

People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research
University Aboubakr Belkaid Tlemcen
Faculty of Technology

Department of Electrical and Electronic Engineering
Specialty: Automation
Option: Automation and Industrial Computing

Master's Thesis

Under Ministerial Decrees No. 1275 and No. 008

Early Failure Detection in Industrial Systems Using
Physics-Informed Neural Networks (PINNs):
Focused on Short-Term Prediction

Presented by

Rayane BELBACHIR

Publicly defended on 22 juin 2026

<i>PRESIDENT:</i>	Mr. BENYAHIA Boumediene	Pr, University of Tlemcen
<i>EXAMINER:</i>	Mr. CHERKI Brahim	Pr, ESSAT Tlemcen
<i>EXAMINER:</i>	Mrs. KELKOUL Bahia	MCB, University of Tlemcen
<i>SUPERVISOR:</i>	Mrs. HANDOUZI Wahida	MCA, University of Tlemcen
<i>EXPERT I2E:</i>	Mr. MAACHOU Omar	Pr, University of Tlemcen
<i>SOCIOECONOMIC PARTNER:</i>	Mr. AMEUR Zouaoui	NAFTAL

Abstract

Industrial digital twins for oil and gas production systems require forecasting models that are both accurate and physically consistent, yet purely data-driven neural networks often violate underlying dynamical constraints over extended forecast horizons. A SINDy-regularised neural forecasting framework is proposed for downhole pressure prediction in gas-lifted wells. The approach integrates sparse system identification with physics-informed deep learning. A convolutional encoder compresses multi-variate telemetry windows into compact latent representations, from which a multi-layer perceptron decoder predicts future pressure at lead times of 30 to 120 seconds. Separately, the Sparse Identification of Nonlinear Dynamics (SINDy) discovers a parsimonious 22-term ordinary differential equation from smoothed operational data, selected via ℓ_1 -regularised regression followed by Ridge refitting from a 34-term observable-space candidate library incorporating polynomial, boundary-flow, and thermo-hydraulic surrogate terms.

The SINDy-discovered ODE is embedded in the training procedure as a physics-based regularisation penalty. Random collocation points are evaluated across the forecast horizon, where the mismatch between the network-predicted temporal derivative and the SINDy ODE is penalised. Evaluation is performed on Well 00019 of the 3W dataset under normal operating conditions. The baseline configuration achieves a validation R^2 of 0.9286 and a test R^2 of 0.8675 at the 120-second forecast horizon, with RMSE remaining below 0.098 MPa across all four horizons.

At $\lambda = 10^{-7}$, the validation data loss decreases by 24.28% and physical consistency improves by 17.41% relative to the unregularised baseline. Varying λ reveals a narrow effective regime: at $\lambda = 10^{-6}$, the 30 s test R^2 rises to 0.9731 at the expense of long-horizon accuracy; at $\lambda = 10^{-5}$, the test-set R^2 collapses to negative values.

SINDy-based physics regularisation, at an appropriately small weight, yields pressure forecasts with improved temporal consistency and long-horizon stability, without requiring a priori specification of the governing equations.

Keywords : Physics-informed machine learning, sparse identification of nonlinear dynamics (SINDy), neural forecasting, gas-lift oil wells, industrial digital twin, signal preprocessing, observability diagnostics.

Résumé

Les jumeaux numériques industriels pour les systèmes de production pétrolière et gazière nécessitent des modèles de prévision à la fois précis et physiquement cohérents. Cependant, les réseaux de neurones purement basés sur les données violent souvent les contraintes dynamiques sous-jacentes sur des horizons de prévision étendus. Un cadre de prévision neuronale régularisé par SINDy est proposé pour la prédiction de la pression de fond dans les puits activés par gaz (gas-lift). L'approche intègre l'identification parcimonieuse de systèmes avec l'apprentissage profond informé par la physique. Un encodeur convolutif compresse les fenêtres de télémétrie multivariées en représentations latentes compactes, à partir desquelles un décodeur perceptron multicouche prédit la pression future à des horizons de 30 à 120 secondes. Séparément, l'Identification Parsemée des Dynamiques Non Linéaires (SINDy) découvre une équation différentielle ordinaire (EDO) parcimonieuse à 22 termes à partir de données opérationnelles lissées, sélectionnée via une régression avec pénalité ℓ_1 suivie d'un ajustement Ridge à partir d'une bibliothèque candidate de 34 termes dans l'espace des observables intégrant des termes polynomiaux, d'écoulement aux limites et des substituts thermo-hydrauliques.

L'EDO découverte par SINDy est intégrée dans la procédure d'entraînement comme pénalité de régularisation basée sur la physique. Des points de collocation aléatoires sont évalués sur l'horizon de prévision, où l'écart entre la dérivée temporelle prédite par le réseau et l'EDO SINDy est pénalisé. L'évaluation est effectuée sur le Puits 00019 de la base de données 3W dans des conditions de fonctionnement normales. La configuration de base atteint un R^2 de validation de 0.9286 et un R^2 de test de 0.8675 à l'horizon de prévision de 120 secondes, avec un RMSE restant inférieur à 0.098 MPa sur l'ensemble des quatre horizons.

À $\lambda = 10^{-7}$, l'erreur de validation des données diminue de 24.28% et la cohérence physique s'améliore de 17.41% par rapport à la configuration de base non régularisée. La variation de λ révèle un régime effectif étroit : à $\lambda = 10^{-6}$, le R^2 de test à 30 s s'élève à 0.9731 au détriment de la précision à long horizon ; à $\lambda = 10^{-5}$, le R^2 de l'ensemble de test s'effondre à des valeurs négatives.

La régularisation physique basée sur SINDy, pondérée avec un poids adéquatement faible, permet d'obtenir des prévisions de pression avec une cohérence temporelle et une stabilité à long horizon améliorées, sans nécessiter la spécification *a priori* des équations directrices.

Mots-clés : Apprentissage automatique informé par la physique, identification parsemée des dynamiques non linéaires (SINDy), prévision neuronale, puits de pétrole gas-lift, jumeau numérique industriel, prétraitement du signal, diagnostics d'observabilité.

ACKNOWLEDGEMENTS

First and foremost, I thank **Allah**, the Most Merciful, for granting me the strength, patience, and perseverance to complete this work.

I wish to express my deepest gratitude to my supervisor, **Mrs. HANDOUZI Wahida**, for her continuous guidance, insightful corrections, and unwavering support throughout every stage of this project. Her expertise shaped the direction of this thesis.

I sincerely thank the members of the jury — **Mr. BENYAHIA Boumediene** (President), **Mr. CHERKI Brahim** (Examiner), and **Mrs. KELKOUL Bahia** (Examiner) — for accepting to evaluate this work and for their constructive feedback.

I also wish to thank **Mr. MAACHOU Omar**, expert from the i2E Tlemcen Incubator, for his mentorship and for facilitating the bridge between academic research and entrepreneurship.

I am deeply thankful to **Mr. AMEUR Zouaoui** of NAFTAL, the socio-economic partner, whose practical guidance during my internship brought the industrial perspective to this project.

I am grateful to **University Aboubakr Belkaid Tlemcen** and the Faculty of Technology for providing the academic environment and resources necessary for this research.

Rayane BELBACHIR

Tlemcen, June 2026

DEDICATION

I humbly dedicate this thesis:

*To my beloved parents,
whose endless love, patience, and structural sacrifices
have paved the way for my education and guided me to this milestone.*

*To my family,
for their continuous support and quiet encouragement
throughout my academic journey.*

*To my friends and classmates,
for the shared moments of study, collaborative efforts,
and constant motivation and friction.*

Contents

List of Abbreviations	VII
General Introduction	1
1 Theoretical Background and State of the Art	2
1.1 Machine Learning and Deep Learning in Scientific Computing	2
1.1.1 The Limitation of Pure Data-driven Models In Scientific Computing	2
1.1.2 Emergence of Physics-Informed Learning Approaches	2
1.1.3 Methodologies for Physics-Guided Learning	2
1.2 Physics-Informed Neural Networks	3
1.2.1 The Initial Framework	3
1.2.2 Architectural and Algorithmic Enhancements	4
1.3 Sparse Identification of Nonlinear Dynamics SINDy	7
1.3.1 Sparse System Identification	7
1.4 Numerical Foundations of Signal Preprocessing	9
1.4.1 Hampel Filter and Robust Scale Estimation	9
1.4.2 Savitzky–Golay Smoothing and Numerical Differentiation	11
1.4.3 Monotone Piecewise Cubic Hermite Interpolation (PCHIP)	12
1.4.4 Role in the Hybrid Identification Pipeline	13
1.5 Conclusion	14
2 Methodology	15
2.1 Methodology Overview	15
2.2 Data Acquisition and Processing Pipeline	16
2.2.1 Dataset Description and Operational Context	16
2.2.2 Well Selection and Data Quality Assessment	16
2.2.3 Data Cleaning and Filtering	20
2.2.4 Feature Engineering	23
2.2.5 Data Structuring for Learning	23
2.3 Architectural Design	24
2.3.1 Parametric Tanh Activation (PTanh)	24
2.3.2 Context Encoder (ConvEncoder)	25
2.3.3 Forecasting Decoder (ForecastingDecoder)	25
2.4 Physics Integration via SINDy	26
2.4.1 Sparse Model Identification via SINDy	26
2.4.2 Loss Function Design	28
2.5 Training	29
2.5.1 Training Loop and Hyperparameters	29
2.5.2 Training and Selection Strategy	29
2.5.3 Summary	30

3 Results and Discussion	31
3.1 Evaluation Protocol and Validation Metrics	31
3.1.1 Validation of Offline SINDy Identification	31
3.1.2 Neural Forecasting Performance Metrics	31
3.2 Offline Sparse System Identification	32
3.3 Neural Forecasting Model: Architecture Comparison and Training	34
3.3.1 Model Variants and Training Setup	34
3.3.2 Training Convergence	35
3.3.3 Architecture Selection	37
3.4 Effect of Physics Regularisation Weight	37
3.4.1 Regularization Saturation and Model Collapse	38
3.5 Test-Set Evaluation	38
3.5.1 Multi-Horizon Forecasting Performance	38
3.5.2 Chronological Rollout	38
3.6 Discussion	39
3.7 Conclusion	40
General conclusion	41
Appendix A: Domain Physics, SINDy Feature Library, and Observability Diagnostics	42
A.1 Measured Variables and Sensor Availability	42
A.2 Well Selection and Dynamic Excitation	43
A.3 Physics-Guided SINDy Feature Library	44
A.3.1 Baseline States and Control Variables (Terms 1–7)	45
A.3.2 Polynomial and Interaction Basis (Terms 8–21)	46
A.3.3 Boundary-Flow and Empirical Nonlinearities (Terms 22–30)	46
A.3.4 Thermodynamic and Hydraulic Surrogates (Terms 31–34)	46
Appendix B: Industrial Deployment and Business Perspectives	48
1. Value Proposition	49
2. Customer Segments	49
3. Customer Relationships	49
4. Channels	50
5. Key Partnerships	50
6. Key Activities	50
7. Key Resources	51
8. Cost Structure	51
9. Revenue Streams	51

List of Figures

1.1	Elements of physics-informed machine learning: data, model, and optimization integration pathways. Adapted from [1].	3
1.2	PINN architecture with automatic differentiation and composite data–PDE residual loss [2].	4
1.3	SINDy workflow from measured data to sparse coefficient selection [3].	7
2.1	Gas-lift actuation: downhole pressure P_{PDG} and gas injection rate Q_{GL}	18
2.2	Choke orifice dynamics: pressure drop and choke valve opening.	19
2.3	Reservoir-wellbore coupling: downhole pressure P_{PDG} and topside tree pressure P_{TPT}	19
2.4	Gas-lift clean profiles: downhole pressure P_{PDG} and gas lift rate Q_{GL}	21
2.5	Choke clean orifice dynamics: pressure drop ΔP and choke valve opening.	22
2.6	Reservoir-wellbore coupling clean profiles: downhole pressure P_{PDG} and tree pressure P_{TPT}	22
2.7	Forecasting model architecture with 1D CNN encoder and MLP decoder.	26
2.8	SINDy model identification pipeline from smoothed telemetry.	27
3.1	Non-overlapping evaluation sequence ($W = 60$ s, $H_{\max} = 120$ s).	32
3.2	Forward simulation of the SINDy ODE against measured downhole pressure.	33
3.3	Training curves for Config A (baseline). Data loss and physics loss (log scale). The dashed lines mark the physics-weight ramp (epochs 150–200).	35
3.4	Training curves for Config B (wider MLP). Data loss and physics loss (log scale). The dashed lines mark the physics-weight ramp (epochs 150–200).	35
3.5	Training curves for Config C (deeper CNN). Data loss and physics loss (log scale). The dashed lines mark the physics-weight ramp (epochs 150–200).	36
3.6	Test-set chronological rollout for measured and predicted pressure.	39
A.1	Well 00003: smoothed downhole pressure P_{PDG} over 143 231 samples.	44
A.2	Well 00007: smoothed P_{PDG} over 21 600 samples.	44
B.1	Pioneer AI: Digital twin intelligence for energy assets.	48

List of Tables

2.1	Percentage of valid samples for each key variable after cleaning.	17
2.2	Dynamic richness and actuation excitation of candidate wells.	18
3.1	Selected SINDy features and Ridge coefficients.	34
3.2	Validation metrics for the architecture variants ($\lambda_{\text{phys}} = 10^{-7}$ at best checkpoint). . . .	37
3.3	Physics loss weight sweep for Config A.	37
3.4	Test-set forecasting performance of Config A (MPa).	38
A.1	Sensor variables available in the 3W dataset.	42
A.2	Sensor availability across the candidate wells.	43
A.3	Observable-space SINDy candidate library.	45

List of Abbreviations

AD : Automatic Differentiation.

AdamW : Adaptive Moment Estimation with Decoupled Weight Decay.

AI : Artificial Intelligence.

ANN : Artificial Neural Network.

B-PINN : Bayesian Physics-Informed Neural Network.

CNN : Convolutional Neural Network.

DL : Deep Learning.

FFNN : Feedforward Neural Network.

IQR : Interquartile Range.

LASSO : Least Absolute Shrinkage and Selection Operator.

L-BFGS : Limited-memory Broyden–Fletcher–Goldfarb–Shanno.

LHS : Latin Hypercube Sampling.

MAD : Median Absolute Deviation.

MAE : Mean Absolute Error.

ML : Machine Learning.

MLP : Multi-Layer Perceptron.

MSE : Mean Squared Error.

ODE : Ordinary Differential Equation.

PCHIP : Piecewise Cubic Hermite Interpolating Polynomial.

PDE : Partial Differential Equation.

PINN : Physics-Informed Neural Network.

PTanh : Parametric Hyperbolic Tangent.

R² : Coefficient of Determination.

ReLU : Rectified Linear Unit.

RMSE : Root Mean Squared Error.

RNN : Recurrent Neural Network.

SG : Savitzky–Golay.

SINDy : Sparse Identification of Nonlinear Dynamics.

STLS : Sequential Thresholded Least Squares.

WSINDy : Weak-form Sparse Identification of Nonlinear Dynamics.

General Introduction

Industrial digital twins rely on forecasting models that are both accurate and physically consistent, yet purely data-driven neural networks often produce predictions that violate the underlying dynamical constraints of the systems they aim to represent [4]. In oil and gas production, gas-lifted wells exhibit complex thermo-hydraulic behaviour governed by coupled mass, momentum, and energy balances. Traditional high-fidelity simulators capture these dynamics but incur computational costs that preclude real-time state estimation and control [5]. Conversely, machine learning surrogates offer rapid inference but lack guarantees of physical plausibility over extended forecast horizons. Physics-informed machine learning has emerged as a response to this tension, embedding domain knowledge into neural architectures, loss functions, or training procedures to constrain predictions within physically admissible regimes [1].

This thesis investigates whether a sparse system-identified ordinary differential equation, discovered via the Sparse Identification of Nonlinear Dynamics (SINDy) framework [3], can serve as a physics-based regulariser during neural network training for downhole pressure forecasting. The approach combines a convolutional encoder and multi-layer perceptron decoder that maps multi-variate telemetry windows to future pressure estimates, with a SINDy-discovered ODE embedded as a continuous-time collocation constraint in the training objective. Candidate dynamics are drawn from an observable-space library of 34 nonlinear terms, using only the measured variables (pressure, flow rate, choke position) rather than unobserved latent mass states. Evaluation is performed on Well 00019 of the 3W dataset over approximately 40 hours of normal-operating-condition telemetry [6].

The manuscript is organised as follows: **Chapter 1** surveys the theoretical foundations of physics-informed machine learning, including the evolution of Physics-Informed Neural Networks (PINNs) from their initial formulation through architectural enhancements, loss balancing strategies, and causal training techniques. The chapter also reviews sparse system identification via SINDy and the mathematical principles of robust signal preprocessing Hampel filtering, Savitzky-Golay smoothing and differentiation, and monotonic piecewise cubic interpolation.

Chapter 2 describes the data pipeline: the 3W dataset and well selection criteria, the three-stage preprocessing chain (Hampel filtering, PCHIP interpolation, Savitzky-Golay smoothing), the encoder-decoder forecasting architecture, the SINDy identification procedure, the physics-informed collocation loss formulation, and the training strategy with a linear physics-weight ramp schedule.

Chapter 3 reports the experimental results: offline validation of the identified 22-term ODE, architecture comparison across three model variants, a regularisation-weight sensitivity analysis revealing a narrow effective range, and multi-horizon test evaluation through chronological rollout.

Appendix A provides the complete 34-term observable-space SINDy candidate library, together with the physical motivation and surrogate derivations for each term category.

Chapter 1

Theoretical Background and State of the Art

Traditional high-fidelity simulations incur computational costs that preclude real-time estimation in industrial settings [4], while standard optimisation-based controllers encounter latency bottlenecks from embedded numerical solvers [5]. This chapter surveys the transition from purely data-driven models to physics-informed methodologies, organised by the integration pathway through which physical knowledge is incorporated. The analysis traces the evolution of structural constraints, loss weighting strategies, and causal training techniques that enable convergence on industrially relevant dynamical systems.

1.1 Machine Learning and Deep Learning in Scientific Computing

1.1.1 The Limitation of Pure Data-driven Models In Scientific Computing

Machine learning (ML) and deep learning (DL) have become foundational technologies in fields such as computer vision, speech recognition, and natural language processing [4]. However, implementing these data-centric methods in scientific and engineering problems reveals fundamental limitations. As noted by Yu and Wang [7], they do not guarantee consistency with physical constraints and require substantial quantities of representative data to generalize reliably.

1.1.2 Emergence of Physics-Informed Learning Approaches

In response to these constraints, research has shifted toward physics-informed techniques, where neural models serve as surrogate state estimators or as part of hybrid architectures for nonlinear system identification [1]. These approaches provide an alternative to standard control methods in the observation layer, as demonstrated by Antonelo *et al.* [5] and Kittelsen *et al.* [8]. Industrial frameworks utilize these architectures to accelerate multi-physics simulations, reporting computational speedups compared to traditional numerical solvers [9].

1.1.3 Methodologies for Physics-Guided Learning

As categorised by Willard *et al.* [4] and Meng *et al.* [1], methodologies for physics-guided learning are organised by their application-centric objectives. These approaches address two principal modelling limitations: where mechanistic models are inadequate, prediction accuracy is enhanced through ML-augmented parameterisation; conversely, for systems that are computationally heavy due to structural complexity, resource efficiency is prioritised through reduced-order models and forward surrogates suitable for real-time control, as demonstrated by Antonelo *et al.* [5] and Hennigh *et al.* [9].

Beyond simulation, the necessity to discover hidden causal connections drives the adoption of inverse modeling, state estimation via governing equations discovery, and synthetic data generation, particularly in environments where direct measurement is challenging [8].

Meng *et al.* [1] along with Yu and Wang [7] organize the implementation methodologies of physics-informed machine learning into three distinct pathways by which physical knowledge is incorporated.

1. **Data-level integration:** physical constraints are used to generate, augment, or structure the training data — often through numerical simulators or synthetic datasets.
2. **Model-level integration:** physical knowledge is built directly into the architecture itself, using computation graphs that mirror numerical solvers, hybrid modules, or inductive biases that reflect conservation laws.
3. **Optimization-level integration:** physical consistency is enforced through constraint-based loss terms — by minimizing the residual of the governing equations, or through regularization strategies based on stability requirements.

Fig. 1.1 illustrates the physics-integration strategies discussed by Meng *et al.* [1]. The diagram provides an overview of how physical knowledge can be incorporated into machine learning systems across different stages of the pipeline, aligning with the three principal integration categories introduced earlier.

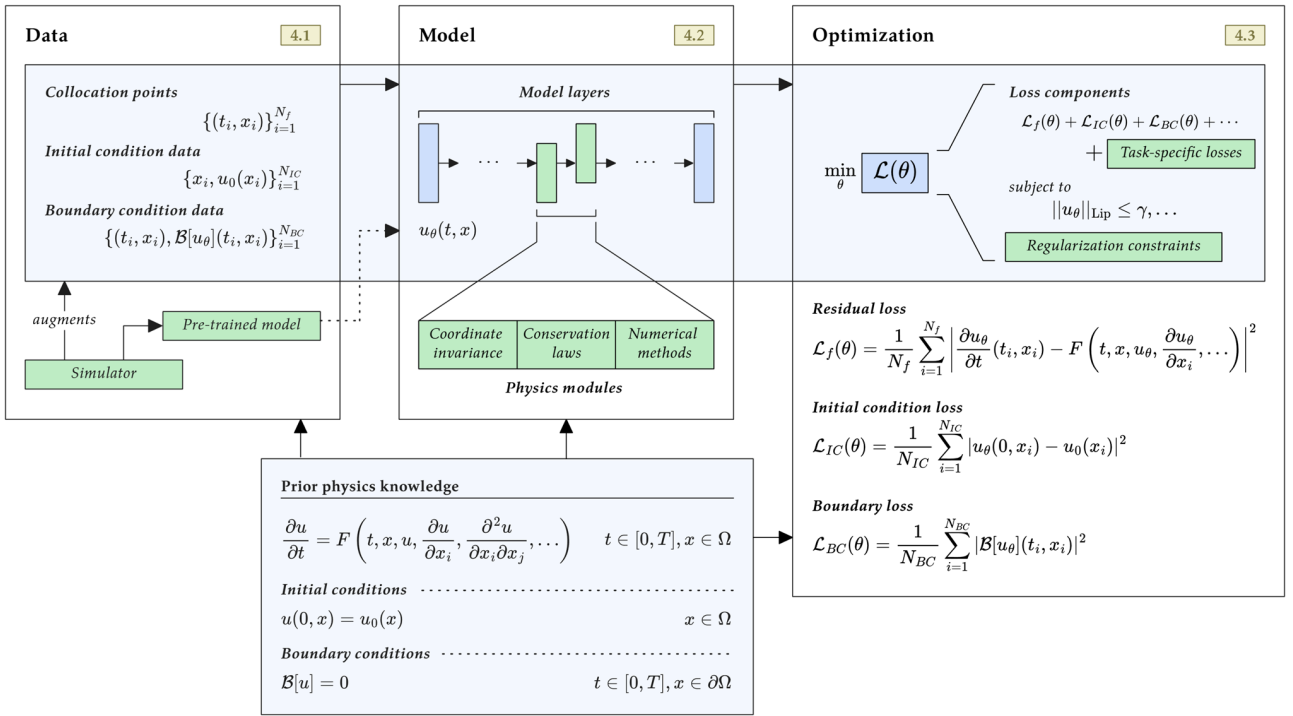


Fig. 1.1. Elements of physics-informed machine learning: data, model, and optimization integration pathways. Adapted from [1].

1.2 Physics-Informed Neural Networks

1.2.1 The Initial Framework

Among physics-informed learning frameworks, physics-informed neural networks (PINNs) provide a methodology for solving forward and inverse problems in scientific computing by designing data-driven algorithms for inferring solutions to nonlinear differential equations. Originally formalized by Raissi *et al.* [2], PINNs embed physical laws directly into the learning objective. By utilizing automatic differentiation to create a composite loss function that minimizes observational data error and the residual of governing differential equations, PINNs transform the solution of ODEs and PDEs into a constrained optimization problem. The general architecture is depicted in Fig. 1.2.

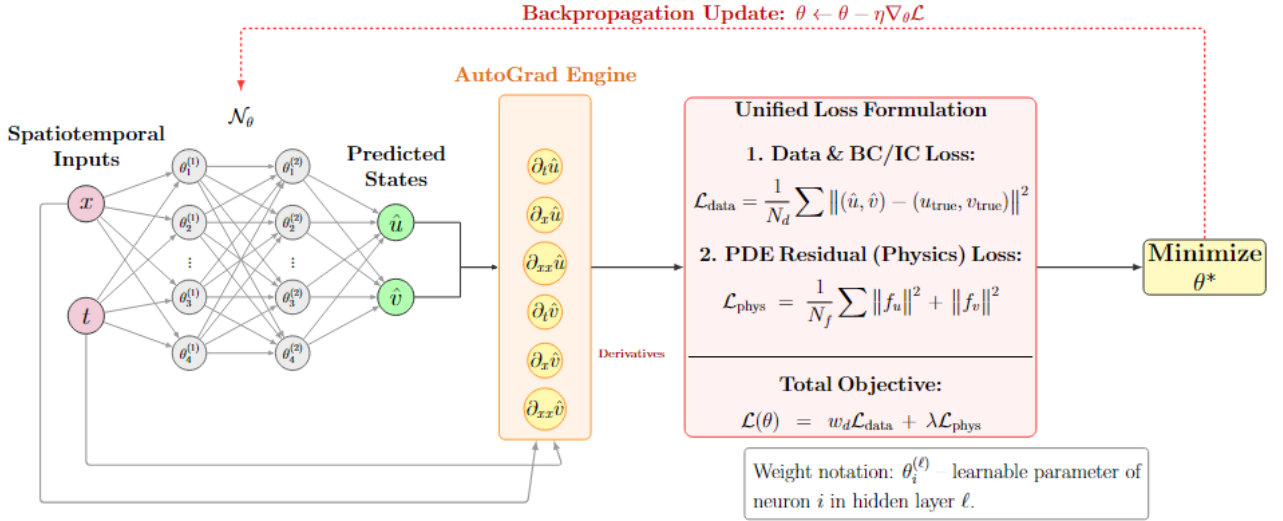


Fig. 1.2. PINN architecture with automatic differentiation and composite data–PDE residual loss [2].

1.2.2 Architectural and Algorithmic Enhancements

While the initial PINN framework demonstrated success on well-posed problems involving the data-driven solution and discovery of differential equations [2], its direct application to complex dynamics revealed significant convergence failures. As systematically illustrated by Krishnapriyan *et al.* [10], standard PINNs frequently exhibit fundamental failure modes in chaotic or moderately challenging regimes. This performance bottleneck is not necessarily attributable to limited neural network capacity, but rather to the formulation of physics as a soft regularisation constraint, coupled with the interaction between activation function curvature and the resulting loss landscape topology. This approach results in an increasingly complex and non-convex loss landscape that standard optimizers struggle to navigate. Furthermore, the authors highlight inefficiencies in the standard training approach where attempting to solve the entire space-time domain simultaneously often precludes convergence, suggesting that global optimization over the full temporal horizon is ill-suited for complex physical modeling.

These initial obstacles motivated subsequent research: in their survey, Cuomo *et al.* [11] documented that most work has focused on customising PINNs through activation functions, gradient optimisation techniques, neural network architectures, and loss function structures.

- Neural Network Architectures

As extensively discussed in their survey [11], researchers experimented with different NN architectures with the majority being feedforward neural networks (FFNN), nonetheless some researchers experimented with recurrent neural networks (RNN), generative adversarial neural networks (GAN) and convolutional neural networks (CNNs) to understand how would that affect the PINNs performance. In Raissi *et al.* paper [2] different DNN topologies were used such as a 5-layer deep neural network with 100 neurons per layer, a DNN with 4 hidden layers and 200 neurons per layer, each for a different problem. Furthermore, a number of research papers have addressed PINNs with multiple fully connected networks blended, such as to approximate specific equations of a larger physical model as did Haghighat *et al.* [12]. Separately, Jagtap *et al.* [13] demonstrated that introducing trainable scaling parameters within activation functions of the form $\sigma(n \cdot a \cdot x)$ accelerates convergence by counteracting the spectral bias of deep networks toward low-frequency components.

- Optimization Strategies and Training Dynamics

The practical deployment of PINNs requires careful handling of the optimisation landscape, which is shaped by the interplay between the data fidelity term and the physics residual term. Raissi *et al.* [2] formulated this as a composite loss function defined over two distinct point sets:

$$\mathcal{L}(\theta) = \mathcal{L}_{\text{data}}(\theta) + \mathcal{L}_{\text{phys}}(\theta), \quad (1.1)$$

where the data loss $\mathcal{L}_{\text{data}}$ is computed over N_u observed points $\{t_u^i, x_u^i, u^i\}_{i=1}^{N_u}$ and the physics loss $\mathcal{L}_{\text{phys}}$ is evaluated over N_f collocation points $\{t_f^i, x_f^i\}_{i=1}^{N_f}$ distributed across the domain:

$$\mathcal{L}_{\text{data}}(\theta) = \frac{1}{N_u} \sum_{i=1}^{N_u} |u_\theta(t_u^i, x_u^i) - u^i|^2, \quad (1.2)$$

$$\mathcal{L}_{\text{phys}}(\theta) = \frac{1}{N_f} \sum_{i=1}^{N_f} \left| \frac{\partial u_\theta}{\partial t}(t_f^i, x_f^i) + \mathcal{N}[u_\theta](t_f^i, x_f^i) \right|^2. \quad (1.3)$$

The neural network $u_\theta(t, x)$ is usually a fully connected feed-forward network with hyperbolic tangent (tanh) activations. tanh stays smooth under repeated differentiation, which is needed to evaluate the differential operator $\mathcal{N}[u_\theta]$ through automatic differentiation (AD). The spatial derivatives $\partial u_\theta / \partial x$, $\partial^2 u_\theta / \partial x^2$, and the temporal derivative $\partial u_\theta / \partial t$ come from back-propagating through the computational graph, eliminating the need for manual derivative computation or mesh discretization. Collocation points are usually sampled with a Latin Hypercube Sampling (LHS) strategy to cover the domain evenly, rather than using uniform grids [11].

Training procedure Training occurs in two stages. First, the Adam optimizer runs for a set number of iterations; its adaptive learning rates and momentum help it move through the non-convex loss landscape without getting stuck early in a poor local minimum. Second, the L-BFGS quasi-Newton optimiser refines the solution using second-order curvature information, converging tightly to a local minimum [2]. For larger datasets, mini-batch stochastic gradient descent can replace full-batch L-BFGS in the first stage to cut memory use. The authors note they used no extra classical regularisers (like L_1/L_2 weight penalties or dropout) — the PDE residual term $\mathcal{L}_{\text{phys}}$ acts as the physical regulariser on its own.

Loss balancing and gradient pathologies A well-documented difficulty arises from the multi-objective nature of the loss function. Because the gradient magnitudes of $\mathcal{L}_{\text{data}}$ and $\mathcal{L}_{\text{phys}}$ can differ by orders of magnitude, the optimiser may prioritise one term at the expense of the other. This imbalance can stall convergence or produce solutions that satisfy the PDE residual while ignoring boundary conditions (or vice versa). As synthesised by Cuomo *et al.* [11], methods such as global learning rate annealing and gradient normalisation have been developed to dynamically balance the weights of these loss terms during training. Specifically, the loss weights $\omega_F, \omega_B, \omega_d$ are adjusted based on the back-propagated gradient magnitudes to maintain balanced training dynamics.

Theoretical error decomposition To formally characterize the accuracy of a trained PINN predictor $\hat{u}_\theta^*$, Cuomo *et al.* [11] decompose the global approximation risk $\mathcal{R}[\hat{u}_\theta^*]$, measured in the L^2 -norm over the continuum domain, into three distinct error components:

$$\mathcal{R}[\hat{u}_\theta^*] \leq \mathcal{E}_O + 2\mathcal{E}_G + \mathcal{E}_A, \quad (1.4)$$

where $\mathcal{E}_O$ is the optimisation error arising from the inability of gradient-based solvers to locate the global minimum of the non-convex empirical risk landscape, $\mathcal{E}_G$ is the generalisation error representing the supremum of the discrepancy between the true continuum risk and its discretised empirical estimate, and $\mathcal{E}_A$ is the approximation error reflecting the fundamental limitation of the chosen network hypothesis class in expressing the exact solution. This decomposition clarifies that improving PINN performance requires addressing all three error sources simultaneously, not merely increasing network capacity.

Hard constraint encoding An alternative to soft penalty-based boundary enforcement is the direct embedding of boundary and initial conditions into the network ansatz. In this approach, known as Physics-Constrained Neural Networks (PCNNs), the predicted solution is expressed as:

$$\hat{u}_\theta(x) = A(x) + D(x)\mathcal{N}_\theta(x), \quad (1.5)$$

where $A(x)$ is an analytic function satisfying the prescribed boundary values, $D(x)$ is a smooth distance function that vanishes on the domain boundary, and $\mathcal{N}_\theta(x)$ is the unconstrained neural network output. By construction, $\hat{u}_\theta(x)$ satisfies the boundary conditions exactly, which removes the boundary loss term from the optimisation and simplifies the loss landscape [11]. This hard-constraint formulation eliminates the competing gradient problem between boundary and interior loss terms.

Causality and temporal propagation : When time is treated as an additional spatial coordinate, as in standard continuous-time PINN formulations, the optimiser may violate physical causality. Information from the initial condition at $t = 0$ does not necessarily propagate forward in time; the network may converge to a trivial steady-state solution in the interior while satisfying boundary conditions at the domain edges. This failure mode is particularly severe for convection-dominated and stiff reaction systems. Krishnapriyan *et al.* [10] demonstrated that for the one-dimensional convection equation $u_t + \beta u_x = 0$, the relative L_2 error approaches 100% when $\beta \geq 10$, and for the reaction-diffusion equation $u_t - \nu u_{xx} - \rho u(1 - u) = 0$, the error reaches 93% at $\rho = 5, \nu = 5$. Through numerical perturbation analysis, they derived that the condition number of the convection operator scales as $\mathcal{O}(\beta N)$, and because PINNs employ an L_2 loss, the optimisation conditioning scales quadratically as $\mathcal{O}(\beta N)^2$. For diffusion operators, the scaling is even steeper at $\mathcal{O}(\nu N^2)^2$.

To restore temporal causality, Wang *et al.* [14] introduced causal training algorithms that introduce a weighted loss:

$$\mathcal{L}_{\text{phys}}(\theta) = \frac{1}{N_t} \sum_{i=1}^{N_t} w_i \mathcal{L}_{\text{phys}}(t_i, \theta), \quad (1.6)$$

where the weights w_i depend on the cumulative residual error at all preceding time steps, ensuring that the network learns the solution sequentially from the initial condition forward. This reformulation has demonstrated accuracy improvements of 10–100× over non-causal baselines.

Curriculum training and time-marching. An orthogonal strategy to stabilise training is curriculum regularisation [10], in which the physical coefficients (e.g., the convection velocity β in $u_t + \beta u_x = 0$) are gradually increased from a diffusive, well-behaved regime to the target physical value. The network first learns the solution on a simpler problem, and the converged weights serve as the initialisation for the next, more challenging regime. Empirically, this cuts the relative L_2 error from about 90% down to below 3% on benchmark convection problems. Similarly, sequence-to-sequence time-marching breaks a long temporal domain into short segments: the network is trained on each segment in turn, and the prediction at the end of one segment becomes the initial condition for the next. This makes the local optimisation problem less stiff and avoids the error build-up that comes with training over the whole space-time domain at once. For the reaction-diffusion problem at $\rho = 5, \nu = 5$, moving from full space-time training to a seq2seq formulation with $\Delta t = 0.05$ reduces the relative L_2 error from 93.5% to 2.36%. However, both methods carry caveats: curriculum sequences must be manually tuned, and time-marching is susceptible to sequential error accumulation over long horizons.

Summary of optimisation landscape : The optimisation of PINNs therefore requires a multi-faceted strategy: (i) a two-stage optimiser schedule (Adam $\rightarrow$ L-BFGS) to navigate non-convexity, (ii) dynamic loss balancing to prevent gradient dominance by any single term, (iii) hard constraint encoding or causal weighting to enforce boundary and temporal consistency, and (iv) curriculum or

time-marching schemes to reduce the stiffness of the underlying differential operators. These techniques, developed across the PINN literature, collectively transform an otherwise brittle optimisation problem into a mathematical framework for solving forward and inverse problems governed by non-linear PDEs.

1.3 Sparse Identification of Nonlinear Dynamics SINDy

1.3.1 Sparse System Identification

Physics-informed methodologies depend on governing equations that are often partially unknown in complex industrial environments. Sparse Identification of Nonlinear Dynamics (SINDy) addresses this through a sparse regression formulation over a library of candidate nonlinear terms [3]. Under the assumption of structural parsimony, threshold-based optimization isolates the dominant physical features, yielding closed-form ordinary differential equations (ODEs) directly from telemetry. Figure 1.3 illustrates the transformation of noisy time series into a sparse ODE.

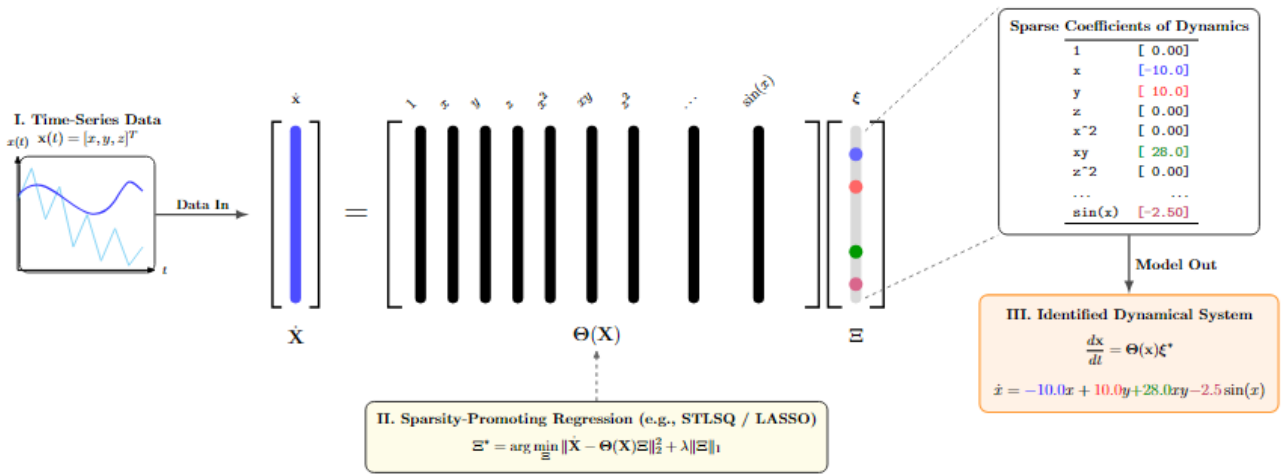


Fig. 1.3. SINDy workflow from measured data to sparse coefficient selection [3].

Algorithmic Formulation

The SINDy procedure is defined by the following sequential operations.

Step 1: State and derivative matrices. Consider a dynamical system with state vector $\mathbf{x}(t) = [x_1(t), x_2(t), \dots, x_n(t)]^T \in \mathbb{R}^n$. From m discrete time snapshots $t_1, t_2, \dots, t_m$, the measurements are arranged into the state history matrix $\mathbf{X}$ and the corresponding time-derivative matrix $\dot{\mathbf{X}}$:

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}^T(t_1) \\ \mathbf{x}^T(t_2) \\ \vdots \\ \mathbf{x}^T(t_m) \end{bmatrix} \in \mathbb{R}^{m \times n}, \quad \dot{\mathbf{X}} = \begin{bmatrix} \dot{\mathbf{x}}^T(t_1) \\ \dot{\mathbf{x}}^T(t_2) \\ \vdots \\ \dot{\mathbf{x}}^T(t_m) \end{bmatrix} \in \mathbb{R}^{m \times n}. \quad (1.7)$$

If the derivatives are not measured directly, they must be estimated numerically from $\mathbf{X}$ (e.g., by total-variation regularised differentiation or polynomial interpolation); the quality of this estimation critically affects the subsequent identification [3].

Step 2: Candidate function library. An overcomplete library $\Theta(\mathbf{X}) \in \mathbb{R}^{m \times p}$ is constructed from the columns of $\mathbf{X}$. The library typically includes:

- a constant term $\mathbf{1} \in \mathbb{R}^{m \times 1}$,

- linear state terms $\mathbf{X}$,
- quadratic nonlinearities $\mathbf{X}^{\otimes 2}$ containing all unique monomials $x_i x_j$,
- higher-order polynomials up to a chosen degree,
- trigonometric functions (e.g., $\sin(\mathbf{X})$, $\cos(\mathbf{X})$), and other physically motivated basis functions.

For n states and a polynomial degree d , the number of candidate functions p grows combinatorially; the library is intentionally overcomplete to increase the likelihood that the true governing terms are contained within it.

Step 3: Linear system formulation. The unknown dynamics are expressed as a linear combination of the library functions,

$$\dot{\mathbf{X}} = \Theta(\mathbf{X}) \Xi, \quad (1.8)$$

where $\Xi = [\xi_1 \ \xi_2 \ \dots \ \xi_n] \in \mathbb{R}^{p \times n}$ is a coefficient matrix. Each column ξ_k contains the coefficients for the k -th state equation. Because physical systems are typically governed by only a few dominant terms, Ξ is assumed to be sparse.

Step 4: Sparse regression. To obtain a parsimonious model, a sparsity-promoting regression is solved column-wise. The ℓ_1 -regularised least-squares (LASSO) formulation is

$$\xi_k = \arg \min_{\xi_k} \left\| \dot{\mathbf{X}}_k - \Theta(\mathbf{X}) \xi_k \right\|_2^2 + \alpha \|\xi_k\|_1, \quad (1.9)$$

where α controls the trade-off between reconstruction fidelity and sparsity. The regularisation parameter α governs the sparsity of the recovered coefficient vector: larger values of α produce fewer active terms, enforcing structural parsimony. In practice, the optimal α is chosen using K -fold cross-validation [15]: this tests a grid of candidate values and keeps the one that gives the lowest prediction error across the data partitions, balancing model complexity against how well it reconstructs the data.

Step 5: Equation recovery and validation. Once the sparse coefficient matrix Ξ is obtained, the governing equations are extracted from the non-zero entries. For the k -th state,

$$\dot{x}_k = \sum_{j \in \mathcal{A}_k} \xi_{jk} \Theta_j(\mathbf{x}). \quad (1.10)$$

The identified model is validated by integrating the closed-loop system from initial conditions and comparing the simulated trajectories against measured data. To ensure unbiased coefficients, the terms selected via LASSO are refitted using Ridge regression (ℓ_2 -penalty) prior to final evaluation.

Illustrative example: Lorenz system. The chaotic Lorenz attractor,

$$\begin{aligned} \dot{x} &= \sigma(y - x), \\ \dot{y} &= x(\rho - z) - y, \\ \dot{z} &= xy - \beta z, \end{aligned} \quad (1.11)$$

is a canonical benchmark. Given time-series data for (x, y, z) , SINDy with a polynomial library up to degree 2 (including terms $x, y, z, x^2, xy, xz, y^2, yz, z^2$) correctly identifies the non-zero coefficients: the linear terms $-\sigma x, \sigma y$, the quadratic terms $xy, -xz$, and the linear damping $-\beta z$. The algorithm recovers the attractor geometry, demonstrating that capturing the structural form of the dynamics is more critical than precise pointwise trajectory prediction [3].

The complete SINDy pipeline can therefore be summarised as follows:

1. Acquire state time-series data $\mathbf{X}$ and, if possible, derivatives $\dot{\mathbf{X}}$ (or estimate them numerically).

2. (Optional) Denoise $\mathbf{X}$ using singular-value hard thresholding or total-variation regularisation to improve derivative estimates.
3. Construct the candidate library $\Theta(\mathbf{X})$ from the chosen basis functions.
4. Solve the sparse regression $\dot{\mathbf{X}} = \Theta(\mathbf{X})\Xi$ via LASSO (1.9).
5. Tune the sparsity parameter λ (or α) by examining the Pareto frontier of model complexity versus accuracy.
6. Recover the symbolic differential equations and validate through forward integration.

The final output is a symbolic differential equation that is interpretable, ensuring that discovered models respect physical boundaries and can be explicitly validated. This algorithmic framework provides a systematic, data-driven route to interpretable dynamical models, requiring only that the true dynamics be sparse in the chosen function library.

The base SINDy framework described above, employing LASSO screening followed by Ridge re-fitting, constitutes the identification pipeline employed in this work.¹

1.4 Numerical Foundations of Signal Preprocessing

Industrial telemetry—such as pressure, flow rate, and valve position measurements—is subject to sensor noise, intermittent dropouts, and impulsive artifacts. In the context of system identification, these imperfections are amplified during numerical differentiation, which can destabilize sparse regression and degrade gradient computations. The signal-conditioning pipeline utilized in this framework involves three complementary techniques: the Hampel filter for outlier rejection, the Savitzky–Golay filter for smoothing and derivative estimation, and piecewise cubic Hermite interpolation for gap filling.

1.4.1 Hampel Filter and Robust Scale Estimation

The detection and removal of impulsive outliers rely on estimators that are resistant to the very anomalies they seek to eliminate. Hampel [18] introduced a general framework for robust estimation based on the *influence curve* (IC), which quantifies the asymptotic bias introduced by an infinitesimal point-mass contamination at a point x :

$$\text{IC}_{T,F}(x) = \lim_{\epsilon \downarrow 0} \frac{T[(1 - \epsilon)F + \epsilon\delta_x] - T(F)}{\epsilon}, \quad (1.12)$$

where T is an estimator functional, F is the nominal distribution, and δ_x is a Dirac measure at x . From the influence curve, three practical sensitivity measures are derived:

- **Gross-Error Sensitivity** $\gamma^* = \sup_x |\text{IC}_{T,F}(x)|$, the worst-case bias caused by a single outlier;
- **Local-Shift Sensitivity** $\lambda^* = \sup_{x \neq y} |\text{IC}_{T,F}(x) - \text{IC}_{T,F}(y)|/|x - y|$, which captures sensitivity to rounding or small perturbations;
- **Rejection Point** ρ^* , the distance beyond which $\text{IC}_{T,F}(x) = 0$, i.e., where observations are entirely discounted.

A central concept in robust scale estimation is the **Median Absolute Deviation** (MAD), defined as

$$\text{MAD} = \text{median}_i |X_i - \text{median}_j X_j|. \quad (1.13)$$

¹For completeness, related extensions include the Weak SINDy (WSINDy) framework [16], which improves noise robustness via integration by parts against compactly supported test functions, and SINDyc [17], which incorporates external control inputs as additional library terms. These extensions are not employed in the present implementation.

Unlike the classical standard deviation, which has a breakdown point of zero (a single extreme value can drive it to infinity), the MAD achieves the theoretical maximum breakdown point of $\delta^* = 0.5$. This means that up to half of the data can be contaminated before the MAD becomes arbitrarily biased. Moreover, while an outlier's influence on the sample variance grows quadratically, its influence on the MAD is strictly bounded. These properties make the MAD the preferred scale estimator for rolling-window outlier detection in industrial signals.

To combine high efficiency under Gaussian noise with strict outlier rejection, Hampel proposed the three-part descending M -estimator (the *Hampel ψ -function*). Scaled by the local MAD, the influence of an observation r is moderated as

$$\psi_{abc}(r) = \begin{cases} r, & |r| \leq a, \\ a \cdot \text{sign}(r), & a < |r| \leq b, \\ a \cdot \frac{r - c \cdot \text{sign}(r)}{b - c}, & b < |r| \leq c, \\ 0, & |r| \geq c, \end{cases} \quad (1.14)$$

where the thresholds $a < b < c$ are typically expressed in multiples of the local MAD. Observations in the central region are treated linearly; those in the intermediate band are Huberised (their influence is bounded); those in the taper zone have their weight smoothly reduced to zero; and observations beyond c are rejected entirely. This smooth transition avoids the numerical instabilities that a hard step-function rejection would introduce.

For a discrete time series $\{x_t\}_{t=1}^T$, the Hampel filter operates over a rolling window of width w . Let $\mathcal{W}_t = [t - \lfloor w/2 \rfloor, t + \lfloor w/2 \rfloor]$ denote the indices inside the window centred at t . The local median and the local MAD are computed as

$$m_t = \text{median}\{x_\tau \mid \tau \in \mathcal{W}_t\}, \quad (1.15)$$

$$\text{MAD}_t = 1.4826 \cdot \text{median}\{|x_\tau - m_t| \mid \tau \in \mathcal{W}_t\}, \quad (1.16)$$

where the factor 1.4826 scales the MAD to approximate the standard deviation of a normal distribution [19]. A sample x_t is classified as an outlier if its deviation from the local median exceeds a dynamic threshold:

$$|x_t - m_t| > \kappa \cdot \max(\text{MAD}_t, \lambda \sigma(\mathbf{x})), \quad (1.17)$$

where $\sigma(\mathbf{x})$ is the global standard deviation of the signal, κ is a detection threshold (typically $\kappa = 3$), and λ is a floor multiplier that prevents hypersensitivity in steady-state (shut-in) operating regimes. Flagged outliers are replaced by the local median m_t .

Algorithm 1 Iterative Hampel Filter for Telemetry Reconstruction

Require: Time series $\mathbf{x} \in \mathbb{R}^T$, window w , threshold κ , floor multiplier λ , iterations n_{iter}

Ensure: Filtered series $\mathbf{x}$

```

1:  $\sigma_0 \leftarrow \lambda \cdot \text{std}(\mathbf{x})$ 
2: for  $i = 1$  to  $n_{\text{iter}}$  do
3:    $\mathbf{m} \leftarrow \text{RollingMedian}(\mathbf{x}, w)$ 
4:    $\mathbf{d} \leftarrow |\mathbf{x} - \mathbf{m}|$ 
5:    $\mathbf{s} \leftarrow 1.4826 \cdot \text{RollingMedian}(\mathbf{d}, w)$ 
6:    $\mathbf{s} \leftarrow \max(\mathbf{s}, \sigma_0)$  {Enforce variance floor}
7:    $\text{mask} \leftarrow \mathbf{d} > (\kappa \cdot \mathbf{s})$ 
8:    $\mathbf{x}[\text{mask}] \leftarrow \mathbf{m}[\text{mask}]$ 
9: end for
10: return  $\mathbf{x}$ 
```

Algorithm 1 formalises the detection and replacement procedure. By applying the filter iteratively (typically 2–3 passes), nested and clustered artifacts are progressively removed while preserving genuine transient events. The theoretical properties reviewed in this section— bounded gross-error sensitivity, maximum breakdown point, and smooth rejection—make the Hampel filter a principled choice for the first stage of industrial data conditioning.

1.4.2 Savitzky–Golay Smoothing and Numerical Differentiation

After impulsive outliers have been removed, the signal remains contaminated by ambient, low-amplitude measurement noise. Savitzky and Golay [20] showed that local polynomial least-squares fitting within a symmetric sliding window can simultaneously suppress high-frequency noise and preserve the higher-order statistical moments of the signal—something that simple moving-average filters fail to do.

Given a uniformly sampled sequence $\{y_j\}$ with spacing Δx , a window of $2m + 1$ points centred at index j is considered. Within this local window, a polynomial of degree n ($n < 2m + 1$) is fitted:

$$f_i = \sum_{k=0}^n b_k i^k, \quad i \in \{-m, \dots, 0, \dots, m\}, \quad (1.18)$$

where i is a local coordinate with $i = 0$ at the central point. The coefficients b_k are determined by minimising the sum of squared residuals:

$$\sum_{i=-m}^m (f_i - y_{j+i})^2 = \sum_{i=-m}^m \left(\sum_{k=0}^n b_k i^k - y_{j+i} \right)^2. \quad (1.19)$$

Setting the partial derivatives with respect to each b_r to zero yields the normal equations

$$\sum_{k=0}^n b_k S_{r+k} = F_r, \quad r = 0, \dots, n, \quad (1.20)$$

where $S_{r+k} = \sum_{i=-m}^m i^{r+k}$ are the moment sums (which vanish for odd $r + k$ due to symmetry) and $F_r = \sum_{i=-m}^m i^r y_{j+i}$ are the weighted data sums.

Because the moment sums S_{r+k} depend only on m and n , the normal equations can be solved analytically once, yielding fixed, integer convolution weights c_j that are independent of the data. The smoothed value at the central point is then simply

$$\tilde{x}_t = \sum_{j=-\lfloor w/2 \rfloor}^{\lfloor w/2 \rfloor} c_j x_{t+j}, \quad (1.21)$$

where $w = 2m + 1$ is the window size and c_j are the pre-computed convolution coefficients. The convolution weights for quadratic ($n = 2$) and cubic ($n = 3$) polynomials are identical, so a cubic fit—which provides better tracking of curvature—incurs no additional computational cost.

The local polynomial framework also yields analytical derivatives at the central point without further numerical approximation. Evaluating the s -th derivative of the fitted polynomial at $i = 0$ gives

$$\left. \frac{d^s f}{di^s} \right|_{i=0} = s! b_s, \quad (1.22)$$

and, incorporating the physical step Δx ,

$$\left. \frac{d^s u}{dx^s} \right|_{x=x_j} = \frac{s!}{\Delta x^s} \frac{\sum_{i=-m}^m C_i^{(s)} y_{j+i}}{N^{(s)}}. \quad (1.23)$$

The derivative convolution weights $C_i^{(s)}$ are tabulated for derivatives up to the fifth order and for window lengths from 5 to 25 points [20]. For example, the first-derivative weights for a quadratic polynomial exhibit a linear antisymmetric profile, which allows the convolution to be updated sequentially using simple subtraction.

The Savitzky–Golay filter supports two modes of derivative extraction. The first computes the derivative analytically by evaluating the fitted polynomial at the central point, directly yielding $\dot{x}_t$ via the pre-computed convolution weights $C_i^{(1)}$ (the `deriv=1` mode in standard implementations). This approach is employed in the SINDy identification pipeline, where the derivative is obtained

simultaneously with smoothing in a single pass. The second mode, adopted in the neural forecasting preprocessing pipeline, first constructs a smoothed signal manifold $\tilde{\mathbf{x}}$ and subsequently applies a second-order accurate central difference scheme to $\tilde{\mathbf{x}}$ to estimate $\dot{x}_t$. While this two-step procedure introduces a slight decoupling between smoothing and differentiation, it provides independent control over each operation and is algorithmically summarised in Algorithm 2.

Algorithm 2 Savitzky–Golay Smoothing and Central-Difference Gradient Extraction

Require: Array $\mathbf{x} \in \mathbb{R}^T$, window w , polynomial order p , time step dt

Ensure: Smoothed array $\tilde{\mathbf{x}}$, gradient array $\dot{\mathbf{x}}$

```

1: if  $T \leq p + 2$  then
2:   return  $\mathbf{x}, \mathbf{0}$ 
3: end if
4: if  $w \geq T$  then
5:    $w \leftarrow (T \text{ if } T \text{ is odd else } T - 1)$ 
6: end if
7: if  $w$  is even then
8:    $w \leftarrow w - 1$ 
9: end if
10: Compute Savitzky–Golay convolution coefficients
11: if filter fails due to ill-conditioned matrix then
12:    $\tilde{\mathbf{x}} \leftarrow \text{CenteredMovingAverage}(\mathbf{x}, 3)$ 
13: else
14:    $\tilde{\mathbf{x}} \leftarrow \text{SavGol}(\mathbf{x}, w, p)$ 
15: end if
16:  $\dot{\mathbf{x}} \leftarrow \text{gradient}(\tilde{\mathbf{x}}, dt)$ 
17: return  $\tilde{\mathbf{x}}, \dot{\mathbf{x}}$ 

```

Algorithm 2 formalises the smoothing and differentiation procedure. The key trade-off in the Savitzky–Golay filter is the choice of window width: a window that is too short retains too much noise; a window that is too long over-smooths and distorts genuine transient features. When the window length is chosen such that at most one inflection point of the underlying signal falls inside it, the filter provides an optimal balance between noise suppression and signal fidelity. In the context of sparse identification and physics-informed learning, this ability to compute smooth, analytically derived derivatives is indispensable.

1.4.3 Monotone Piecewise Cubic Hermite Interpolation (PCHIP)

Gaps caused by outlier removal or sensor dropouts must be filled before smoothing and derivative estimation can be applied. Standard C^2 cubic splines enforce global second-derivative continuity, but a single sharp transition can propagate oscillations throughout the entire domain—a manifestation of Runge’s phenomenon. Fritsch and Carlson [21] developed the Piecewise Cubic Hermite Interpolating Polynomial (PCHIP) to preserve the monotonicity of the data while providing a continuously differentiable (C^1) interpolant.

On each subinterval $[x_i, x_{i+1}]$ of length $h_i = x_{i+1} - x_i$, the interpolant is expressed in cubic Hermite form:

$$p(x) = f_i H_1(s) + f_{i+1} H_2(s) + d_i H_3(s) + d_{i+1} H_4(s), \quad (1.24)$$

where $s = (x - x_i)/h_i \in [0, 1]$, f_i, f_{i+1} are the data values, d_i, d_{i+1} are the prescribed endpoint slopes, and $H_k(s)$ are the standard cubic Hermite basis functions. The slopes d_i are initially estimated using a centred three-point difference formula (the Bessel method) and then adjusted to guarantee monotonicity.

Let $\Delta_i = (f_{i+1} - f_i)/h_i$ denote the secant slope of the data on the interval. If the signs of adjacent secant slopes indicate a local extremum or plateau ($\Delta_{i-1}\Delta_i \leq 0$ or $\Delta_i = 0$), the derivative at x_i is set

to zero:

$$d_i = 0 \quad \text{if} \quad \Delta_{i-1}\Delta_i \leq 0 \quad \text{or} \quad \Delta_i = 0. \quad (1.25)$$

This prevents the interpolant from overshooting flat or extremal regions.

For intervals where the data are strictly monotone ($\Delta_i \neq 0$), Fritsch and Carlson derived the necessary and sufficient conditions for the cubic polynomial to remain monotone. Defining the dimensionless slope ratios $\alpha_i = d_i/\Delta_i$ and $\beta_i = d_{i+1}/\Delta_i$, the monotonicity region $\mathcal{M}$ in the (α, β) plane is bounded by the ellipse

$$(\alpha - 1)^2 + (\alpha - 1)(\beta - 1) + (\beta - 1)^2 - 3(\alpha + \beta - 2) = 0. \quad (1.26)$$

If the initial pair (α_i, β_i) lies outside $\mathcal{M}$, the slopes are rescaled radially toward the origin until they lie within a safe subregion $\mathcal{S} \subset \mathcal{M}$ (typically a circle of radius 3):

$$\tau_i = \frac{3}{\sqrt{\alpha_i^2 + \beta_i^2}}, \quad d_i^* = \tau_i d_i, \quad d_{i+1}^* = \tau_i d_{i+1}. \quad (1.27)$$

This scaling preserves the relative magnitudes of the slopes while ensuring that the cubic segment is free of extraneous oscillations.

Algorithm 3 Monotonic Gap Imputation via PCHIP

Require: Matrix $\mathbf{X}$ containing telemetry signals with missing samples

Ensure: Continuous matrix $\mathbf{X}$ with gaps filled

```

1: for each column  $\mathbf{x} \in \mathbf{X}$  do
2:   if  $\mathbf{x}$  contains NaN then
3:     Attempt PCHIP interpolation
4:   if PCHIP fails then
5:     Apply linear interpolation
6:   end if
7:    $\mathbf{x} \leftarrow \text{bfill}(\text{ffill}(\mathbf{x}))$  {Resolve remaining edge NaNs}
8: end if
9: end for
10: return  $\mathbf{X}$ 

```

Algorithm 3 describes the gap-filling procedure. For the small number of extended gaps where PCHIP cannot be applied reliably, linear interpolation is used as a fallback, followed by forward- and backward-filling of any remaining boundary values. By substituting global C^2 continuity for local C^1 continuity and strict monotonicity preservation, PCHIP eliminates the unphysical wiggles that plague standard splines. When applied after outlier removal and before polynomial smoothing, it provides a continuous, physically plausible signal that respects the intrinsic shape of the industrial measurements.

1.4.4 Role in the Hybrid Identification Pipeline

The sequential application of these three techniques—Hampel filtering to reject impulsive artifacts, PCHIP to fill gaps while preserving monotonicity, and Savitzky–Golay smoothing to suppress ambient noise and compute stable derivatives—transforms raw, artifact-corrupted telemetry into a differentiable signal manifold suitable for both sparse system identification and physics-informed neural network training. This preprocessing strategy is not a cosmetic step; it is a necessary condition for the numerical stability of the downstream algorithms. The theoretical properties reviewed in this section—bounded gross-error sensitivity (Hampel), moment-preserving local polynomial approximation (Savitzky–Golay), and strict monotonicity enforcement (PCHIP)—together make industrial system identification feasible under realistic measurement conditions.

1.5 Conclusion

This chapter detailed the transition from purely data-driven models to frameworks that embed physical structure as an inductive bias. Physics-guided machine learning operates across data-level, model-level, and optimization-level integration pathways. Physics-Informed Neural Networks (PINNs) exemplify optimization-level integration, requiring adaptive loss balancing and causal weighting to navigate the non-convex optimization of complex physical systems.

Furthermore, SINDy approaches model discovery as a sparse regression problem, extracting governing equations from measurements by isolating dominant terms in a candidate function library. Extensions such as WSINDy and SINDYc address noise stability and control inputs, respectively. Finally, the mathematical foundations of signal conditioning—comprising median-based outlier rejection, polynomial smoothing, and monotonic interpolation—establish the numerical consistency required for industrial system identification in noisy environments. The theoretical principles analyzed here provide the basis for the implementation of the SINDy-regularised neural forecasting framework presented in the following chapters.

These developments offer a broad set of tools for scientific machine learning. Building upon this theoretical baseline, the subsequent chapter describes the data processing, neural network design, and training methodology adopted for industrial state estimation.

Chapter 2

Methodology

2.1 Methodology Overview

The preceding chapter established the theoretical foundations of physics-informed machine learning and sparse system identification, highlighting the limitations of purely data-driven models and the motivation for incorporating physical structure into learning algorithms. The present work implements a supervised learning pipeline for short-term forecasting of downhole pressure in a gas-lifted well. Historical measurements of pressure and control signals are used to predict the future evolution of the target variable. The pipeline combines two components: a neural forecasting model that maps a context window of recent observations and a lead time to a future pressure estimate, and a Sparse Identification of Nonlinear Dynamics (SINDy) model that provides a physics-based regularisation term during training.

The neural forecasting model consists of a one-dimensional convolutional encoder that compresses a multi-variate context window into a fixed-size latent vector, and a multi-layer perceptron decoder that maps the latent representation and a normalised forecast horizon to a predicted pressure value. The SINDy model is identified offline from smoothed telemetry, using a two-stage sparse regression procedure (LASSO followed by Ridge) to discover a parsimonious ordinary differential equation that approximates the dominant downhole pressure dynamics.

During training, the mean squared error between the network predictions and the measured pressure values is minimised jointly with a physics-informed collocation loss. The physics loss is computed by evaluating the SINDy-discovered ODE at randomly sampled points along the forecast horizon and penalising discrepancies between the network-derived time derivative and the SINDy-implied rate of change. This encourages the learned trajectories to remain consistent with the identified physical structure without requiring a pre-specified mechanistic model.

The objective is to evaluate whether the inclusion of such a data-driven physics regularizer improves the temporal consistency and long-horizon stability of pressure forecasts under normal operating conditions.

The remainder of this chapter is organised in three blocks.

Data pipeline (Sections 2.2–2.2.5): the 3W dataset and well selection criteria are described, followed by the three-stage preprocessing sequence—Hampel filtering, PCHIP interpolation, and Savitzky–Golay smoothing—that transforms raw telemetry into differentiable state trajectories. Feature engineering steps, including temporal derivative computation and the separation of state and control variables, are then detailed, and the procedures for chronological splitting, robust scaling, and fixed-length sequence construction are specified.

Model architecture and physics integration (Sections 2.3–2.4.2: the neural forecasting architecture—parametric activation, convolutional encoder, and MLP decoder is presented, together with the SINDy system identification pipeline, from candidate library construction through sparse regression to ODE validation. The physics-informed collocation loss and its coupling with the data loss are then formulated.

Training (Section 2.5): the optimisation algorithm, the linear ramp schedule for the physics loss weight, model checkpointing, and the monitored performance metrics are specified.

2.2 Data Acquisition and Processing Pipeline

The available data impose significant constraints on the modeling process. In oil-well operations, sensor networks are primarily deployed for monitoring and operational safety rather than for system identification or machine learning purposes, which results in datasets that are often weakly informative with respect to the underlying dynamics.

Consequently, the task is not only to predict future observations but also to infer the hidden dynamics responsible for the measured behaviour, enabling the trained network to capture the underlying physical relationships rather than superficial temporal correlations. Several characteristics of the data complicate model development:

1. Only a subset of the relevant process variables is measured directly or partially available. Consequently, internal states must be estimated from indirect observations.
2. Measurement noise can significantly affect derivative estimation, which is problematic for SINDy that relies on accurate temporal gradients.
3. Industrial systems are predominantly maintained under optimal nominal operating conditions, yielding datasets that are not only volumetrically constrained but also limited in dynamic richness. This scarcity of off-nominal or transient regimes severely restricts the model's ability to generalize across the full operational envelope.
4. Differences in reservoir characteristics, equipment configurations, and control mean that data collected from one well may not accurately represent the behaviour of another, limiting transferability without fine-tuning.

Addressing these limitations requires several preprocessing and restructuring steps before model training. The following sections describe how raw measurements are transformed into temporally consistent sequences suitable for the implemented modeling framework.

2.2.1 Dataset Description and Operational Context

The empirical foundation of this methodology is derived from the 3W dataset, a publicly available industrial repository published by Petrobras [6]. Originally engineered to benchmark anomaly detection algorithms in offshore naturally flowing oil wells, the dataset comprises multivariate time-series data extracted directly from physical production environments. The recordings capture the internal states of the wells at a standardized sampling frequency of 1 Hz. To represent the complex thermo-hydraulic behavior of the system, each instance is characterized by eight continuous process variables. These include critical pressure measurements—specifically the Permanent Downhole Gauge P_{PDG} , topside wellhead pressure P_{TPT} , and pressures upstream and downstream of the production choke $P_{MON-CKP}$ and $P_{JUS-CKP}$ —alongside topside temperature T_{TPT} and active control signals, including gas lift rate Q_{GL} and choke valve opening percentage $ABER - CKP$. Together, these variables provide a partial yet physically meaningful representation of the system dynamics.

Systemic Scope: Confinement to Normal Operating Conditions

Characterizing reliable state estimation under nominal dynamics is a prerequisite for identifying anomalous deviations. By isolating the normal operational regime, the methodology ensures that the governing equations extracted by the SINDy framework reflect the intrinsic system physics rather than transient or failure-induced behaviors. This approach establishes a physically consistent baseline for future extensions toward predictive maintenance.

2.2.2 Well Selection and Data Quality Assessment

Although all data correspond to normal operating conditions, the wells differ substantially in terms of sensor availability, control activity, and data quality. More importantly, these wells present varying degrees of:

1. *Partial Observability*: Several wells do not have measurement of important quantities such as downhole pressure P_{PDG} or exhibit prolonged sensor freezing, rendering supervised state estimation infeasible.
2. *Control Input Variability*: Insufficient excitation for identifying causal relationships, with some wells operating under nearly static conditions, limiting the identifiability of causal dynamics.
3. *Available Data Duration*: The quantity of usable, non-corrupted data varies widely, with some wells having a large percentage of data absent.

To select a well suitable for model development, each candidate is evaluated according to these criteria. Negative values in telemetry, representing non-physical sensor malfunctions, are filtered out and cast to null values prior to interpolation:

$$x_{\text{raw}} < 0 \implies x = \text{NaN} \quad (2.1)$$

Table 2.1 summarizes the percentage of valid samples after preprocessing. Wells lacking the target variable P_{PDG} , downstream choke pressure $P_{\text{JUS-CKP}}$, or control signals (Q_{GL} , $\text{ABER} - \text{CKP}$) are excluded, because these variables are required for both state prediction and the identification of dynamical relationships.

TABLE 2.1. Percentage of valid samples for each key variable after cleaning.

Well	P-PDG	P-TPT	P-MON	P-JUS	QGL	ABER
WELL-00002	0.0	100.0	0.0	0.0	0.0	0.0
WELL-00001	0.0	100.0	100.0	0.0	100.0	0.0
WELL-00006	73.4	100.0	100.0	0.0	100.0	0.0
WELL-00005	0.0	100.0	100.0	0.0	0.0	100.0
WELL-00008	0.0	0.0	0.0	0.0	0.0	0.0
WELL-00003	100.0	100.0	100.0	0.0	100.0	0.0
WELL-00004	0.0	100.0	100.0	0.0	0.0	100.0
WELL-00007	100.0	100.0	100.0	0.0	100.0	100.0
WELL-00019	100.0	100.0	100.0	100.0	99.9	98.9

Among the candidates, WELL-00019 has complete or near-complete coverage of all the required variables. The other wells are missing at least one essential measurement, which rules them out for this learning framework.

Sensor availability alone does not guarantee identifiability; the data must also exhibit sufficient dynamic excitation. Three physics-based criteria are assessed:

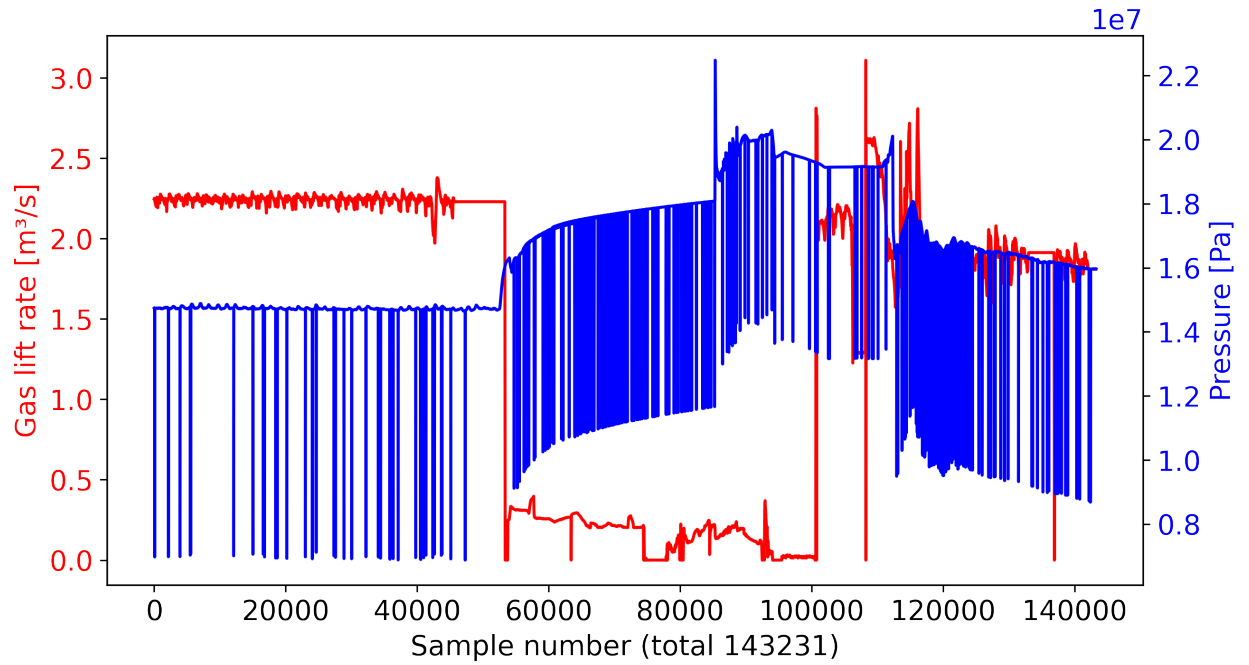
1. **Pressure Gradient Dynamics**: The relationship between downhole pressure P_{PDG} and top-side tree pressure P_{TPT} characterises pressure propagation through the production system, reflecting flow conditions and fluid-column dynamics.
2. **Choke-Induced Pressure Dynamics**: The pressure drop across the choke, $\Delta P = P_{\text{MON}} - P_{\text{JUS}}$, together with the choke opening $\text{ABER} - \text{CKP}$, quantifies how choke actuation affects flow conditions and provides a source of system excitation.
3. **Gas-Lift Response**: The coupling between Q_{GL} and P_{PDG} reflects the effect of gas-lift injection on hydrostatic pressure inside the production tubing. Changes in Q_{GL} produce delayed responses in P_{PDG} , making this interaction relevant for system identification and control-oriented modelling.

Table 2.2 quantifies these criteria: the standard deviation of the smoothed downhole pressure ($\sigma(P_{\text{PDG}})$) measures pressure variability; the number of actuation events (significant changes in Q_{GL} and $\text{ABER} - \text{CKP}$) reflects how much control excitation there is; and the data volume in hours shows how much clean, usable data is available.

Only WELL-00019 exhibits significant pressure variation and frequent control actuation, providing the input variability required for system identification. Physical validity is confirmed in Figs. 2.1, 2.2, and 2.3.

TABLE 2.2. Dynamic richness and actuation excitation of candidate wells.

Well	$\sigma(P_{\text{PDG}})$ [Pa]	Actuation events	Data volume [h]
WELL-00002	0.00×10^0	0	1220
WELL-00001	0.00×10^0	1	556
WELL-00006	1.99×10^7	0	675
WELL-00005	—	7	353
WELL-00008	—	0	337
WELL-00003	3.23×10^5	0	155
WELL-00004	0.00×10^0	0	36
WELL-00007	5.55×10^3	0	6
WELL-00019	1.77×10^6	28	40

**Fig. 2.1.** Gas-lift actuation: downhole pressure P_{PDG} and gas injection rate Q_{GL} .

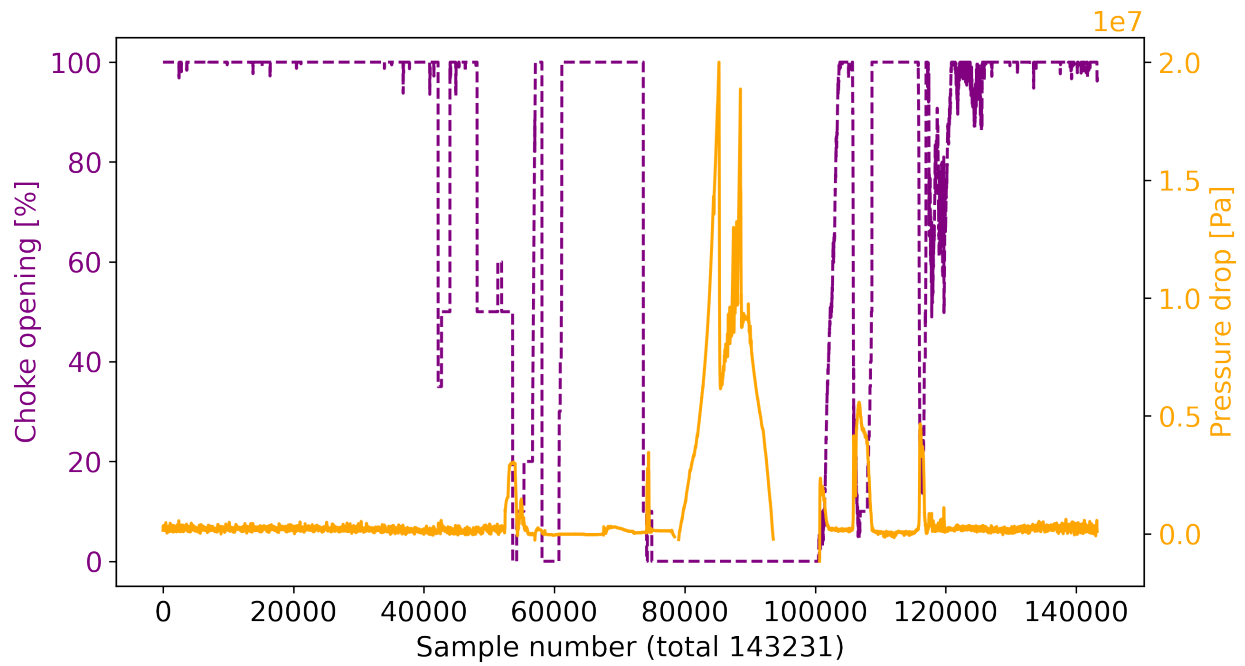


Fig. 2.2. Choke orifice dynamics: pressure drop and choke valve opening.

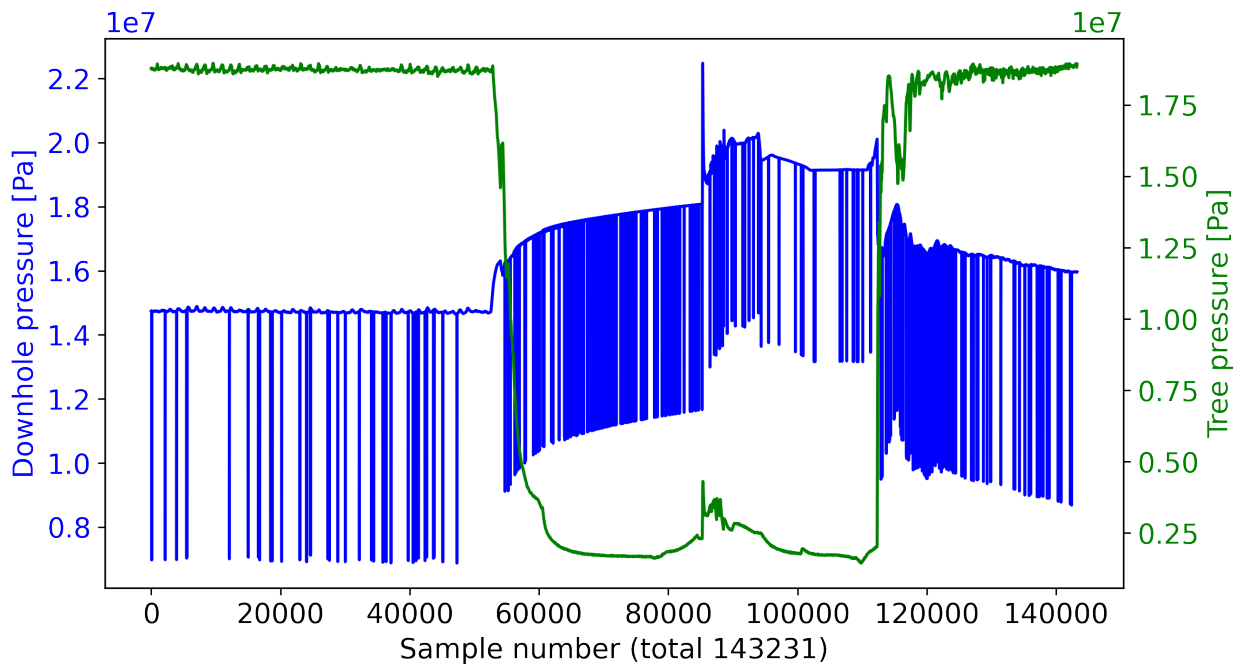


Fig. 2.3. Reservoir-wellbore coupling: downhole pressure P_{PDG} and topside tree pressure P_{TPT} .

2.2.3 Data Cleaning and Filtering

The raw measurements reflect a superposition of true thermo-hydraulic dynamics and non-physical operational perturbations. As shown in Figs. 2.1, 2.2, and 2.3, the dataset contains sensor dropouts, signal freezing, and high-frequency noise. These artifacts are detrimental to physics-informed neural forecasting; if untreated, they are amplified during numerical differentiation, destabilizing SINDy regression and degrading gradient computations. The preprocessing strategy employs a sequential filtering pipeline to obtain smooth, differentiable signals while preserving transient behaviors.

Hampel Filtering for Sensor Fault Rejection

The first preprocessing step removes obvious sensor artifacts caused by telemetry errors and intermittent hardware faults. These faults typically appear as isolated spikes or abrupt jumps that are inconsistent with the expected temporal evolution of pressure and flow measurements.

To eliminate these anomalies, a Hampel filter [18] based on the Median Absolute Deviation (MAD) is applied across a rolling temporal window. For a discrete univariate sequence x_t , a rolling window of width $W_{\text{Hampel}} = 180$ is defined centered at index t as:

$$\mathbf{W}_t = \left\{ x_i \mid i \in \left[t - \left\lfloor \frac{W_{\text{Hampel}}}{2} \right\rfloor, t + \left\lfloor \frac{W_{\text{Hampel}}}{2} \right\rfloor \right] \right\} \quad (2.2)$$

The rolling median $\tilde{x}_t$ and the MAD are formulated as:

$$\tilde{x}_t = \text{median}(\mathbf{W}_t) \quad (2.3)$$

$$\text{MAD}_t = \text{median}(\{|x_i - \tilde{x}_t| \mid \forall x_i \in \mathbf{W}_t\}) \quad (2.4)$$

To compute a consistent estimator of the standard deviation under a normal distribution, the local standard deviation is scaled as $\sigma_t = 1.4826 \cdot \text{MAD}_t$. To prevent division-by-zero or hypersensitivity in flat regions, a safety floor is applied based on the global standard deviation: $\sigma_t = \max(\sigma_t, \gamma \cdot \sigma_{\text{global}})$, where $\gamma = 0.03$. An observation is flagged as an outlier and replaced with $\tilde{x}_t$ if:

$$|x_t - \tilde{x}_t| > n_\sigma \cdot \sigma_t \quad (2.5)$$

In the forecasting data processing pipeline, the threshold multiplier is configured as $n_\sigma = 12.0$. However, during the SINDy identification preprocessing stage, an offline two-pass Hampel strategy is executed with more stringent parameters, using $(W_{\text{Hampel}} = 150, n_\sigma = 8.0, \gamma = 0.03)$ on the first pass, and $(W_{\text{Hampel}} = 150, n_\sigma = 3.0, \gamma = 0.03)$ on the second pass. This ensures highly sparse trajectories for structural ODE regression.

PCHIP Interpolation for Gap Filling and Data Continuity

After outlier rejection, the time series may still contain missing samples as the Hampel filtering replaces spikes with NaN values or the original sensor data had inherent dropouts. Since subsequent smoothing and derivative estimation procedures require continuous observations, these gaps must be reconstructed before further processing.

Missing samples are interpolated using Piecewise Cubic Hermite Interpolating Polynomials (PCHIP) [21]. For each subinterval $[t_i, t_{i+1}]$, PCHIP constructs a cubic polynomial $H(t)$ matching boundary values $H(t_i) = x_i, H(t_{i+1}) = x_{i+1}$ and first derivatives $H'(t_i) = d_i, H'(t_{i+1}) = d_{i+1}$. Monotonicity is preserved by enforcing boundary slope limitations: if local slopes $S_i = \frac{x_{i+1} - x_i}{t_{i+1} - t_i}$ change sign, the derivative is set to zero ($d_i = 0$). Otherwise, it is computed as a weighted harmonic mean:

$$d_i = \frac{w_1 + w_2}{\frac{w_1}{S_{i-1}} + \frac{w_2}{S_i}} \quad (2.6)$$

where weights are $w_1 = 2h_i + h_{i-1}$ and $w_2 = h_i + 2h_{i-1}$, with grid step $h_i = t_{i+1} - t_i$. This derivative limiting constraint ensures under monotonicity conditions that the interpolant avoids artificial local extrema (overshoots). Linear interpolation is used as a fallback strategy, followed by forward and backward filling at the signal boundaries when required.

Savitzky–Golay Filtering for Dynamical Manifold Reconstruction

Following the rejection of outliers, the signal remains corrupted by ambient, low-amplitude measurement noise. While seemingly negligible in the observational space, this high-frequency variance is amplified by numerical differentiation, a fundamental prerequisite for both identifying the governing ODEs via SINDy and evaluating the physics-informed collocation loss within the neural forecasting model.

To resolve this, a Savitzky–Golay filter [20] is employed to smooth the trajectory for reliable differentiation and subsequent temporal feature aggregation. By performing a local polynomial regression within a moving window, this filter fits a low-degree polynomial to the local data manifold. The specific advantage of this methodology is its ability to suppress high-frequency noise while strictly preserving the higher-order moments of the signal, including the critical curvature and peak amplitudes of transient pressure ramps.

For the neural forecasting pipeline, the filter uses a window size of $W_{SG} = 31$ and polynomial degree $d = 3$. For SINDy identification, where derivative estimation is sensitive, a wider window of $W_{SG} = 51$ is selected to provide higher noise regularization.

The preprocessing pipeline transforms raw telemetry into a smooth, differentiable manifold. Reconstructed signal profiles are presented in Figs. 2.4, 2.5, and 2.6.

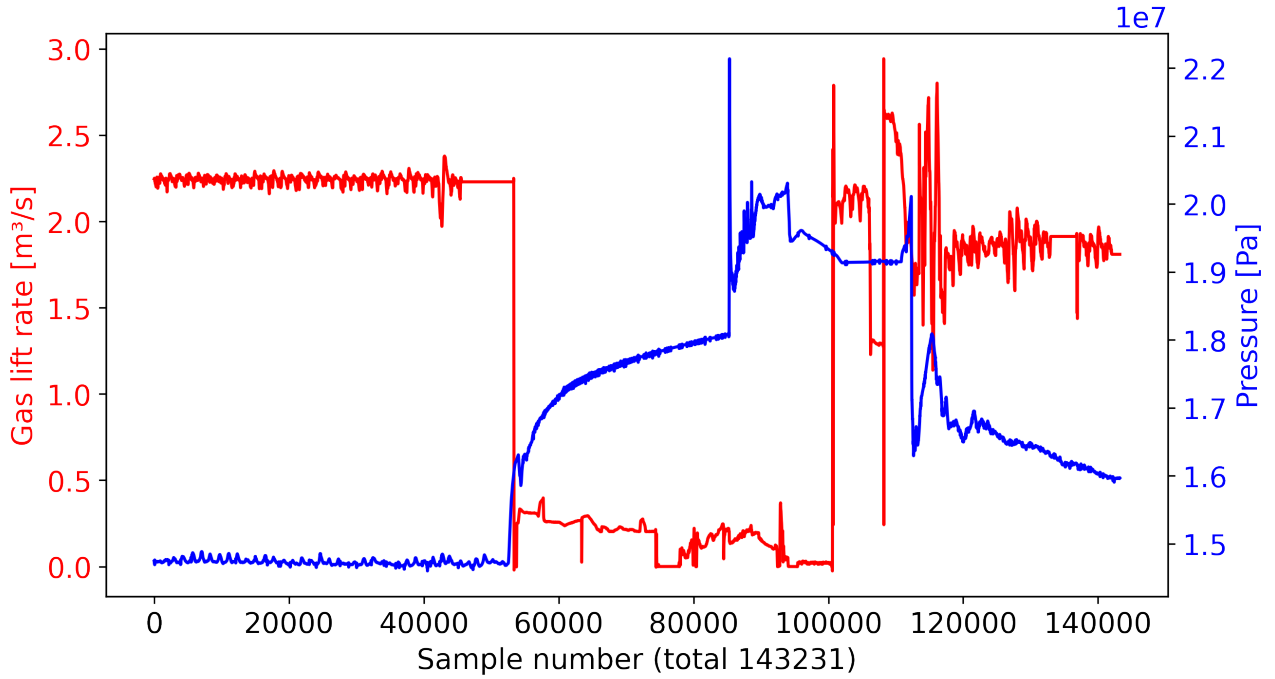


Fig. 2.4. Gas-lift clean profiles: downhole pressure P_{PDG} and gas lift rate Q_{GL} .

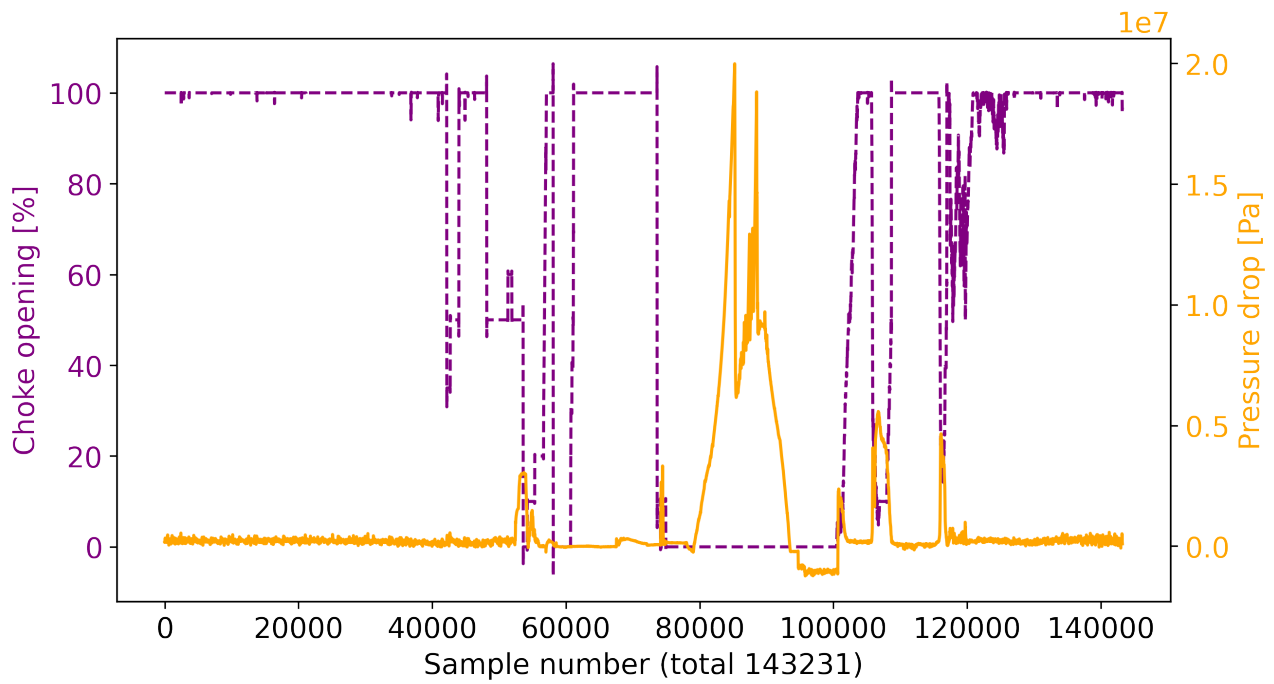


Fig. 2.5. Choke clean orifice dynamics: pressure drop ΔP and choke valve opening.

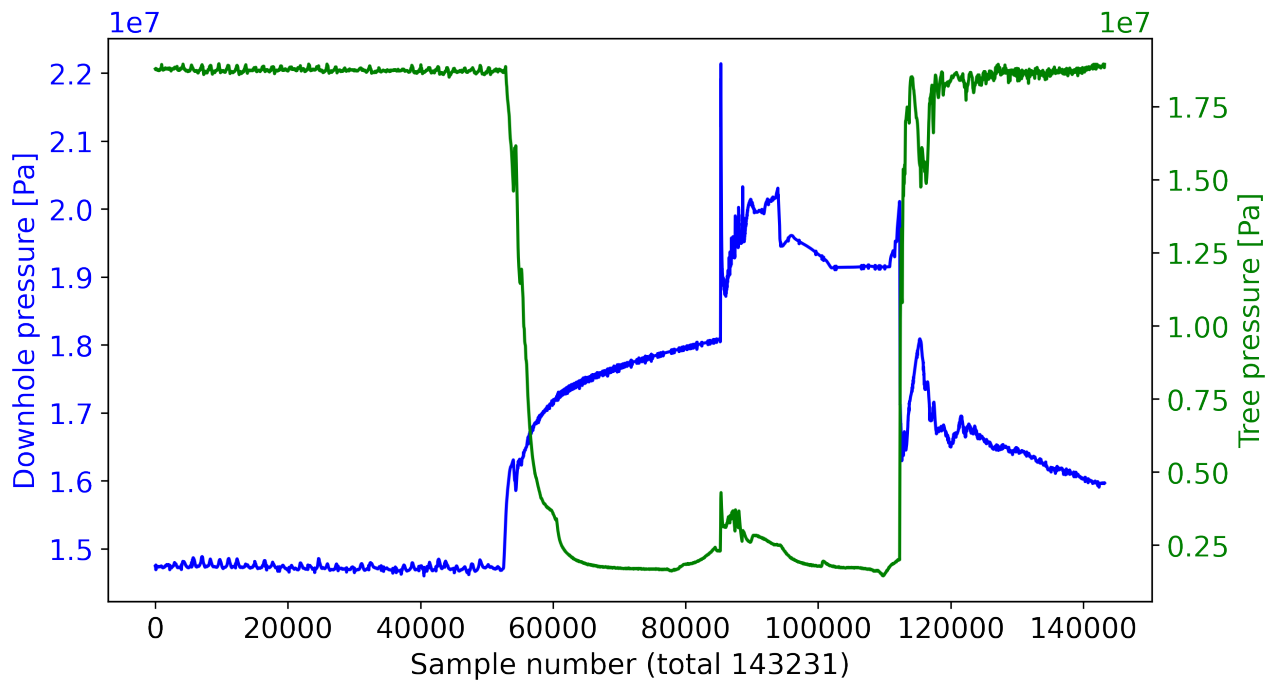


Fig. 2.6. Reservoir-wellbore coupling clean profiles: downhole pressure P_{PDG} and tree pressure P_{TPT} .

2.2.4 Feature Engineering

After preprocessing, additional features are extracted to support both system identification and neural network training. The selected features are derived from variables that are directly related to the physical behavior of a gas-lifted well. Rather than introducing a large number of generic statistical descriptors, the feature set is constructed using quantities that have a direct physical interpretation within the production system.

Temporal Derivative Computation

Both SINDy and the physics-informed loss formulation require estimates of the state derivatives $\dot{\mathbf{x}}$. Because the dynamics of the gas-lifted well are inherently expressed as differential equations, these derivatives are not auxiliary features but necessary.

However, direct differentiation of noisy industrial measurements is sensitive to measurement errors and could lead to gradient explosion, destroying the sparse regression process. This vulnerability is bypassed by computing derivatives exclusively on the Savitzky-Golay filtered signals. Since the data has already been approximated locally by low-order polynomials, first-order derivatives are numerically approximated using central difference schemes applied to the smoothed representation:

$$\dot{x}_t \approx \frac{x_{t+1}^{\text{sm}} - x_{t-1}^{\text{sm}}}{2 \cdot \Delta t} \quad (2.7)$$

where sampling interval $\Delta t = 1.0$ s. This provides smoother derivative estimates that are less sensitive to measurement noise.

State and Control Variable Representation

To properly condition the data for the neural forecasting architecture, the feature space is explicitly partitioned based on the physical role of each variable.

- **State Variables $\mathbf{X}$:** Internal thermodynamic quantities, including the upstream and downstream choke pressures $P_{\text{MON-CKP}}$, $P_{\text{JUS-CKP}}$ and topside pressure P_{TPT} . These variables describe the current operating state of the production system.
- **Control Variables $\mathbf{U}$:** The external actuation signals, specifically the gas lift injection rate Q_{GL} and choke opening fractional value $u \in [0, 1]$, transformed from percentage as:

$$u = \frac{\text{ABER} - \text{CKP}}{100} \quad (2.8)$$

These variables represent external actions that influence the evolution of the system state.

This separation follows the standard representation of controlled dynamical systems, formulated as $\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u})$. Both the downhole pressure P_{PDG} and the state and control variables listed above are concatenated to form the six-channel input to the neural network. Additionally, the raw control and boundary pressure sequences are carried alongside each sample to enable evaluation of the SINDy right-hand side at collocation points during physics-informed training.

2.2.5 Data Structuring for Learning

Once the feature space is established, the continuous time series must be reformatted into a structure compatible with supervised learning. Improper structuring at this stage can introduce temporal leakage or artificially bias the learned dynamics.

Chronological Split and Scaling

The available record for WELL-00019 spans approximately 40 hours of continuous operation at 1 Hz. The data are divided chronologically into training (70%), validation (15%), and test (15%) subsets, preserving the temporal order of industrial operations. This ordering ensures that no future information leaks into the training process; all model selection and hyperparameter tuning are performed exclusively on the validation split, with the test split held out for final evaluation.

Each of the six input channels is normalised independently using a `RobustScaler`, which centres the data on the median and scales by the interquartile range (IQR):

$$\tilde{x} = \frac{x - \text{median}(x_t)}{\text{IQR}(x_t)} \quad (2.9)$$

where $\text{IQR}(x_t) = Q_3(x_t) - Q_1(x_t)$. The scaler is fitted on the training split alone and then applied identically to the validation and test splits. Robust statistics are chosen over min-max normalisation to reduce sensitivity to extreme transient excursions. The target variable P_{PDG} is scaled separately using the same procedure.

Sequence Construction

The continuous time series is cut into fixed-length samples. Each sample has a context window of $W = 60$ consecutive time steps (1 minute) and a forecast horizon h taken from the set $\{30, 60, 90, 120\}$ seconds. The window stride is set to $S = W + H_{\text{max}} = 180$ steps, where $H_{\text{max}} = 120$ is the maximum forecast horizon. With this stride, consecutive windows never overlap, so there's no risk of temporal leakage between training instances.

For a window starting at index s , the context is the six smoothed and scaled variables over the interval $[s, s + W)$. The target is the scaled downhole pressure at time $s + W + h - 1$. The normalised lead time is $h/H_{\text{max}} \in (0, 1]$. Each sample also carries extra information needed for the physics-informed loss: the raw (unscaled) downhole pressure at the last context time step, the raw control and boundary pressure values at that same instant, and the full sequences of these raw variables over the H_{max} -step forecast horizon. These raw sequences enable the evaluation of the SINDy right-hand side at collocation points without pulling in extra data during training.

Samples are grouped into batches of size $B = 256$ using a custom collation function that stacks the individual tensors along the batch dimension. The training `DataLoader` shuffles samples each epoch using a fixed deterministic seed, while the validation and test `DataLoaders` keep the windows in chronological order. All tensors are kept in double precision (`float64`) to keep gradient computation of the physics loss term numerically stable.

2.3 Architectural Design

A neural operator surrogate is implemented for forecasting across continuous horizons. The model maps past state-control sequences to future state predictions. The architecture consists of three components: a parametric activation layer, a 1D convolutional neural network (CNN) encoder, and a multi-layer perceptron (MLP) decoder.

2.3.1 Parametric Tanh Activation (PTanh)

To prevent gradient vanishing during automatic differentiation, a custom parametric hyperbolic tangent activation layer (PTanh) is implemented:

$$f(x) = a \cdot \tanh(b \cdot x) \quad (2.10)$$

where $a, b \in \mathbb{R}$ are learnable, scalar parameters registered as PyTorch parameters (`nn.Parameter`) and initialized to 1.0.

The analytical derivative of the activation function with respect to the input x is derived as:

$$f'(x) = \frac{d}{dx} [a \tanh(b \cdot x)] = a \cdot b \cdot (1 - \tanh^2(b \cdot x)) \quad (2.11)$$

Substituting $\tanh(b \cdot x) = \frac{f(x)}{a}$ into the derivative yields:

$$f'(x) = a \cdot b \left(1 - \frac{f(x)^2}{a^2}\right) \quad (2.12)$$

In standard neural networks, backpropagating through deep networks often suffers from vanishing gradients because the derivative of standard Tanh is bounded by 1.0 and decays rapidly outside the active region $[-2, 2]$. By making a and b learnable, the network dynamically scales its slopes during training. The parameter b acts as a frequency scaler, stretching the input domain to keep the neurons in their active, non-saturated gradient zones. This dynamic scaling is critical for stabilizing the automatic differentiation steps required by the continuous-time collocation physics loss.

2.3.2 Context Encoder (ConvEncoder)

The convolutional encoder maps multi-channel context sequences of shape (B, C_{in}, W) to a compact latent embedding $\mathbf{z} \in \mathbb{R}^{B \times \text{latent_dim}}$. The input dimension is $C_{\text{in}} = 6$ (comprising the six smoothed and scaled variables), and the context window is $W = 60$ steps (seconds).

The encoder consists of two sequential 1D convolutional blocks. Each block applies a 1D convolution, a ‘PTanh’ activation, and a local average pooling layer. For a given input tensor $\mathbf{X}_{\text{in}}$, the i -th convolution layer computes:

$$\mathbf{X}_{\text{conv},i} = \text{PTanh}(\mathbf{W}_i * \mathbf{X}_{i-1} + \mathbf{b}_i) \quad (2.13)$$

where $*$ denotes the 1D discrete convolution operator, $\mathbf{W}_i \in \mathbb{R}^{C_i \times C_{i-1} \times K_i}$ represents the kernel weights, and $\mathbf{b}_i \in \mathbb{R}^{C_i}$ represents the bias vectors. The kernel size is fixed at $K_i = 5$, and a padding of $p_i = \lfloor K_i/2 \rfloor = 2$ is applied to maintain the temporal output dimension.

Following each convolution, a local average pooling layer downsamples the temporal dimension:

$$\text{AvgPool1D}(\mathbf{X}_{\text{conv},i})_{b,c,\tau} = \frac{1}{K_{\text{pool}}} \sum_{m=0}^{K_{\text{pool}}-1} X_{\text{conv},i, b, c, \tau \cdot S_{\text{pool}} + m} \quad (2.14)$$

where the pooling window size is $K_{\text{pool}} = 2$ and the stride is $S_{\text{pool}} = 2$.

To guarantee that the extracted latent embedding is invariant to the input context window length W , a final adaptive average pooling layer collapses the temporal axis completely to size 1:

$$\mathbf{z}_{b,c} = \frac{1}{L_{\text{final}}} \sum_{t=1}^{L_{\text{final}}} X_{\text{final}, b, c, t} \quad (2.15)$$

yielding the latent context vector $\mathbf{z} \in \mathbb{R}^{B \times \text{latent_dim}}$, where the latent dimension equals the channel count of the last convolutional layer ($\text{latent_dim} = 16$).

2.3.3 Forecasting Decoder (ForecastingDecoder)

The decoder is a fully-connected multi-layer perceptron (MLP) with hidden layers of size $[64, 64]$ that maps the concatenated state-horizon vector to a future scalar pressure.

The decoder takes the concatenated vector of the latent embedding $\mathbf{z}$ and the normalized forecast horizon $h_{\text{norm}} \in (0, 1]$:

$$\mathbf{y}^{(0)} = [\mathbf{z}, h_{\text{norm}}] \in \mathbb{R}^{B \times (\text{latent_dim} + 1)} \quad (2.16)$$

For each hidden layer $l \in \{1, \dots, L_{\text{layers}}\}$ (where $L_{\text{layers}} = 2$), the forward propagation is formulated as:

$$\mathbf{y}^{(l)} = \text{PTanh}(\mathbf{W}^{(l)} \mathbf{y}^{(l-1)} + \mathbf{b}^{(l)}) \quad (2.17)$$

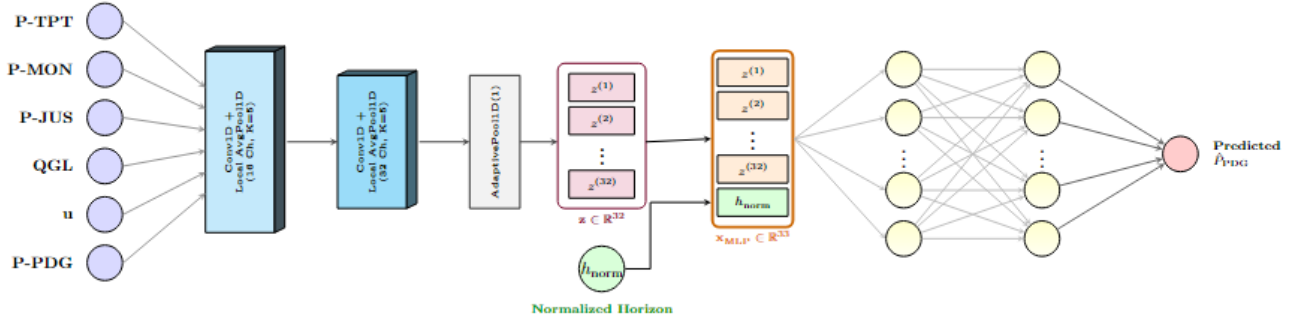


Fig. 2.7. Forecasting model architecture with 1D CNN encoder and MLP decoder.

where $\mathbf{W}^{(l)} \in \mathbb{R}^{N_l \times N_{l-1}}$ and $\mathbf{b}^{(l)} \in \mathbb{R}^{N_l}$ represent the weight matrices and bias vectors of layer l , and the hidden dimensions are $N_1 = N_2 = 64$.

The final layer projects the last hidden layer representation to the predicted scaled downhole pressure $\hat{y} \in \mathbb{R}^{B \times 1}$:

$$\hat{y} = \mathbf{W}^{(\text{out})} \mathbf{y}^{(L_{\text{layers}})} + \mathbf{b}^{(\text{out})} \quad (2.18)$$

Every linear layer inside the decoder is explicitly cast to double precision (`float64`) using `‘.double()’`. While standard models typically use single precision (`float32`), physics-informed neural networks computing collocation derivatives require double precision to prevent machine epsilon underflow ($\approx 1.19 \times 10^{-7}$) during backpropagation of second-order gradients, providing the numerical resolution ($\approx 2.22 \times 10^{-16}$) required for stable training.

The complete data flow, tensor shapes, and layer configurations of the joint forecasting model are illustrated in Figure 2.7.

2.4 Physics Integration via SINDy

A purely data-driven forecasting model can produce predictions that violate the physical constraints of the gas-lift system, particularly over longer forecast horizons. To mitigate this, a physics-based regularisation term is constructed from a SINDy-discovered dynamical model and added to the training objective.

The procedure consists of two stages. First, a Sparse Identification of Nonlinear Dynamics (SINDy) model is identified offline from the smoothed telemetry. This stage yields an explicit ordinary differential equation that approximates the dominant pressure dynamics. Second, during neural network training, the SINDy ODE is evaluated at randomly chosen collocation points along the forecast horizon, and the discrepancy between the network-derived time derivative and the SINDy-implied rate of change is penalised. The offline identification step is described in Section 2.4.1; the collocation-based physics loss is formulated in Section 2.4.2.

2.4.1 Sparse Model Identification via SINDy

The objective of the model discovery phase is to identify a parsimonious ordinary differential equation that governs the temporal evolution of the downhole pressure (P_{PDG}). This process uses the Sparse Identification of Nonlinear Dynamics (SINDy) framework, which formulates system identification as an overdetermined linear inverse problem. Given a trajectory of the system state and control inputs, the temporal derivative is assumed to be a linear combination of a small number of active terms drawn from a large library of candidate nonlinear functions.

The candidate library is built from the six smoothed variables produced by the preprocessing pipeline (Section 2.2.3). Let $P = P_{\text{PDG}}$, $Q = Q_{\text{GL}}$, $u = \text{ABER}/100$, and let the three boundary pressures be P_{TPT} , P_{MON} , P_{JUS} . The library comprises 34 terms organised into four categories—baseline states and controls, polynomial and interaction terms, boundary-flow and empirical nonlinearities, and thermo-hydraulic surrogates—as fully detailed in Appendix B. Four representative surrogates are summarised here:

1. Choke Orifice Flow Proxy:

$$\phi_{\text{orifice}} = u \cdot \sqrt{\rho_{\text{mix}} \cdot \max(P_{\text{TPT}} - P_{\text{JUS}}, 0)} + \epsilon \quad (2.19)$$

where $\rho_{\text{mix}} = \max\left(\frac{P_{\text{TPT}} \cdot M_G}{R \cdot T_{\text{tubing}}}, 10^{-9}\right)$, $M_G = 0.0167$, $R = 8.314$, $T_{\text{tubing}} = 369.4$, and $\epsilon = 10^{-12}$.

2. Tubing Frictional Loss Proxy:

$$\phi_{\text{friction}} = \rho_{\text{proxy}} \cdot U_{\text{proxy}}^2 \quad (2.20)$$

where $U_{\text{proxy}} = \max\left(\frac{|Q|}{A_{\text{tubing}} + \epsilon}, 0\right)$, $A_{\text{tubing}} = \frac{\pi D_{\text{tubing}}^2}{4}$, $D_{\text{tubing}} = 0.134$, and mixture density $\rho_{\text{proxy}} = \max(0.1\rho_{\text{mix}} + 0.9\rho_{\text{liquid}}, 10^{-6})$ with $\rho_{\text{liquid}} = 760.0$.

3. Reservoir Inflow Relationship:

$$\phi_{\text{inflow}} = J \cdot \max(P_{\text{reservoir}} - P, 0) \quad (2.21)$$

where $J = 2.47 \times 10^{-6}$ and $P_{\text{reservoir}} = 1.6 \times 10^7$.

4. Gas-Lift Weight Injection Proxy:

$$\phi_{\text{injection}} = \left(\frac{P_{\text{injection}} \cdot M_G}{R \cdot T_{\text{annulus}}}\right) \cdot Q \quad (2.22)$$

where $P_{\text{injection}} = 1.6 \times 10^7$ and $T_{\text{annulus}} = 348.0$.

The overall SINDy pipeline, from smoothed telemetry to a sparse ODE, is illustrated in Fig. 2.8.

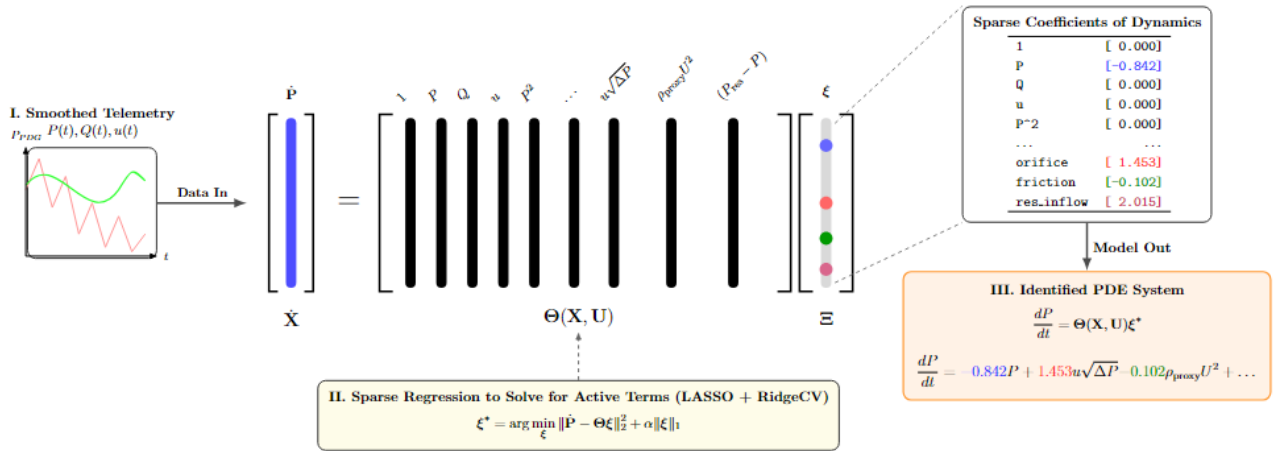


Fig. 2.8. SINDy model identification pipeline from smoothed telemetry.

Sparse Regression and Term Selection

The identification problem is posed as the overdetermined linear system

$$\dot{\mathbf{X}} = \Theta(\mathbf{X}) \Xi, \quad (2.23)$$

where Ξ is the coefficient vector. To obtain a parsimonious model, a two-stage sparsity-promoting strategy is employed.

First, LASSO (ℓ_1 -regularised regression) is applied to select a sparse set of active terms:

$$\min_{\Xi} \frac{1}{2N} \|\mathbf{y}_{\text{scaled}} - \Theta_{\text{std}}(\mathbf{X})\Xi\|_2^2 + \alpha_1 \|\Xi\|_1 \quad (2.24)$$

where $\mathbf{y}_{\text{scaled}} = \dot{\mathbf{P}}/10^6$ in MPa/s. The regularisation hyperparameter α_1 is chosen via 5-fold cross-validation on a uniformly subsampled set of 30 000 time points. To retain a sufficiently expressive model, the cross-validated α_1 is halved, and a final LASSO fit with this relaxed penalty is applied to the full dataset. The $k = 22$ terms with the largest absolute coefficients are retained.

Because the ℓ_1 penalty biases coefficients toward zero, the selected terms are subsequently refitted with Ridge regression (ℓ_2 -regularised) to estimate the physical scales accurately:

$$\min_{\mathbf{c}_{\text{std}}} \|\mathbf{y}_{\text{scaled}} - \mathbf{\Theta}_{\text{sel,std}}(\mathbf{X})\mathbf{c}_{\text{std}}\|_2^2 + \alpha_2 \|\mathbf{c}_{\text{std}}\|_2^2 \quad (2.25)$$

where the Ridge regularisation parameter α_2 is chosen independently via cross-validation. This yields the final coefficient vector $\mathbf{c}_{\text{std}}$ and an intercept term.

Identified Dynamical Model

The resulting ordinary differential equation for the downhole pressure is

$$\frac{dP}{dt} [\text{MPa/s}] = \text{intercept} + \sum_{i=1}^{22} c_{\text{std},i} \cdot \text{std}(\theta_i(\mathbf{X})), \quad (2.26)$$

where $\text{std}(\cdot)$ denotes z -score standardisation using the mean and variance computed from the training data. To validate the discovered equation, it is integrated forward in time using a stiff implicit Runge–Kutta (Radau) solver, with the initial condition taken from the first smoothed pressure value. The root-mean-square error between the simulated trajectory and the smoothed pressure is approximately 6×10^5 Pa (0.6 MPa), indicating that the parsimonious 22-term model reproduces the dominant pressure evolution with sufficient fidelity for use as a regularisation constraint.

2.4.2 Loss Function Design

The network is trained with a loss function composed of two terms: a data-fit term and a physics-informed regularisation term. The relative influence of the physics term is controlled by a scalar weight λ that is progressively increased during training.

Data Loss

The data loss is the mean squared error (MSE) between the network’s scaled prediction and the scaled target pressure:

$$\mathcal{L}_{\text{data}} = \frac{1}{B} \sum_{b=1}^B (\hat{y}_b - y_b)^2,$$

where $\hat{y}_b$ is the predicted scaled pressure for sample b , y_b is the corresponding true scaled value, and B is the batch size. Both predictions and losses are kept fully differentiable, allowing joint gradients to update network parameters.

Physics-Informed Collocation Loss

To encourage consistency with the SINDy-discovered dynamics, a physics loss is evaluated at randomly chosen collocation points along the forecast horizon. For each sample in a batch, $N_{\text{colloc}} = 10$ integer time steps $\tau_1, \dots, \tau_{N_{\text{colloc}}}$ are sampled uniformly from $\{1, \dots, h\}$, where h is the target forecast horizon of the sample. The normalised lead times are τ_j/H_{max} with $H_{\text{max}} = 120$ s.

Let $\hat{P}_{\text{scaled}}$ denote the network output at a normalised lead time. Automatic differentiation yields the derivative with respect to the normalised horizon, which is mapped to a physical rate of pressure change (Pa s^{-1}) via the continuous multivariable chain rule:

$$\frac{dP}{dt} = \frac{\partial P}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial h} \cdot \frac{dh}{dt} = \frac{\partial \hat{y}}{\partial h} \cdot \left(\frac{S_{\text{rob}}}{H_{\text{max}}} \right) \quad (2.27)$$

where S_{rob} is the scale factor of the robust scaler applied to the target pressure.

At each collocation point, the predicted pressure (inverse-scaled to Pascals) and the corresponding control and boundary pressure values (gathered from the raw future sequences stored in the sample as 1D tensors) are fed into the SINDy right-hand-side function f_{SINDy} , which returns the implied pressure rate of change. The physics loss is then defined as the dimensionless, scale-invariant error

$$\mathcal{L}_{\text{phys}} = \frac{1}{B \cdot N_{\text{colloc}}} \sum_{b=1}^B \sum_{j=1}^{N_{\text{colloc}}} \left(\frac{\frac{d\hat{P}_{b,j}}{dt} - f_{\text{SINDy}}(\hat{P}_{b,j}, \mathbf{u}_{b,j})}{\text{RMS}(f_{\text{SINDy}}) + \epsilon} \right)^2, \quad (2.28)$$

where $\mathbf{u}_{b,j}$ denotes the control vector at collocation point j of sample b , $\text{RMS}(f_{\text{SINDy}})$ is the root-mean-square of the SINDy predictions over the collocation points, and $\epsilon = 10^{-12}$ prevents division by zero.

Total Loss

The total training objective is a weighted sum

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{data}} + \lambda \cdot \mathcal{L}_{\text{phys}}. \quad (2.29)$$

2.5 Training

The data structuring procedure—chronological split, robust scaling, and fixed-length sequence construction—is detailed in Section 2.2.5. The resulting `DataLoader` objects provide the training, validation, and test batches used throughout the optimization process described below.

2.5.1 Training Loop and Hyperparameters

The model parameters are optimized using the AdamW algorithm [22] with a base learning rate of $\eta = 10^{-3}$ and weight decay 10^{-5} . The learning rate is reduced by a plateau scheduler (patience 20 epochs, decay factor 0.98) down to a minimum of 4×10^{-4} . Gradient clipping was evaluated as a precautionary measure but proved unnecessary: the linear ramp schedule of the physics loss weight maintains healthy gradient magnitudes throughout training.

The physics loss weight λ follows a linear ramp schedule to ensure numerical stability during the transition from data-driven to physics-informed optimization. In the baseline configuration ($\lambda_{\text{final}} = 10^{-7}$), $\lambda = 0$ for the first 150 epochs, followed by a linear ramp to the terminal weight over 50 epochs (totaling 1000 epochs). To assess the trade-off between data fidelity and physical consistency (the multi-objective Pareto front), two higher terminal weights are investigated: $\lambda_{\text{final}} \in \{10^{-6}, 10^{-5}\}$. These sensitivity runs explore a compressed 600-epoch schedule, with the linear ramp occurring between epochs 100 and 150. After the ramp is complete, λ remains constant at λ_{final} for the remainder of the training session.

After every epoch, the validation loss is computed. If it improves upon the best value seen so far, the current model parameters are saved as a checkpoint. At the end of training, the model is restored to the best checkpoint. This strategy selects the model with the best forecasting accuracy on unseen data under the specified levels of physics regularization.

2.5.2 Training and Selection Strategy

The model selection process was conducted in two sequential stages to isolate the effects of architectural capacity from those of physics-informed regularisation:

1. **Architectural Grid Search:** First, three different architecture configurations (varying in encoder depth and decoder capacity) were trained under the baseline physics weight of $\lambda_{\text{final}} = 10^{-7}$. The performance of these three variants was evaluated, and the configuration yielding the lowest validation loss was selected as the optimal architecture.

2. Regularisation Sensitivity Analysis: Second, this best-performing architecture was then systematically validated across three terminal weights: $\lambda_{\text{final}} \in \{10^{-7}, 10^{-6}, 10^{-5}\}$.

The single best configuration emerging from these evaluations was then selected as the final model and subjected to the comprehensive multi-horizon test-set evaluation.

2.5.3 Summary

The methodology presented follows the pipeline from industrial telemetry to a trained forecasting model. Preprocessing via Hampel filtering, PCHIP interpolation, and Savitzky–Golay smoothing produces differentiable trajectories. The neural architecture compresses context windows into latent representations, which are decoded into future pressure estimates. SINDy identifies a parsimonious ODE that acts as a physical regularizer through a normalized collocation loss. The linear ramp schedule for the physics weight introduces physical constraints gradually to preserve data-fit quality. This framework provides the basis for assessing forecast stability across different scales of observable-space SINDy regularization (10^{-7} to 10^{-5}).

Chapter 3

Results and Discussion

This chapter evaluates the SINDy-regularised neural forecasting framework on downhole pressure prediction for well WELL-00019 from the 3W dataset [6] under normal operating conditions. The objective is to assess whether the SINDy-based continuous collocation regularisation term improves forecasting stability across multiple horizons while maintaining pointwise accuracy.

The evaluation is organised as follows. Section 3.1 defines the validation protocol and metrics. Section 3.2 reports the offline sparse identification of the pressure dynamics and the selection of the SINDy model with 22 terms. Section 3.3 examines the training behaviour of three architecture variants and identifies the best-performing configuration. Section 3.4 analyses the effect of the physics regularisation weight λ_{phys} on the trade-off between data fidelity and physical consistency. Section 3.5 provides the final test-set evaluation across multiple forecast horizons and a chronological rollout over the full test window. Section 3.6 contextualises the findings, evaluates operational limitations, and outlines future research directions.

3.1 Evaluation Protocol and Validation Metrics

The evaluation framework is divided into two stages that mirror the pipeline structure: offline validation of the SINDy-discovered ordinary differential equation (ODE), and assessment of the neural forecasting model’s predictive performance.

3.1.1 Validation of Offline SINDy Identification

The physics-informed collocation loss depends on a closed-form ODE that approximates the dominant downhole pressure dynamics. Before this ODE is embedded in the training objective, its ability to reproduce the observed pressure evolution must be established. The identified dynamical system is evaluated against three criteria:

1. **Forward simulation accuracy:** the root-mean-square error (RMSE) between the smoothed pressure trajectory P_{sm} and the trajectory obtained by integrating the discovered ODE with a stiff implicit Runge–Kutta (Radau) solver, starting from the observed initial condition.
2. **Sparsity:** the number of active terms k retained after ℓ_1 -regularised selection; a parsimonious model is preferred to avoid over-fitting.
3. **Coefficient stability:** the variance of the estimated coefficients under bootstrap resampling of the training data, indicating sensitivity to local fluctuations.

3.1.2 Neural Forecasting Performance Metrics

All forecasting metrics are computed in physical units (MPa) after inverse-scaling the network outputs with the robust scaler parameters S_{rob} and C_{rob} (see Chapter 2):

$$P(t) = \frac{\hat{y}(t) \cdot S_{\text{rob}} + C_{\text{rob}}}{10^6} \quad [\text{MPa}] \quad (3.1)$$

The following metrics are reported for each forecast horizon $h \in \{30, 60, 90, 120\}$ s:

- **Mean Absolute Error (MAE):**

$$\text{MAE} = \frac{1}{M} \sum_{i=1}^M |P_i - \hat{P}_i| \quad [\text{MPa}] \quad (3.2)$$

- **Root Mean Squared Error (RMSE):**

$$\text{RMSE} = \sqrt{\frac{1}{M} \sum_{i=1}^M (P_i - \hat{P}_i)^2} \quad [\text{MPa}] \quad (3.3)$$

- **Coefficient of determination (R^2):**

$$R^2 = 1 - \frac{\sum_{i=1}^M (P_i - \hat{P}_i)^2}{\sum_{i=1}^M (P_i - \bar{P})^2} \quad (3.4)$$

where M represents the total number of evaluation sequences, and $\bar{P}$ represents the mean of the physical target validation pressures.

To avoid inflated performance estimates from overlapping windows, the test set is constructed with a strictly non-overlapping stride $S = W + H_{\text{max}} = 180$ s, where $W = 60$ s is the context length and $H_{\text{max}} = 120$ s is the maximum forecast horizon. Figure 3.1 illustrates this partitioning.

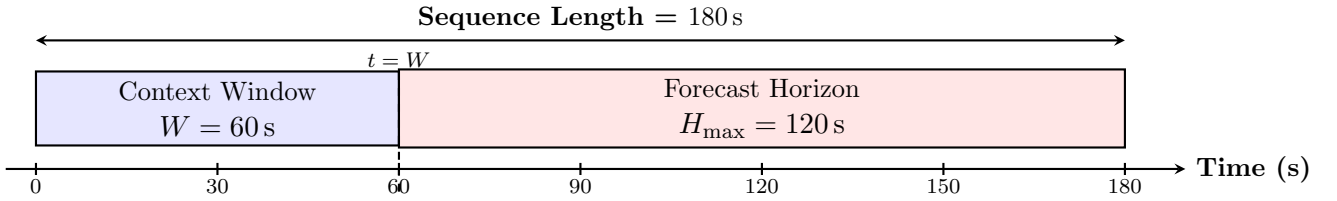


Fig. 3.1. Non-overlapping evaluation sequence ($W = 60$ s, $H_{\text{max}} = 120$ s).

3.2 Offline Sparse System Identification

The physics-informed regularisation term relies on a SINDy-discovered ODE that approximates the pressure dynamics of the gas-lifted well. Following the structural parsimony principles of sparse system identification [3], the identification was performed offline on the smoothed telemetry of WELL-00019 using the preprocessing pipeline described in Chapter 2. A 34-term candidate library was constructed from the six measured variables, comprising polynomial, interaction, boundary-flow, and thermo-hydraulic surrogate terms (see Appendix A).

Term selection was performed using LASSO (ℓ_1 -regularised) regression on a subsample of 30 000 points, with the regularisation parameter chosen by 5-fold cross-validation. The 22 terms with the largest absolute coefficients were retained and subsequently refitted with Ridge regression (ℓ_2 -regularised) to obtain unbiased coefficient estimates. The resulting model balances expressiveness and stability: a forward simulation RMSE of 6.05×10^5 Pa, an R^2 of 0.884, and a low long-horizon drift of 2.10×10^5 Pa.

Figure 3.2 displays the forward integration of the identified ODE over the full operational record using a stiff implicit Runge–Kutta (Radau) solver, with the initial condition set to the first smoothed pressure value. The simulated trajectory (dashed line) closely follows the smoothed pressure (solid

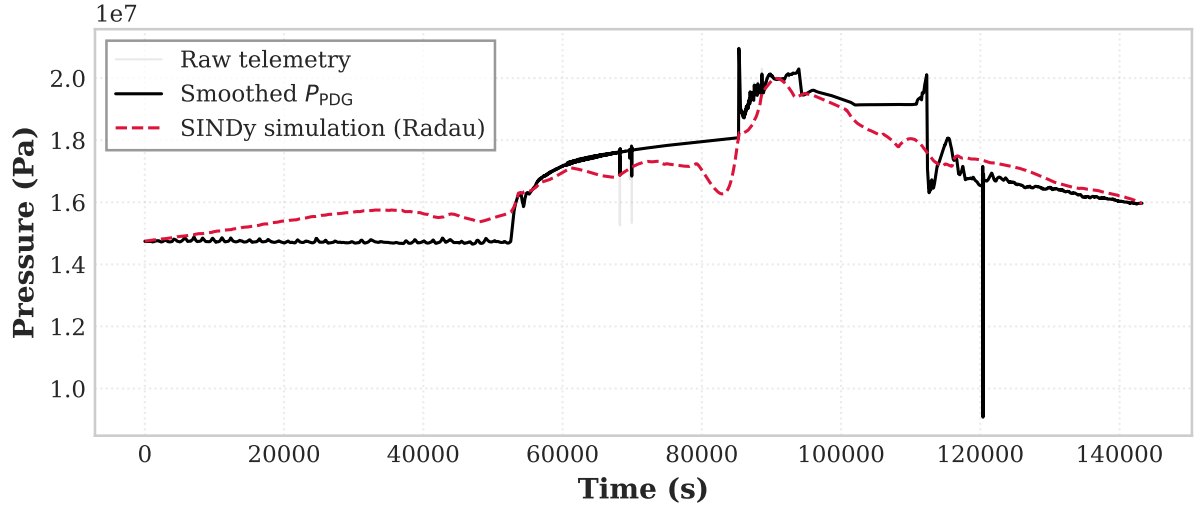


Fig. 3.2. Forward simulation of the SINDy ODE against measured downhole pressure.

line), confirming that the 22-term parsimonious model captures the dominant downhole dynamics. The root-mean-square error between the two trajectories is approximately 6×10^5 Pa (0.6 MPa).

The identified ODE takes the form:

$$\frac{dP_{\text{PDG}}}{dt} [\text{MPa/s}] = \beta_0 + \sum_{i=1}^{22} c_{\text{std},i} \cdot \text{std}(\theta_i(\mathbf{X})), \quad (3.5)$$

where $\beta_0 = 8.5607 \times 10^{-6}$ and $\text{std}(\cdot)$ denotes z -score standardisation. The 22 selected terms, standardisation statistics, and their corresponding Ridge coefficients are listed in Table 3.1.

TABLE 3.1. Selected SINDy features and Ridge coefficients.

Feature (θ_i)	Mean (μ_i)	Std. Dev. (σ_i)	Coefficient ($c_{\text{std},i}$)
sqrt_delta	6.31×10^2	7.63×10^2	-5.3127×10^{-4}
PMON	2.11×10^6	2.37×10^6	$+3.1615 \times 10^{-4}$
delta	9.32×10^5	2.56×10^6	$+3.0239 \times 10^{-4}$
$1/(1+ Q)$	5.04×10^{-1}	2.60×10^{-1}	$+1.2980 \times 10^{-4}$
$u \times \text{sqrt_delta}$	2.83×10^2	2.05×10^2	$+1.8941 \times 10^{-4}$
Q^3	6.43×10^0	5.10×10^0	$+3.5405 \times 10^{-4}$
$Q \times u^2$	1.25×10^0	9.87×10^{-1}	$+2.5301 \times 10^{-4}$
P^3	4.76×10^{21}	1.55×10^{21}	$+3.9450 \times 10^{-4}$
u^3	6.61×10^{-1}	4.55×10^{-1}	-4.6554×10^{-5}
P^2	2.80×10^{14}	6.03×10^{13}	$+5.8529 \times 10^{-5}$
PTPT	1.20×10^7	7.96×10^6	$+1.3303 \times 10^{-4}$
PJUS	1.17×10^6	5.09×10^5	-5.2017×10^{-5}
u	7.05×10^{-1}	4.22×10^{-1}	-2.5056×10^{-4}
bias	1.00×10^0	1.00×10^0	$+0.0000 \times 10^0$
Q	1.46×10^0	9.33×10^{-1}	-7.0493×10^{-4}
P	1.66×10^7	1.77×10^6	-5.7887×10^{-4}
$Q \times u$	1.30×10^0	9.63×10^{-1}	-3.3886×10^{-4}
$P \times u$	1.13×10^7	6.80×10^6	$+4.1105 \times 10^{-4}$
$P \times Q$	2.32×10^7	1.47×10^7	$+4.2480 \times 10^{-4}$
u^2	6.76×10^{-1}	4.43×10^{-1}	-2.0697×10^{-4}
Q^2	3.00×10^0	2.22×10^0	-9.6526×10^{-5}
$Q \times \text{sqrt_delta}$	7.25×10^2	6.40×10^2	$+4.6094 \times 10^{-5}$

3.3 Neural Forecasting Model: Architecture Comparison and Training

3.3.1 Model Variants and Training Setup

Three configurations of the ConvEncoder–ForecastingDecoder architecture (Section ??) were trained to assess the influence of encoder and decoder capacity on forecasting performance. All variants share the same input dimensions (6 channels, context window $W = 60$), the same output structure (scalar pressure prediction), and the same training hyperparameters described in Chapter 2 (Section 2.5.1). The variants differ in the number of convolutional channels in the encoder and the number of hidden layers and neurons in the decoder:

- **Config A (baseline):** two convolutional layers with 16 and 16 channels; decoder with two hidden layers of 64 neurons each ($\sim 7\,200$ parameters).
- **Config B (wider MLP):** same encoder as Config A; decoder with three hidden layers of 128, 128, and 64 neurons ($\sim 29\,000$ parameters).
- **Config C (deeper CNN):** three convolutional layers with 16, 32, and 64 channels; decoder identical to Config A ($\sim 22\,000$ parameters).

All configurations were trained for 1000 epochs with the linear λ_{phys} ramp schedule: $\lambda_{\text{phys}} = 0$ for epochs 0–150, increasing linearly to $\lambda_{\text{phys}} = 10^{-7}$ over epochs 150–200, and constant thereafter. The AdamW optimiser with a base learning rate of 10^{-3} , weight decay 10^{-5} , and a plateau learning rate scheduler (patience 20, decay factor 0.98, minimum learning rate 4×10^{-4}) was used throughout.

3.3.2 Training Convergence

Figures 3.3, 3.4, and 3.5 present the evolution of the data loss and physics loss for Configurations A, B, and C over the 1000-epoch training run.

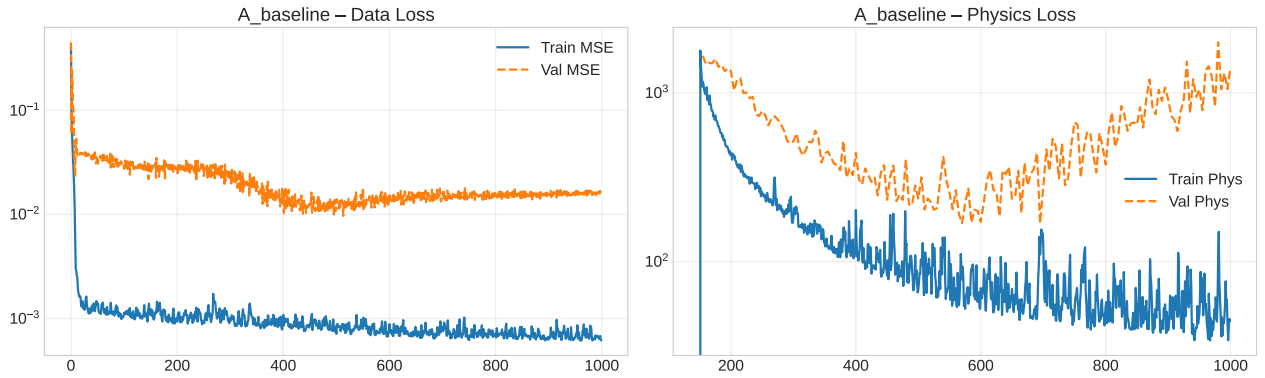


Fig. 3.3. Training curves for Config A (baseline). Data loss and physics loss (log scale). The dashed lines mark the physics-weight ramp (epochs 150–200).

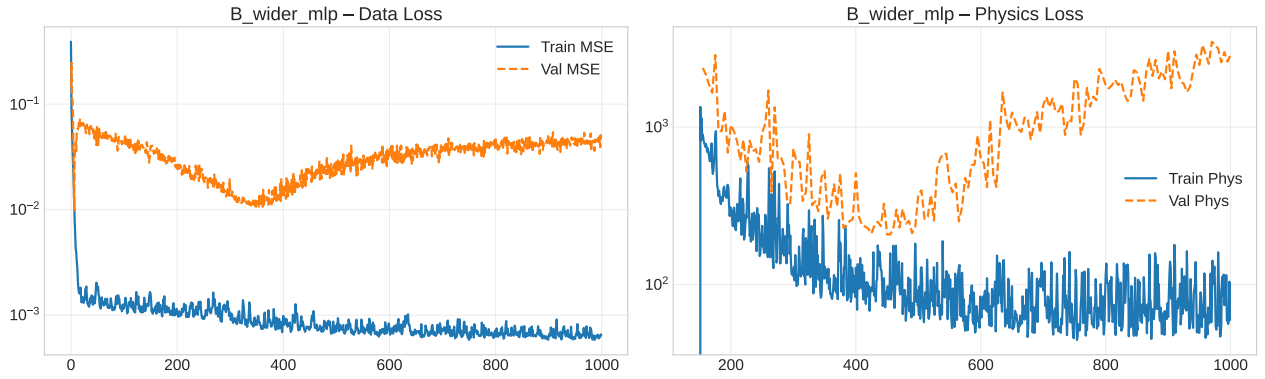


Fig. 3.4. Training curves for Config B (wider MLP). Data loss and physics loss (log scale). The dashed lines mark the physics-weight ramp (epochs 150–200).

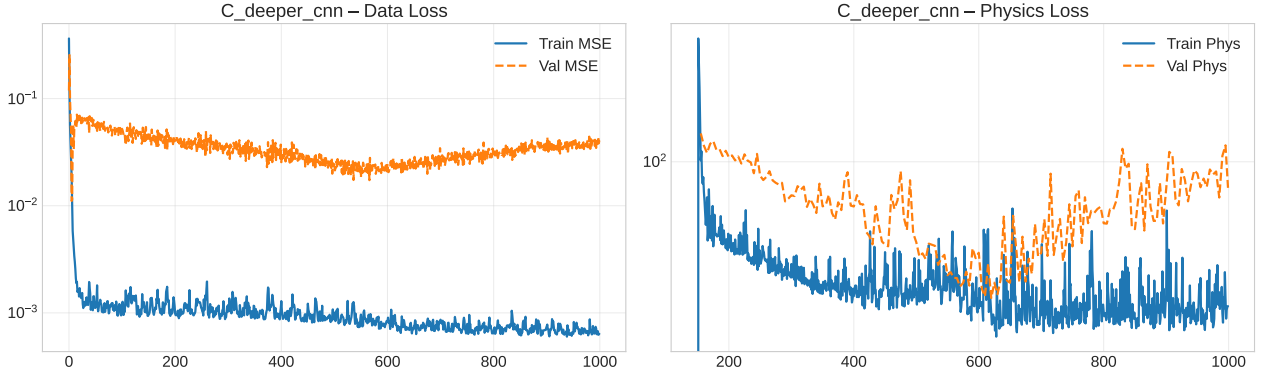


Fig. 3.5. Training curves for Config C (deeper CNN). Data loss and physics loss (log scale). The dashed lines mark the physics-weight ramp (epochs 150–200).

All three configurations converge stably, indicating that the training pipeline converges across different architectural capacities. The empirical data-fitting loss (L_{data}) decreases monotonically during the initial 150 epochs ($\lambda_{\text{phys}} = 0.0$), showing that the networks quickly capture the boundary dynamics. However, upon the activation of the physics loss at Epoch 150, the validation curves undergo a highly distinct non-monotonic “V-shape” or “U-shape” response:

- **Configuration B (Wider MLP):** Reaches its absolute validation data loss minimum of 9.9284×10^{-3} **exactly at Epoch 350**, after which it undergoes a distinct upward bounce, stabilizing near 4.0×10^{-2} .
- **Configuration A (Baseline):** Delays its validation minimum until **Epoch 450** (9.6571×10^{-3}), followed by a soft, highly bounded rise that plateaus near 1.6×10^{-2} .
- **Config C (Deeper CNN):** Prolongs its empirical descent until **Epoch 580** (1.1076×10^{-2}), after which it undergoes a very gradual, asymptotic upward drift.

This validation bounce is not an artifact of learning rate scheduling. Because the plateau scheduler uses a decay factor of 0.98 with a patience of 20 epochs, the learning rate declines continuously without the sharp, step-like drops that typically cause sudden validation plateaus.

This V-shape response represents a physical multi-objective regularization trade-off:

1. *Cooperative Minimization (Epochs 150–350):* During the early stage of the physics ramp, the network’s parameters are flexible. The model minimizes the empirical boundary errors (L_{data}) and the continuous-time SINDy derivative residuals (L_{phys}) simultaneously.
2. *Regularization Conflict (Epochs 350+):* As training progresses past the validation “valley,” the continuous-time collocation constraint becomes rigid. Since the SINDy-discovered ODE is an approximation, the optimization objectives begin to compete.
3. *Regularization Bias:* To continue minimizing training losses, the optimizer shifts parameters to satisfy SINDy derivatives. This causes the validation data loss to increase as the network departs from the empirical data manifold to satisfy physical consistency.

The rate of this transition is directly linked to the model’s structural capacity. Configuration B, which utilises a wider MLP decoder ([128, 128, 64] hidden layers), possesses a higher capacity to fit local, high-frequency gradient shapes. This makes it more sensitive to the SINDy regulariser, causing its validation curve to drift earliest (Epoch 350). Conversely, Configuration C, which compresses the context window using a deeper CNN encoder, delays this transition to Epoch 580, suggesting that convolutional feature extraction provides a natural buffering effect against gradient-based regularization stiffness.

Since Configuration A (baseline) achieved the lowest validation data loss (9.6571×10^{-3}) and the most stable, bounded validation plateau among all three models, it was selected as the optimal architecture for subsequent physical evaluations..

Table 3.2 reports the final validation metrics for each configuration, evaluated with $\lambda_{\text{phys}} = 10^{-7}$ at the best checkpoint (lowest validation data loss).

TABLE 3.2. Validation metrics for the architecture variants ($\lambda_{\text{phys}} = 10^{-7}$ at best checkpoint).

Configuration	Val. Data Loss	Val. RMSE [MPa]	Val. MAE [MPa]	Val. R^2	Parameters
Config A (baseline)	9.6571×10^{-3}	0.3026	0.2056	0.9286	$\sim 7\,200$
Config B (wider MLP)	9.9284×10^{-3}	0.3070	0.2161	0.9265	$\sim 29\,000$
Config C (deeper CNN)	1.1076×10^{-2}	0.3266	0.2471	0.9168	$\sim 22\,000$

3.3.3 Architecture Selection

Config A achieved the lowest validation data loss (9.6571×10^{-3}) and the highest R^2 (0.9286), and was therefore selected as the best architecture for the subsequent λ_{phys} sensitivity analysis and test-set evaluation. The selection is based on forecasting performance under the baseline physics regularization ($\lambda_{\text{phys}} = 10^{-7}$), ensuring that the chosen model maintains physical consistency during optimization. On the held-out test set, the gap between Config A and the higher-capacity variants was larger than the validation metrics alone suggest: at the 30 s horizon, Config B and Config C achieved R^2 values of 0.663 and 0.822, respectively, versus 0.840 for Config A, confirming that the additional parameters did not improve generalisation on this single-well dataset.

3.4 Effect of Physics Regularisation Weight

To quantify the influence of the physics loss on the trade-off between data fidelity and physical consistency, Config A (baseline, conv = [16,16], decoder hidden = [64,64]) was retrained with two larger values of the final physics weight: $\lambda_{\text{phys}} = 10^{-6}$ and $\lambda_{\text{phys}} = 10^{-5}$. Each run lasted 600 epochs; the linear ramp started at epoch 100 and ended at epoch 150. All other optimization and architectural hyperparameters remained identical.

To provide a complete, thesis-ready view of these multi-objective optimization dynamics, these sweeps are evaluated across both training-time validation losses and physical test-set metrics in Table 3.3.

TABLE 3.3. Physics loss weight sweep for Config A.

Model Identifier	λ_{final}	Val. Data Loss	Val. R^2	$\Delta\text{ValData}$ [%]	$\Delta\text{ValPhys}$ [%]
A_baseline	10^{-7}	9.6571×10^{-3}	0.9286	-24.28	+17.41
lambda_e-6	10^{-6}	1.0330×10^{-2}	0.9218	+167.08	+82.37
lambda_e-5	10^{-5}	1.0209×10^{-2}	0.9283	+142.96	+99.13

Model Identifier	λ_{final}	30s RMSE [MPa]	30s R^2	120s RMSE [MPa]	120s R^2
A_baseline	10^{-7}	0.09795	0.83978	0.08850	0.86751
lambda_e-6	10^{-6}	0.04011	0.97313	0.17552	0.47885
lambda_e-5	10^{-5}	0.34182	-0.95120	0.37482	-1.37670

3.4.1 Regularization Saturation and Model Collapse

Analysis of the validation metrics suggests that increasing λ_{phys} reduces the physics residual; at $\lambda = 10^{-5}$, the physical ODE mismatch is reduced by 99.13%. However, independent test-set evaluation reveals regularization saturation:

1. *Optimal Regularization* ($\lambda = 10^{-7}$): At this weight, the physics loss regularizes the forecast without dominating the data-fit term. It achieves a forecasting generalization gain ($\Delta\text{ValData} = -24.28\%$) and maintains stable accuracy across all horizons, peaking at $R^2 = 0.8675$ at 120 s.
2. *Short-Term Stabilization* ($\lambda = 10^{-6}$): Increasing the weight to 10^{-6} enforces a tighter physical constraint. The model achieves a short-term forecasting R^2 of 0.9731 at the 30-s horizon, indicating that moderate regularisation stabilizes short-term transients. Long-term performance degrades to $R^2 = 0.4789$.
3. *Degenerate Solution Collapse* ($\lambda = 10^{-5}$): When λ is raised to 10^{-5} , the physics constraint dominates the optimization. The optimizer reduces the physics residual by flattening the predicted pressure curves. This flat-lining satisfies the ODE residuals but eliminates forecasting capability, causing test-set R^2 to collapse.

These results are consistent with regularisation saturation in multi-objective training. Validation loss evaluated under physics-informed constraints ($\lambda = 10^{-7}$) captures the coupling between data and dynamics; however, only independent multi-horizon evaluation distinguishes improved generalisation from degenerate flat-lined solutions.

3.5 Test-Set Evaluation

Config A (baseline, trained with $\lambda_{\text{phys}} = 10^{-7}$) was evaluated on the held-out chronological test set (15% of the full record, approximately 6 hours of continuous operation).

3.5.1 Multi-Horizon Forecasting Performance

Table 3.4 reports the RMSE, MAE, and R^2 at the four forecast horizons.

TABLE 3.4. Test-set forecasting performance of Config A (MPa).

Horizon	RMSE [MPa]	MAE [MPa]	R^2
30 s	0.09795	0.09281	0.83978
60 s	0.09622	0.09089	0.84437
90 s	0.09371	0.08728	0.85044
120 s	0.08850	0.08210	0.86751

The model achieves an RMSE below 0.098 MPa across all horizons, corresponding to a relative error of less than 0.6% with respect to the observed pressure range (≈ 16.0 – 16.85 MPa). The R^2 values increase monotonically with horizon, reaching 0.86751 at 120 s.

3.5.2 Chronological Rollout

Figure 3.6 presents the continuous chronological rollout of the model predictions over the full test window.

The forecasts track the measured pressure throughout the six-hour test window, following the global pressure depletion from 16.85 to 15.97 MPa without systematic drift. The 30 s forecast resolves short-term fluctuations associated with choke cycling, while the 120 s forecast produces a smoother trajectory that follows the underlying pressure trend. Both forecasts converge to the same endpoint as the measured signal.

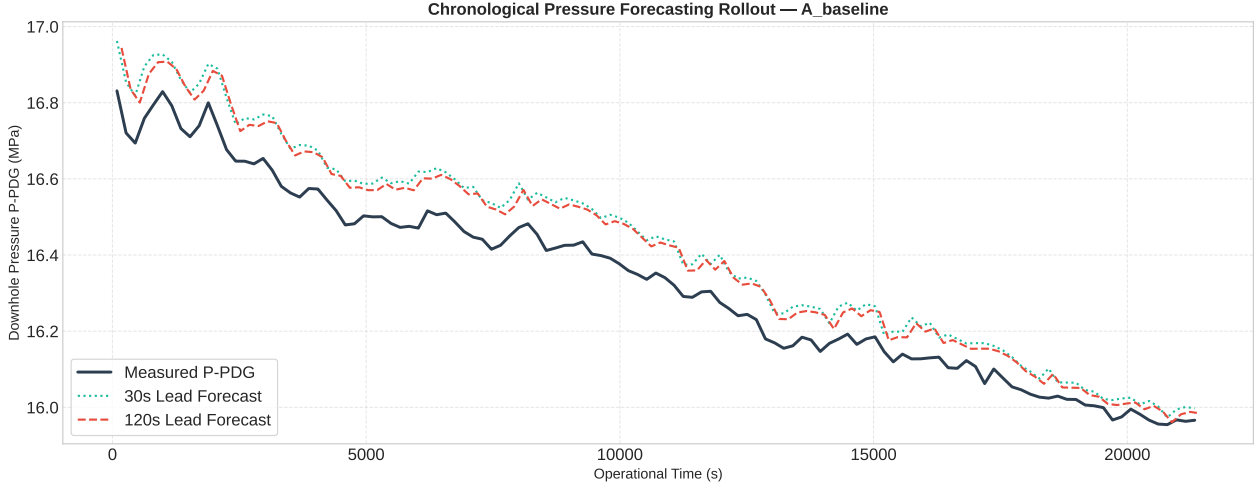


Fig. 3.6. Test-set chronological rollout for measured and predicted pressure.

3.6 Discussion

The experimental results show that the SINDy-regularised neural forecasting model achieves stable predictions across multiple forecast horizons on industrial well data. Several observations are noted below.

Effect of physics regularisation on data fitting. The training curves (Figures 3.3–3.5) indicate that the validation data loss continued to decrease after the physics loss was introduced. This suggests that the SINDy-based penalty did not interfere with the primary forecasting objective.

Architecture capacity. The architecture comparison (Table 3.2) shows that the additional parameters in Configurations B and C did not yield improvements in validation accuracy on this dataset. Config A, with approximately 7 200 parameters, achieved the best results, indicating that a compact architecture is adequate when the dominant pressure dynamics are sufficiently smooth.

Trade-off under varying λ_{phys} . The λ_{phys} sweep (Section 3.4) quantifies the trade-off between data fidelity and physical consistency. When λ_{phys} was increased from 10^{-7} to 10^{-5} , the physics loss improvement rose from +17.41% to +99.13%, while the validation data loss increased from 9.6571×10^{-3} to 1.0209×10^{-2} . However, at the extreme setting of $\lambda = 10^{-5}$, the test-set forecasting R^2 collapses to negative values, confirming that over-regularisation compromises forecasting capability.

Long-horizon stability. The chronological rollout (Fig. 3.6) shows no systematic drift over the test period. The model performance improves at longer horizons ($h = 120$ s) compared to shorter ones ($h = 30$ s). At short horizons, pressure dynamics are influenced by sensor noise and high-frequency fluid fluctuations. At longer horizons, these fluctuations average out, and the trajectory follows the physical depletion trend captured by the SINDy ODE.

Relation to prior work. The over-regularisation collapse seen at $\lambda = 10^{-5}$ matches the stiff gradient flow dynamics described by Wang *et al.* [23], where the PDE-residual Hessian eigenvalues dominate the data-fit Hessian, pushing the optimiser toward solutions that satisfy the physics constraint at the cost of data fidelity. The linear ramp schedule used here follows the curriculum regularisation idea from Krishnapriyan *et al.* [10], where physical constraints are introduced gradually, after the network has already converged on the data-fit objective — this avoids the rugged loss landscapes that come from optimising both at once.

Limitations. This evaluation only covers a single well (WELL-00019) under normal operating conditions. The SINDy ODE was identified from this specific record and may not transfer directly to wells with different geometries or reservoir properties. The model assumes the availability of all six input variables; sensor dropout scenarios were not evaluated. The λ_{phys} schedule was fixed a priori; adaptive scheduling strategies could potentially improve the trade-off further.

3.7 Conclusion

Offline SINDy identification on WELL-00019 yielded a 22-term ODE from a 34-term candidate library; forward integration reproduced the 40-hour pressure record with an RMSE of 6.05×10^5 Pa and $R^2 = 0.884$. Among three architecture variants, the baseline ConvEncoder–MLP decoder ($\sim 7\,200$ parameters) achieved the lowest validation data loss (9.6571×10^{-3} , $R^2 = 0.9286$) at $\lambda = 10^{-7}$, with test-set RMSE remaining below 0.098 MPa across all four forecast horizons (30, 60, 90, 120 s). Increasing λ to 10^{-6} raised the 30-second test R^2 to 0.9731 but reduced the 120-second R^2 to 0.4789; at $\lambda = 10^{-5}$, the 120-second R^2 fell to -1.377 . Across the optimal configuration, the 120-second forecast consistently outperformed the 30-second forecast: RMSE decreased from 0.09795 to 0.08850 MPa, while R^2 increased from 0.8398 to 0.8675, corresponding to the SINDy ODE’s capture of the net pressure depletion from 16.85 to 15.97 MPa over the six-hour test window.

General conclusion

A 22-term ordinary differential equation was identified from the smoothed operational record of WELL-00019 in the 3W dataset [6], using ℓ_1 -regularised regression followed by Ridge refitting over a 34-term candidate library restricted to the six measured variables. Forward integration of this ODE reproduced the 40-hour pressure trajectory with an RMSE of 6.05×10^5 Pa ($R^2 = 0.884$). The equation was embedded as a continuous-time collocation constraint during training of a ConvEncoder–MLP decoder model ($\sim 7\,200$ parameters), with the physics weight λ ramped linearly after a data-only pretraining phase.

At $\lambda = 10^{-7}$, validation R^2 reached 0.9286 and test-set RMSE remained below 0.098 MPa across all forecast horizons; at the 120-second horizon, test R^2 was 0.8675. Raising λ to 10^{-6} increased the 30-second test R^2 to 0.9731 at the expense of long-horizon accuracy, while at $\lambda = 10^{-5}$ the 120-second R^2 fell to -1.377 . Across the $\lambda = 10^{-7}$ configuration, the 120-second forecast yielded lower RMSE (0.08850 versus 0.09795 MPa) and higher R^2 (0.8675 versus 0.8398) than the 30-second forecast, consistent with the SINDy ODE capturing the net pressure depletion from 16.85 to 15.97 MPa over the six-hour test window.

These results indicate that a governing equation discovered from data, applied as a collocation-based regulariser at low weight, can reduce validation loss and preserve forecasting accuracy at extended horizons on this well without a pre-specified mechanistic model. The evaluation is confined to a single well under normal operating conditions. Extensions include: evaluating the SINDy-regularised approach on wells with different reservoir properties and geometries; replacing the fixed-shape ConvEncoder–MLP architecture with a neural operator that maps arbitrary-length context windows to continuous-time forecasts without retraining per horizon; substituting the single-point collocation scheme with multi-step Runge–Kutta integration of the SINDy ODE over the forecast interval; and replacing the fixed linear λ ramp with an adaptive weighting strategy that balances data and physics gradients during training.

Appendix A: Domain Physics, SINDy Feature Library, and Observability Diagnostics

A.1 Measured Variables and Sensor Availability

The 3W dataset [6] contains, in addition to valve-state flags and an operational status indicator, the 16 continuous physical variables listed in Table A.1. Valve flags and the “class”/“state” labels are not used in this work, which targets state estimation under normal operating conditions.

TABLE A.1. Sensor variables available in the 3W dataset.

Field name	Symbol	Unit	Description
P-PDG	P_{PDG}	Pa	Downhole pressure (permanent downhole gauge)
P-TPT	P_{TPT}	Pa	Subsea Xmas-tree pressure (TPT)
P-MON-CKP	P_{MON}	Pa	Upstream pressure of production choke
P-JUS-CKP	P_{JUS}	Pa	Downstream pressure of production choke
QGL	Q_{GL}	m ³ /s	Gas-lift injection flow rate
ABER-CKP	ABER	%	Production choke opening
P-ANULAR	P_{an}	Pa	Annulus pressure
P-MON-CKGL	$P_{\text{mon,gl}}$	Pa	Upstream pressure of gas-lift choke
P-JUS-CKGL	$P_{\text{jus,gl}}$	Pa	Downstream pressure of gas-lift choke
P-MON-SDV-P	$P_{\text{mon,sdv}}$	Pa	Upstream pressure of production SDV
PT-P	P_{tp}	Pa	Pressure downstream of PWV
QBS	Q_{bs}	m ³ /s	Flow rate at service pump
T-PDG	T_{PDG}	°C	Downhole temperature
T-TPT	T_{TPT}	°C	Subsea Xmas-tree temperature
T-MON-CKP	T_{mon}	°C	Upstream temperature of production choke
T-JUS-CKP	T_{jus}	°C	Downstream temperature of production choke

Not every sensor is present in every well. Table A.2 reports the percentage of valid (non-negative, non-NaN) samples after the cleaning pipeline of Chapter II. Cells are coloured according to the thresholds used throughout this appendix: $\geq 60\%$ (green) is sufficient, 20–60% (yellow) is partial, $< 20\%$ (red) is essentially absent.

TABLE A.2. Sensor availability across the candidate wells.

Variable	00019	00007	00001	00006	00003	00005	00004	00002
P-PDG	100.0	100.0	71.4	100.0	99.3	0.0	100.0	99.7
P-TPT	100.0	100.0	100.0	100.0	99.3	100.0	100.0	100.0
P-MON-CKP	94.6	100.0	100.0	100.0	99.9	100.0	100.0	0.0
P-JUS-CKP	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
QGL	100.0	100.0	100.0	100.0	99.9	0.0	0.0	0.0
ABER-CKP	100.0	100.0	0.0	0.0	0.0	100.0	100.0	0.0
P-ANULAR	0.0	0.0	100.0	100.0	99.3	100.0	100.0	100.0
P-MON-CKGL	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
P-JUS-CKGL	98.3	100.0	72.3	73.4	0.0	100.0	0.0	0.0
P-MON-SDV-P	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
PT-P	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
QBS	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
T-PDG	100.0	100.0	71.4	100.0	0.0	0.0	100.0	0.0
T-TPT	0.0	100.0	100.0	100.0	99.3	100.0	100.0	100.0
T-MON-CKP	99.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0
T-JUS-CKP	0.0	100.0	100.0	100.0	0.0	100.0	100.0	0.0

Only WELL-00019 offers sufficient coverage of the six variables required for the hybrid operator (P_{PDG} , P_{TPT} , P_{MON} , P_{JUS} , Q_{GL} , $ABER$). The remaining wells lack P_{JUS} and P_{PDG} , rendering them unsuitable for physics-constrained identification.

A.2 Well Selection and Dynamic Excitation

The criteria for retaining a well are described in Chapter II: persistent pressure variability, frequent control actuation, and virtually complete sensor coverage. Among the eight wells, only Well 00019 satisfies all three. The remaining candidates exhibit either quasi-steady dynamics, missing boundary variables, or insufficient data volume, preventing a meaningful sparse identification.

Figures A.1 and A.2 illustrate two typical inadequate behaviours. Well 00003 (Fig. A.1) has a rich record ($> 140\,000$ samples) but the downhole pressure evolves in a smooth, predictable manner with low variance ($\sigma = 3.2 \times 10^5$ Pa) and no visible control-induced transients; the lack of P_{JUS} further compromises the choke boundary feature. Well 00007 (Fig. A.2) possesses only 21 000 valid samples, insufficient to capture the system's dynamic range, and its pressure fluctuations are minimal ($\sigma = 5.6 \times 10^3$ Pa) while the downstream choke pressure is entirely absent.

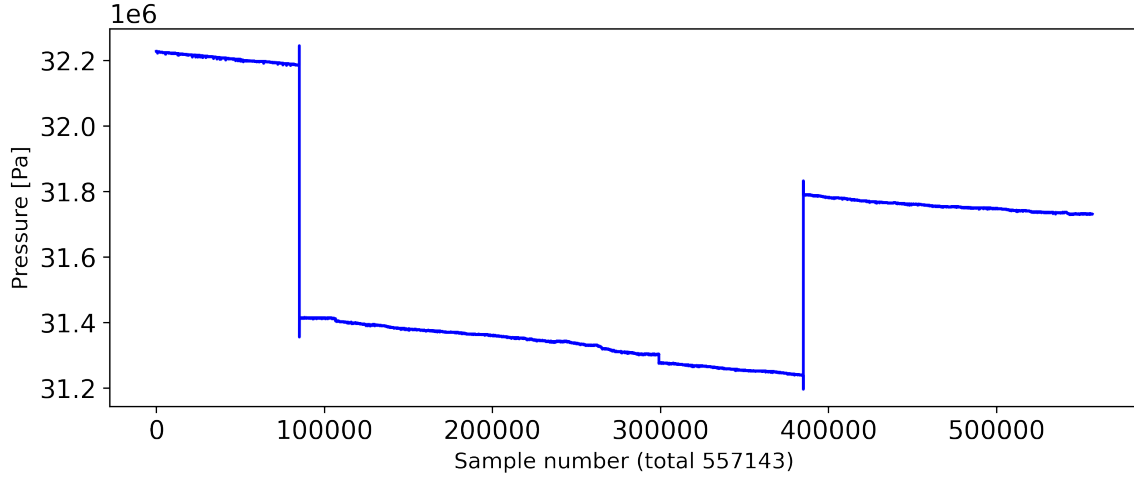


Fig. A.1. Well 00003: smoothed downhole pressure P_{PDG} over 143 231 samples.

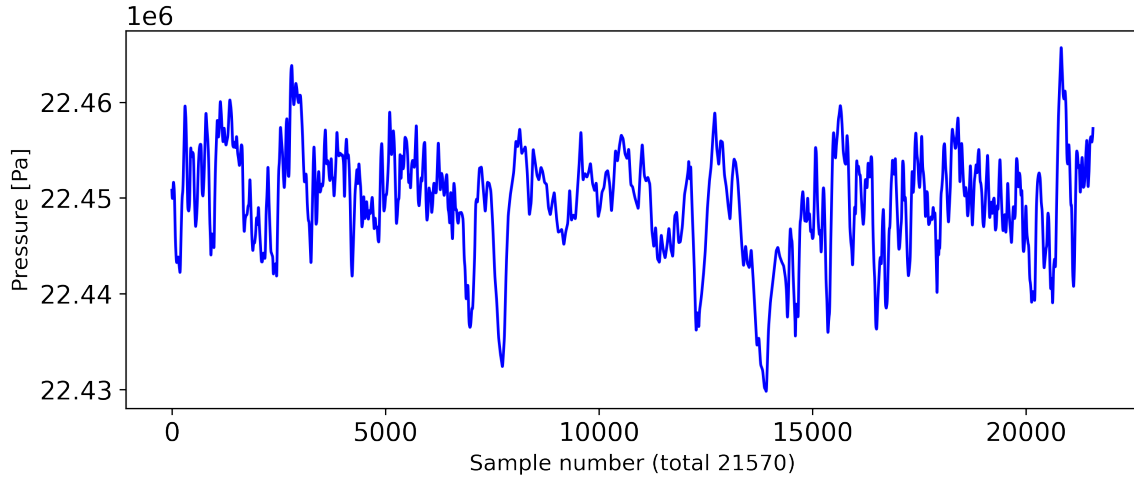


Fig. A.2. Well 00007: smoothed P_{PDG} over 21 600 samples.

A.3 Physics-Guided SINDy Feature Library

The Sparse Identification of Nonlinear Dynamics (SINDy) framework [3] identifies parsimonious governing dynamics by selecting a sparse active subset from an overcomplete candidate library. In the present gas-lift system, the canonical thermo-hydraulic formulation is fundamentally expressed in terms of latent mass states such as the annulus gas mass (m_{ga}), tubing gas mass (m_{gt}), and tubing liquid mass (m_{lt}) [24]. However, these internal states are not directly observable within the operational 3W telemetry dataset [6]. Consequently, the identified dynamics cannot be interpreted as an exact reduced-order reconstruction of the original mechanistic equations.

Instead, the identification problem is reformulated in observable space by projecting the latent thermo-hydraulic behaviour onto the measured sensor manifold:

$$\mathcal{O} = \{P_{\text{PDG}}, P_{\text{TPT}}, P_{\text{MON}}, P_{\text{JUS}}, Q, u\}, \quad (\text{A.1})$$

where $u = \text{ABER-CKP}/100$ denotes the normalised choke opening.

Under partial observability, the mapping between the hidden thermo-hydraulic states and the measured variables is not assumed to preserve the exact algebraic structure of the original governing equations after projection onto the observable manifold. Accordingly, the candidate library is not interpreted as a direct mechanistic derivation, but rather as a physically-informed observable-space basis intended to span the dominant nonlinear behaviours expected within the measured operational regime.

The resulting 34-term library therefore combines:

- affine observable states and control variables,
- low-order polynomial interaction terms,
- boundary-flow motivated nonlinearities,
- empirical attenuation and sublinear-response candidates,
- and simplified thermo-hydraulic surrogate representations.

This hybrid construction intentionally balances physical interpretability with empirical flexibility. Candidate functions are introduced as plausible observable-space representations of the underlying dynamics, while sparse regression subsequently determines which nonlinear structures remain supported by the operational data.

For implementation clarity, Table A.3 summarises the complete observable-space candidate library together with the corresponding functional interpretation and parser identifiers used during feature construction.

TABLE A.3. Observable-space SINDy candidate library.

i	Candidate θ_i	Mathematical Form	Functional Role	Code ID
Baseline Observable States and Controls				
1	P	P_{PDG}	Target State	P
2	Q	Q_{GL}	Injection Signal	Q
3	u	ABER/100	Choke Actuation	u
4–6	$P_{\text{TPT}}, P_{\text{MON}}, P_{\text{JUS}}$	Topside and choke pressures	Boundary Observables	PTPT, PMON, PJUS
7	bias	1	Constant Offset	bias
Polynomial and Interaction Basis				
8–13	P^n, Q^n, u^n	$x^2, x^3 \quad \forall x \in \{P, Q, u\}$	Smooth Nonlinear Basis	P^2, Q^3 , etc.
14–21	Cross Terms	$PQ, Pu, P^2Q, Qu^2, \dots$	Coupled Observable Interactions	$P*Q, P^2*Q$, etc.
Boundary-Flow and Empirical Nonlinearities				
22–23	$\sqrt{u}, \sqrt{Q}$	$\sqrt{\max(x, 0) + \epsilon}$	Sublinear Response Candidates	$u^{0.5}, Q^{0.5}$
24–25	Inverse Terms	$1/(1 + x + \epsilon)$	Attenuation Candidates	$1/(1+ Q)$, etc.
26–27	$\Delta P, \sqrt{\Delta P}$	$P_{\text{MON}} - P_{\text{JUS}}$	Pressure-Drop Candidates	delta, sqrt_delta
28–30	Flow-Coupling Terms	$x\sqrt{\max(\Delta P, 0) + \epsilon}$	Boundary-Flow Interactions	$u*\text{sqrt_delta}$, etc.
Thermo-Hydraulic Surrogates				
31	θ_{orifice}	Eq. (A.2)	Boundary-Flow Surrogate	orifice
32	θ_{friction}	Eq. (A.3)	Transport Surrogate	friction
33	$\theta_{\text{res_inflow}}$	Eq. (A.4)	Reservoir Coupling Surrogate	res_inflow
34	$\theta_{\text{w_inj}}$	Eq. (A.5)	Injection Surrogate	w_inj

A.3.1 Baseline States and Control Variables (Terms 1–7)

The first subset of the library contains the directly measured observable states, operational control variables, and a constant bias term. The downhole pressure P_{PDG} is included because the pressure derivative is expected to depend on the current operating state through pressure-sensitive transport and inflow behaviour. The gas-lift injection rate Q and choke opening u represent the principal operational actuation variables governing artificial-lift behaviour.

The additional pressure measurements $P_{\text{TPT}}, P_{\text{MON}}, P_{\text{JUS}}$ provide observable information regarding upstream tubing conditions and choke-boundary pressure evolution. A constant bias term is additionally included to account for persistent offsets and slowly varying operating contributions not explicitly represented by the remaining nonlinear candidates.

A.3.2 Polynomial and Interaction Basis (Terms 8–21)

After elimination of the latent mass states through observable substitutions and simplified closure approximations, the projected dynamics may contain smooth nonlinear couplings between the measured variables. Low-order polynomial terms are therefore introduced as a compact basis capable of approximating smooth observable-space interactions over the operating range represented within the dataset.

The candidate set includes quadratic and cubic powers ($P^2, P^3, Q^2, Q^3, u^2, u^3$) as well as bilinear and mixed interaction terms such as PQ, Pu, P^2Q, Qu^2 . These functions are not interpreted as individual physical laws. Rather, they provide a low-order nonlinear basis intended to approximate smooth couplings generated after projection of the latent thermo-hydraulic dynamics onto the observable manifold.

The polynomial expansion is intentionally restricted to second- and third-order interactions to balance representational flexibility against excessive feature proliferation, parameter collinearity, and regression ill-conditioning.

A.3.3 Boundary-Flow and Empirical Nonlinearities (Terms 22–30)

Certain transport mechanisms encountered in gas-lift operation exhibit non-smooth or asymptotic behaviour that may not be efficiently represented through low-order polynomial interactions alone. To improve representational coverage of such effects, the library incorporates additional nonlinear candidates motivated by boundary-flow relations and empirical operational trends.

Boundary-flow motivated candidates. Fluid transport through production and injection restrictions is commonly represented using orifice-type relations involving square-root pressure-drop dependencies [25]. Accordingly, pressure-drop-based candidates constructed from

$$\Delta P = P_{\text{MON}} - P_{\text{JUS}}$$

and the associated square-root term $\sqrt{\Delta P}$ are included as observable-space representations consistent with expected choke-flow behaviour.

Additional interaction terms such as $u\sqrt{\Delta P}$ and $Q\sqrt{\Delta P}$ are introduced to represent possible coupling between valve actuation, injection intensity, and boundary-flow response.

These terms are included as physically motivated nonlinear candidates intended to approximate observable manifestations of boundary-flow dynamics.

Empirical attenuation and sublinear-response candidates. The inverse terms

$$(1 + |Q|)^{-1}, \quad (1 + |u|)^{-1}$$

are introduced as bounded nonlinear enrichments capable of representing attenuation-like or saturation-like behaviour that may emerge under high injection or restricted-flow operating conditions.

Similarly, the square-root terms $\sqrt{Q}$ and $\sqrt{u}$ are included as empirical nonlinear candidates that may capture sublinear operational responses not efficiently represented through polynomial expansions alone.

These functions provide phenomenological observable-space hypotheses whose retention is determined through sparse regression.

A.3.4 Thermodynamic and Hydraulic Surrogates (Terms 31–34)

The final subset of the library contains mechanistically motivated surrogate terms derived from simplified thermo-hydraulic considerations while remaining compatible with the observable-space limitations imposed by the available measurements.

Choke-flow surrogate (Term 31). Canonical gas-lift formulations commonly express outlet flow as a function of valve opening, mixture density, and choke pressure drop [8]. Since the mixture density is not directly observable, it is approximated using a proportional observable proxy motivated by ideal-gas scaling under approximately stable topside operating conditions:

$$\rho_{\text{mix}} \propto P_{\text{TPT}}.$$

The downstream separator pressure appearing in the original mechanistic formulation is replaced by the nearest available downstream measurement P_{JUS} , which is employed as an operational proxy under the production regime considered. The resulting observable-space surrogate becomes:

$$\theta_{\text{orifice}} = u \sqrt{\rho_{\text{mix}} \max(P_{\text{TPT}} - P_{\text{JUS}}, 0) + \epsilon}. \quad (\text{A.2})$$

This candidate is intended to embed dominant boundary-flow structure into the observable feature space while explicitly acknowledging the approximations introduced by latent-state elimination and proxy substitution.

Tubing-friction surrogate (Term 32). Friction-related pressure losses in multiphase transport commonly scale with a density-weighted velocity magnitude. Because detailed phase velocities are unavailable, the observable injection rate is used to construct a simplified velocity proxy:

$$U_{\text{proxy}} = \frac{|Q|}{A_t}.$$

Within the studied operating regime, the injected gas signal is assumed to capture a dominant component of the observable transport variability. The resulting surrogate is therefore defined as:

$$\theta_{\text{friction}} = \rho_{\text{proxy}} U_{\text{proxy}}^2, \quad \rho_{\text{proxy}} = 0.1 \rho_{\text{mix}} + 0.9 \rho_L. \quad (\text{A.3})$$

The velocity proxy does not explicitly incorporate reservoir-fluid transport contributions; rather, it is intended as a simplified observable-space approximation of dominant friction-related trends.

Reservoir inflow surrogate (Term 33). Assuming a linear productivity-index approximation, reservoir inflow is expected to vary with the pressure drawdown relative to the reservoir pressure. Because the true flowing bottom-hole pressure is unavailable, the measured downhole pressure P_{PDG} is employed as an observable proxy:

$$\theta_{\text{res_inflow}} = \text{PI} \max(P_{\text{res}} - P_{\text{PDG}}, 0). \quad (\text{A.4})$$

This surrogate is intended to provide a simplified observable representation of reservoir coupling effects while remaining compatible with the available telemetry.

Gas-injection surrogate (Term 34). The injected gas contribution is approximated using a proportional relationship between gas injection rate and effective injected mass flow:

$$\theta_{\text{w_inj}} = c_{\text{inj}} Q. \quad (\text{A.5})$$

This simplified surrogate assumes approximately stable upstream injection conditions and idealised gas behaviour over the operating regime considered.

A small regularisation constant $\epsilon = 10^{-12}$ is applied to non-smooth operations throughout the library to maintain numerical stability during automatic differentiation and sparse optimisation.

Appendix B: Industrial Deployment and Business Perspectives

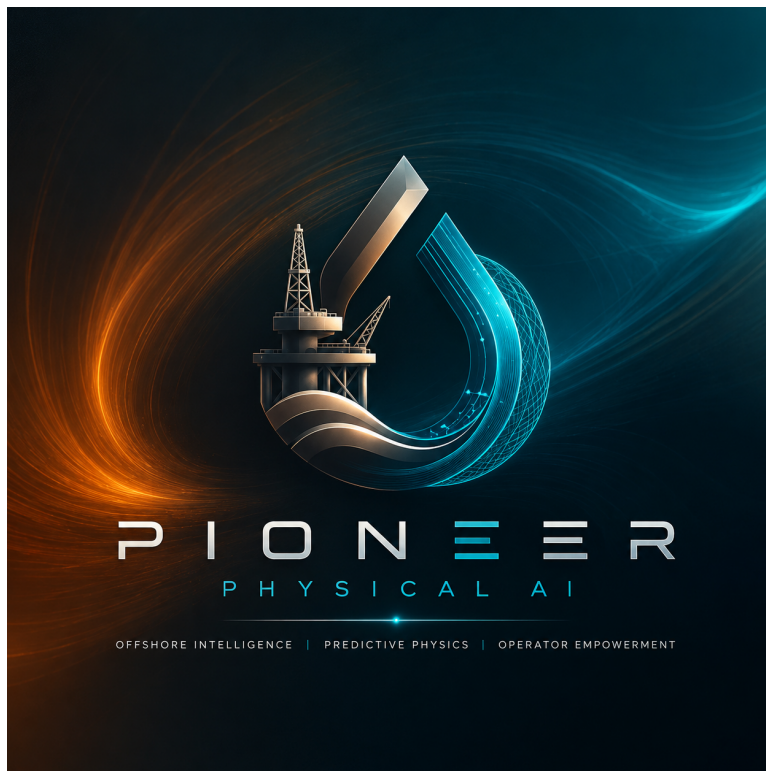


Fig. B.1. Pioneer AI: Digital twin intelligence for energy assets.

Physics-Informed AI for Heavy Industry
Digital Twin Intelligence for Energy Assets

1. Value Proposition

What problems do we solve for our customers?

We address unplanned downtime, sensor failures, and operational blind spots across energy infrastructure. Existing AI often hallucinates and lacks physical consistency, whereas traditional physics simulations are computationally too slow for real-time decisions.

What customer needs do our services satisfy?

There is a critical need for a resilient, physics-grounded digital twin that provides real-time state estimation, fault detection, and predictive insights—independent of perfect sensor data. This methodology is applicable to any energy asset, including wells, pipelines, refineries, geothermal systems, and hydrogen storage.

How is our offering different from competitors?

Our method uses hybrid physics-informed deep learning. By embedding physical laws directly into the AI, our outputs avoid hallucinations and are always physically possible. Unlike traditional simulations (CFD/FEA), inference occurs in milliseconds.

What is our unique value proposition?

“See the unseen. Decide with physics.” We provide a real-time, physics-guaranteed digital intelligence layer that turns any energy asset into a self-aware system. This reduces downtime, eliminates unnecessary sensor deployments, and enables robust proactive decision support.

2. Customer Segments

Who are our main customers?

Operators of critical energy infrastructure, spanning oil and gas (offshore and onshore), pipelines, storage facilities, refineries, geothermal plants, hydrogen systems, and thermal power stations.

What are the different customer segments we target?

1. **Oil & gas upstream:** Wells and offshore platforms.
2. **Midstream:** Transmission pipelines and LNG terminals.
3. **Downstream:** Refineries and storage facilities.
4. **Renewable & industrial energy:** Geothermal plants, concentrated solar facilities, and hydrogen electrolyzers.

What are the specific needs of each customer segment?

All segments require real-time, reliable monitoring of physical variables (pressure, temperature, flow, vibration) when sensors fail or data is noisy. Upstream operators need to avoid multi-million dollar downtimes. Pipeline asset managers require rapid leak detection. Renewable operators must monitor extreme physical conditions without deploying expensive additional hardware.

How do we categorize our customers?

Customers are categorized by asset type (rotating vs. static), environment (offshore vs. onshore), operational criticality (high-risk vs. low-risk), and adoption readiness (early adopters vs. followers).

3. Customer Relationships

What type of relationship does each segment expect?

Clients expect a high-touch, trust-based partnership. Risk-averse energy operators demand offline pilot validation using their historical data, dedicated integration support, and clear return-on-investment (ROI) demonstration before full field deployment.

How do we maintain these relationships?

Relationships are maintained through technical presentations at major industry events (ADIPEC,

OTC, SPE Summits), peer-reviewed scientific publications, direct introductions via Society of Petroleum Engineers (SPE) local chapters, and dedicated pilot partnerships with early adopters.

How can we personalize our interactions?

We offer customised digital twin dashboards per asset class, provide white-labelled API access, assign a dedicated technical account manager for each pilot facility, and co-develop use-case extensions with operator partners.

4. Channels

Through which channels do our customers want to be reached?

Clients prefer trusted industry channels such as SPE/IADC conferences, peer-reviewed journals (JPT, OnePetro), and direct technical introductions from established automation vendors or academic research partners.

Which channels are most effective for each target?

Large conferences (ADIPEC, OTC) are highly effective for connecting with Majors. Technical publications reach engineering decision-makers, LinkedIn case studies attract executives, direct pilot partnerships engage early adopters, and automation service providers facilitate mid-stream integration.

How do we integrate these channels?

A successful integration path involves: Publishing a white paper → Presenting at SPE → Inviting targets to a live demo → Offering an offline pilot on their historical data → Converting the pilot into a single-facility license → Expanding fleet-wide. All channels culminate in a unified API/plugin delivery.

5. Key Partnerships

Who are our key partners?

Pilot operator partners (providing data and field validation), automation service providers (for SCADA integration), cloud providers (AWS/Azure for High-Performance Computing), academic research labs (providing neural operator monitoring and peer review), and equipment vendors (for joint R&D).

How do partnerships improve the value proposition?

Pilot partners provide essential free data access, drastically reducing acquisition costs. Cloud providers offer compute credits. Automation vendors supply crucial API toolkits. Research labs supply cutting-edge algorithms without internal R&D overhead.

How do we align interests?

We offer pilot operators first-right of refusal on fleet licenses and share validated case studies with cloud providers for their marketing. We co-author papers with academic labs and offer automation vendors a revenue share on joint integration projects.

6. Key Activities

What are the main actions required to deliver value?

Executing customised system analysis for each client, developing secure API and SCADA integrations, running offline pilots on historical data, training domain-specific physics-informed models, filing patents, and continuously upgrading the core algorithmic framework.

What are the essential operations?

Continuous model R&D, data pipeline maintenance, cybersecurity hardening, dedicated customer support, technical publishing, and business development via industry events.

Which activities create the most value?

Real-time fault detection, physics-guaranteed state estimation, seamless SCADA integration, and

predictive maintenance alerts. These activities directly reduce unplanned downtime and extend asset lifetimes.

7. Key Resources

What are our essential assets?

- **Intangible:** Proprietary hybrid algorithm (validated on the Petrobras 3W benchmark), preparation of patent filings, and the core API codebase.
- **Human:** Physics-informed ML engineers, energy systems domain experts, and SCADA specialists.
- **Tangible:** Computing infrastructure (cloud scale).

What tools and technologies are needed?

PyTorch/TensorFlow, Docker, Railway/AWS cloud infrastructure, and industrial SCADA protocols (Modbus, OPC UA, MQTT).

What is our primary competitive advantage?

Being a first-mover in deploying physics-aware digital twins across continuous energy assets. The methodology is validated on chaotic real-world data (Petrobras 3W) and represents a rare convergence of deep learning, thermodynamics, and control theory. The agile core team absorbs research advances faster than legacy software providers.

8. Cost Structure

What are the associated costs?

Fixed costs: Enterprise-grade cloud infrastructure (scales with pilot count), intellectual property patent filing fees, and engineering salaries for the core team.

Variable costs: Compute overhead per API call (scales with user volume), travel to industry conferences, data access fees for pilot integrations, and third-party API integration costs.

What are the most significant business costs?

Acquiring specialised engineering talent (intersection of physics and machine learning) and cloud computing resources required to train large-scale digital twin models.

How can we improve operational efficiency?

Utilising open-source scientific frameworks, negotiating computational credits with cloud partners, sharing pilot validation costs with the operators, automating CI/CD pipelines, and maintaining a heavily lean core team by outsourcing non-critical tasks.

9. Revenue Streams

What are customers willing to pay for?

An annual SaaS license for digital twin API access, one-time fees for custom integration and deployment, ongoing support and model retraining, and potentially outcome-based contracts (e.g., a share of the downtime savings generated).

How do we generate revenue?

1. **Enterprise SaaS license:** Estimated \$500k–\$1M Annual Contract Value (ACV) per major operator.
2. **Custom integration:** \$100k–\$200k one-time commissioning and training fee.
3. **Support and maintenance:** 15–20% of the base license per annum.

4. **Pilot fees:** Non-disclosure R&D partnerships.

What is the pricing model?

Contracts are tiered based on the number of deployed assets, data volume, and the degree of customisation required. The progression typically follows: Pilot → Facility-level license → Fleet-wide rollout. Volume discounts are structured for multi-facility operations.

Bibliography

- [1] C. Meng, S. Griesemer, D. Cao, S. Seo, and Y. Liu, “When physics meets machine learning: A survey of physics-informed machine learning,” *Machine Learning for Computational Science and Engineering*, vol. 1, no. 1, p. 20, Jun. 2025.
- [2] M. Raissi, P. Perdikaris, and G. Karniadakis, “Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations,” *Journal of Computational Physics*, vol. 378, pp. 686–707, Feb. 2019.
- [3] S. L. Brunton, J. L. Proctor, and J. N. Kutz, “Discovering governing equations from data by sparse identification of nonlinear dynamical systems,” *Proceedings of the National Academy of Sciences*, vol. 113, no. 15, pp. 3932–3937, Apr. 2016.
- [4] J. Willard, X. Jia, S. Xu, M. Steinbach, and V. Kumar, “Integrating Scientific Knowledge with Machine Learning for Engineering and Environmental Systems,” *ACM Computing Surveys*, vol. 55, no. 4, pp. 1–37, Apr. 2023.
- [5] E. A. Antonelo, E. Camponogara, L. O. Seman, J. P. Jordanou, E. R. De Souza, and J. F. Hübner, “Physics-informed neural nets for control of dynamical systems,” *Neurocomputing*, vol. 579, p. 127419, Apr. 2024.
- [6] R. E. V. Vargas, C. J. Munaro, P. M. Ciarelli, A. G. Medeiros, B. G. D. Amaral, D. C. Barionuevo, J. C. D. D. Araújo, J. L. Ribeiro, and L. P. Magalhães, “A realistic and public dataset with rare undesirable real events in oil wells,” *Journal of Petroleum Science and Engineering*, vol. 181, p. 106223, Oct. 2019.
- [7] R. Yu and R. Wang, “Learning dynamical systems from data: An introduction to physics-guided deep learning,” *Proceedings of the National Academy of Sciences*, vol. 121, no. 27, p. e2311808121, Jul. 2024.
- [8] J. E. Kittelsen, E. A. Antonelo, E. Camponogara, and L. S. Imsland, “Physics-Informed Neural Networks with skip connections for modeling and control of gas-lifted oil wells,” *Applied Soft Computing*, vol. 158, p. 111603, Jun. 2024.
- [9] O. Hennigh, S. Narasimhan, M. A. Nabian, A. Subramaniam, K. Tangsali, M. Rietmann, J. d. A. Ferrandis, W. Byeon, Z. Fang, and S. Choudhry, “NVIDIA SimNetTM: An AI-accelerated multi-physics simulation framework,” Dec. 2020.
- [10] A. S. Krishnapriyan, A. Gholami, S. Zhe, R. M. Kirby, and M. W. Mahoney, *Characterizing Possible Failure Modes in Physics-Informed Neural Networks*. Curran Associates, Inc., 2021, vol. 34. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2021/file/df438e5206f31600e6ae4af72f2725f1-Paper.pdf
- [11] S. Cuomo, V. S. Di Cola, F. Giampaolo, G. Rozza, M. Raissi, and F. Piccialli, “Scientific Machine Learning Through Physics-Informed Neural Networks: Where we are and What’s Next,” *Journal of Scientific Computing*, vol. 92, no. 3, p. 88, Sep. 2022.
- [12] E. Haghighat, M. Raissi, A. Moure, H. Gomez, and R. Juanes, “A physics-informed deep learning framework for inversion and surrogate modeling in solid mechanics,” *Computer Methods in Applied Mechanics and Engineering*, vol. 379, p. 113741, Jun. 2021.

- [13] A. D. Jagtap, K. Kawaguchi, and G. E. Karniadakis, “Adaptive activation functions accelerate convergence in deep and physics-informed neural networks,” *Journal of Computational Physics*, vol. 404, p. 109136, Mar. 2020.
- [14] S. Wang, S. Sankaran, and P. Perdikaris, “Respecting causality is all you need for training physics-informed neural networks,” 2022. [Online]. Available: <https://arxiv.org/abs/2203.07404>
- [15] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and É. Duchesnay, “Scikit-learn: Machine Learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
- [16] D. A. Messenger and D. M. Bortz, “Weak SINDy For Partial Differential Equations,” *Journal of Computational Physics*, vol. 443, p. 110525, Oct. 2021.
- [17] S. L. Brunton, J. L. Proctor, and J. N. Kutz, “Sparse identification of nonlinear dynamics with control (sindyc),” *IFAC-PapersOnLine*, vol. 49, no. 18, pp. 710–715, 2016.
- [18] F. R. Hampel, “The influence curve and its role in robust estimation,” *J. Am. Stat. Assoc.*, vol. 69, no. 346, pp. 383–393, 1974.
- [19] P. J. Huber, *Robust Statistics*. New York: John Wiley & Sons, 1981. [Online]. Available: <https://doi.org/10.1002/0471725250>
- [20] A. Savitzky and M. J. E. Golay, “Smoothing and differentiation of data by simplified least squares procedures,” *Analytical Chemistry*, vol. 36, no. 8, pp. 1627–1639, Jul. 1964.
- [21] F. N. Fritsch and R. E. Carlson, “Monotone piecewise cubic interpolation,” *SIAM Journal on Numerical Analysis*, vol. 17, no. 2, pp. 238–246, 1980.
- [22] I. Loshchilov and F. Hutter, “Decoupled weight decay regularization,” in *International Conference on Learning Representations (ICLR)*, 2017, published online: 14 Nov 2017. [Online]. Available: <https://openreview.net/forum?id=Bkg6RiCqY7>
- [23] S. Wang, Y. Teng, and P. Perdikaris, “Understanding and mitigating gradient pathologies in physics-informed neural networks,” 2020. [Online]. Available: <https://api.semanticscholar.org/CorpusID:210473546>
- [24] E. Jahanshahi and S. Skogestad, “Simplified Dynamical Models for Control of Severe Slugging in Multiphase Risers,” *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 1634–1639, Jan. 2011.
- [25] O. Adukwu, D. Odloak, and F. Kassab Junior, “Gas-lifted systems: A hands-on tutorial on modelling and analysis,” *Franklin Open*, vol. 4, p. 100036, Sep. 2023.