. Their performance therefore represents the limit of exogenous-only forecasting under the considered setting.

Among standalone deep sequence baselines, Causal Transformer and GRU are the strongest entries in the retained primary benchmark, but both remain substantially less accurate than the ARX-based models. This indicates that sequence-model complexity alone is not sufficient for the present short-horizon FCCU task. The later ARX-residual comparator experiment (Section 4.8) further shows that simpler residual branches can match or outperform the full proposed branch in a single-seed diagnostic.

The lower performance of Random Forest and multilayer perceptron indicates that static nonlinear models without a specialized temporal structure are less suitable for this forecasting problem. Although they can learn nonlinear mappings, they do not explicitly model causal temporal dependencies and process inertia.

4.2. Ablation Study

The ablation results were recomputed using the fixed seeds 11, 42, and 101. The full model has a mean historical-test RMSE of 0.022823 ± 0.000036 over this three-seed subset. Removing the frozen ARX branch causes a large degradation to 0.087400, whereas removing individual deep components changes the RMSE by less than 10^{-4} and several simplified variants have slightly lower point errors. Consequently, an independent positive contribution of each deep component is not statistically established.

Window-length sensitivity was evaluated at the five-minute horizon using explicitly defined histories. Development-only RMSE was 0.021587 for a 15 min lookback, 0.021785 ± 0.000029 for the 30 min reference look-back, and 0.022353 for a 60 min lookback. Window-length sensitivity is therefore reported using explicitly defined lookback durations, while feature effects are assessed through explicit component ablations.

4.3. Computational Efficiency

The proposed model contains 183,256 total parameters, of which 178,332 are trainable after freezing the 4924-parameter ARX branch. The ARX-like Ridge reference contains 4924 fitted coefficients. Under the unified timing protocol, the median single-window latency is 15.221 ms for the hybrid and 0.0248 ms for ARX-like Ridge, making the hybrid approximately 613 times slower per individual forecast.

The absolute latency of the hybrid model remains far below the one-minute sampling interval and is therefore compatible with near-real-time inference on the tested hardware.

However, the additional computational cost is not accompanied by a statistically significant RMSE improvement. ARX-like Ridge is therefore the preferred engineering option when minimum latency, simplicity, and comparable point accuracy are the primary requirements. Practical deployment of either autoregressive model also assumes timely availability of the four product-yield histories; minute-level availability on an operating FCCU would require an online analyzer or a separately validated soft sensor.

Table 5 summarizes the computational cost and inference-time characteristics of the proposed model and the ARX-like Ridge reference under the unified timing protocol.

Table 5. Computational cost of the proposed model and ARX-like Ridge.

Measure	Proposed Model	ARX-like Ridge
Total parameters	183,256	4924
Trainable parameters	178,332	4924
Total training time (s)	89.969	34.394
Median single-window latency (ms)	15.221	0.0248
95th-percentile latency (ms)	17.645	0.0314
Median batch-256 latency (ms)	20.358	0.291
Throughput (windows/s)	12,575	879,916

Timing used one warm-up run followed by 50 measured repetitions for each model. Hybrid training time includes construction of the ARX branch plus the median neural-training time across the ten seeds. The comparison is intended to characterize the deployment trade-off, not to claim that the slower architecture is preferable when its predictive accuracy is statistically indistinguishable from the ARX reference.

4.4. Scenario-Stratified Evaluation

To prevent the long NOC varying-feed simulation from dominating the aggregate score, metrics were computed separately for each of the seven operating scenarios and four target outputs. Scenario-balanced aggregation assigns equal weight to each scenario regardless of its number of available windows. Table 6 reports the RMSE for seed 42; the complete RMSE, MAE, R2, and sMAPE matrix for all 28 scenario-target combinations is provided in Supplementary Table S15.

Table 6. RMSE by product output and operating scenario for seed 42.

Scenario	Gasoline Proxy	LCO	LPG	Slurry	Macro-RMSE
CAB valve leak	0.043778	0.017618	0.022241	0.010932	0.023642
Condenser efficiency decrease	0.090903	0.016466	0.059075	0.010444	0.044222
FHN sensor drift	0.041489	0.016558	0.021996	0.010711	0.022689
Heat exchanger fouling	0.039000	0.015249	0.020490	0.009880	0.021155
NOC stable feed	0.035334	0.013267	0.018102	0.008699	0.018851
NOC varying feed	0.040997	0.013651	0.011724	0.008683	0.018764
Pressure drop increase	0.042836	0.017190	0.022435	0.011009	0.023368
Scenario-balanced mean	0.047762	0.015714	0.025152	0.010051	0.024670

The largest known-scenario error occurs under condenser efficiency decrease, where the macro-RMSE reaches 0.044222 because gasoline-proxy and LPG errors increase to 0.090903 and 0.059075, respectively. The two normal scenarios have macro-RMSE values of 0.018851 and 0.018764. The scenario-balanced mean is 0.024670; the normal-only and faulty-only scenario-balanced macro-RMSE values are 0.018807 and 0.027015, and giving the normal and faulty groups an equal overall weight yields 0.022911.

4.5. Multi-Seed Reproducibility and Dependence-Aware Statistical Comparison

The proposed model was independently trained with ten prespecified seeds: 11, 23, 42, 71, 101, 131, 173, 211, 257, and 307. No run was selected on the basis of historical-test error; each checkpoint was selected only by the minimum validation macro-RMSE. The full seed-level results are given in Supplementary Table S14.

Table 7 summarizes the reproducibility results across the ten prespecified independent runs, including the mean, standard deviation, and 95% confidence interval for each evaluation metric.

Table 7. Reproducibility of the proposed model across ten independent runs.

Metric	Mean	SD	95% Confidence Interval
Validation RMSE	0.020590	0.000061	[0.020547, 0.020634]
Historical-test RMSE	0.022809	0.000116	[0.022726, 0.022892]
Historical-test MAE	0.017378	0.000081	[0.017319, 0.017436]
Historical-test R2	0.967491	0.000344	[0.967246, 0.967737]
Historical-test sMAPE (%)	0.134715	0.000557	[0.134317, 0.135113]

The narrow intervals show low sensitivity to random initialization. Nevertheless, the mean RMSE of the proposed model (0.022809) is slightly higher than that of ARX-like Ridge (0.022751), while MAE and sMAPE are marginally lower and R2 is marginally lower. The direction of the comparison therefore depends on the metric.

Because the 30 min windows overlap, forecast losses are temporally dependent. We therefore used a scenario-paired Wilcoxon test, a Diebold–Mariano analysis with HAC variance, and an overlapping moving-block bootstrap. The bootstrap used 5000 replicates with block length 35, and the HAC lag was 34.

For the Diebold–Mariano analysis, the window-level loss difference was defined as the mean squared error of the proposed model minus the mean squared error of ARX-like Ridge, averaged across the four target outputs. Loss differences were ordered chronologically within each of the seven operating scenarios. Newey–West/Bartlett HAC long-run variances were estimated separately within scenarios using a prespecified lag of 34 (=30-step lookback + 5-step forecast horizon − 1), after which scenario means and their variance estimates were combined with equal scenario weights. Positive statistics therefore indicate lower loss for ARX-like Ridge, whereas negative statistics indicate lower loss for the proposed model. The analysis was repeated separately for all ten prespecified model seeds, and the corresponding statistics and two-sided p-values are reported in Supplementary Table S20.

Table 8 summarizes the dependence-aware statistical comparison between the proposed model and the ARX-like Ridge reference.

The scenario-paired Wilcoxon test did not indicate a difference, and the moving-block bootstrap interval includes zero. The HAC-corrected Diebold–Mariano outcome was seed-dependent rather than consistently favoring one model. These analyses do not establish the statistical superiority of the proposed architecture on the historical-test segment; they also should not be interpreted as proof of model equivalence.

For the canonical frozen run (seed 42), an overlapping moving-block bootstrap was additionally applied to the scenario-ordered window-level squared-loss differences. Resampling was performed independently within each operating scenario so that no bootstrap block crossed a scenario boundary. For each scenario, block starting positions were sampled with replacement from all valid starting locations for overlapping blocks of length 35. Blocks were concatenated until the original scenario-specific sequence length was restored, with the final block truncated when necessary. This procedure was repeated 5000 times

using a fixed NumPy random-generator seed of 20,260,802. For each bootstrap replicate, the mean loss difference was first computed within each scenario and the final statistic was then obtained as the unweighted mean across the seven scenarios. This construction preserves local serial dependence while preventing the longest scenario from dominating the aggregate estimate.

Table 8. Dependence-aware statistical comparison of the proposed model and ARX-like Ridge for the canonical frozen run (seed 42).

Statistical Comparison	Result
ARX-like Ridge RMSE	0.022751
Proposed-model RMSE, seed 42	0.022796
RMSE difference, seed 42 (Proposed − ARX)	+0.000045
Scenario-paired Wilcoxon statistic	13.0
Scenario-paired Wilcoxon two-sided p-value	0.9375
Scenario-balanced mean squared-loss difference (Proposed − ARX)	4.895587×10^{-7}
Diebold–Mariano HAC statistic, seed 42	0.049800
Diebold–Mariano two-sided p-value	0.960282
Diebold–Mariano HAC lag	34
95% moving-block bootstrap interval	$[-2.3 \times 10^{-5}, 1.1 \times 10^{-5}]$
Bootstrap repetitions	5000
Overlapping block length	35

For seed 42, the observed scenario-balanced mean squared-loss difference (proposed model minus ARX-like Ridge) was 4.895587×10^{-7}. The corresponding 95% percentile moving-block bootstrap interval was $[-2.3 \times 10^{-5}, 1.1 \times 10^{-5}]$, which includes zero. This result is consistent with the seed-42 DM-HAC statistic of 0.049800 (two-sided $p = 0.960282$) and does not support a statistically significant advantage of either model for this run.

4.6. Rolling-Origin and Leave-One-Scenario-Out Evaluation

Temporal transport within the known scenario families was evaluated using a development-only rolling-origin design. The historical-test subset was not used in this experiment. At each origin, preprocessing parameters, the ARX branch, and the neural model were re-estimated using only earlier temporal intervals.

The rolling-origin experiment used only the original development portion (train + validation) of each scenario; no historical-test rows were used. Three expanding origins were defined within each scenario. At Origin 1, the first 40% of development rows were used for training, the next 15% for validation, and the following 15% for assessment. Origins 2 and 3 expanded the training prefix to 55% and 70%, respectively, while retaining subsequent non-overlapping 15% validation and 15% assessment intervals. Preprocessing, ARX fitting, neural-model training, and validation-based checkpoint selection were repeated independently at each origin using only information available up to the corresponding training boundary. Exact integer cutoffs differ with scenario length and are reported in Supplementary Table S21.

Table 9 reports the results of the development-only rolling-origin evaluation for the proposed model and ARX-like Ridge.

The percentage intervals denote nominal within-scenario fractions of the development data. Exact integer row and time-minute boundaries for every scenario and origin are reported in Supplementary Table S21. Window counts are obtained after causal window construction with a 30-step lookback and a 5-step horizon.

The point RMSE and MAE are close at all three origins; the mean RMSE difference is 0.000033. This result characterizes temporal transport within known scenario families and does not establish statistical equivalence.

Table 9. Development-only rolling-origin evaluation.

Origin	Train Interval	Validation Interval	Assessment Interval	Train Windows	Validation Windows	Assessment Windows	Proposed RMSE	ARX RMSE
1	0–40%	40–55%	55–70%	6613	2329	2329	0.040070	0.040109
2	0–55%	55–70%	70–85%	9185	2329	2329	0.021406	0.021417
3	0–70%	70–85%	85–100%	11752	2329	2329	0.021766	0.021815

Generalization to a completely withheld operating scenario was assessed in seven leave-one-scenario-out (LOSO) experiments. In each fold, one scenario was excluded from imputation fitting, scaling, training, and validation. Full target-wise LOSO metrics are reported in Supplementary Table S16.

Table 10 summarizes the leave-one-scenario-out results obtained when each operating scenario was completely withheld from model development.

Table 10. Leave-one-scenario-out evaluation by macro-RMSE.

Completely Withheld Scenario	Proposed Model Macro-RMSE	ARX-like Ridge Macro-RMSE
CAB valve leak	0.499780	0.500460
Condenser efficiency decrease	0.314032	0.313496
FHN sensor drift	0.042815	0.042610
Heat exchanger fouling	0.265642	0.265510
NOC stable feed	0.019337	0.019297
NOC varying feed	0.250613	0.250491
Pressure drop increase	257.287292	257.287354
Scenario-balanced mean	36.954216	36.954174

LOSO does not support reliable extrapolation to unseen regimes. When the pressure-drop scenario is entirely withheld, both models fail catastrophically. For the proposed model, 68.4388% of the predicted components become negative and 22.1906% exceed 100 wt%. The architecture should therefore be regarded as an interpolation-oriented forecaster within represented simulated regimes, not as a validated predictor of previously unseen faults.

4.7. Forecast-Horizon Sensitivity

The architecture was retrained independently for 1, 5, 15, and 30 min horizons using the same 30 min lookback. For each horizon, causal windows, train-only preprocessing, ARX initialization, and validation-based checkpoint selection were recomputed. The complete sample-count and training audit is provided in Supplementary Table S17.

Table 11 reports the forecast-horizon sensitivity results for the 1, 5, 15, and 30 min prediction horizons.

At one minute, the models are nearly indistinguishable. At five minutes, ARX has a slightly lower RMSE while the hybrid has a slightly lower MAE. At 15 and 30 min, ARX has lower errors and higher R2. Under the current configuration, the deep residual branch therefore does not improve longer-horizon forecasting; the empirically supported application scope is short-horizon prediction at approximately 1–5 min.

Table 11. Forecasting performance across different prediction horizons.

Horizon	Proposed RMSE	ARX RMSE	Proposed MAE	ARX MAE	Proposed R2	ARX R2
1 min	0.008602	0.008608	0.006736	0.006741	0.996366	0.996361
5 min	0.022796	0.022751	0.017353	0.017394	0.967755	0.967819
15 min	0.061341	0.056038	0.046359	0.041839	0.804916	0.845666
30 min	0.101168	0.091227	0.073200	0.066379	0.593360	0.677493

4.8. Simpler ARX-Residual Comparator Models

To test whether the complete nonlinear branch is necessary, three simpler residual architectures were evaluated at seed 42 using the same ARX-based formulation. These are diagnostic single-seed comparisons rather than a statistically ranked benchmark. Table 12 compares the full proposed architecture with the simpler ARX+MLP, ARX+GRU, and ARX+TCN residual variants and the ARX-like Ridge reference at seed 42.

Table 12. Nonlinear residual-branch variants for seed 42.

Model	Validation RMSE	Historical-Test RMSE	MAE	R2	Residual-Branch Parameters
ARX+MLP residual	0.020527	0.022180	0.017170	0.968083	166,084
ARX+GRU residual	0.020576	0.022360	0.017209	0.967929	7812
ARX+TCN residual	0.020553	0.022558	0.017335	0.967923	14,020
Full proposed model	0.020643	0.022796	0.017353	0.967755	178,332 trainable
ARX-like Ridge	0.020699	0.022751	0.017394	0.967819	4924

All three simplified residual models have a similar or lower point RMSE than the complete architecture, with the lowest seed-42 value obtained by ARX+MLP. Because these variants were not repeated across multiple seeds, this experiment does not establish statistical ordering. It does, however, reinforce the central importance of the ARX-based formulation and shows that an independent advantage of the complete deep residual branch has not been demonstrated.

4.9. Consistency and Bounds Diagnostics

The auxiliary total-yield target is exactly the sum of the four primary targets; it is therefore not an independent physical measurement. The auxiliary, consistency, and bounds terms are treated as soft regularization components. Direct violation metrics were computed on the historical-test segment, on a three-seed ablation without these terms, and on the LOSO pressure-drop stress test. Table 13 summarizes the consistency and admissible-bound diagnostics for the historical-test evaluation, the regularization ablation, and the leave-one-scenario-out pressure-drop stress test.

Table 13. Consistency and admissible-bound diagnostics.

Configuration/Evaluation Domain	Negative Predictions (%)	Outputs > 100% (%)	Sum > 100% (%)	Prediction Range	Total-Yield MAE
Full model, historical test, 10 seeds	0	0	0	2.0066–48.3292	0.0596–0.0606
Without auxiliary/ consistency/bounds terms, 3 seeds	0	0	0	2.0097–48.3332	0.0601–0.0609
LOSO: pressure-drop scenario withheld	68.4388	22.1906	0	−329.9371–503.4765	197.4894

No bound violations occur on the historical-test segment either with the full regularized model or with the regularization terms removed. The corresponding three-seed RMSE values are also nearly identical (0.022823 versus 0.022819). Hence, the additional losses are not presented as a demonstrated source of accuracy or admissibility within known regimes.

The LOSO pressure-drop failure further shows that consistency is not guaranteed outside the training distribution.

4.10. Direct Model Interpretation and Exploratory Feature-Target Analysis

Interpretation is presented in two explicitly separated layers. First, scenario means, Pearson's correlation, mutual information, and Random-Forest TreeSHAP are used only to characterize exploratory feature-target associations in the simulated FCCU benchmark. Second, the proposed hybrid model itself is explained directly through grouped permutation-Shapley attribution. This separation avoids treating a separate tree model as a faithful surrogate of the hybrid architecture.

Figure 3 shows the scenario-wise mean values of the four product-yield targets. Gasoline proxy and LPG exhibit the strongest scenario-dependent variation, whereas light cycle oil and slurry remain within narrower ranges. In particular, the condenser-efficiency-decrease scenario is associated with a lower gasoline proxy yield and a higher LPG yield, indicating a redistribution of product outputs under this operating condition. This confirms that the product targets respond differently to scenario changes and supports the multi-output formulation of the forecasting task.

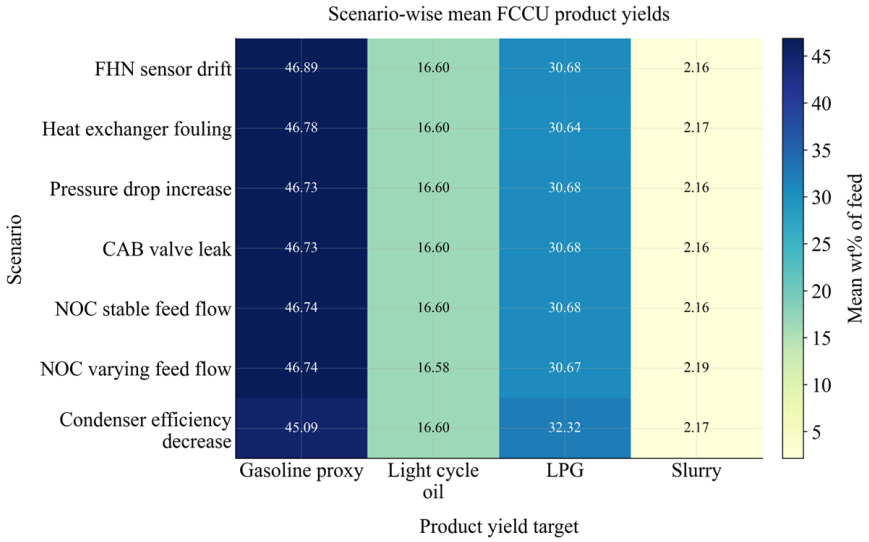


Figure 3. Scenario-wise mean FCC product yields.

Figure 4 presents the Pearson correlation matrix for selected FCC process variables and product-yield indicators. The matrix reveals interpretable linear association patterns between flow, catalyst-inventory, pressure, gas-composition, and product-yield variables. Several product outputs show opposite correlation directions, particularly gasoline proxy and LPG, which reflects the competitive redistribution of product fractions in the FCC process. However, some variables exhibit weak linear correlations, suggesting that linear dependence alone is insufficient to fully describe the process dynamics.

Figure 5 further examines feature–target dependence using mutual information, which can capture nonlinear relationships beyond the Pearson correlation. The results show that several process variables retain meaningful dependence with product-yield targets even when their linear correlations are limited. Variables related to feed flow, catalyst circulation, regenerator and stack-gas behavior, compressor operation, accumulator level, and fractionator valve control show notable associations with the target outputs. This indicates that the forecasting task involves both linear and nonlinear process relationships.

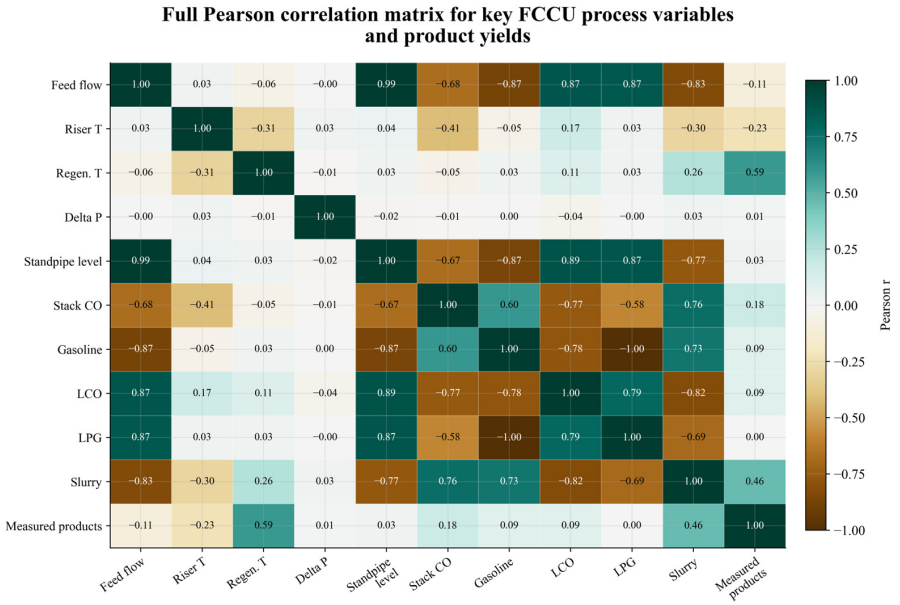


Figure 4. Pearson correlation matrix for key FCC process variables and product yields.

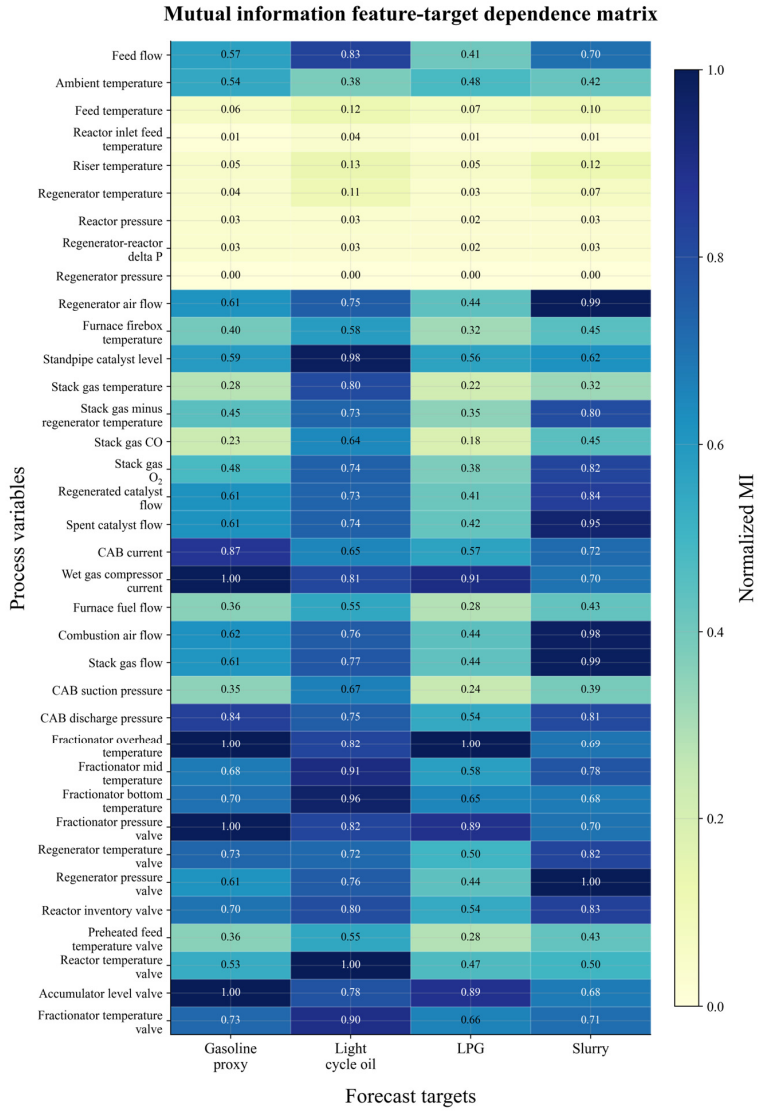


Figure 5. Mutual-information feature-target dependence matrix.

Figures 3–5 show structured scenario and feature-target relationships, but these exploratory associations do not by themselves establish that the full hybrid architecture is necessary. They are used to describe the forecasting problem rather than to claim direct internal attribution of the proposed model.

The TreeSHAP analyses in Figures 6–9 were computed from target-specific RandomForestRegressor models trained against the observed targets. These tree models do not reproduce the hybrid model predictions and are therefore not treated as surrogates of the proposed architecture. Their plots are retained only as exploratory summaries of nonlinear feature-target associations.

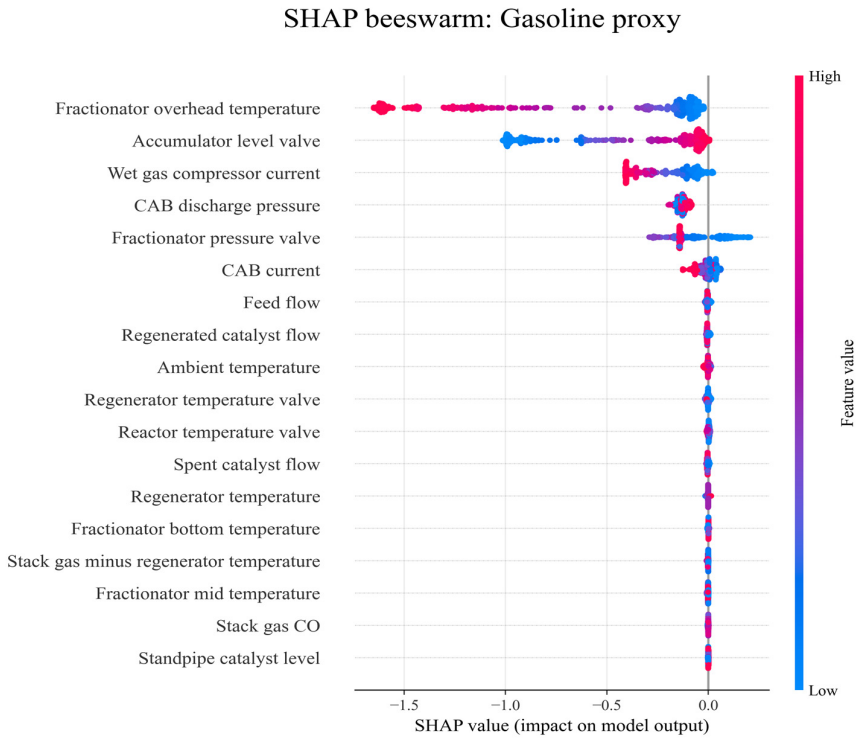


Figure 6. Exploratory TreeSHAP feature-target association summary for gasoline proxy using a target-specific RandomForestRegressor.

Figures 6–9 present the exploratory TreeSHAP association summaries for gasoline proxy, LCO, LPG, and slurry. The visual patterns describe relationships learned by the target-specific Random Forest models and should not be interpreted as explanations of the hybrid network's internal decisions.

Figure 10 shows the representative pressure-drop fault trajectory and the corresponding product-yield response. The trajectory illustrates that gasoline proxy gradually decreases and then partially recovers, whereas LPG demonstrates an opposite response pattern. Slurry remains within a narrower range but still reflects the operating-regime transition. This delayed and product-specific behavior supports the use of causal sequence models capable of representing process inertia and lagged effects.

Direct grouped permutation-Shapley analysis was then applied to the frozen proposed model (seed 42). Each of the 41 input channels was permuted as a group containing all 30 lags. The analysis used 28 scenario-stratified windows, 64 permutations, and 32 antithetic permutation pairs. The maximum Monte Carlo error was 0.001473 wt%-points and the mean error was 0.000148; the local-additivity error was zero to numerical precision.

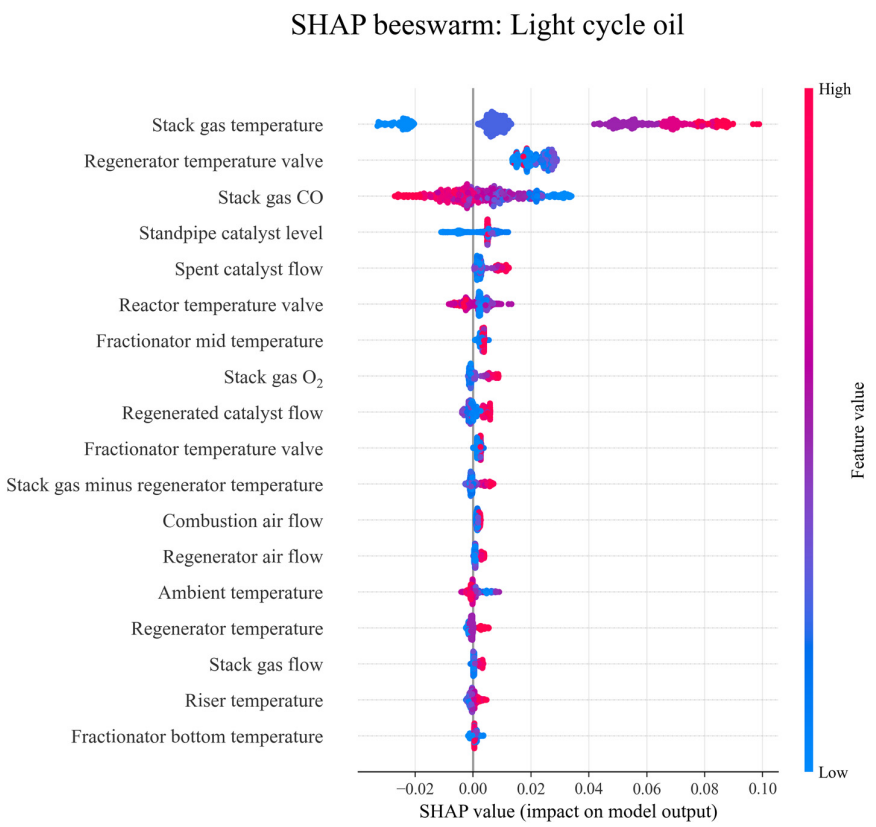


Figure 7. Exploratory TreeSHAP feature-target association summary for light cycle oil using a target-specific RandomForestRegressor.

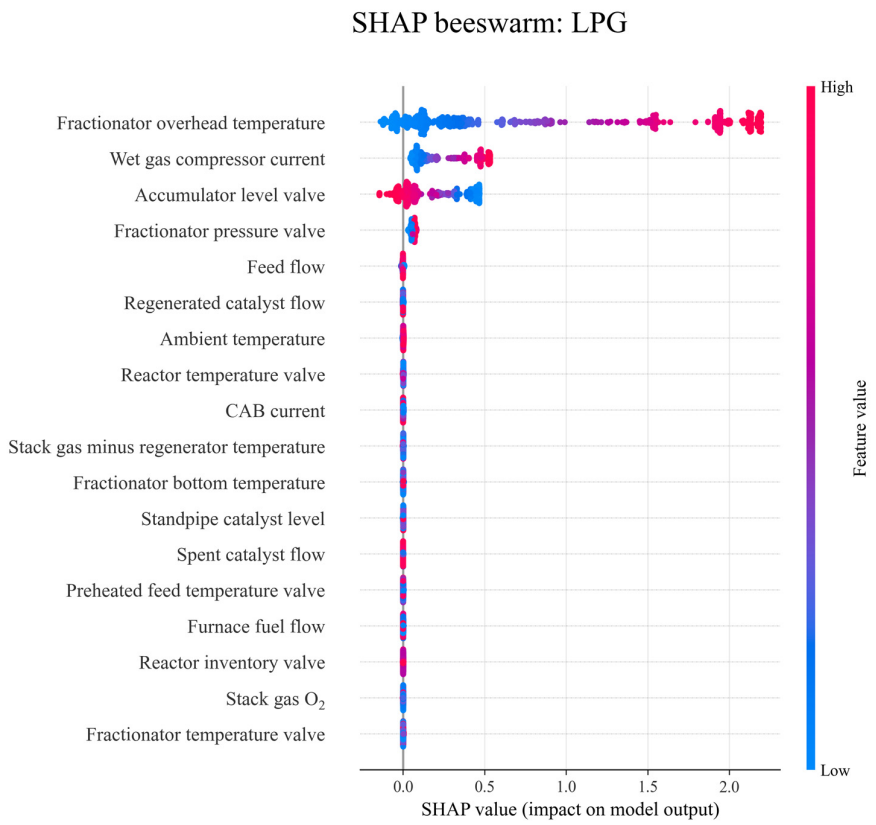


Figure 8. Exploratory TreeSHAP feature-target association summary for LPG using a target-specific RandomForestRegressor.

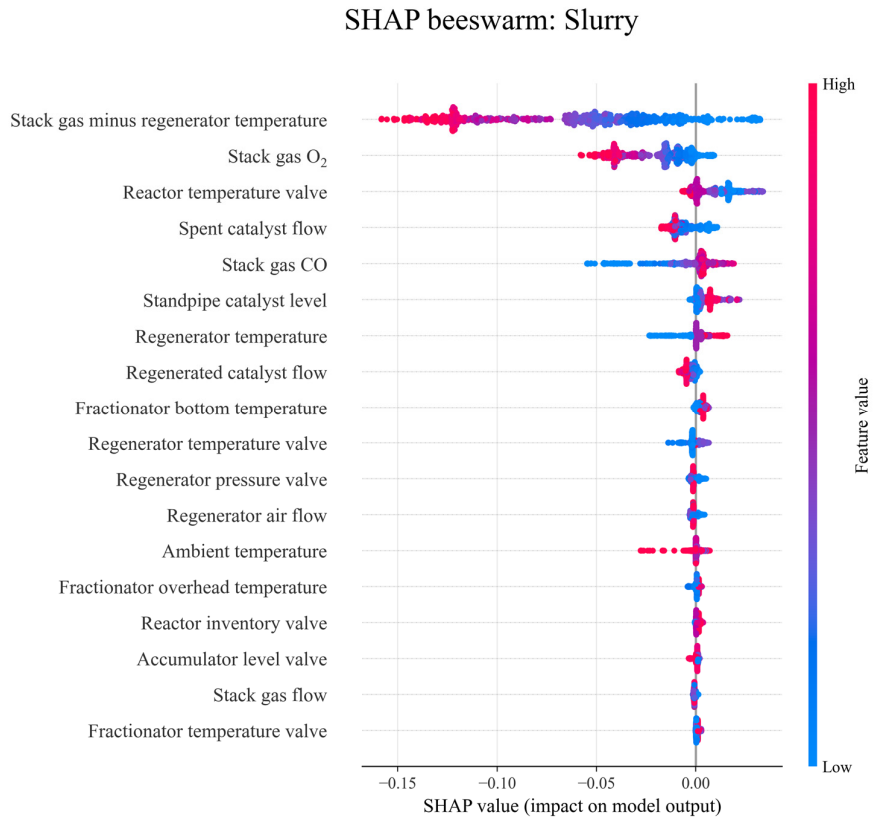


Figure 9. Exploratory TreeSHAP feature-target association summary for slurry using a target-specific RandomForestRegressor.

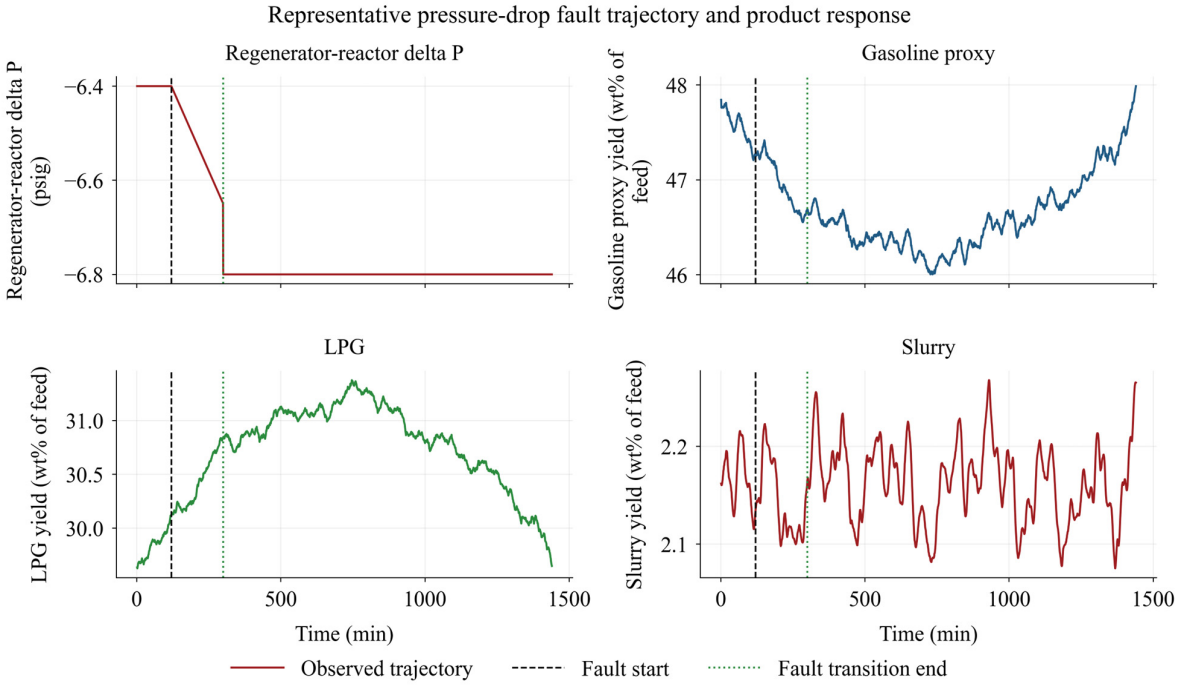


Figure 10. Representative pressure-drop fault trajectory and product-yield response.

Figure 11 shows the mean absolute grouped Shapley importance averaged across the four targets. The largest contribution is the historical gasoline-proxy channel (approximately 0.207 wt%-points), followed by accumulator-level valve V9 (0.110), historical LPG (0.089), fractionator-overhead temperature Tfra (0.072), and CAB discharge pressure P2

(0.054). The ranking indicates predictive sensitivity, not causal process effects. Target-specific top-15 group contributions are given in Supplementary Table S18.

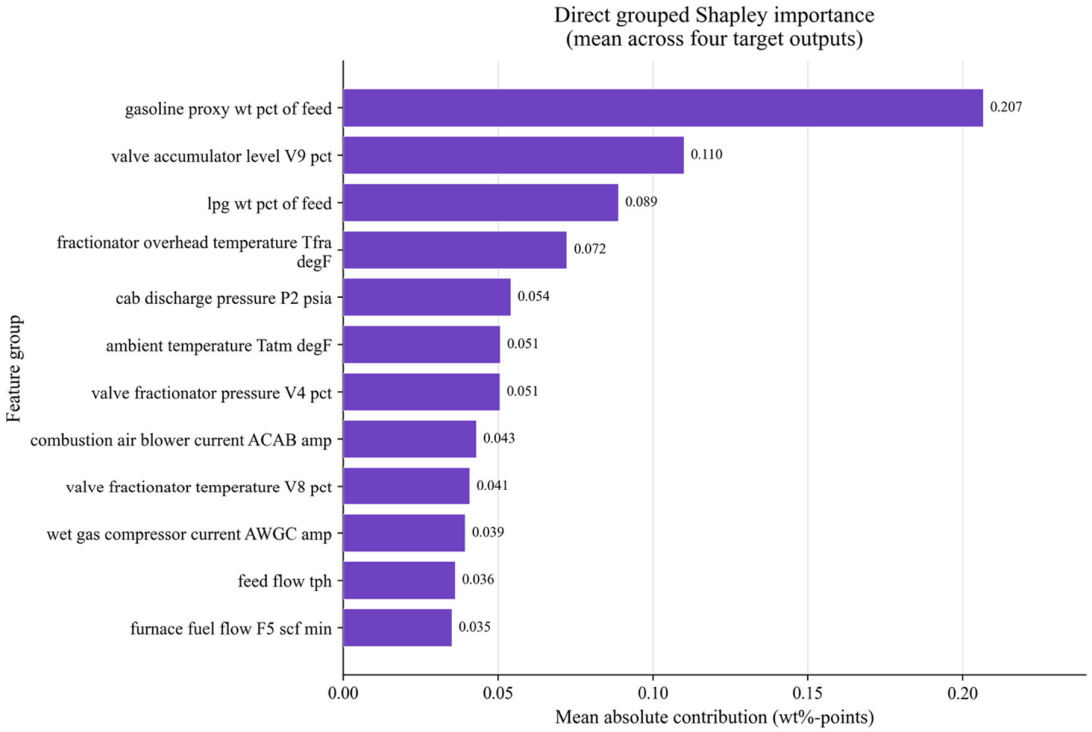


Figure 11. Direct grouped permutation-Shapley importance for the proposed model, averaged across four target outputs.

A separate ExtraTrees model was fitted only to quantify surrogate fidelity against the hybrid predictions. Its target-wise R2 values were 0.989549 for gasoline proxy, 0.668737 for LCO, 0.995762 for LPG, and 0.805008 for slurry (Supplementary Table S19). Because fidelity is substantially weaker for LCO and slurry, the surrogate is not used for the principal interpretability conclusions.

5. Discussion

The experimental results show that five-minute FCCU product-yield dynamics in this simulated benchmark are dominated by a strong autoregressive–exogenous structure. ARX-like Ridge obtains historical-test RMSE 0.022751, compared with 0.022809 ± 0.000116 for the proposed model across ten seeds. Dependence-aware analyses do not establish a consistent statistical advantage for either model across the evaluated runs. Learned residual-scaling coefficients remain close to zero, and removing the frozen ARX branch causes a large error increase. In contrast, removing individual deep components changes the RMSE only slightly and sometimes improves the point estimate.

The additional ARX-residual comparator results strengthen this interpretation. At seed 42, ARX+MLP, ARX+GRU, and ARX+TCN all match or improve on the full model's point RMSE, with ARX+MLP reaching 0.022180. Because these simplified comparisons were not repeated across seeds, they do not support a formal ranking, but they show that the complete deep residual branch is not required to obtain strong accuracy once the ARX prior is present. The contribution of this study is therefore framed as a carefully evaluated process-informed ARX-residual formulation rather than as a demonstration of architectural superiority.

Rolling-origin evaluation shows similarly close performance between the proposed model and ARX on later intervals of known scenario families. The more demanding LOSO evaluation changes the conclusion: both models fail severely when the pressure-drop regime is entirely absent from development data. This failure demonstrates that high interpolation accuracy within represented simulations does not imply robustness to unseen operating regimes.

The computational results also favor a cautious deployment interpretation. The hybrid model is compatible with the one-minute sampling cycle in absolute latency, but its median single-window latency is approximately 613 times that of ARX-like Ridge and this extra cost is not accompanied by a significant RMSE gain. If simplicity, latency, and comparable accuracy are primary engineering objectives, ARX-like Ridge is the more efficient choice. Use of either autoregressive formulation in a real refinery would additionally require timely availability of the four historical product-yield channels.

The consistency objective should also be interpreted conservatively. The auxiliary total yield is algebraically determined by the four primary targets and is not independent physical supervision. No historical-test bound violations occurred even after removing the auxiliary, consistency, and bounds terms, while the LOSO stress test produced severe violations outside the training distribution. The model is therefore described as process-informed and consistency-regularized rather than as a mechanistically constrained model claiming independent physical supervision.

Direct grouped permutation-Shapley analysis indicates that the frozen hybrid model relies strongly on autoregressive product histories while also responding to fractionator, accumulator, compressor, pressure, and other process-state variables. The separate Random-Forest TreeSHAP plots are retained only as exploratory feature-target associations. The limited ExtraTrees fidelity for LCO and slurry further justifies basing model-specific interpretation on direct attribution rather than on a surrogate.

The study has important limitations. It uses one simulated FCCU benchmark rather than independent industrial plant data and therefore does not capture the full range of analyzer delays, feed-composition changes, sensor noise, maintenance states, and unit-to-unit variability. LOSO evaluation reveals a severe extrapolation failure for an unseen pressure-drop scenario. The proposed model does not demonstrate statistically significant superiority over ARX-like Ridge, the residual-comparator and multi-horizon diagnostics are seed-specific, and the regularization weights are fixed structural coefficients rather than globally optimized physical parameters. Minute-level availability of historical product yields in an operating FCCU is also not established. Confirmatory validation requires newly prespecified simulation trajectories or independent industrial data.

6. Conclusions

This study developed a process-informed Hybrid ARX-Residual Deep Sequence Model for short-horizon multi-output forecasting of gasoline proxy, LCO, LPG, and slurry on an open simulated FCCU benchmark. The architecture combines a frozen ARX dynamic prior with causal temporal convolution, a three-layer GRU, causal self-attention, attention pooling, residual gating, and a consistency-regularized objective. Across ten independent seeds, the proposed model achieved historical-test RMSE 0.022809 $\pm$ 0.000116, MAE 0.017378 $\pm$ 0.000081, and R2 0.967491 $\pm$ 0.000344, while ARX-like Ridge achieved RMSE 0.022751. Dependence-aware statistical analyses did not establish a consistent statistical advantage of the proposed model over ARX-like Ridge. The ablation results identify the frozen ARX branch as the critical component, whereas the independent benefit of the remaining deep components is not established. Rolling-origin evaluation shows similar performance on later intervals of known scenarios, but LOSO evaluation does not support

reliable generalization to unseen regimes. The principal conclusion is therefore that correct autoregressive–exogenous structure is more important than added deep-sequence complexity for this short-horizon simulated FCCU task. External confirmation requires newly prespecified trajectories or independent industrial data.

Supplementary Materials: The following supporting information can be downloaded at https://www.mdpi.com/article/10.3390/a19090727/s1: Supplementary Tables S1–S21, including descriptive statistics, data-quality and leakage-control audits, model configurations, reproducibility results, scenario-resolved evaluations, forecast-horizon analyses, and interpretability diagnostics. Table S1: Descriptive statistics of target product-yield variables; Table S2: Data quality, chronology, and engineering consistency checks; Table S3: Target mean comparison between normal and faulty scenarios; Table S4: Target means across train, validation, and test subsets; Table S5: Largest covariate mean shifts relative to the training set; Table S6: Training configuration of the proposed neural sequence model; Table S7: Provenance and transformation of the FCCU dataset structure; Table S8: Complete list of 37 exogenous process features; Table S9: Full accounting of 75 columns of the analytical dataset; Table S10: Fixed configuration of the proposed model; Table S11: Distribution of learned residual-scaling coefficients across ten seeds; Table S12: Registry of classical and baseline model configurations; Table S13: Sequence-model configurations, hyperparameter-search space, tuning budget, and selection procedure; Table S14: Complete results for ten independent runs of the proposed model; Table S15: Complete historical-test metrics by product output and operating scenario (seed 42); Table S16: Scenario-balanced leave-one-scenario-out metrics by target output; Table S17: Sample-count audit and macro-metrics for multiple forecast horizons; Table S18: Fifteen largest direct grouped permutation-Shapley contributions for each target output; Table S19: ExtraTrees surrogate fidelity to the proposed-model predictions; Table S20: Seed-wise Diebold–Mariano HAC statistics for the proposed model versus ARX-like Ridge; Table S21: Exact scenario-wise cutoffs for the development-only rolling-origin evaluation.

Author Contributions: Conceptualization, G.U. and S.T. (Salimzhan Tassanbayev); methodology, K.K.; software, G.T. and S.T. (Saltanat Turgyn); validation, G.T., A.S., and A.M.; formal analysis, K.K. and G.A.; investigation, G.U.; resources, A.M.; data curation, S.T. (Salimzhan Tassanbayev) and S.T. (Saltanat Turgyn); writing—original draft preparation, G.T.; writing—review and editing, G.T. and G.A.; visualization, G.T.; supervision, G.U.; project administration, A.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Committee of Science of the Ministry of Science and Higher Education of the Republic of Kazakhstan, grant number AP23489120.

Data Availability Statement: The processed dataset, preprocessing and model implementations, experimental configurations, and executable code supporting the 10-trial Optuna search, multi-seed reproducibility analysis, dependence-aware Diebold–Mariano and moving-block-bootstrap comparisons, development-only rolling-origin evaluation, leave-one-scenario-out analysis, simplified ARX-residual comparators, multi-horizon evaluation, and direct grouped permutation-Shapley analysis are available in the authors' public GitHub repository: https://github.com/Guldek1987/Intelligent-Product-Yield-Forecasting-in-Fluid-Catalytic-Cracking-Using-Multivariate-Process-Data (accessed on 16 August 2026). Notebook 3 documents the executed hyperparameter search, whereas Notebook 4 contains the reviewer-closure experiments and associated saved outputs. Dependency information is provided in requirements.txt.

Acknowledgments: The authors acknowledge the use of ChatGPT (GPT-4/GPT-4o), an AI-based language model, and Grammarly (version v.1.2.256.1883) solely to improve the clarity, grammar, and readability of the manuscript. The authors reviewed, verified, and edited all AI-assisted content and take full responsibility for the final version of the manuscript.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ARX	Autoregressive with exogenous inputs
ANN	Artificial neural network
FCC	Fluid catalytic cracking
FCCU	Fluid catalytic cracking unit
GRU	Gated recurrent unit
LCO	Light cycle oil
LPG	Liquefied petroleum gas
MAE	Mean absolute error
ML	Machine learning
MLP	Multilayer perceptron
NLinear	Normalized linear model
R^2	Coefficient of determination
RMSE	Root mean squared error
SHAP	SHapley Additive exPlanations
sMAPE	Symmetric mean absolute percentage error
TCN	Temporal convolutional network
TreeSHAP	Tree-based SHapley Additive exPlanations
HAC	Heteroskedasticity and autocorrelation consistent
LOSO	Leave-one-scenario-out
NARX	Nonlinear autoregressive with exogenous inputs

References

1. Kadlec, P.; Gabrys, B.; Strandt, S. Data-driven soft sensors in the process industry. *Comput. Chem. Eng.* **2009**, *33*, 795–814. [CrossRef]
2. Qin, S.J. Survey on data-driven industrial process monitoring and diagnosis. *Annu. Rev. Control* **2012**, *36*, 220–234. [CrossRef]
3. Pinheiro, C.I.; Fernandes, J.L.; Domingues, L.; Chambel, A.J.; Graça, I.; Oliveira, N.M.; Cerqueira, H.S.; Ribeiro, F.R. Fluid catalytic cracking (FCC) process modeling, simulation, and control. *Ind. Eng. Chem. Res.* **2012**, *51*, 1–29. [CrossRef]
4. Machine Learning for Process Systems Engineering (ML-PSE). Fluid Catalytic Cracking Unit (FCCU) Dataset. Available online: https://mlforpse.com/fccu-dataset/ (accessed on 3 June 2026).
5. Machine Learning for Process Systems Engineering (ML-PSE). Fluid-Catalytic-Cracking-Unit-Dataset-for-Process-Monitoring-Evaluation. GitHub Repository. Available online: https://github.com/ML-PSE/Fluid-Catalytic-Cracking-Unit-Dataset-for-Process-Monitoring-Evaluation (accessed on 3 June 2026).
6. Santander, O.; Kuppuraj, V.; Harrison, C.A.; Baldea, M. An Open Source Fluid Catalytic Cracker-Fractionator Model to Support the Development and Benchmarking of Process Control, Machine Learning and Operation Strategies. *Comput. Chem. Eng.* **2022**, *164*, 107900. [CrossRef]
7. Baldea Group. "FCC-Fractionator," Open-Source Repository. Available online: https://github.com/Baldea-Group/FCC-Fractionator (accessed on 3 June 2026).
8. Long, J.; Li, T.; Yang, M.-L.; Hu, G.; Zhong, W. Hybrid Strategy Integrating Variable Selection and a Neural Network for Fluid Catalytic Cracking Modeling. *Ind. Eng. Chem. Res.* **2019**, *58*, 247–258. [CrossRef]
9. Yang, F.; Dai, C.; Tang, J.; Xuan, J.; Cao, J. A hybrid deep learning and mechanistic kinetics model for the prediction of fluid catalytic cracking performance. *Chem. Eng. Res. Des.* **2020**, *155*, 202–210. [CrossRef]
10. Yang, F.; Sang, Y.; Lv, J.; Cao, J. Prediction of gasoline yield in fluid catalytic cracking based on multiple level LSTM. *Chem. Eng. Res. Des.* **2022**, *185*, 119–129. [CrossRef]
11. Wei, B.; Tan, S.; Zhang, Q.; Zhou, H. A hybrid soft sensor for key product yield of FCC unit based on deep learning framework driven by data and process mechanism. *Chem. Eng. Res. Des.* **2024**, *202*, 429–443. [CrossRef]
12. He, G.; Luo, L.; Zhou, L.; Dai, Y.; Ji, X.; Guo, C.; Lu, Z. Deep learning prediction of yields of fluid catalytic cracking via differential evolutionary dual-stage attention-based LSTM. *Fuel* **2024**, *370*, 131826. [CrossRef]
13. Khaldi, M.K.; Al-Dhaifallah, M.; Taha, O. Artificial intelligence perspectives: A systematic literature review on modeling, control, and optimization of fluid catalytic cracking. *Alex. Eng. J.* **2023**, *80*, 294–314. [CrossRef]
14. Yang, F.; Xu, M.; Lei, W.; Lv, J. Artificial intelligence methods applied to catalytic cracking processes. *Big Data Min. Anal.* **2023**, *6*, 361–380. [CrossRef]

15. Acosta-López, J.G.; de Lasa, H. Artificial intelligence for hybrid modeling in fluid catalytic cracking (FCC). *Processes* **2024**, *12*, 61. [CrossRef]
16. Jiang, Y.; Yin, S.; Dong, J.; Kaynak, O. A review on soft sensors for monitoring, control, and optimization of industrial processes. *IEEE Sens. J.* **2021**, *21*, 12868–12881. [CrossRef]
17. Perera, Y.S.; Ratnaweera, D.A.A.C.; Dasanayaka, C.H.; Abeykoon, C. The role of artificial intelligence-driven soft sensors in advanced sustainable process industries: A critical review. *Eng. Appl. Artif. Intell.* **2023**, *121*, 105988. [CrossRef]
18. Bai, S.; Kolter, J.Z.; Koltun, V. An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling. *arXiv* **2018**, arXiv:1803.01271. [CrossRef]
19. Zhou, H.; Zhang, S.; Peng, J.; Zhang, S.; Li, J.; Xiong, H.; Zhang, W. Informer: Beyond efficient Transformer for long sequence time-series forecasting. *Proc. AAAI Conf. Artif. Intell.* **2021**, *35*, 11106–11115. [CrossRef]
20. Wu, H.; Xu, J.; Wang, J.; Long, M. Autoformer: Decomposition Transformers with Auto-Correlation for Long-Term Series Forecasting. In Proceedings of the 35th International Conference on Neural Information Processing Systems (NeurIPS 2021), Virtual Event, 6–14 December 2021; pp. 22419–22430. Available online: https://dl.acm.org/doi/10.5555/3540261.3541978 (accessed on 10 June 2026).
21. Zhou, T.; Ma, Z.; Wen, Q.; Wang, X.; Sun, L.; Jin, R. FEDformer: Frequency Enhanced Decomposed Transformer for Long-Term Series Forecasting. In Proceedings of the 39th International Conference on Machine Learning, Baltimore, MD, USA, 17–23 July 2022; PMLR, Volume 162, pp. 27268–27286. Available online: https://proceedings.mlr.press/v162/zhou22g.html (accessed on 10 June 2026).
22. Wen, Q.; Zhou, T.; Zhang, C.; Chen, W.; Ma, Z.; Yan, J.; Sun, L. Transformers in time series: A survey. In Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, Macao, China, 19–25 August 2023; pp. 6778–6786. [CrossRef]
23. Zeng, A.; Chen, M.; Zhang, L.; Xu, Q. Are Transformers effective for time series forecasting? *Proc. AAAI Conf. Artif. Intell.* **2023**, *37*, 11121–11128. [CrossRef]
24. Nie, Y.; Nguyen, N.H.; Sinthong, P.; Kalagnanam, J. A Time Series Is Worth 64 Words: Long-Term Forecasting with Transformers. In Proceedings of the Eleventh International Conference on Learning Representations (ICLR), Kigali, Rwanda, 1–5 May 2023; Available online: https://openreview.net/forum?id=Jbdc0vTOcol (accessed on 10 June 2026).
25. Lundberg, S.M.; Lee, S.-I. A Unified Approach to Interpreting Model Predictions. In *Advances in Neural Information Processing Systems*; Curran Associates, Inc.: Red Hook, NY, USA, 2017; Volume 30, pp. 4765–4774.
26. Narendra, K.S.; Parthasarathy, K. Identification and control of dynamical systems using neural networks. *IEEE Trans. Neural Netw.* **1990**, *1*, 4–27. [CrossRef]
27. Lai, G.; Chang, W.-C.; Yang, Y.; Liu, H. Modeling long- and short-term temporal patterns with deep neural networks. In Proceedings of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, Ann Arbor, MI, USA, 8–12 July 2018; pp. 95–104. [CrossRef]
28. Zhang, G.P. Time series forecasting using a hybrid ARIMA and neural network model. *Neurocomputing* **2003**, *50*, 159–175. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.