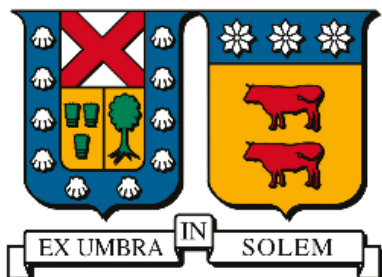


UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE INGENIERÍA QUÍMICA Y AMBIENTAL
VALPARAÍSO - CHILE



**DATA-DRIVEN MODELING OF ANAEROBIC
DIGESTION PROCESSES WITH A VIEW TO
PROCESS CONTROL: A FIRST APPROACH**

Andrés Ignacio Pino Santana

**MEMORIA DE TITULACIÓN PARA OPTAR AL TÍTULO DE INGENIERO CIVIL
QUÍMICO**

**THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF CHEMICAL CIVIL ENGINEER**

Advisor: Dr. Santiago García

Co-advisor: Dr. Alain Vande Wouwer

March – 2025

Material de referencia, su uso no involucra responsabilidad del autor o de la Institución

Reference material, its use does not involve liability on the part of the author or the institution.



Acknowledgments

Sincere gratitude is extended to my parents for their unwavering support and tireless efforts that have carried me through life's many challenges. Special thanks are also due to my friends for their invaluable advice, steadfast companionship, and continuous support that have enriched my journey beyond measure.

Heartfelt appreciation is conveyed to the extended family in Belgium for welcoming me as one of their own during my stay. Deep thanks are also due to Professor Santiago Garcia, Professor Alain Vande Wouwer, and the WBI for broadening my horizons and providing the opportunity to experience life in a new country—an experience that has profoundly impacted my personal growth. Finally, warm thanks are extended to the members of the SECO laboratory for their camaraderie and support during this journey.



Abstract

This thesis presents a data-driven modeling approach for predicting the dynamics of anaerobic digestion processes with the aim of informing process control strategies. Three neural network architectures—NARX and two ways of using LSTM— are first evaluated using a simple system—emulated by a mass balance of a second-order reaction—as a baseline, including a sensitivity analysis of parameters such as the number of excitations, sampling rate, step duration, and network complexity. The insights obtained from this analysis are then applied to guide the training of networks for predicting the behavior of anaerobic digestion, as emulated by the Hill and AM2 models. Although reliable multi-step forecasts can be achieved with abundant training data, this study examines how prediction accuracy varies as the available data and model complexity change. The results indicate that it is possible to train data-driven models that mimic the nonlinear dynamics of anaerobic digestion, considering the trade-offs between data availability and model complexity. These findings provide a foundation for subsequent investigations focused on integrating such models into control frameworks, particularly Model Predictive Control (MPC).

Keywords:

Data-driven modeling, anaerobic digestion, neural networks, NARX, LSTM, Model Predictive Control (MPC).



Glossary

A. Process and Reaction Parameters (Simple Second-Order Reaction Model):

- **D** : Dilution rate (1/h). Defines the rate at which reactor contents are renewed.
- **C** : Reactor concentration (g/L). The current concentration of the reactant within the reactor.
- **k** : Reaction rate constant (L/(g·h)). The constant governing the second-order reaction consuming the reactant.
- **C_{in}** : Inlet concentration (g/L). The concentration of the reactant entering the reactor.

B. Hill Model Parameters (Anaerobic Digestion Emulation):

- **T** : Operating temperature (°C).
- **μ_m** : Maximum specific growth rate of acidogenic bacteria (1/d), often defined as $0.013 \cdot T - 0.129$.
- **μ_{mc}** : Maximum specific growth rate of methanogenic bacteria (1/d), set equal to μ_m .
- **K_{dm}** : Decay rate for acidogenic bacteria (1/d), set equal to μ_m .
- **K_{dmc}** : Decay rate for methanogenic bacteria (1/d), set equal to μ_{mc} .
- **K_{id}** : Inhibition constant for acidogenesis (g VFA/L).
- **K_{idc}** : Inhibition constant for methanogenesis (g VFA/L).
- **Y** : Yield coefficient for acidogenic bacteria (g organism/g BVS).
- **Y_c** : Yield coefficient for methanogenic bacteria (g organism/g VFA).
- **K_s** : Half-saturation constant for substrate (g BVS/L).
- **K_{sc}** : Half-saturation constant for volatile fatty acids (g VFA/L).
- **K_i** : Inhibition constant for acidogenic bacteria (g VFA/L).



- K_{ic} : Inhibition constant for methanogenic bacteria (g VFA/L).
- S_{in} : Influent concentration of biodegradable volatile solids (g BVS/L).
- VFA_{in} : Influent concentration of volatile fatty acids (g VFA/L).
- D_0 : Nominal dilution rate (1/d).
- S_0 : Initial substrate concentration in the reactor (g BVS/L).
- X_0 : Initial concentration of acidogenic bacteria (g/L).
- X_{c0} : Initial concentration of methanogenic bacteria (g/L).
- VFA_0 : Initial concentration of volatile fatty acids (g VFA/L).

C. AM2 Model Parameters (Advanced Anaerobic Digestion Model):

- k_1 : Yield for substrate degradation (mmol/g).
- k_2 : Yield for volatile fatty acids (VFA) production (mmol/g).
- k_3 : Yield for VFA consumption (mmol/g).
- k_4 : Yield for CO₂ production (mmol/g).
- k_5 : Alternate yield for CO₂ production (mmol/g).
- k_6 : Yield for CH₄ production (mmol/g).
- k_b : Equilibrium constant (mol/L).
- k_H : Henry's constant (mmol/(L·atm)).
- k_{La} : Liquid–gas transfer constant (d⁻¹).
- k_{I2} : Inhibition constant (mmol/L).
- k_{S1} : Half-saturation constant for substrate S1 (g/L).
- k_{S2} : Half-saturation constant for VFA S2 (mmol/L).



- $\mu_{1,max}$: Maximum specific growth rate of acidogens (d^{-1}).
- $\mu_{2,max}$: Maximum specific growth rate of methanogens (d^{-1}).
- α : Fraction of bacteria in the liquid phase (dimensionless).
- P_C : CO₂ partial pressure (atm).
- P_T : Total pressure (atm).
- D_0 (AM2): Nominal dilution rate (d^{-1}) in the AM2 model.
- $X_{1,0}$: Initial concentration of acidogens (g/L).
- $X_{2,0}$: Initial concentration of methanogens (g/L).
- $S_{1,0}$: Initial substrate concentration (gCOD/L).
- $S_{2,0}$: Initial VFA concentration (mmol/L).
- Z_0 : Initial total alkalinity (mmol/L).
- C_0 : Initial total inorganic carbon concentration (mmol/L).
- S_{1in} : Influent substrate concentration (gCOD/L).
- S_{2in} : Influent VFA concentration (mmol/L).
- Z_{in} : Influent total alkalinity (mmol/L).
- C_{in} (AM2): Influent total inorganic carbon concentration (mmol/L).

D. Neural Network and System Identification Variables:

- $u(t)$: Input signal at time t (exogenous input, e.g., dilution rate changes).
- $y(t)$: System output at time t (measured response such as concentration or methane flow).
- $\epsilon(t)$: Prediction error at time t (difference between predicted and measured output).
- **MSE**: Mean Squared Error; the average of the squared prediction errors.



- **MAE:** Mean Absolute Error; the average of the absolute prediction errors.
- **AIC:** Akaike's Information Criterion; a metric balancing model fit and complexity.
- **AICc:** Small sample-size corrected AIC; adjusts AIC for biases in small datasets.

E. LSTM (Long Short-Term Memory) Specific Variables:

- x_t : Input at time t (the current input vector for the LSTM cell).
- h_t : Hidden state at time t (the LSTM cell's output or "memory" passed to the next step).
- c_t : Cell state at time t (internal memory carrying information over time).
- f_t : Forget gate activation; controls retention of the previous cell state.
- i_t : Input gate activation; determines how much new information is added.
- o_t : Output gate activation; regulates the exposure of the cell state as output.
- g_t : Candidate cell update; the new information computed to potentially update the cell state.
- **W, R, b (for i, f, o, g):** Weight matrices and bias vectors for the respective LSTM gates.

F. Training and Experimental Setup Parameters:

- **Number of Steps:** The number of consecutive excitations applied during training.
- **Step Duration Factor:** A multiplier (divided by the maximum dilution rate) that defines the duration each step is maintained.
- **Sampling Rate Factor:** A factor used with the maximum dilution rate to set the data sampling frequency.
- **Number of Delays / Window Size:** The number of previous time steps used as input to the model, providing historical context for predictions.



- **Hidden Neurons / Hidden Units:** The number of neurons in hidden layers (or the dimension of LSTM states), affecting the model's capacity to learn complex relationships.
- **Number of Layers:** The number of hidden layers in the neural network.
- **Noise Level:** The standard deviation of Gaussian noise added to the simulation outputs to mimic measurement uncertainty.



Table of contents

Acknowledgments	i
Abstract.....	ii
Glossary	iii
A. Process and Reaction Parameters (Simple Second-Order Reaction Model):	iii
B. Hill Model Parameters (Anaerobic Digestion Emulation):	iii
C. AM2 Model Parameters (Advanced Anaerobic Digestion Model):.....	iv
D. Neural Network and System Identification Variables:	v
E. LSTM (Long Short-Term Memory) Specific Variables:	vi
F. Training and Experimental Setup Parameters:	vi
Table of contents.....	viii
List of figures	xiii
List of Tables	xix
1. Introduction	1
1.1 Context and Background	1
1.2 Problem Statement and Motivation	1
1.3 Objectives	2
1.3.1 General Objective	2
1.3.2 Specific Objectives	2
1.4 Approach and tools used	3
2. Theoretical Foundations and Model descriptions.....	5
2.1 Nonlinear System Identification and Neural Network Structures	5
2.1.1 Introduction to Nonlinear System Identification	5
2.1.2 Basic Neural Network Structures	6
2.1.3 Overview of NN-Based System Identification Structures.....	6
2.1.4 Recurrent Neural Networks (RNN).....	8
2.2 Proposed Neural Networks for the Study	11
2.2.1 NARX Architecture	11
2.2.2 LSTM-A Architecture.....	12
2.2.3 LSTM-B Architecture.....	13



2.3	Overview of Process Models Used.....	14
2.3.1	Simple Second-Order Reaction Model.....	14
2.3.2	Hill Model	16
2.3.3	AM2 Model	18
3.	Experimental Methodology	21
3.1	Definition of Key Parameters	21
3.1.1	Number of Steps):.....	21
3.1.2	Step Duration Factor:.....	21
3.1.3	Sampling Rate Factor:	21
3.1.4	Number of Delays / Window Size:	21
3.1.5	Hidden Neurons / Hidden Units:	22
3.1.6	Number of Layers:.....	22
3.2	Data Set Division and Experimental Setup	22
3.2.1	Overview of Data Generation.....	22
3.2.2	Data Splits	22
3.2.3	Practical Details.....	23
3.3	Overview of Experimental Configuration	24
3.3.1	For the Simple System:	24
3.3.2	For the Advanced Anaerobic Digestion Models (Hill and AM2):.....	24
3.4	Sensitivity Analysis (For the Symple System)	25
3.4.1	Strategy for Sweeping Each Parameter	25
3.5	Training Modes.....	25
3.5.1	Training Modes.....	25
3.5.2	Training Configuration	25
3.5.3	Performance Metrics and Loss Functions	26
3.5.4	Validation and Model Selection.....	27
4.	Analysis of the Simple System.....	28
4.1	Experimental Data and Dataset Organization	28
4.2	NARX Architecture Analysis	30
4.2.1	NARX Analysis Overview	30
4.2.2	Experiments with NARX	30



4.2.3	Summary of NARX Analysis	36
4.3	LSTM-A Architecture Analysis	36
4.3.1	LSTM-A Analysis Overview	36
4.3.2	Experiments with LSTM-A	36
4.3.3	Summary of LSTM-A Analysis	41
4.4	Analysis on the Simple System: LSTM-B Architecture	41
4.4.1	LSTM-B Analysis Overview	41
4.4.2	Experiments with LSTM-B	41
4.4.3	Summary of LSTM-B Analysis	46
4.5	Overall Comparative Analysis	46
4.5.1	Training Data Diversity: Single vs. Multiple Excitations	47
4.5.2	Extrapolation (Inside vs. Outside the Training Range)	47
4.5.3	Influence of Step Duration and Sampling Rate	47
4.6	Observations on MSE and MAE	48
4.6.1	Multi-Step Performance	48
4.6.2	Overall Accuracy	48
4.6.3	Extrapolation Errors	48
4.7	Iterations and Training Efficiency	50
4.8	Simulation and Training Times	51
4.9	Discussion and Conclusions for the Simple System	52
5.	Analysis of the Anaerobic Digestion System	54
5.1	Introduction	54
5.2	Experimental Setup	54
5.2.1	Dataset Division and Training Details	55
5.3	Evaluation of LSTM-W on Anaerobic Digestion Models	56
5.3.1	Analysis on the Hill Model	56
5.3.2	Summary of MAE comparisons	63
5.4	Discussion and Comparative Analysis	65
6.	Conclusions and Future Work	67
6.1	Key Findings	67
6.1.1	Performance on the Simple System	67



6.2	Application to Anaerobic Digestion Models	67
6.3	General Insights.....	68
6.4	Limitations.....	68
6.5	Future Work	69
6.6	Final Remarks.....	69
7.	References	70
8.	Appendix	72
A.	Symple System Parameters	72
B.	Hill Model Parameters.....	72
C.	Hill Model Parameters.....	73
D.	Detailed Experimental Configurations	74
E.	NARX Experiment Figures	76
a.	Experiment A – Single-Step Excitation	76
b.	Experiment B – Multiple-Step Excitation (15-step configuration)	78
c.	Experiment C – Sensitivity Analysis	79
d.	Experiment D– Minimal Configuration Selection.....	82
F.	LSTM-A Experiment Figures.....	84
e.	Experiment A – Single-Step Excitation	84
f.	Experiment B – Multiple-Step Excitation (15-step configuration)	86
g.	Experiment C – Sensitivity Analysis	87
h.	Experiment D– Minimal Configuration Selection.....	90
G.	LSTM-B Experiment Figures.....	92
i.	Experiment A – Single-Step Excitation	92
j.	Experiment B – Multiple-Step Excitation (15-step configuration)	94
k.	Experiment C – Sensitivity Analysis	96
l.	Experiment D– Minimal Configuration Selection.....	99
H.	Benchmark Comparison	101
I.	Metrics Distribution.....	102
J.	Hill Model Experiments	106
a.	Experiment A – Two-Step Excitation	106
b.	Experiment B – Six-Step Excitation.....	107



K.	AM2 Model Experiments	108
c.	Experiment A – Two-Step Excitation	108
d.	Experiment B – Six-Step Excitation.....	109
L.	LSTM-A Snippet of Code	110
M.	LSTM-B Snippet of Code	112



List of figures

Fig. 1 Illustration of a Multilayer Perceptron (MLP). On the left, a single neuron is shown, receiving inputs ϕl with weights w_i, l and bias b_i . On the right, multiple neurons form a hidden layer (with activation functions f_j) and an output layer (with activation F_i), illustrating the two-layer MLP structure described in the text.	6
Fig. 2: Overview of some neural network-based system identification structures.	7
Fig. 3: Example of a rolled and unrolled RNN representation.	8
Fig. 4: Schematic representation of a single LSTM unit. The figure highlights how the hidden state ht , cell state ct , and gating components (ft , it , ot , and gt), interact to regulate information flow and maintain long-term dependencies.	9
Fig. 5: Illustration of the NARX network in series-parallel (open-loop) mode (left) and parallel (closed-loop) mode (right). In open-loop mode, the network is trained using the true past outputs, while in closed-loop mode it feeds back its own predictions as inputs for multi-step forecasting.	11
Fig. 6: Schematic of the LSTM-A architecture. The bottom LSTM layer receives the delayed inputs of $u(t)$ and $y(t)$, but each set of delays is treated as a single time step. The hidden (h) and cell (c) states from that step are then passed to the next layer, which produces the final prediction $y(t)$. Since the states are reset for each new input, they are not carried over across successive predictions.	13
Fig. 7: Schematic of the LSTM-B architecture. A sliding window of length w (represented by the blocks $q - 1$ through $q - w$) is applied to the input $u(t)$ and output $y(t)$. Within each window, the LSTM layers process the sequence step by step, updating the hidden (h) and cell (c) states at each time increment. The final hidden state of the top layer is then used to generate the prediction $y(t)$. Once the prediction is made, the window is shifted forward one step and the process repeats.	14
Fig. 8: Equilibrium concentration C_{eq} versus dilution rate D (left), with the red lines indicating training/testing ranges and the extended region showing augmented testing data. On the right, the time evolution of $C(t)$ is plotted for multiple dilution rates, illustrating convergence toward each equilibrium.	15
Fig. 9: Equilibrium methane production qCH_4, eq vs. D (left) and time evolution of qCH_4, eq (right), highlighting the onset of washout at higher dilution rates.	17
Fig. 10: Time evolution of the Hill model state variables $\{S, X, X_c, VFA\}$ for different dilution rates.	18
Fig. 11: Equilibrium methane flow qM, eq versus D for the AM2 model (left), and the time evolution of qM (right) under different dilution rates, illustrating stable or washout behavior.	20
Fig. 12: Time evolution of the AM2 model's state variables $\{X1, X2, S1, S2, Z, C\}$ for various dilution rates.	20
Fig. 13: Example of the training dataset (Set 1), showing step changes in the dilution rate (top) and the resulting concentration $C(t)$ (bottom).	29



Fig. 14: Testing datasets for Sets 2–5, illustrating multiple-step excitations (Sets 2 and 3) and single-step excitations (Sets 4 and 5). The left panels display the dilution rate $D(t)$, and the right panels show the corresponding concentration $C(t)$ over time.	29
Fig. 15: Predictions for Set 2 with single-step training (Experiment A). Black: noisy ground truth; red: one-step; magenta: multi-step. Multi-step forecasts diverge over time.....	31
Fig. 16: NARX predictions for Set 3 with single-step training (Experiment A). Black: noisy ground truth; red: one-step; magenta: multi-step. Limited training causes large multi-step errors, highlighting need for more excitation.	31
Fig. 17: NARX predictions for Set 2 with multi-step training (Experiment B). Black: noisy ground truth; red: one-step; magenta: multi-step. Richer training improves multi-step forecasts vs. single-step.	33
Fig. 18: NARX predictions for Set 3 with multi-step training (Experiment B). Black: noisy ground truth; red: one-step; magenta: multi-step. Multiple excitations reduce long-horizon drift, though some errors persist at extremes.	33
Fig. 19: NARX predictions for Set 2 using minimal config (Experiment D). Black: noisy ground truth; red: one-step; magenta: multi-step. Setup preserves stability and accuracy in multi-step forecasts.	35
Fig. 20: NARX predictions for Set 3 using minimal config (Experiment D). Black: noisy ground truth; red: one-step; magenta: multi-step. Optimized setup ensures stable, accurate performance under challenging conditions.....	35
Fig. 21: LSTM-A predictions for Set 2 with single-step training (Experiment A). Black: noisy ground truth; red: one-step; magenta: multi-step. Single-step training limits diversity, causing drift in multi-step forecasts.....	37
Fig. 22: LSTM-A predictions for Set 3 with single-step training (Experiment A). Black: noisy ground truth; red: one-step; magenta: multi-step. Limited dynamics from single excitation cause multi-step divergence.	37
Fig. 23: LSTM-A predictions for Set 2 with multi-step training (Experiment B). Black: noisy ground truth; red: one-step; magenta: multi-step. Multiple excitations improve long-term dynamic capture vs. single-step.....	38
Fig. 24: LSTM-A predictions for Set 3 with multi-step training (Experiment B). Black: noisy ground truth; red: one-step; magenta: multi-step. Richer training reduces long-horizon drift vs. single-step setup.	39
Fig. 25: LSTM-A predictions for Set 2 using minimal config (Experiment D). Black: noisy ground truth; red: one-step; magenta: multi-step. Tuning preserves multi-step stability despite less data and simpler architecture.	40
Fig. 26: LSTM-A predictions for Set 2 using minimal config (Experiment D). Black: noisy ground truth; red: one-step; magenta: multi-step. Parameter tuning ensures stable multi-step forecasts despite limited data and simpler model.	40
Fig. 27: LSTM-B predictions for Set 2 with single-step training (Experiment A). Black: noisy ground truth; red: one-step; magenta: multi-step. Single excitation limits dynamics, causing multi-step forecasts to drift over time.	42



Fig. 28: LSTM-B predictions for Set 3 with single-step training (Experiment A). Black: noisy ground truth; red: one-step; magenta: multi-step. Limited excitation leads to poor long-range dynamics and divergence in multi-step forecasts.	42
Fig. 29: LSTM-B predictions for Set 2 with multi-step training (Experiment B). Black: noisy ground truth; red: one-step; magenta: multi-step. Added excitations reduce drift vs. single-step training.	43
Fig. 30: LSTM-B predictions for Set 3 with multi-step training (Experiment B). Black: noisy ground truth; red: one-step; magenta: multi-step. Multiple excitations improve transition tracking, though some errors persist at extremes.	44
Fig. 31: LSTM-B predictions for Set 2 using minimal config (Experiment D). Black: noisy ground truth; red: one-step; magenta: multi-step. Optimized setup maintains stable and accurate multi-step forecasts.	45
Fig. 32: LSTM-B predictions for Set 3 using minimal config (Experiment D). Black: noisy ground truth; red: one-step; magenta: multi-step. Model remains robust even under challenging conditions.	46
Fig. 33: One-step (top row) and multi-step (bottom row) MAE for NARX, LSTM-A, and LSTM-B across Experiments A, B, and D, evaluated on five different sets (Set1–Set5).	49
Fig. 34: One-step-ahead (top row) and multi-step-ahead (bottom row) MSE (left) and MAE (right) for NARX, LSTM-A, and LSTM-B across Sets 1–5 in the Experiment D.	50
Fig. 35: Example training/validation dataset (left) and testing dataset (right) for the Hill model. The top row shows the dilution rate profiles, while the bottom row illustrates the corresponding methane flow rate (qCH_4 in $L\ CH_4/L \cdot d$).....	55
Fig. 36: Example AM2 datasets: training/validation (left) and testing (right). Top: dilution rate D ; bottom: methane flow qM . Training spans 30 days with step changes; testing covers 140 days.	56
Fig. 37: LSTM-B on Hill model with two-step training (Experiment A). Blue: ground truth; red: one-step; magenta: multi-step. Overestimation at mid-to-high methane flow suggests limited training fails to capture nonlinear dynamics.....	58
Fig. 38: Segmented multi-step prediction with reinitialization every 10 steps (gray: seed data). Slightly reduces overestimation vs. continuous mode, but high methane flow remains challenging due to limited training diversity.	58
Fig. 39: Hill model on Dataset 2 with six-step training. Black: ground truth; red: one-step; magenta: multi-step. Overestimation at high methane flow is reduced vs. two-step case.	59
Fig. 40: Segmented multi-step prediction with restarts. Black: ground truth; orange: forecasts; shaded: seed data. Further reduces overestimation, but errors remain at high methane flow. Analysis on the AM2 Model.....	60
Fig. 41: AM2 model on Dataset 2 with two-step training (Experiment A). Blue: ground truth; red: one-step; magenta: multi-step. One-step tracks well, but multi-step overestimates at mid-to-high methane flow.	61



Fig. 42: Segmented multi-step forecasting with restarts. Black: ground truth; orange: short-window forecasts; shaded: seed data. Reduces overestimation slightly, but large errors persist at high flow.	62
Fig. 43: AM2 model on Dataset 2 with six-step training (Experiment B). Black: ground truth; red: one-step; magenta: multi-step predictions. Overestimation at high methane flow is reduced vs. two-step setup, showing benefits of richer training.	63
Fig. 44: Segmented multi-step forecasting on the same dataset. Black: ground truth, orange: forecast. Shaded areas indicate seed data. This approach reduces overestimation vs. continuous forecasting, though some errors remain at high flow.	63
Fig. 45: Mean Absolute Error (MAE) for the Hill model under one-step-ahead (left) and multi-step-ahead (right) predictions, comparing Experiment A (blue) and Experiment B (orange) across training, validation, and testing sets	64
Fig. 46: Mean Absolute Error (MAE) for the AM2 model under one-step-ahead (left) and multi-step-ahead (right) predictions, comparing Experiment A (blue) and Experiment B (orange) across training, validation, and testing sets.	65
Fig. 47: Predictions for Set 1 (One-Step: left, Multi-Step: right) – Single-Step Training.....	76
Fig. 48: Predictions for Set 4 (One-Step: left, Multi-Step: right) – Single-Step Training.....	76
Fig. 49: Predictions for Set 5 (One-Step: left, Multi-Step: right) – Single-Step Training.....	77
Fig. 50: Training and validation loss curves over 45 iterations – Single-Step Training	77
Fig. 51: Predictions for Set 1 (One-Step: left, Multi-Step: right) – 15-Step Training.....	78
Fig. 52: Predictions for Set 4 (One-Step: left, Multi-Step: right) – 15-Step Training.....	78
Fig. 53: Predictions for Set 5 (One-Step: left, Multi-Step: right) – 15-Step Training.....	79
Fig. 54: Training and validation loss curves over 80 iterations – 15-Step Training.....	79
Fig. 55: MAE vs. Forecast Steps (One-Step: left, Multi-Step: right)	79
Fig. 56: MAE vs. Step Duration Factor (One-Step: left, Multi-Step: right).	80
Fig. 57: MAE vs. Sampling Rate Factor (One-Step: left, Multi-Step: right).	80
Fig. 58: MAE vs. Number of Delays (One-Step: left, Multi-Step: right)	80
Fig. 59: AIC vs. Delays (left), AICc vs. Delays (right) – One-Step	81
Fig. 60: MAE vs. Hidden Units (One-Step: left, Multi-Step: right).....	81
Fig. 61: AIC vs. Hidden Units (left), AICc vs. Hidden Units (right) – One-Step	81
Fig. 62: MAE vs. Number of Layers (One-Step: left, Multi-Step: right).....	82
Fig. 63: AIC vs. Layers (left), AICc vs. Layers (right) – One-Step	82
Fig. 64: Predictions for Set 1 (One-Step: left, Multi-Step: right) – Minimal Config	82
Fig. 65: Predictions for Set 4 (One-Step: left, Multi-Step: right) – Minimal Config	83
Fig. 66: Predictions for Set 5 (One-Step: left, Multi-Step: right) – Minimal Config	83
Fig. 67: Training and validation loss curves over 150 iterations – Minimal Config.....	83
Fig. 68: Predictions for Set 1 (One-Step: left, Multi-Step: right) – Single-Step	84
Fig. 69: Predictions for Set 4 (One-Step: left, Multi-Step: right) – Single-Step	84
Fig. 70: Predictions for Set 5 (One-Step: left, Multi-Step: right) – Single-Step	85
Fig. 71: Training and validation loss curves over 60 iterations – Single-Step	85
Fig. 72: Predictions for Set 1 (One-Step: left, Multi-Step: right) – 15-Step	86



Fig. 73: Predictions for Set 4 (One-Step: left, Multi-Step: right) – 15-Step	86
Fig. 74: Predictions for Set 5 (One-Step: left, Multi-Step: right) – 15-Step	87
Fig. 75: Training and Validation Loss curves over 50 iterations – 15-Step.....	87
Fig. 76: MAE vs. Forecast Steps (One-Step: left, Multi-Step: right).....	87
Fig. 77: MAE vs. Step Duration Factor (One-Step: left, Multi-Step: right)	88
Fig. 78: MAE vs. Sampling Rate Factor (One-Step: left, Multi-Step: right)	88
Fig. 79: MAE vs. Number of Delays (One-Step: left, Multi-Step: right)	88
Fig. 80: AIC vs. Delays (left), AICc vs. Delays (right) – One-Step	89
Fig. 81: MAE vs. Hidden Units (One-Step: left, Multi-Step: right).....	89
Fig. 82: AIC vs. Hidden Units (left), AICc vs. Hidden Units (right) – One-Step	89
Fig. 83: MAE vs. Layers (One-Step: left, Multi-Step: right)	90
Fig. 84: AIC vs. Layers (left), AICc vs. Layers (right) – One-Step	90
Fig. 85: One-step-ahead (left) and multi-step-ahead (right) predictions for Set 1 (LSTM-A)..	90
Fig. 86: One-step-ahead (left) and multi-step-ahead (right) predictions for Set 4 (LSTM-A)..	91
Fig. 87: One-step-ahead (left) and multi-step-ahead (right) predictions for Set 5 (LSTM-A)..	91
Fig. 88: Training and validation loss curves over 50 iterations (LSTM-A).	92
Fig. 89: One-step-ahead (left) and multi-step-ahead (right) predictions for Set 1 (LSTM-B)..	92
Fig. 90: One-step-ahead (left) and multi-step-ahead (right) predictions for Set 4 (LSTM-B)..	93
Fig. 91: One-step-ahead (left) and multi-step-ahead (right) predictions for Set 5 (LSTM-B)..	93
Fig. 92: Training and validation loss curves over 35 iterations (LSTM-B).	94
Fig. 93: One-step-ahead (left) and multi-step-ahead (right) predictions for Set 1 (AM2 Model, Single-Step Training).....	94
Fig. 94: One-step-ahead (left) and multi-step-ahead (right) predictions for Set 4 (AM2 Model, Single-Step Training).....	95
Fig. 95: One-step-ahead (left) and multi-step-ahead (right) predictions for Set 5 (AM2 Model, Single-Step Training).....	95
Fig. 96: MAE vs Steps (One-Step-Ahead – left, Multi-Step-Ahead – right)	96
Fig. 97: MAE vs Step Factor (One-Step-Ahead – left, Multi-Step-Ahead – right)	96
Fig. 98: MAE vs Sample Time Factor (One-Step-Ahead – left, Multi-Step-Ahead – right)	96
Fig. 99: MAE vs Window Size (One-Step-Ahead – left, Multi-Step-Ahead – right)	97
Fig. 100: AIC (left) and AICc (right) vs Window Size.....	97
Fig. 101: MAE vs Hidden Units (One-Step-Ahead – left, Multi-Step-Ahead – right)	97
Fig. 102: AIC (left) and AICc (right) vs. number of hidden units for one-step-ahead predictions.	98
Fig. 103: MAE vs. number of layers for one-step-ahead (left) and multi-step-ahead (right) predictions.	98
Fig. 104: AIC (left) and AICc (right) vs. number of layers for one-step-ahead predictions.	98
Fig. 105: One-step-ahead (left) and multi-step-ahead (right) predictions for Set 1 (LSTM-B, minimal configuration).	99
Fig. 106: One-step-ahead (left) and multi-step-ahead (right) predictions for Set 4 (LSTM-B, minimal configuration).	99



Fig. 107: One-step-ahead (left) and multi-step-ahead (right) predictions for Set 5 (LSTM-B, minimal configuration).	100
Fig. 108: Training and validation loss vs. iteration for the minimal LSTM-B configuration.	100
Fig. 109: Comparison of selected iterations (left) and total iterations (right).	101
Fig. 110: Comparison of total simulation time (left) and training time (right).	101
Fig. 111: MAE distributions for Set 1 – One-Step-Ahead Predictions.	102
Fig. 112: MAE distributions for Set 1 – Multi-Step-Ahead Predictions.	102
Fig. 113: MAE distributions for Set 2 – One-Step-Ahead Predictions.	102
Fig. 114: MAE distributions for Set 2 – Multi-Step-Ahead Predictions.	103
Fig. 115: MAE distributions for Set 3 – One-Step-Ahead Predictions.	103
Fig. 116: MAE distributions for Set 3 – Multi-Step-Ahead Predictions.	103
Fig. 117: MAE distributions for Set 4 – One-Step-Ahead Predictions.	104
Fig. 118: MAE distributions for Set 4 – Multi-Step-Ahead Predictions.	104
Fig. 119: MAE distributions for Set 5 – One-Step-Ahead Predictions.	104
Fig. 120: MAE distributions for Set 5 – Multi-Step-Ahead Predictions.	105
Fig. 121: Total MAE distributions – One-Step-Ahead Predictions (All Sets).	105
Fig. 122: Total MAE distributions – Multi-Step-Ahead Predictions (All Sets).	105
Fig. 123: One-step-ahead (left) and multi-step-ahead (right) predictions for Dataset 1 under two-step excitation (Hill Model).	106
Fig. 124: Training and validation loss vs. iteration for the Hill Model (two-step excitation).	106
Fig. 125: One-step-ahead (left) and multi-step-ahead (right) predictions for Dataset 1 (Hill Model, six-step excitation).	107
Fig. 126: Training and validation loss vs. iteration (Hill Model, six-step excitation).	107
Fig. 127: One-step-ahead (left) and multi-step-ahead (right) predictions for Dataset 1 (AM2 Model, two-step excitation).	108
Fig. 128: Training and validation loss vs. iteration (AM2 Model, two-step excitation).	109
Fig. 129: One-step-ahead (left) and multi-step-ahead (right) predictions for Dataset 1 (AM2 Model, six-step excitation).	109
Fig. 130: Training and validation loss vs. iteration (AM2 Model, six-step excitation).	110



List of Tables

Table 1: Iteration Performance Metrics for NARX, LSTM-A, and LSTM-B.....	51
Table 2: Simulation and Training Times for NARX, LSTM-A, and LSTM-B	51
Table 3: Parameters of the Simple Reaction System.....	72
Table 4: Hill Model Parameter Equations	72
Table 5: AM2 Model Parameter Values and Definitions (Bernard et al., 2001).....	73
Table 6: Base Configuration for the Simple System (Sensitivity Experiments)	74
Table 7: Parameter Variation Ranges for the Simple System (Experiment C)	75
Table 8: Dilution Rate Settings and Noise Levels.....	75
Table 9: Anaerobic Digestion Model Base Configurations (Hill and AM2)	75
Table 10: MSE and MAE for one-step-ahead and multi-step-ahead predictions (Hill Model, two-step excitation).....	107
Table 11: MSE and MAE for one-step-ahead and multi-step-ahead predictions (Hill Model, six-step excitation).....	108
Table 12: MSE and MAE for one-step-ahead and multi-step-ahead predictions (AM2 Model, two-step excitation).	109
Table 13: MSE and MAE for one-step-ahead and multi-step-ahead predictions (AM2 Model, six-step excitation).....	110



1. Introduction

1.1 Context and Background

Anaerobic digestion (AD) has increasingly gained attention as a sustainable approach for managing large volumes of municipal organic waste (Piadeh et al., 2024). In contrast to landfills—which generate greenhouse gases and pose risks to soil and water resources—AD offers the dual advantage of reducing waste volumes and producing renewable energy in the form of biogas (Piadeh et al., 2024). Beyond waste reduction, AD also contributes to broader sustainability targets by aligning with circular economy principles, helping meet multiple Sustainable Development Goals (SDGs), and enhancing resource recovery in both urban and rural settings (Piadeh et al., 2024).

Anaerobic digestion is a complex, multi-step process that encompasses both biochemical and physicochemical reactions, starting with the disintegration and hydrolysis of organic matter, followed by acidogenesis, acetogenesis, and methanogenesis (IWA, 2005). Each of these stages involves distinct microbial consortia and metabolic pathways that convert feedstock into intermediate compounds and ultimately produce biogas rich in methane and carbon dioxide (IWA, 2005). The interplay between these reactions, coupled with factors such as pH, temperature, and substrate characteristics, underscores the intricate and dynamic nature of AD processes.

The inherent complexity of anaerobic digestion has spurred extensive efforts to develop dynamic mathematical models that can capture its nonlinear behavior and facilitate process optimization (Bernard et al., 2001; Emebu et al., 2022). These models serve multiple purposes, including the simulation of reactor performance, identification of parameters, and the design of control strategies such as model predictive control (MPC) to ensure stable and efficient operation under varying conditions (Emebu et al., 2022; Haugen et al., 2013; Kil et al., 2017).

1.2 Problem Statement and Motivation

Mathematical modeling of anaerobic digestion presents a significant challenge due to the inherent complexity of its biochemical and physicochemical reactions. While existing models—ranging from descriptive frameworks like ADM1 to control-oriented models such as Hill and AM2—offer valuable insights, technological advances have improved the ability to collect process data (Cruz et al., 2021). However, the inherently slow dynamics of anaerobic digestion and the variability introduced by diverse organic residues may result in a limited amount of high-quality data for any given system. This limitation underscores the need to carefully manage data constraints when exploring data-driven modeling techniques (e.g., NARX and LSTM) as potential alternatives to traditional physics-based approaches. A central challenge is finding the right balance between model complexity and the amount of available training data. In other



words, it is crucial to identify the minimal model complexity required to achieve accurate predictions when data is limited. This balance is especially important because overly complex models may require excessive computational resources and could impair real-time control performance, while attempting to collect large volumes of data may be impractical due to the inherently slow and variable nature of anaerobic digestion.

In this work, a simple second-order reaction model—a basic mass balance with a second-order reaction—was used as a baseline to explore how training data diversity and model complexity interact to produce optimal prediction performance under nearly ideal conditions. The insights gained from this simple model are intended to guide the application of data-driven techniques to anaerobic digestion processes, as emulated by the Hill and AM2 models. Given that the slow dynamics inherent to anaerobic digestion constrain data availability, achieving the right balance between model complexity and data requirements is crucial for developing accurate short-term predictions. This balance ultimately supports the integration of data-driven models into control frameworks such as Nonlinear Model Predictive Control (NMPC).

This exploratory work aims to provide a preliminary view of integrating data-driven models into the monitoring and control framework of anaerobic digestion systems, while acknowledging the inherent challenges of capturing the full complexity of the process.

1.3 Objectives

1.3.1 General Objective

- To evaluate the feasibility and performance of data-driven neural network architectures for predicting nonlinear dynamic systems, with a focus on identifying the optimal balance between model complexity and training data quantity for anaerobic digestion.

1.3.2 Specific Objectives

1.3.2.1 Analysis of a Second-Order Reaction Model

- Analyze a simple system represented by a second-order reaction model.
- Investigate how parameters such as sampling rate, sampling time, number of neurons, delays, layers, and step duration influence prediction accuracy for one-step and multi-step forecasts.

1.3.2.2 Apply Findings to the Hill Model Emulator:

- Use findings from the simple system analysis to define suitable neural network structures for the Hill model emulator.
- Evaluate the neural network performance for one-step and multi-step predictions using constant inlet conditions.



1.3.2.3 Extend the Methodology to the AM2 Model Emulator:

- Transfer findings from the simple system analysis to define suitable neural network structures for the AM2 model emulator.
- Evaluate the neural network performance for one-step and multi-step predictions using constant inlet conditions.

1.4 Approach and tools used

This work uses data-driven modeling to address the challenges associated with predicting nonlinear dynamic behavior in anaerobic digestion systems. The methodology consists of two main phases:

- **Analysis of a Simple System:**

A simple second-order reaction model is employed as a baseline to evaluate how parameters such as sampling rate, sampling time, number of neurons, delays, layers, and step duration influence prediction accuracy. Sensitivity analyses identify suitable model configurations for both one-step and multi-step predictions.

- **Application to Process Emulators:**

Findings from the simple system analysis are applied to more complex process emulators based on the Hill and AM2 models, representing typical anaerobic digestion conditions. Neural network architectures, including NARX and LSTM variants (e.g., LSTM-A and LSTM-B), are evaluated using constant inlet conditions.

Tools and Techniques:

- **Data-Driven Modeling Techniques:**

Neural network architectures, primarily NARX and LSTM-based models, are used to capture system dynamics and provide short-term predictions relevant for control applications

- **Process Emulators:**

The Hill and AM2 models simulate anaerobic digestion conditions, providing a controlled environment to evaluate neural network predictions.

- **Parameter Sensitivity Analysis**

Parameters are systematically varied to assess their impact on model performance, measured using metrics such as Mean Squared Error (MSE) and Mean Absolute Error (MAE).



- **Experimental Setup**

Datasets are divided into training, validation, and testing sets to clearly evaluate predictive performance for both one-step and multi-step predictions.

This combination of methods and tools enables evaluation of the suitability of data-driven neural network models for improving monitoring and control in anaerobic digestion systems.

2. Theoretical Foundations and Model descriptions

2.1 Nonlinear System Identification and Neural Network Structures

2.1.1 Introduction to Nonlinear System Identification

System identification is the process of inferring a mathematical model of a dynamic system from observed input-output data. In many practical scenarios, especially when dealing with real-world processes such as chemical or biological reactions, the true system behavior is nonlinear. Traditional linear identification methods can fail to capture these nonlinearities. Therefore, **nonlinear system identification** extends the classical framework by allowing more flexible model structures, such as artificial neural networks (ANNs) (M. Norgaard et al., 2000).

A typical **system identification procedure** follows these steps (M. Norgaard et al., 2000):

1. **Experiment:** An input signal $u(t)$ is applied to the real system to collect output data $y(t)$. This experiment should capture the relevant operating range of the system.
2. **Model structure selection:** A family of candidate models is chosen (e.g., linear ARX, nonlinear NNARX, etc.), and a more specific subset (e.g., model order, time delays) is defined.
3. **Parameter estimation (training):** The model's parameters (e.g., weights in a neural network) are optimized to fit the observed data, usually by minimizing a prediction error criterion.
4. **Validation:** The trained model is tested on new or withheld data to ensure it meets the required performance. If the model is inadequate, one may return to earlier steps to refine the approach (e.g., choose a larger network, collect different data, etc.).

When focusing on **nonlinear identification**, neural networks offer a powerful approach because:

- They can approximate a wide range of nonlinear mappings (universal approximation property).
- They can be trained directly on input-output data, reducing the need for a deep first-principles (physical) model.
- They integrate well into established ARX/OE/ARMAX frameworks by replacing linear blocks with neural networks (M. Norgaard et al., 2000).

2.1.2 Basic Neural Network Structures

2.1.2.1 Perceptron and MLP

A **perceptron** (or neuron) is the simplest processing element: it sums weighted inputs and passes the result through an activation function. By combining multiple perceptrons in successive layers, we form a **Multilayer Perceptron (MLP)**. A typical MLP (see Fig. 1) has:

1. An **input layer** (no activation function, just data input).
2. One or more **hidden layers** of neurons with nonlinear activation (e.g., tanh).
3. An **output layer**, often linear for regression tasks.

Mathematically, for a two-layer MLP with one hidden layer of n_h units and an output layer of dimension m , the model can be represented (M. Norgaard et al., 2000) (in the single-output case) as:

$$\hat{y}(t) = F_i \left[\sum_{i=1}^{n_h} W_{i,j} f_j \left(\sum_{l=1}^{n_\phi} w_{j,l} \varphi_l(t) + w_{j,0} + W_{i,0} \right) \right]$$

Where $\varphi_l(t)$ are the inputs or regressors, $w_{j,l}$ and $W_{i,j}$ are weights, and f_j, F_i are activation functions.

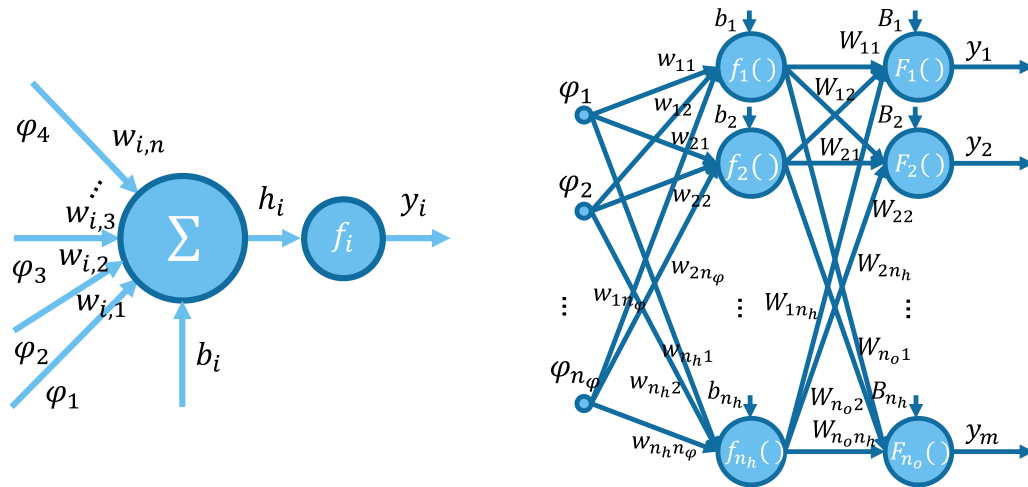


Fig. 1 Illustration of a Multilayer Perceptron (MLP). On the left, a single neuron is shown, receiving inputs ϕ_l with weights $w_{i,l}$ and bias b_i . On the right, multiple neurons form a hidden layer (with activation functions f_j) and an output layer (with activation F_i), illustrating the two-layer MLP structure described in the text.

2.1.3 Overview of NN-Based System Identification Structures

Neural networks can be combined with classical linear model structures—such as FIR, ARX, ARMAX, and OE—to yield variants like **NNFIR**, **NNARX**, **NNARMAX**, and **NNOE**

(M. Norgaard et al., 2000). Each of these variants differs in how past inputs, outputs, or predicted outputs are used as regressors, and in whether any internal feedback loops (via prediction errors or predicted outputs) are present (see Fig. 2). Below is a brief overview of each structure. (Note that **NNARX** will be discussed in more detail in Section 2.2.1)

2.1.3.1 NNFIR (Neural Network FIR)

This is the nonlinear extension of the FIR model. It only uses past inputs as regressors, meaning there is no involvement of past outputs or prediction errors. The architecture is purely feedforward.

2.1.3.2 NNARX (Neural Network AutoRegressive with eXogenous inputs)

The nonlinear counterpart to the linear ARX model. It uses past outputs and past inputs as regressors, with a neural network replacing the linear mapping to capture nonlinear dependencies. This structure is feedforward, as it relies solely on measured signals. (A more detailed discussion of the NNARX model's training and usage can be found in Section 2.2.1)

2.1.3.3 NNARMAX (Neural Network AutoRegressive, Moving Average with eXogenous inputs)

A nonlinear version of the ARMAX model, which employs past inputs, past outputs, and past prediction errors as regressors. The inclusion of prediction errors introduces an internal feedback loop into the model.

2.1.3.4 NNOE (Neural Network Output Error)

The neural-network-based counterpart to the linear OE model. It relies on predicted past outputs (rather than measured outputs) in its regressors, so the model incorporates feedback from its own predictions.

2.1.3.5 NNSSIF (Neural Network State-Space Innovations Form)

A state-space-oriented model where the state update and/or output equations are modeled by neural networks. It includes feedback from both states and prediction errors, leading to a more intricate structure.

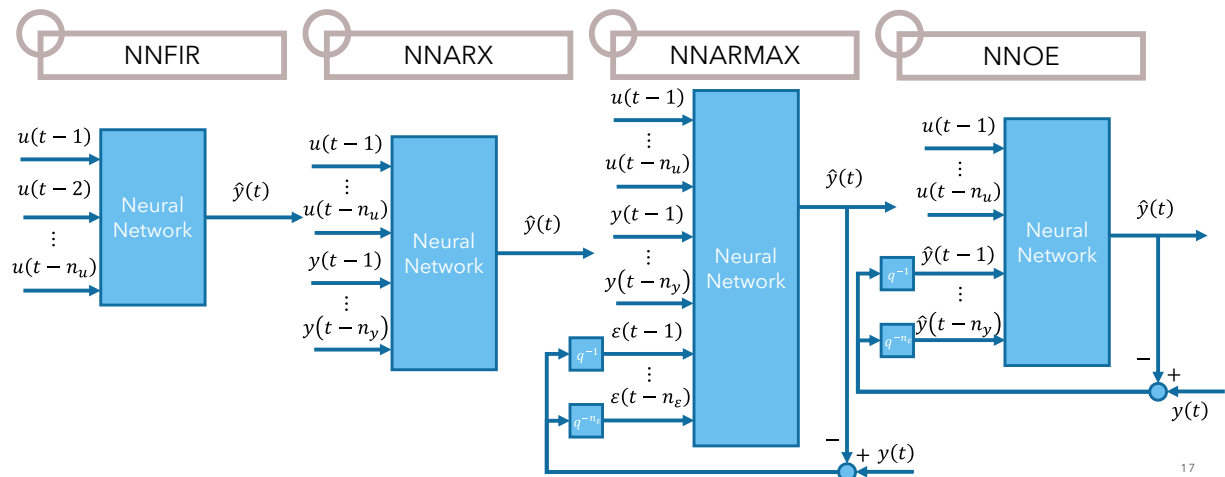


Fig. 2: Overview of some neural network-based system identification structures.

2.1.4 Recurrent Neural Networks (RNN)

RNNs are a type of neural network designed to work with sequential data. Unlike standard feedforward networks, RNNs use the history of inputs to influence the output over time. This makes them well-suited for tasks such as time-series prediction—as in this case—but they can also be applied to speech recognition and language processing.

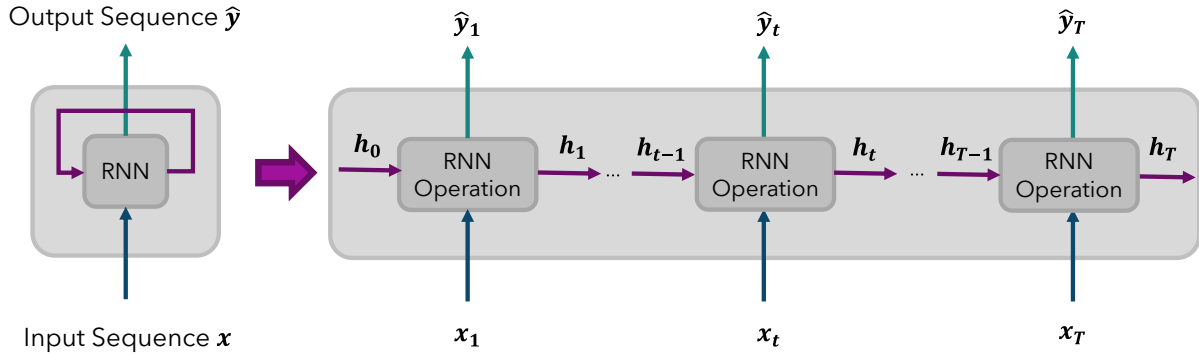


Fig. 3: Example of a rolled and unrolled RNN representation.

Fig. 3 shows an example of two ways to visualize an RNN. On the left, the **rolled RNN** illustrates the feedback loop, where the hidden state is passed from one time step to the next, allowing the network to use past information. On the right, the **unrolled/unfolded RNN** replicates the RNN cell across multiple time steps with shared weights. This view makes it easier to see how the network processes sequences over time.

This distinction between the recurrent and unrolled representations helps explain how RNNs handle sequential information (Brunton & Kutz, 2021).

2.1.4.1 LSTM

A **Long Short-Term Memory (LSTM)** network is a specialized type of Recurrent Neural Network (RNN) designed to learn long-term dependencies in sequential data (Sepp Hochreiter & Jürgen Schmidhuber, 1997). In contrast to conventional RNNs—which suffer from exploding or vanishing gradients over long sequences—LSTM units include internal mechanisms called *gates* to regulate the flow of information and gradients through time (MathWorks, 2024). These gates enable the capture of long-range temporal dependencies without losing the ability to learn short-term patterns (see Fig. 4).

1. Cell State

The cell state c_t represents the “memory” component of the LSTM unit (Sepp Hochreiter & Jürgen Schmidhuber, 1997). It provides a pathway for gradient flow across many time steps, greatly mitigating vanishing or exploding gradients compared to traditional RNNs.

2. Gates

- **Forget Gate** (f_t): Controls how much of the past cell state c_{t-1} is “forgotten” (i.e., set to zero or retained).
- **Input Gate** (i_t): Determines how much new information enters the cell state from the current input.
- **Output Gate** (o_t): Governs how much of the cell state is exposed (output) at each time step, thus regulating how much of the memory is accessed by subsequent network layers.

Each gate is typically implemented with a sigmoid activation, $\sigma(\cdot)$, which outputs values in $[0,1]$. A value of 0 means “closed” (no pass), while 1 means “open” (full pass).

3. Candidate Cell Update ($\tilde{c}_t$)

The new information to potentially add to the cell state is produced by a tanh (hyperbolic tangent) function, often called the candidate update.

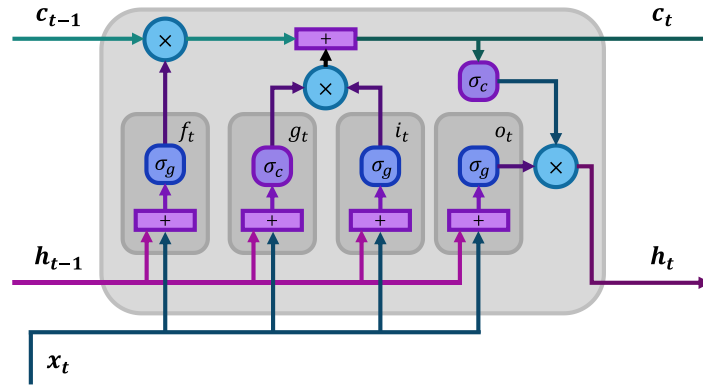


Fig. 4: Schematic representation of a single LSTM unit. The figure highlights how the hidden state h_t , cell state c_t , and gating components (f_t , i_t , o_t , and g_t), interact to regulate information flow and maintain long-term dependencies.

2.1.4.2 LSTM Equations

The formulation of LSTM units, as commonly implemented in tools such as the one used in this work (MathWorks, 2024) is given by:

$$i_t = \sigma_g(W_i x_t + R_i h_{t-1} + b_i),$$

$$f_t = \sigma_g(W_f x_t + R_f h_{t-1} + b_f),$$

$$g_t = \sigma_c(W_g x_t + R_g h_{t-1} + b_g),$$

$$o_t = \sigma_g(W_o x_t + R_o h_{t-1} + b_o),$$

where

- x_t is the input at time t .
- h_{t-1} is the hidden state from the previous time step.
- i_t, f_t, g_t, o_t input, forget, candidate, and output gate activations, respectively.
- $W_{\{\cdot\}}, R_{\{\cdot\}}, b_{\{\cdot\}}$ are weight and bias parameters.
- $\sigma_g(\cdot)$ is the logistic sigmoid
- $\sigma_c(\cdot)$ is the hyperbolic tangent activation.

The **cell state** c_t and the **hidden state** h_t then update as:

$$c_t = f_t \odot c_{t-1} + i_t \odot g_t,$$
$$h_t = o_t \odot \sigma_c(c_t),$$

where $\odot$ denotes element-wise multiplication. In plain words:

- The **forget gate** f_t decides how much of the previous cell state c_{t-1} is carried forward.
- The **input gate** i_t controls how much of the newly generated candidate g_t is integrated into the current cell state.
- The updated cell state c_t thus reflects both past information (scaled by f_t) and new information (scaled by i_t).
- The **output gate** o_t dictates how much of $\sigma_c(c_t)$ becomes the hidden state h_t that is visible to the rest of the network.

By preserving gradients in the cell state, this structure alleviates the exponential decay (or blow-up) of backpropagated gradients, a central problem in standard RNNs (Sepp Hochreiter & Jürgen Schmidhuber, 1997)

2.1.4.3 Integration of LSTM in System Identification

For **nonlinear system identification** of dynamic processes, the LSTM approach is particularly effective when the data exhibit long-term dependencies. Typical scenarios:

- **Sequence-to-One Regression:** Predicting a single future value of a time series (e.g., the system output) after reading a sequence of inputs $\{u(t-n), \dots, u(t-1)\}$.
- **Sequence-to-Sequence Forecasting:** Generating multiple future predictions at each time step.

2.2 Proposed Neural Networks for the Study

2.2.1 NARX Architecture

(See Fig. 5 for a schematic representation of the NARX architecture.)

A **NARX** (Nonlinear AutoRegressive with eXogenous inputs) neural network predicts the current output $y(t)$ using past values of both the output $\{y(t-1), \dots, y(t-n_y)\}$ and the external input (or exogenous input) $\{u(t-1), \dots, u(t-n_u)\}$. This approach has been successfully applied to dynamic systems with significant nonlinearities (Wang & Song, 2014).

- In **series-parallel mode** (sometimes called “open-loop” training or “one-step-ahead” prediction), the network sees the *true* past values of y during training.
- In **parallel mode** (sometimes referred to as “closed-loop”), the network feeds back its own predicted outputs in place of the true past y values once it is deployed for multi-step forecasting. (Note: this configuration is somewhat similar to that of the NNOE architecture described in 2.1.3.4)

During **training**, the network learns only to predict the next time step. For **multi-step forecasting**, it iteratively feeds its predictions back as inputs over multiple steps, “closing the loop” on itself.

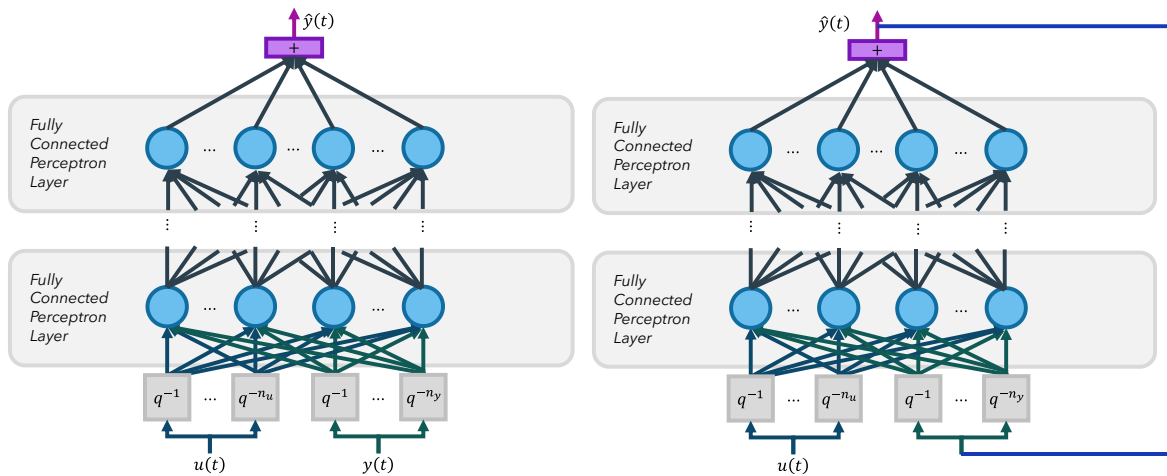


Fig. 5: Illustration of the NARX network in series-parallel (open-loop) mode (left) and parallel (closed-loop) mode (right). In open-loop mode, the network is trained using the true past outputs, while in closed-loop mode it feeds back its own predictions as inputs for multi-step forecasting.



2.2.2 LSTM-A Architecture

(See Fig. 6 for a schematic representation of the LSTM-A architecture.)

See Appendix L for a snippet of code.

The architecture referred to throughout this work as LSTM-A represents an intermediate step, combining aspects of a traditional NARX model with LSTM layers. Essentially, it replaces the nonlinear component of a standard NARX model (typically an MLP) with an LSTM layer.

However, this intermediate design does not fully utilize the LSTM's capability for long-term memory retention. This is because internal states (hidden state "h" and cell state "c") are reset to zero at each new prediction, preventing any transfer of hidden state information across successive predictions.

Each prediction is made as follows:

1. Hidden states start at zero.
2. Inputs and outputs from past time steps (as in a standard NARX model) are provided.
3. Internal states are updated based on this data.
4. Only the hidden state ("h") is subsequently used to compute the output or passed to subsequent layers.

During training, the network learns only to predict the next time step. For multi-step forecasting, it iteratively feeds its predictions back as inputs over multiple steps, "closing the loop" on itself.

It can be thought of as an intermediate architecture positioned between traditional NARX and full sequence-based LSTM models, using an LSTM layer without maintaining internal memory across prediction steps.

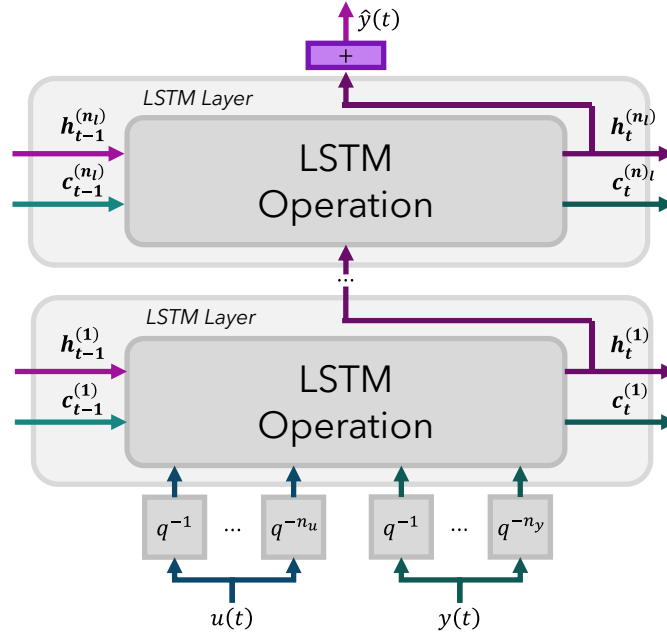


Fig. 6: Schematic of the LSTM-A architecture. The bottom LSTM layer receives the delayed inputs of $u(t)$ and $y(t)$, but each set of delays is treated as a single time step. The hidden ($\mathbf{h}$) and cell ($\mathbf{c}$) states from that step are then passed to the next layer, which produces the final prediction $\hat{y}(t)$. Since the states are reset for each new input, they are **not** carried over across successive predictions.

2.2.3 LSTM-B Architecture

(See Fig. 7 for a schematic unrolled representation of the LSTM-B architecture.)

See Appendix M for a snippet of code.

The architecture referred to throughout this work as LSTM-B uses a different approach, it explicitly leverages LSTM's memory capabilities within a fixed window of data. It processes inputs sequentially rather than treating past data as independent.

The prediction procedure for LSTM-B is as follows:

1. Internal states (hidden state "h" and cell state "c") are initialized at zero for the first time step within each window.
2. At time $t-w$, data inputs are introduced, and internal states are updated.
3. These updated states are carried forward to the next time step, where new data is processed.
4. This sequential processing continues, updating internal states at each step.
5. If multiple layers are used, the hidden state from one layer passes to the next, maintaining its temporal information.

6. Finally, the hidden state from the last time step of the last layer within the window is used to compute the predicted output at time t .

Each prediction in LSTM-B handles one data window. For the next prediction, internal states are reinitialized to zero, and the process is repeated.

During training, the network learns only to predict the next time step. For multi-step forecasting, it iteratively feeds its predictions back as inputs over multiple steps, "closing the loop" on itself.

LSTM-B may capture temporal relationships within the fixed window but does not maintain memory across windows.

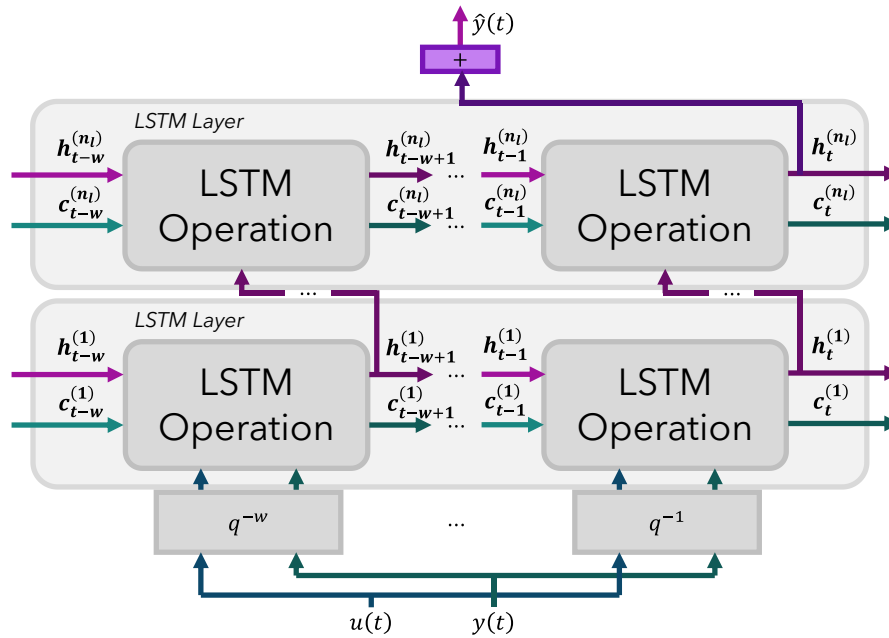


Fig. 7: Schematic of the **LSTM-B** architecture. A sliding window of length w (represented by the blocks q^{-1} through q^{-w}) is applied to the input $u(t)$ and output $y(t)$. Within each window, the LSTM layers process the sequence step by step, updating the hidden (h) and cell (c) states at each time increment. The final hidden state of the top layer is then used to generate the prediction $\hat{y}(t)$. Once the prediction is made, the window is shifted forward one step and the process repeats.

2.3 Overview of Process Models Used

2.3.1 Simple Second-Order Reaction Model

This model illustrates a basic nonlinear dynamic through a simple tank or continuously stirred reactor, where the reactant C (concentration in the reactor) experiences both an inflow and outflow (characterized by the dilution rate D) as well as a second-order reaction (kC^2). Although it is a simple system, it exhibits nonlinear behavior in the relationship between the equilibrium concentration and the dilution rate.

The main differential equation is:

$$\frac{dC}{dt} = D(C_{in} - C) - kC^2$$

where D is the dilution rate (or renewal rate of the fluid), C_{in} is the inlet concentration, C is the current concentration in the reactor, and k is the rate constant for the second-order reaction. The term $D(C_{in} - C)$ models the inflow and outflow, while $-kC^2$ represents the reaction that consumes the reactant at a rate proportional to the square of C . *For a complete list of the parameters used in the simple second-order reaction model (values and units), see Appendix A, Table 3.*

Fig. 8 illustrates two key behaviors of this model. The left panel shows how the **equilibrium concentration** C_{eq} varies with D . As D increases, C_{eq} changes in a nonlinear fashion due to the kC^2 term. As the dilution rate increases, the equilibrium concentration becomes less sensitive to changes in D , whereas at lower D , small variations can lead to more pronounced shifts in C_{eq} .

The right panel of Fig. 8 depicts the **dynamic evolution** of $C(t)$ for multiple dilution rates. Each trajectory starts from an initial condition $C(0) = 0$ and converges toward its corresponding C_{eq} . This confirms that, for each fixed D , there is a stable equilibrium state toward which the system evolves. Overall, the second-order term kC^2 gives rise to the nonlinear response, underscoring that variations in D do not necessarily produce proportional changes in C .

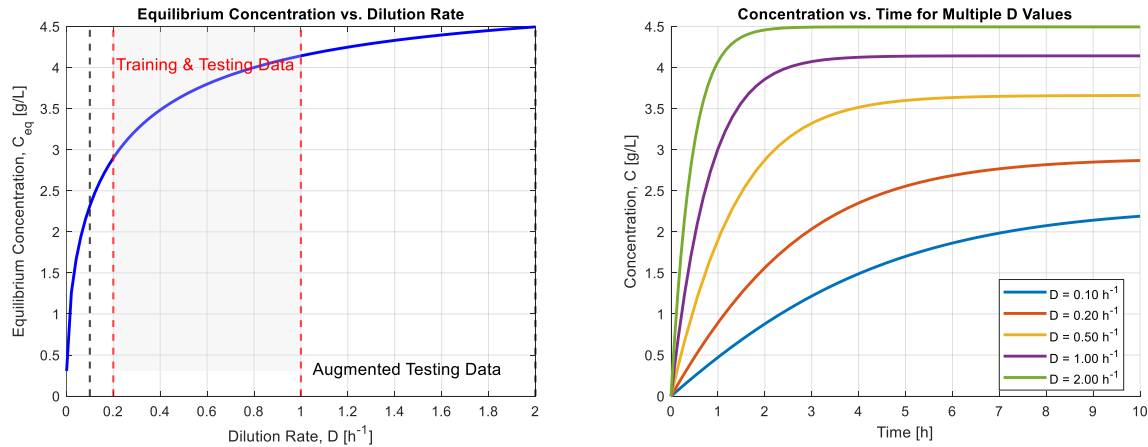


Fig. 8: Equilibrium concentration C_{eq} versus dilution rate D (left), with the red lines indicating training/testing ranges and the extended region showing augmented testing data. On the right, the time evolution of $C(t)$ is plotted for multiple dilution rates, illustrating convergence toward each equilibrium.

2.3.2 Hill Model

The Hill model is a simplified dynamic framework for anaerobic digestion originally proposed to balance accuracy and complexity (Hill, 1983). It reduces the substrate to two key fractions—Biodegradable Volatile Solids (BVS, S) and Volatile Fatty Acids (VFA)—and thereby captures essential nonlinear dynamics, including the potential washout of microbial populations, with relatively few parameters. Husain (1998) later modified Hill's formulation by making the bacterial death rates VFA-dependent, allowing for a more realistic depiction of inhibitory effects at high VFA concentrations (Haugen et al., 2013; Hill, 1983; Husain, 1998).

For a continuously stirred tank reactor (CSTR) operating at dilution rate D , the state variables are:

$$\{S, X, X_c, VFA\}$$

representing the concentrations of BVS, acidogenic bacteria, methanogenic bacteria, and volatile fatty acids, respectively. The model's mass balances are expressed by

$$\begin{aligned}\frac{dS}{dt} &= D(S_{in} - S) - \frac{\mu X}{Y} \\ \frac{dX}{dt} &= (\mu - K_d - D)X \\ \frac{dX_c}{dt} &= (\mu_c - K_{dc} - D)X_c \\ \frac{d(VFA)}{dt} &= D(VFA_{in} - VFA) + \frac{\mu X(1 - Y)}{Y} - \frac{\mu_c X_c}{Y_c}\end{aligned}$$

where μ and μ_c are the specific growth rates of acidogenic and methanogenic bacteria, respectively, given by:

$$\mu = \frac{\mu_m}{\frac{K_s}{S} + 1 + \frac{VFA}{K_i}}, \quad \mu_c = \frac{\mu_{m,c}}{\frac{K_{sc}}{S} + 1 + \frac{VFA}{K_{ic}}}$$

and the VFA-dependent death rates K_d and K_{dc} follow:

$$K_d = \frac{K_{dm}}{1 + \left(\frac{K_{id}}{VFA}\right)}, \quad K_{dc} = \frac{K_{dmc}}{1 + \left(\frac{K_{idc}}{VFA}\right)}$$

These expressions capture both the inhibitory effect of VFA on growth (via μ and μ_c) and the increased decay rate at higher VFA levels (via K_d and K_{dc}). *For a complete list of the Hill model*

parameters and their definitions (including the equations used, such as $\mu_m = 0.013 \times T - 0.129$), see Appendix B, Table 4.

Fig. 9 (left) illustrates the **equilibrium methane production** $q_{CH_4,eq}$ as a function of D . At lower D , $q_{CH_4,eq}$ increases and reaches a maximum, but it declines sharply beyond that point due to washout—where both X and X_c eventually decay to zero. On the right side of Fig. 9, the **time evolution of q_{CH_4}** is shown for different dilution rates, highlighting how higher D values induce a faster decline in methane production.

Fig. 10 depicts the **dynamic trajectories** of the state variables $\{S, X, X_c, VFA\}$ over time for several D values. Under moderate D , the system converges to a stable equilibrium, while at higher D , the microbial populations wash out. Consequently, S and VFA either stabilize near their influent concentrations or reach near-zero values, depending on whether the bacteria can sustain growth. These patterns confirm the model's ability to capture both stable and washout scenarios in anaerobic digestion.

Finally, the Hill model calculates methane production via

$$q_{CH_4} = 0.5 \frac{\mu_c X_c (1 - Y_c)}{Y_c}$$

linking the methanogenic activity directly to VFA consumption. By incorporating VFA-dependent death rates (Husain's modification), the model can better reflect inhibitory conditions and predict the resulting impact on methane output and biomass concentrations.

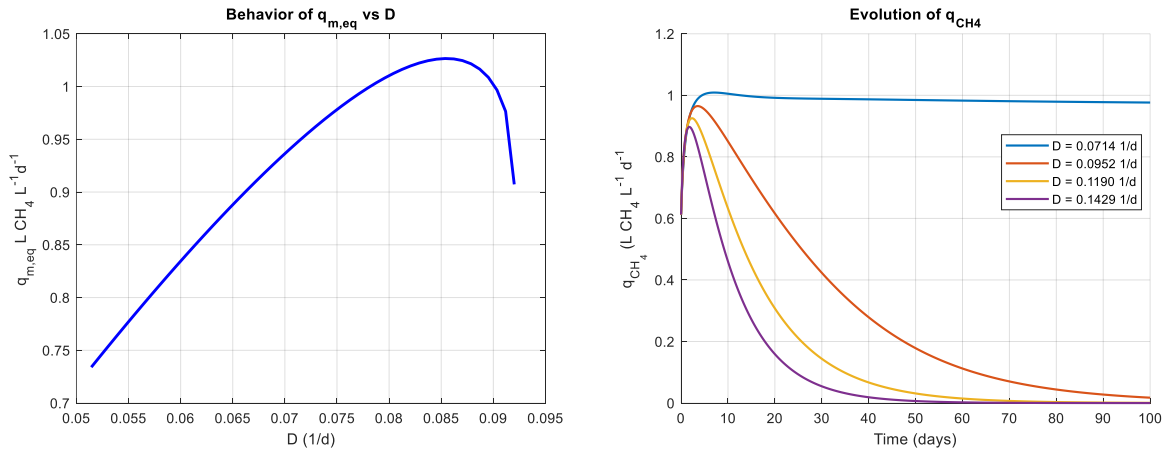


Fig. 9: Equilibrium methane production $q_{CH_4,eq}$ vs. D (left) and time evolution of $q_{CH_4,eq}$ (right), highlighting the onset of washout at higher dilution rates.

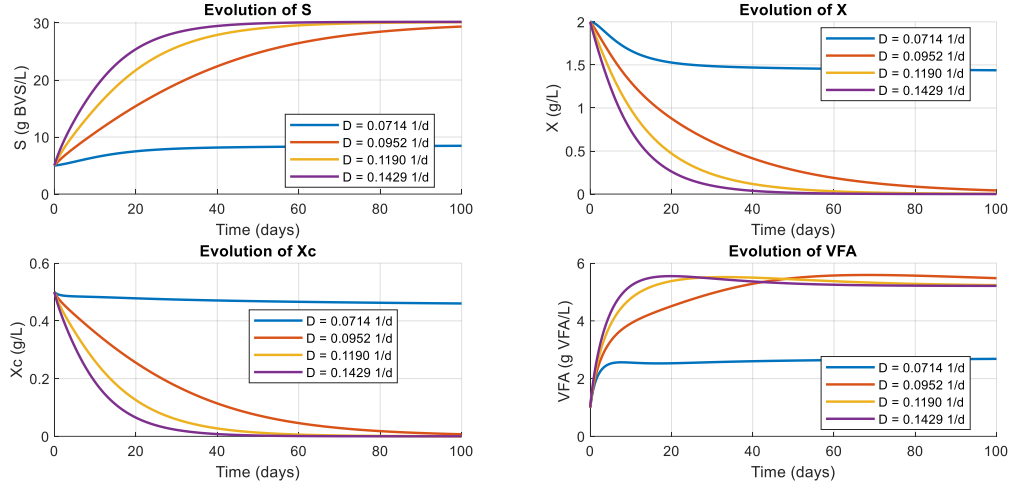


Fig. 10: Time evolution of the Hill model state variables $\{S, X, X_c, VFA\}$ for different dilution rates.

2.3.3 AM2 Model

The AM2 model is a dynamic, mass-balance-based framework developed for anaerobic wastewater treatment processes (Bernard et al., 2001). In contrast to the Hill model, AM2 considers a two-step process—encompassing acidogenesis and methanization—and explicitly incorporates electrochemical equilibria to account for alkalinity.. This formulation enables AM2 to capture both transient and steady-state behaviors under varying dilution rates D .

For a continuously stirred tank reactor (CSTR) operating at dilution rate D , the model tracks six state variables, $\{X_1, X_2, S_1, S_2, Z, C\}$, representing the concentrations of acidogenic bacteria, methanogenic bacteria, organic substrate (BVS), volatile fatty acids (VFAs), total alkalinity, and total inorganic carbon, respectively. Their mass balances are described by

$$\begin{aligned}\frac{dX_1}{dt} &= \mu_1 X_1 - \alpha D X_1 \\ \frac{dX_2}{dt} &= \mu_2 X_2 - \alpha D X_2 \\ \frac{dS_1}{dt} &= D(S_{1,in} - S_1) - k_1 \mu_1 X_1 \\ \frac{dS_2}{dt} &= D(S_{2,in} - S_2) + k_2 \mu_1 X_1 - k_3 \mu_2 X_2 \\ \frac{dZ}{dt} &= D(Z_{in} - Z)\end{aligned}$$

$$\frac{dC}{dt} = D(C_{in} - C) - q_C + k_4\mu_1X_1 + k_5\mu_5X_2$$

The acidogenic growth rate is modeled by a Monod-type expression, $\mu_1 = \mu_{1,max} S_1 / (K_{S1} + S_1)$, while the methanogenic growth rate follows Haldane kinetics to account for VFA inhibition: $\mu_2 = \mu_{2,max} S_2 / (k_{S2} + S_2 + S_2^2 / K_{I2})$. *For a complete list of the AM2 model parameters—including yield coefficients, inhibition and transfer constants, initial and feed conditions—see Appendix C, Table 5.*

Methane production is computed from the methanogenic biomass as

$$q_M = k_6\mu_2X_2$$

and reactor stability is further evaluated using pH, calculated as

$$pH = -\log_{10} \left(\frac{k_b(C - Z + S_2)}{Z - S_2} \right)$$

An auxiliary term, $\phi = (C + S_2 - Z) + k_H P_C + \mu_2 X_2 k_6 / k_{La}$, is introduced to facilitate the determination of the partial pressure of CO₂, which is obtained by enforcing liquid–gas equilibrium:

$$P_C = \frac{\phi - \sqrt{\phi^2 - 4k_H P_T (C + S_2 - Z)}}{2k_H}$$

with P_T representing the total gas pressure. Finally, the CO₂ flow rate is given by

$$q_C = k_{La}((C + S_2 - Z) - k_H P_C)$$

Figures illustrate key model behaviors. Fig. 11 **(left)** shows the equilibrium methane flow $q_{M,eq}$ as a function of D ; at low dilution rates, $q_{M,eq}$ increases nearly linearly before reaching a maximum and then declining sharply due to washout (as X_2 tends to zero). Fig. 11 **(right)** depicts the time evolution of q_M under different D values, highlighting how higher dilution rates induce a faster decline in methane production. Fig. 12 presents the dynamic trajectories of the state variables $\{X_1, X_2, S_1, S_2, Z, C\}$ over time; under moderate D , the variables converge to stable equilibria, whereas at higher D the methanogenic biomass X_2 and inorganic carbon C decline toward zero, reflecting washout.

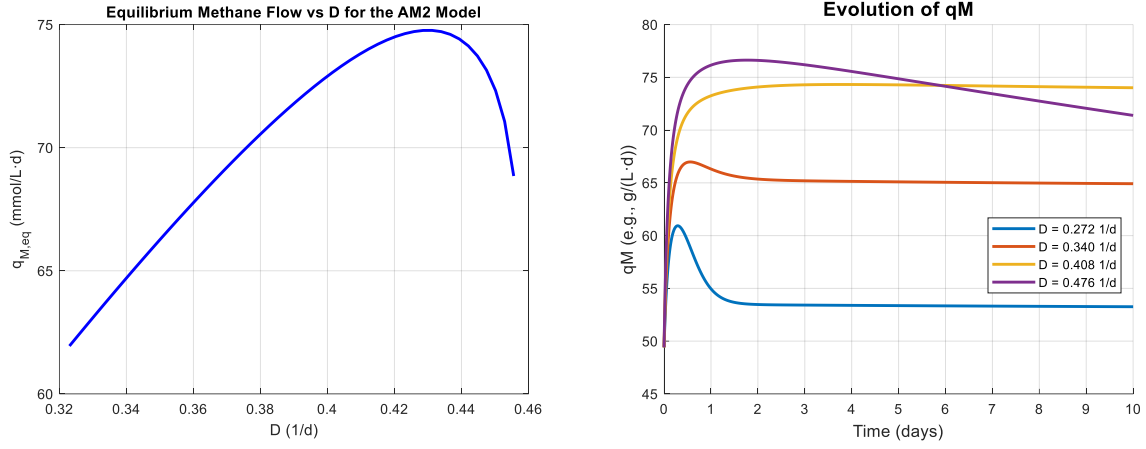


Fig. 11: Equilibrium methane flow $q_{M,eq}$ versus D for the AM2 model (left), and the time evolution of q_M (right) under different dilution rates, illustrating stable or washout behavior.

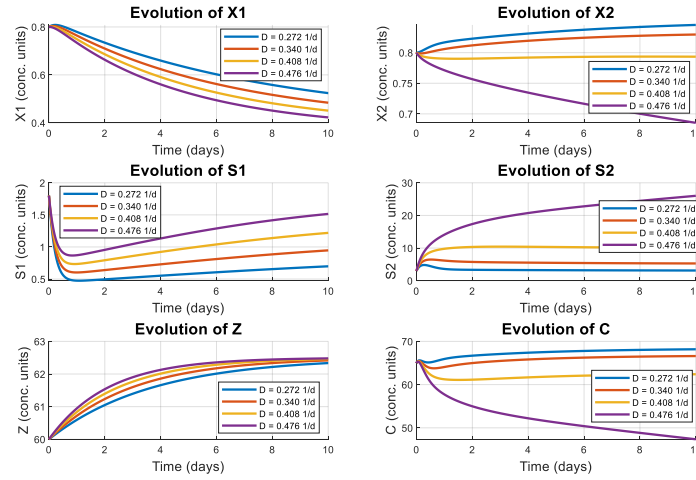


Fig. 12: Time evolution of the AM2 model's state variables $\{X_1, X_2, S_1, S_2, Z, C\}$ for various dilution rates.



3. Experimental Methodology

This chapter describes the data generation procedures, dataset organization, and strategies used for training and evaluating the neural networks. The methodology presented here supports a systematic comparison of different architectures and parameter configurations for both the simple second-order system and the more advanced anaerobic digestion models (Hill and AM2), which serve as emulators of real process data.

3.1 Definition of Key Parameters

In this study, several key parameters were defined and systematically varied throughout the experiments. These parameters, described below, are used in data generation, model training, and sensitivity analysis:

3.1.1 Number of Steps:

This represents the number of consecutive excitations applied to the system, starting from a steady state.

3.1.2 Step Duration Factor:

The duration of each step is defined as the product of a factor and the inverse of the maximum dilution rate inside the range, i.e.,

$$\text{Step Duration} = \frac{\text{Factor}}{D_{\max}}$$

This factor adjusts how long each operating condition is maintained during the excitation.

3.1.3 Sampling Rate Factor:

Similarly, the sampling rate is determined by the inverse of the product of the maximum dilution rate and a factor, i.e.,

$$\text{Sampling Rate} = \frac{1}{D_{\max} \cdot \text{Factor}}$$

3.1.4 Number of Delays / Window Size:

This parameter indicates the number of previous time steps (or delays) fed into the network, typically ranging from $t - 1$ to $t - n$. In architectures like NARX or LSTM-A, it represents the amount of historical information used to predict the next time point. In LSTM-B, the same delays form a sequence window processed step by step, with internal states updated at each step but reset between predictions.



3.1.5 Hidden Neurons / Hidden Units:

For feedforward networks (e.g., NARX), this is the number of neurons in a hidden layer. For recurrent networks such as LSTM, it represents the dimensionality of the hidden state vector h and the cell state vector c . This parameter essentially governs the network's capacity to model complex nonlinear relationships.

3.1.6 Number of Layers:

This is the number of hidden layers in the network. In this analysis, it is assumed that each hidden layer contains the same number of neurons or units for simplicity, even though in practice layers might have different sizes.

For further details—including numerical values, ranges, and definitions—please refer to Appendix D (Table 6 and Table 7).

3.2 Data Set Division and Experimental Setup

3.2.1 Overview of Data Generation

For the **Simple System**, the second-order reaction model described in Section 2.3.1 was simulated under varying operating conditions. In these simulations, step changes in operating parameters were introduced—always starting from steady state—to excite the system. Gaussian noise was added to the output C with a standard deviation that was adjusted and visually inspected.

For the **Hill** and **AM2 Models**, the equations described in Sections 2.3.2 and 2.3.3, respectively, were used to emulate real process data. The models were subjected to step or random changes to the influent conditions. Gaussian noise was also added to the output q_m for both models, with the standard deviation set according to visual assessment.

(Noise levels and dilution rate settings detailed in Appendix D, Table 8 and Table 9)

All simulations were performed in MATLAB using the `ode45` solver.

3.2.2 Data Splits

For the Simple System (5 Sets):

1. Set 1 (Training and Validation):

This set includes multiple step changes within a defined range. Each set always starts from a steady-state condition before being excited. Approximately **30%** for validation.



2. **Set 2 (Test: Same Range, Multiple Steps):**

Contains different step-change patterns within the same parameter bounds as Set 1.

3. **Set 3 (Test: Extended Range):**

Applies wider bounds to test the model's ability to handle partial extrapolation.

4. **Sets 4 and 5 (Single-Step Tests):**

Each set starts from steady state, then a single, sharp step is introduced. Set 4 remains within the original range, while Set 5 extends one or more parameters beyond the training limits.

For the Hill and AM2 Models (2 Sets):

1. **Set 1 (Training and Validation):**

Includes nominal operating conditions with step or random changes, always beginning from steady state.

2. **Set 2 (Test):**

Contains different step-change patterns or disturbances, designed to evaluate model generalization.

3.2.3 Practical Details

- **Software Tools:** MATLAB was used for all simulations and analyses. Data was generated using the `ode45` solver.
- **Data Preprocessing:** Normalization was performed on the training data to avoid any data leakage from the validation set.
- **Neural Network Implementation:** The Deep Learning Toolbox and its function `trainnet` were used to handle data preprocessing, define architectures, and perform training routines.
- **Hardware:** The experiments were conducted on a CPU AMD Ryzen 5 5600H (6 cores) and an RTX 3060 Laptop GPU.
- **Sensitivity Analysis:** The Parallel Computing Toolbox was used with 6 workers to perform the sensitivity analysis.



3.3 Overview of Experimental Configuration

To facilitate a comprehensive evaluation, we implemented different experimental configurations for the simple system and for the advanced anaerobic digestion models.

3.3.1 For the Simple System:

3.3.1.1 Experiment A: Single-Step Excitation

The training dataset is generated using a single abrupt step change from steady state. This configuration tests the network's ability to capture immediate, one-step-ahead dynamics but typically provides limited variety.

3.3.1.2 Experiment B: Multiple-Step Excitation

The training dataset includes 15 consecutive step excitations. This configuration aims to capture a broader range of dynamic responses.

3.3.1.3 Experiment C: Sensitivity Analysis

Key parameters (number of steps, step duration factor, sampling rate factor, number of delays/window size, hidden neurons/units, and number of layers) are systematically varied. For each parameter value, 10 runs were performed and performance metrics (such as MAE) are reported as the median along with the 25th and 75th percentiles. Graphs of MAE versus each parameter were used to identify an optimal configuration that balances prediction accuracy and network complexity.

3.3.1.4 Experiment D: Minimal Configuration

Based on the sensitivity analysis, a minimal configuration is selected.

3.3.2 For the Advanced Anaerobic Digestion Models (Hill and AM2):

3.3.2.1 Experiment A:

A configuration using 2 excitation steps is evaluated to determine whether minimal training data is sufficient to capture the essential dynamics of the respective model.

3.3.2.2 Experiment B:

A configuration using 6 excitation steps is tested. This richer training dataset is expected to yield more stable training and improved prediction accuracy—particularly in regions with strong nonlinearity—compared to the 2-step configuration.

For further details, including specific dilution rate ranges, noise levels, and model base configurations, please refer to Appendix D (Table 8 and Table 9).



3.4 Sensitivity Analysis (For the Symple System)

The performance of the neural networks for the simple system was assessed by systematically varying the key parameters (as defined in Section 3.1). For each configuration, 10 runs were performed, and performance metrics such as the Mean Absolute Error (MAE) were recorded. Detailed parameter ranges are provided in Appendix D (Table 7).

3.4.1 Strategy for Sweeping Each Parameter

A baseline configuration was first identified (see Table 6 in appendix D). Keeping these values fixed, each parameter was varied individually over the following ranges:

- **Number of Steps:** 1 to 20
- **Duration Factor:** 1 to 15
- **Sample Time Factor:** 2 to 30
- **Window Size / Delay:** 1 to 15
- **Hidden Neurons/Units:** 1 to 20
- **Number of Layers:** 1 to 4

Graphs plotting MAE versus each parameter guided the selection of a “minimal” configuration that provides an optimal balance between prediction accuracy and network complexity.

3.5 Training Modes

3.5.1 Training Modes

Networks were primarily trained in a one-step-ahead fashion (i.e., predicting the immediate next time step) using true past outputs as inputs. During testing, multi-step-ahead predictions were obtained by feeding the network’s own outputs back as inputs in a closed-loop manner.

3.5.2 Training Configuration

3.5.2.1 Optimization algorithms and Settings

Multiple training algorithms were tested, including Adam, L-BFGS, and—for NARX architectures—the Levenberg-Marquardt method. Manual tuning indicated that L-BFGS yielded better performance than Adam, although no exhaustive tuning was conducted. An example of the training options used is as follows:

```
options = trainingOptions('lbfgs', ...
    "ExecutionEnvironment", "gpu", ...
    "ValidationData", {XValD1, YVal}, ...
    "ValidationFrequency", 2, ...
    "ValidationPatience", 20, ...
    "InputDataFormats", "CTB", ...
    "Plots", "none", ...
    'OutputNetwork', "best-validation", ...
    "Verbose", false);
```

3.5.3 Performance Metrics and Loss Functions

In the experiments, several performance metrics were employed both as loss functions during training and as benchmarks for model selection. These include the Mean Squared Error (MSE), Mean Absolute Error (MAE), Akaike's Information Criterion (AIC), and its small sample-size corrected version (AICc).

3.5.3.1 Mean Squared Error (MSE):

MSE is used as the loss function during training. It measures the average squared difference between the predicted values and the actual observations. Formally, if $\varepsilon(t)$ denotes the prediction error at time t and there are N observations, then:

$$MSE = \frac{1}{N} \sum_{t=1}^N \varepsilon(t)^2$$

Minimizing MSE is standard in regression problems; however, it is sensitive to outliers due to the squaring of errors.

3.5.3.2 Mean Absolute Error (MAE):

MAE is used in this work as a benchmark metric to evaluate model performance because it is less sensitive to extreme values compared to MSE. It is defined as:

$$MAE = \frac{1}{N} \sum_{t=1}^N |\varepsilon(t)|$$

MAE often provides a more representative measure of typical error magnitude, facilitating clearer graphical comparisons between experiments.

3.5.3.3 Akaike's Information Criterion (AIC):

AIC (Ljung, 1999) provides a measure of model quality by simulating the model's predictive ability on different data sets, balancing goodness-of-fit against model complexity. Under the assumption of Gaussian errors, the raw AIC is defined as:

$$AIC = N \cdot \log \left(\det \left(\frac{1}{N} \sum_{t=1}^N \varepsilon(t) \varepsilon(t)^T \right) \right) + 2n_p + N \cdot (n_y \cdot \log(2\pi) + 1)$$



where:

- N is the number of observations,
- $\varepsilon(t)$ is the prediction error vector at time t ,
- n_p is the number of estimated parameters, and
- n_y is the number of model outputs.

According to Akaike's theory, the model with the smallest AIC value is considered the most accurate.

3.5.3.4 *Small Sample-Size Corrected AIC (AICc):*

AICc (Ljung, 1999) adjusts AIC to account for bias when the sample size is small. It is computed as:

$$AICc = AIC + \frac{2n_p(n_p + 1)}{N - n_p - 1}$$

This correction is particularly important when the number of parameters is not negligible compared to the number of observations.

3.5.4 *Validation and Model Selection*

The validation portion of Set 1 (for the simple system or for the Hill/AM2 models) was used to monitor training progress and select the best-performing model based on validation error. Specifically, the network state yielding the lowest validation loss during training was retained (*OutputNetwork* = "best-validation"). Signs of overfitting—such as continued improvement in training loss without corresponding improvements in validation loss, or a rising validation error—were used as criteria to stop training early. In addition, model generalization was assessed using fully separate testing sets, typically Set 2 in the case of the AD models.



4. Analysis of the Simple System

This chapter presents an evaluation of three neural network architectures—NARX, LSTM-A, and LSTM-B—using the simple second-order reaction model described in Section 2.3.1. The main objectives are to compare one-step-ahead and multi-step-ahead prediction performance, investigate the influence of key training parameters, and ultimately select a minimal configuration that balances accuracy with model simplicity.

4.1 Experimental Data and Dataset Organization

- **Set 1 (Training and Validation):**

Used exclusively to train and validate each network. Although the specific step sequences and parameter variations may differ among the architectures, the overall concept remains the same.

- **Sets 2 to 5 (Testing):**

- **Set 2:** Multiple-step responses within the training range.
- **Set 3:** Multiple-step responses extending beyond the training range (for extrapolation testing).
- **Set 4:** A single-step test within the training range.
- **Set 5:** A single-step test beyond the training range.

All networks were trained in a one-step-ahead manner on Set 1, with each training sample predicting the immediate next time point. Multi-step predictions were then generated by recursively feeding the network's own outputs back as inputs. A sensitivity analysis was carried out by varying six key parameters—(i) number of steps (multi-step horizon), (ii) sampling rate, (iii) duration of each step, (iv) number of neurons, (v) number of layers, and (vi) delays—to assess their influence on prediction accuracy.

Fig. 13 shows an example of the training dataset (Set 1), highlighting the step changes in the dilution rate and the corresponding response in concentration $C(t)$. The testing datasets for Sets 2–5, which include multiple-step and single-step excitations, are depicted in Fig. 14.

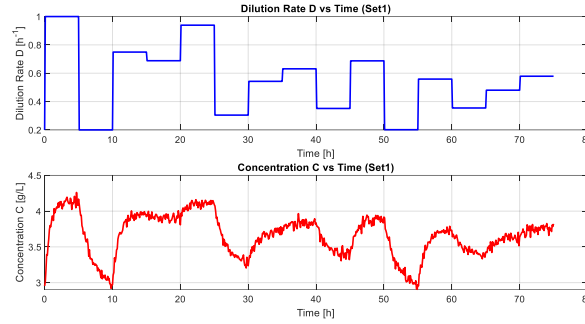


Fig. 13: Example of the training dataset (Set 1), showing step changes in the dilution rate (top) and the resulting concentration $C(t)$ (bottom).

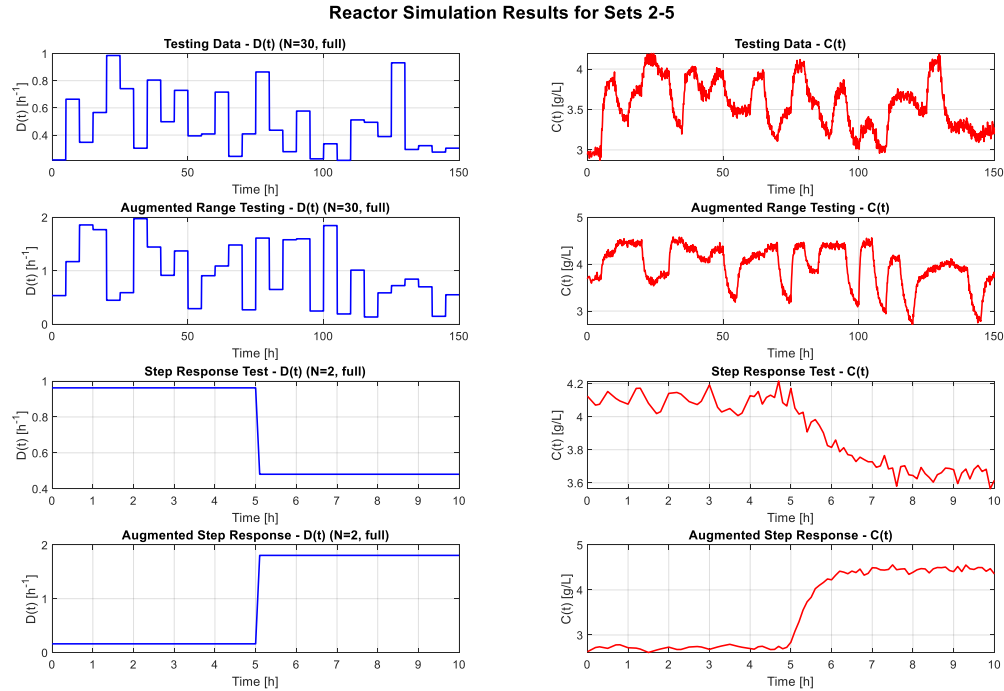


Fig. 14: Testing datasets for Sets 2–5, illustrating multiple-step excitations (Sets 2 and 3) and single-step excitations (Sets 4 and 5). The left panels display the dilution rate $D(t)$, and the right panels show the corresponding concentration $C(t)$ over time.

The following sections detail the experimental procedures and findings for each architecture—NARX, LSTM-A, and LSTM-B—covering single-step vs. multiple-step excitations, sensitivity analyses, and minimal configuration selections.



4.2 NARX Architecture Analysis

4.2.1 NARX Analysis Overview

An evaluation of the NARX architecture on the simple second-order reaction model is presented. All networks are trained in a one-step-ahead manner using Set 1, where each training sample predicts the immediate next time point. For testing, both one-step responses (from a single step excitation) and multi-step responses (obtained by recursively feeding back the model's predicted outputs) are analyzed.

4.2.2 Experiments with NARX

4.2.2.1 Experiment A: Single-Step Excitation

Objective:

Evaluate whether training the NARX model on a single abrupt step change from steady state is sufficient to learn both immediate (one-step-ahead) and extended (multi-step-ahead) dynamics.

Method:

The training set includes the response to one step change, with the tail end of that transient reserved for validation. Training is performed in a one-step-ahead manner, and the model state corresponding to the lowest validation error is selected rather than the final training iteration.

Observations:

- **Training and Validation Simplicity:**

Both the training and the 30% validation segments are derived from a single excitation, making it easy to learn immediate dynamics; however, the validation data offer limited diversity.

- **Benefit of the Validation Set:**

Despite its small size, the validation set helps to avoid overfitting, resulting in better performance than simply using the final training iteration.

- **One-Step vs. Multi-Step Performance:**

One-step-ahead predictions are generally more accurate; however, errors quickly accumulate in multi-step forecasts, leading to divergence (see Fig. 15 and Fig. 16).

- **Limited Utility for Longer Horizons:**

Exposure to only one step change means the network lacks the dynamic variety needed for stable multi-step predictions, occasionally impacting even one-step performance.

Conclusions:

A single abrupt step does not provide sufficient dynamic variety for stable multi-step forecasting. While one-step-ahead predictions may sometimes be acceptable, the overall performance remains inconsistent. Incorporating additional step excitations is necessary to achieve reliable short- and long-term predictions. *Further details on the other test sets and training outcomes are provided in Appendix E.*

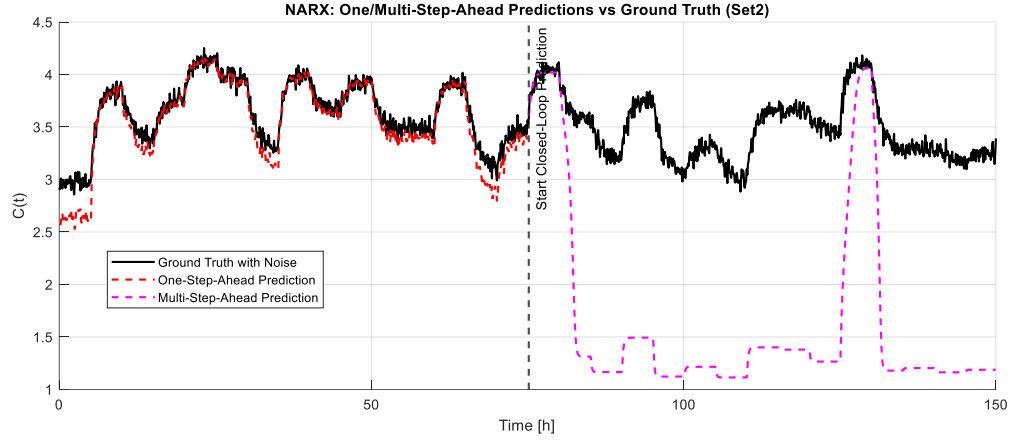


Fig. 15: Predictions for Set 2 with single-step training (Experiment A). Black: noisy ground truth; red: one-step; magenta: multi-step. Multi-step forecasts diverge over time.

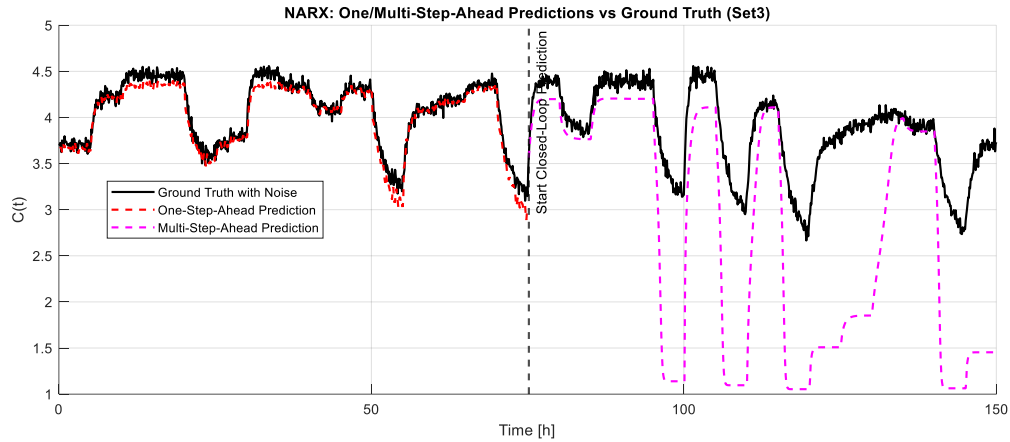


Fig. 16: NARX predictions for Set 3 with single-step training (Experiment A). Black: noisy ground truth; red: one-step; magenta: multi-step. Limited training causes large multi-step errors, highlighting need for more excitation.

4.2.2.2 Experiment B: Multiple-Step Excitation (15-Step Configuration)

Assess how the NARX model's predictive accuracy changes when the training dataset is expanded from a single abrupt step change to **15 distinct step changes**, while keeping other training parameters and test sets the same as in the 1-step-excitation experiment.



Objective:

Assess how expanding the training dataset from a single step to 15 distinct step excitations affects the NARX model's prediction accuracy.

Method:

The training dataset now consists of 15 successive step excitations, while all other training parameters and test sets remain unchanged.

Observations:

- **Improved Forecasting Accuracy:**

The expanded dataset significantly improves both one-step-ahead and multi-step-ahead predictions by better capturing the range of dynamic responses and reducing drift over longer horizons.

- **Robustness Across Test Sets:**

The network handles partial extrapolation (Set 3) and large single steps (Sets 4 and 5) more reliably, indicating that exposure to diverse excitations enhances generalization.

- **Sustained Multi-Step Tracking:**

With multiple training excitations, the network maintains accurate forecasts for extended prediction windows.

- **Toward a Minimal Configuration:**

Although 15 steps yield clear improvements, further analysis is needed to determine if fewer excitations can achieve similar performance.

Conclusions:

Expanding the training dataset to 15 step excitations results in a richer representation of system dynamics and improves both one-step-ahead and multi-step-ahead prediction accuracy (see Fig. 17 and Fig. 18). The next step is to identify the minimal number of excitations required for robust performance. *Further details on the other test sets and training outcomes are provided in Appendix E.*

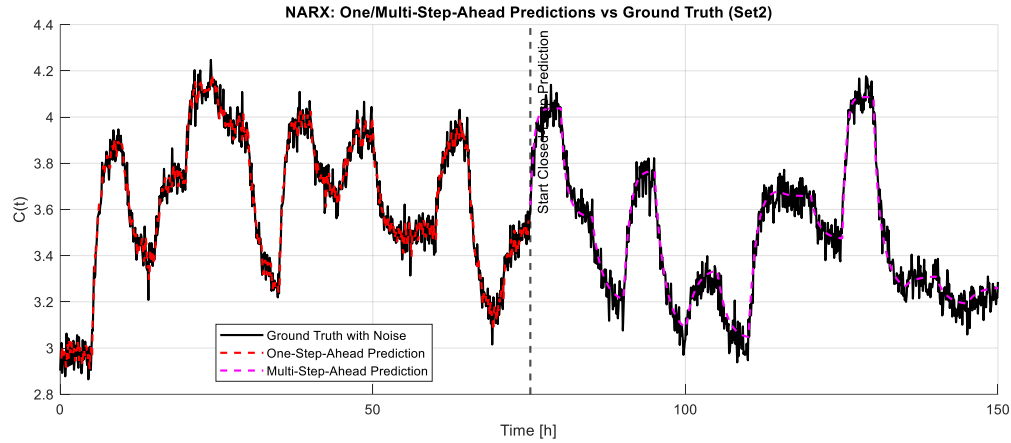


Fig. 17: NARX predictions for Set 2 with multi-step training (Experiment B). Black: noisy ground truth; red: one-step; magenta: multi-step. Richer training improves multi-step forecasts vs. single-step.

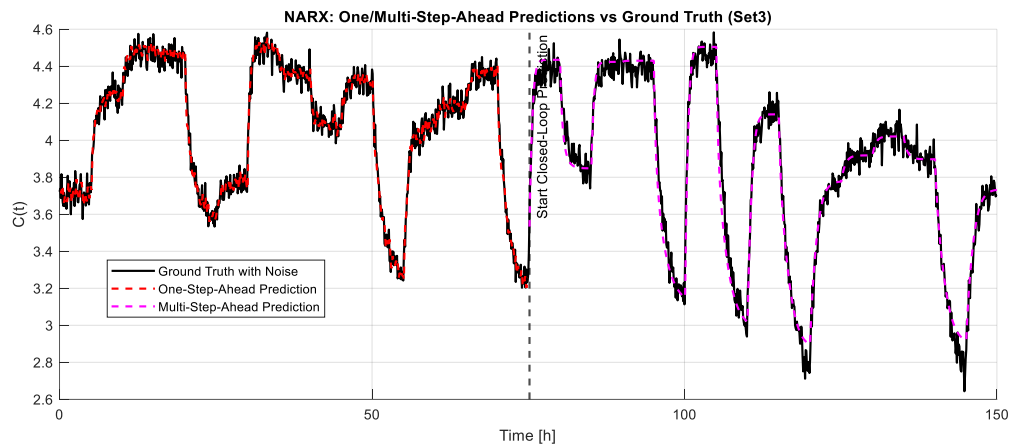


Fig. 18: NARX predictions for Set 3 with multi-step training (Experiment B). Black: noisy ground truth; red: one-step; magenta: multi-step. Multiple excitations reduce long-horizon drift, though some errors persist at extremes.

4.2.2.3 Experiment C: Sensitivity Analysis

This experiment evaluates how varying six key parameters impacts NARX predictions:

- **Number of Steps:**

Errors and variance are high with a low number of training steps and stabilize at approximately 5–6 steps for both one-step-ahead and multi-step-ahead predictions.

- **Step Duration:**

Shorter step durations limit the observed transient, especially at low dilution rates, leading to degraded performance; longer durations improve predictions, with a plateau observed around 8 for one-step-ahead and around 6 for multi-step-ahead.



- **Sampling Rate:**

A moderately slow sampling rate is beneficial, as faster sampling reduces the initial information window and increases errors in multi-step forecasting.

- **Delays:**

Increasing delays improves multi-step accuracy by providing more historical context, but excessive delays may cause overfitting in one-step predictions.

- **Number of Hidden Units:**

Beyond a modest baseline (around 4–6 neurons), further increases yield diminishing returns and may even cause overfitting.

- **Number of Layers:**

Additional layers do not improve performance and can introduce instability, indicating that a single hidden layer is sufficient.

Conclusions:

The sensitivity analysis highlights that training with 5–6 excitations, appropriate step durations, and moderate sampling rates, delays, and network complexity is key to achieving balanced, low-error forecasts. *Further details on the effect of different parameters (sensitivity analysis) are presented in Appendix E.*

4.2.2.4 Experiment D: Selection of the Minimal Configuration

Chosen Configuration:

- **Training Dataset (Set 1):**

- 6 step excitations
- Step duration factor: 6
- Sampling-time factor: 10

- **NARX Base Parameters:**

- Delays: 6
- Hidden neurons: 5
- Layers: 1

Observations:

- **Stable Multi-Step Tracking:** The network maintains accurate predictions over extended horizons.

- **Effective Generalization:**

- In Set 2 (training range), the network reproduces known dynamics accurately (see Fig. 19).
- In Set 3 (extrapolation), the model continues to predict coherently (see Fig. 20).
- In Sets 4 and 5 (abrupt single steps), the network transitions smoothly between steady states.

Conclusions:

This minimal configuration effectively balances simplicity and performance, providing reliable forecasts across routine and extrapolated conditions. *Further details on the other test sets and training outcomes are provided in Appendix E.*

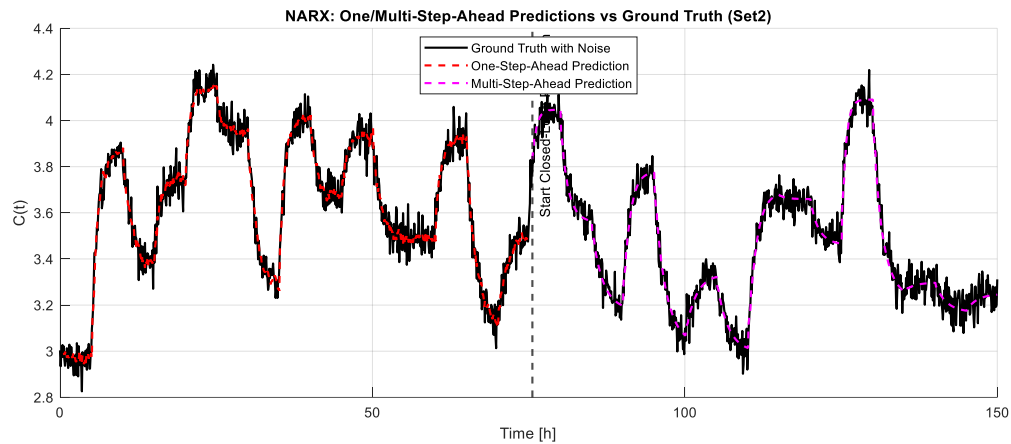


Fig. 19: NARX predictions for Set 2 using minimal config (Experiment D). Black: noisy ground truth; red: one-step; magenta: multi-step. Setup preserves stability and accuracy in multi-step forecasts.

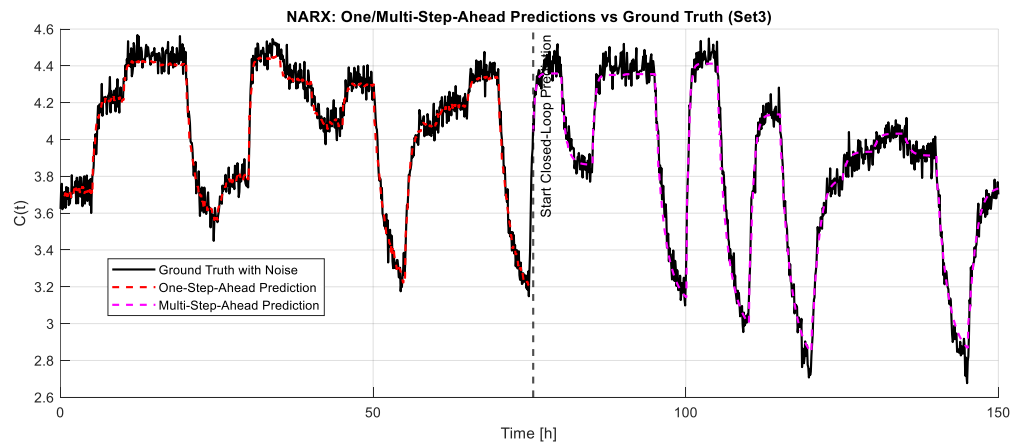


Fig. 20: NARX predictions for Set 3 using minimal config (Experiment D). Black: noisy ground truth; red: one-step; magenta: multi-step. Optimized setup ensures stable, accurate performance under challenging conditions.



4.2.3 *Summary of NARX Analysis*

The experiments with the NARX architecture reveal that:

- **Single-Step Excitation** somewhat captures immediate dynamics but is insufficient for stable multi-step forecasting.
- **Multiple-Step Excitation** (15 steps) significantly enhances both one-step-ahead and multi-step-ahead performance by exposing the network to a broader range of dynamic behaviors.
- **Sensitivity Analysis** confirms that a training set with approximately 5–6 step excitations, along with moderate step durations, sampling rates, and network parameters (delays, hidden units, and layers), is effective under the tested conditions.
- **Minimal Configuration** (6 steps, step duration factor 6, sampling-time factor 10, 6 delays, 5 hidden neurons, 1 layer) yields consistent forecasts with low error across all test sets.

In summary, the NARX architecture benefits substantially from diverse training data and careful tuning of parameters to achieve reliable one-step-ahead and multi-step-ahead predictions.

4.3 *LSTM-A Architecture Analysis*

4.3.1 *LSTM-A Analysis Overview*

An evaluation of the LSTM-A architecture on the simple second-order reaction model is presented. All networks are trained in a one-step-ahead manner using Set 1, where each training sample predicts the immediate next time point. For testing, both one-step responses (from a single step excitation) and multi-step responses (obtained by recursively feeding back the model's predicted outputs) are analyzed. This approach is consistent with the methodology used for NARX, allowing a direct comparison of their performance.

4.3.2 *Experiments with LSTM-A*

4.3.2.1 *Experiment A: Single-Step Excitation*

Objective:

Determine if a single abrupt step is sufficient for LSTM-A to learn both one-step-ahead and multi-step-ahead dynamics.

Method:

Training is performed in a one-step-ahead manner on a single step change, with a portion of that transient reserved for validation. Multi-step forecasts are generated by recursively feeding the output back as input.

Observations:

- Immediate one-step-ahead predictions are moderately accurate.
- Multi-step-ahead predictions quickly diverge due to the limited training data, resulting in unstable forecasts.
- This behavior mirrors that of the NARX model when trained on only one excitation.

Conclusions:

A single excitation does not provide sufficient dynamic variety for reliable multi-step forecasting in LSTM-A (see Fig. 21 and Fig. 22). Multiple excitations might be required for generalization *Further details on the other test sets and training outcomes are provided in Appendix F.*

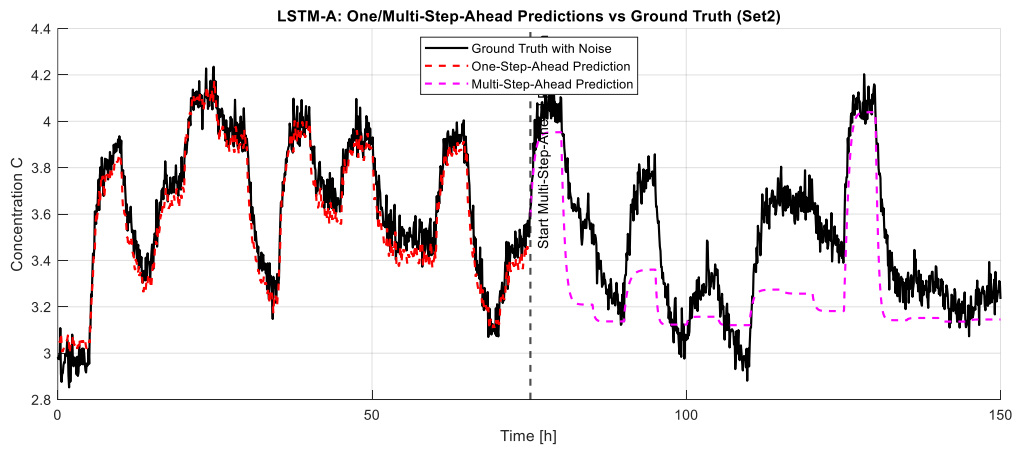


Fig. 21: LSTM-A predictions for Set 2 with single-step training (Experiment A). Black: noisy ground truth; red: one-step; magenta: multi-step. Single-step training limits diversity, causing drift in multi-step forecasts.

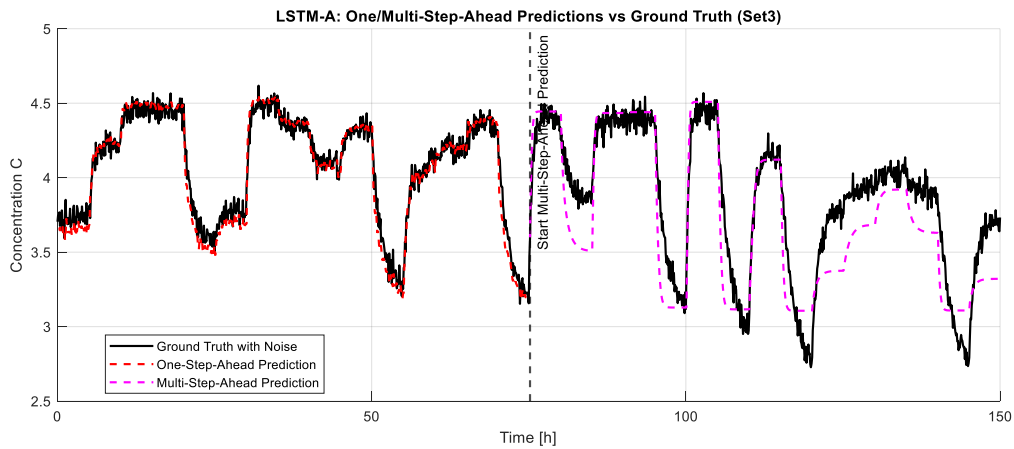


Fig. 22: LSTM-A predictions for Set 3 with single-step training (Experiment A). Black: noisy ground truth; red: one-step; magenta: multi-step. Limited dynamics from single excitation cause multi-step divergence.

4.3.2.2 Experiment B: Multiple-Step Excitation

Objective:

Assess improvements in LSTM-A predictions when trained with 15 distinct step excitations.

Observations:

- Both one-step-ahead and multi-step-ahead forecasts improve markedly compared to the single-step scenario.
- The model demonstrates stable predictions across all test sets, including extrapolation (Set 3) and abrupt transitions (Sets 4 and 5).
- Performance is comparable to that observed in NARX, with no clear advantage for either architecture.

Conclusions:

Training with multiple excitations enhances the model's ability to generalize and produce reliable multi-step forecasts (see Fig. 23 and Fig. 24), supporting the need for diverse training data. *Further details on the other test sets and training outcomes are provided in Appendix F.*

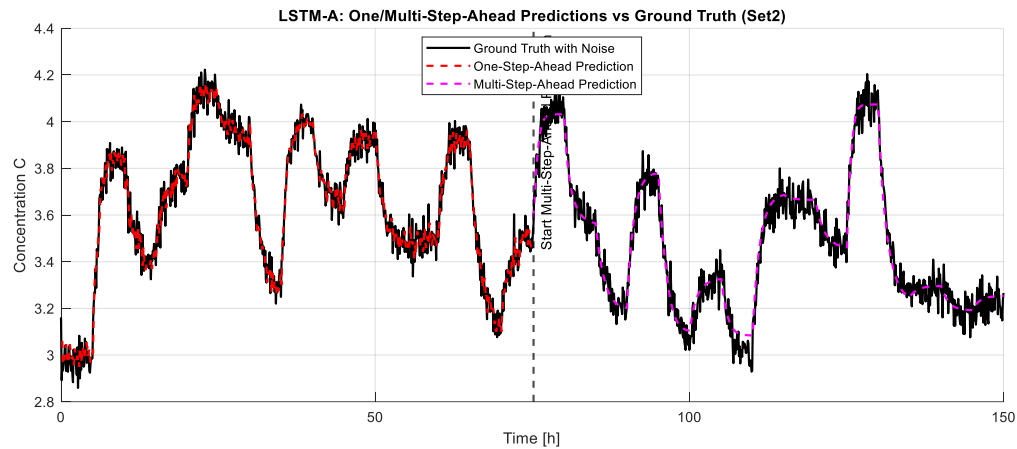


Fig. 23: LSTM-A predictions for Set 2 with multi-step training (Experiment B). Black: noisy ground truth; red: one-step; magenta: multi-step. Multiple excitations improve long-term dynamic capture vs. single-step.

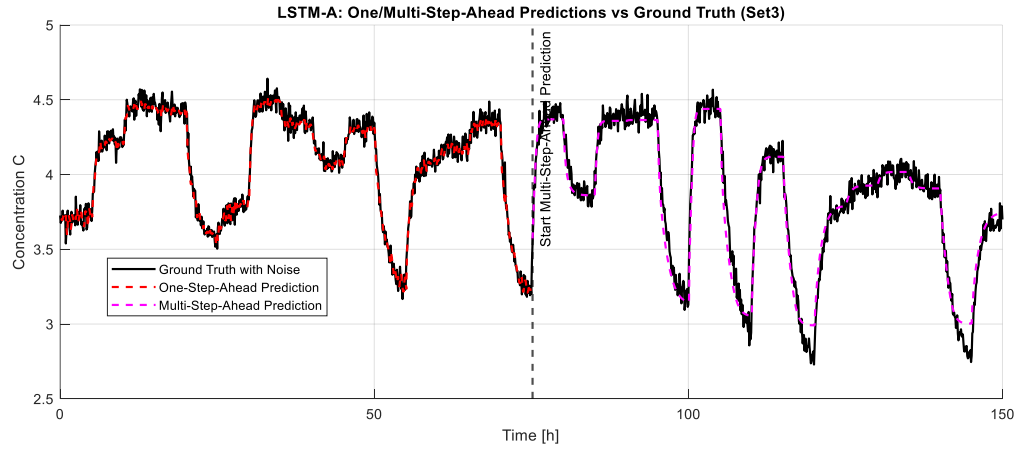


Fig. 24: LSTM-A predictions for Set 3 with multi-step training (Experiment B). Black: noisy ground truth; red: one-step; magenta: multi-step. Richer training reduces long-horizon drift vs. single-step setup.

4.3.2.3 Experiment C: Sensitivity Analysis

Key observations from the sensitivity analysis include:

- **Number of Steps:** Errors stabilize at around 5–6 excitations.
- **Step Duration:** One-step-ahead accuracy improves with longer durations (plateauing near 8), while multi-step accuracy peaks at around 6.
- **Sampling Time Factor:** Faster sampling reduces the available information window, increasing errors in multi-step mode, particularly for extrapolation.
- **Delays, Hidden Units, and Layers:**
 - Increasing delays improves multi-step forecasts up to a point (optimal around 6 delays).
 - Hidden units yield erratic effects, and extra layers tend to degrade performance.

Conclusions:

Optimal performance is achieved with a moderate number of excitations, appropriate step durations, and careful tuning of delays and network complexity. *Further details on the effect of different parameters (sensitivity analysis) are presented in Appendix F.*

4.3.2.4 Experiment D: Minimal Configuration Selection

Chosen Configuration:

- Training dataset: 6 step excitations, step duration factor 7, sampling-time factor 10
- LSTM-A Base Parameters: 6 delays, 3 hidden neurons, 1 layer

Observations:

- The minimal configuration yields stable one-step-ahead and multi-step-ahead forecasts (see Fig. 25 and Fig. 26) across all test sets, including effective extrapolation (Set 3) and smooth handling of abrupt transitions (Sets 4 and 5).

Conclusions:

This configuration strikes a balance between simplicity and predictive accuracy, ensuring robust performance under both routine and extrapolated conditions. *Further details on the other test sets and training outcomes are provided in Appendix F.*

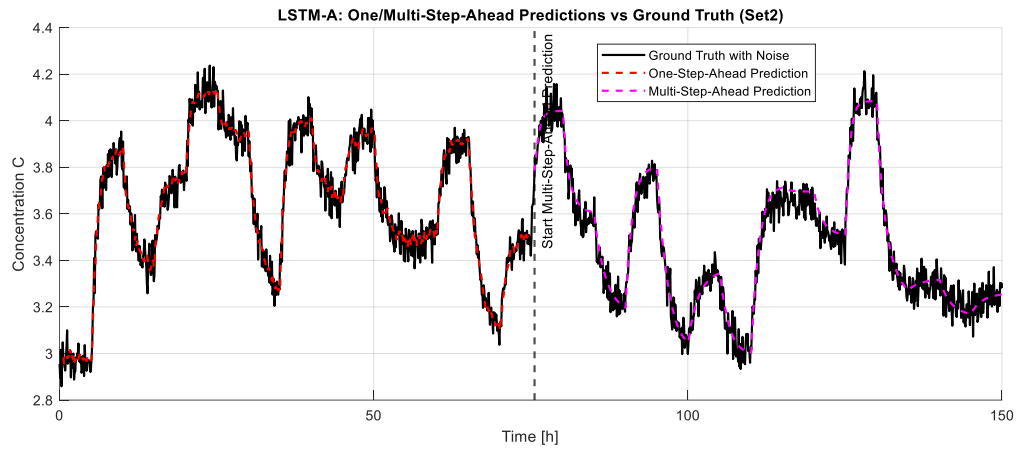


Fig. 25: LSTM-A predictions for Set 2 using minimal config (Experiment D). Black: noisy ground truth; red: one-step; magenta: multi-step. Tuning preserves multi-step stability despite less data and simpler architecture.

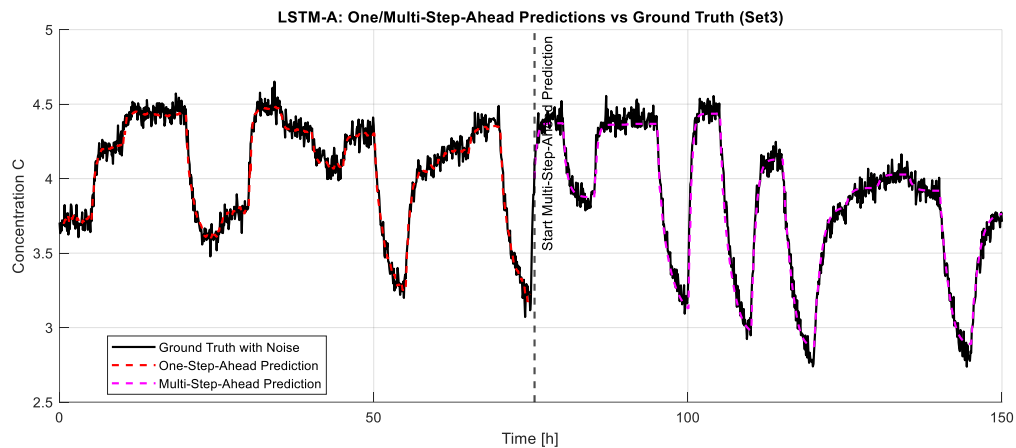


Fig. 26: LSTM-A predictions for Set 2 using minimal config (Experiment D). Black: noisy ground truth; red: one-step; magenta: multi-step. Parameter tuning ensures stable multi-step forecasts despite limited data and simpler model.



4.3.3 Summary of LSTM-A Analysis

The experiments with LSTM-A indicate that:

- Training on a single excitation is insufficient, leading to unreliable multi-step forecasts.
- Multiple excitations (15 steps) markedly improve performance across one-step-ahead and multi-step-ahead predictions.
- Sensitivity analysis confirms that approximately 5–6 excitations, moderate step durations, an appropriate sampling rate, and careful tuning of delays and network complexity are key.
- The minimal configuration (5 step excitations, 6 delays, 3 hidden neurons in a single layer) provides robust and reliable forecasts, comparable to NARX.

Overall, LSTM-A demonstrates strong potential when provided with diverse training data and optimized parameters.

4.4 Analysis on the Simple System: LSTM-B Architecture

4.4.1 LSTM-B Analysis Overview

The LSTM-B model processes a continuous window of past data, and its evaluation follows the same overall procedure as the other architectures. The network is trained in a one-step-ahead manner using Set 1, and its performance is assessed using both one-step and multi-step forecasting on Sets 2–5.

4.4.2 Experiments with LSTM-B

4.4.2.1 Experiment A: Single-Step Excitation (One-Step Configuration)

Objective:

Assess whether training with a single abrupt step from steady state is sufficient for LSTM-B to capture immediate (one-step-ahead) and extended (multi-step-ahead) predictions.

Observations:

- **One-Step Performance:**

The model does not accurately predict the immediate response following the excitation.

- **Multi-Step Divergence:**

When predictions are generated recursively, errors accumulate rapidly, leading to significant divergence.

- **General Reliability:**

Although the model fits the training data well, it fails to generalize beyond the single excitation.

Conclusions:

A single step excitation is inadequate for stable multi-step forecasting in LSTM-B (see Fig. 27 and Fig. 28), indicating the need for additional training excitations. *Further details on the other test sets and training outcomes are provided in Appendix G.*

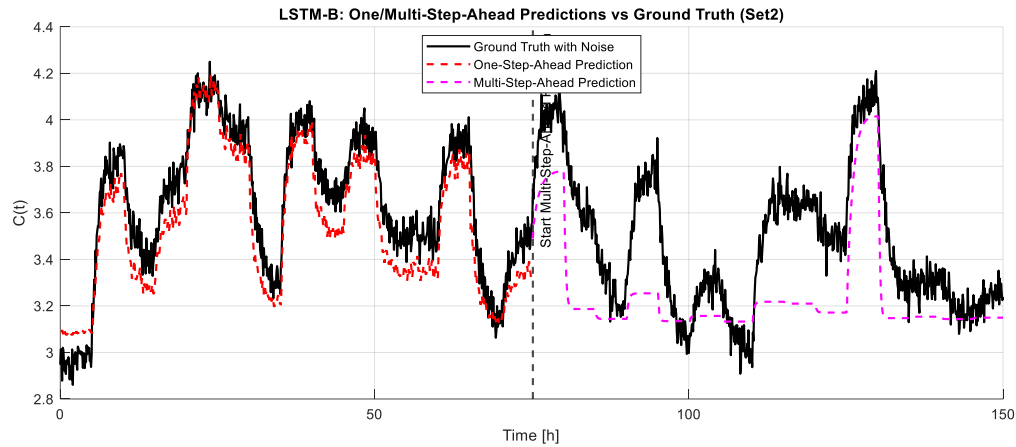


Fig. 27: LSTM-B predictions for Set 2 with single-step training (Experiment A). Black: noisy ground truth; red: one-step; magenta: multi-step. Single excitation limits dynamics, causing multi-step forecasts to drift over time.

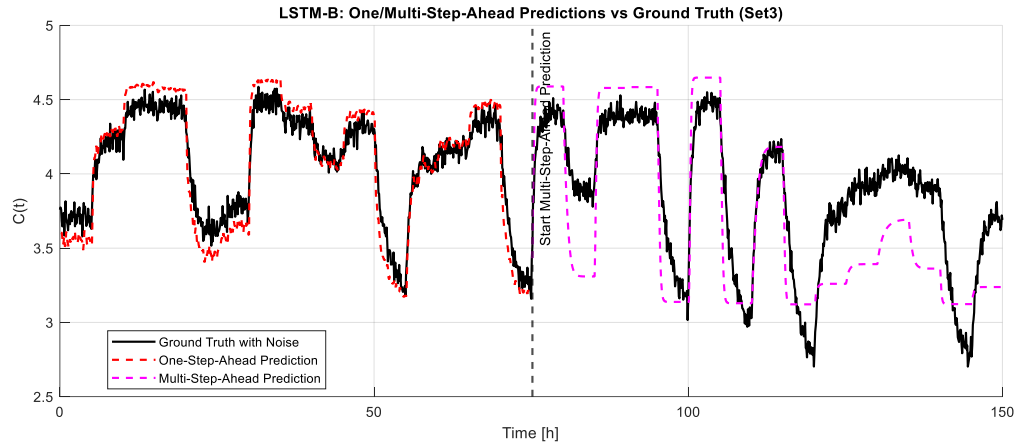


Fig. 28: LSTM-B predictions for Set 3 with single-step training (Experiment A). Black: noisy ground truth; red: one-step; magenta: multi-step. Limited excitation leads to poor long-range dynamics and divergence in multi-step forecasts.

4.4.2.2 Experiment B: Multiple-Step Excitation (15-Step Configuration)

Observations:

- The model demonstrates substantially improved performance in both one-step-ahead and multi-step-ahead predictions when trained with 15 excitations.
- Performance across all test sets is strong; although there is a slight decrease in extrapolation accuracy (Set 3), overall reliability is maintained.
- This indicates that training with multiple excitations is essential for capturing the full temporal dynamics.

Conclusions:

Multiple excitations enable LSTM-B to generate stable forecasts (see Fig. 29 and Fig. 30), supporting its potential when provided with diverse training data. *Further details on the other test sets and training outcomes are provided in Appendix G.*

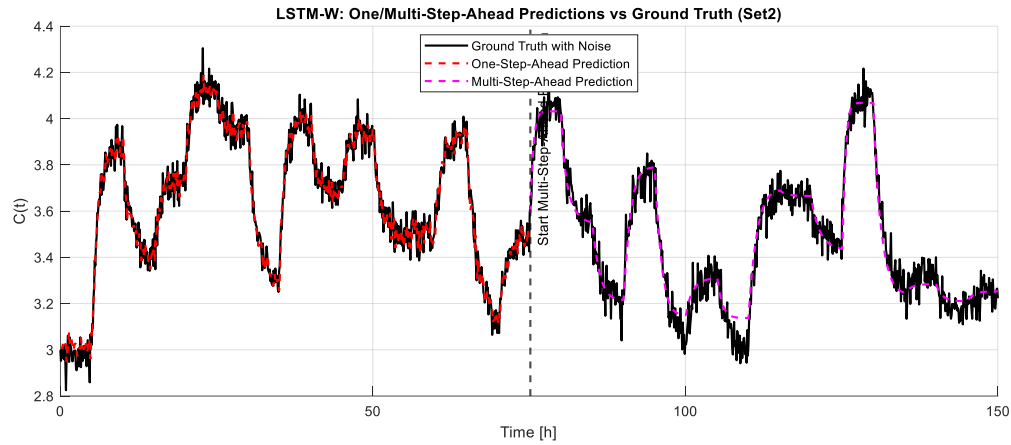


Fig. 29: LSTM-B predictions for Set 2 with multi-step training (Experiment B). Black: noisy ground truth; red: one-step; magenta: multi-step. Added excitations reduce drift vs. single-step training.

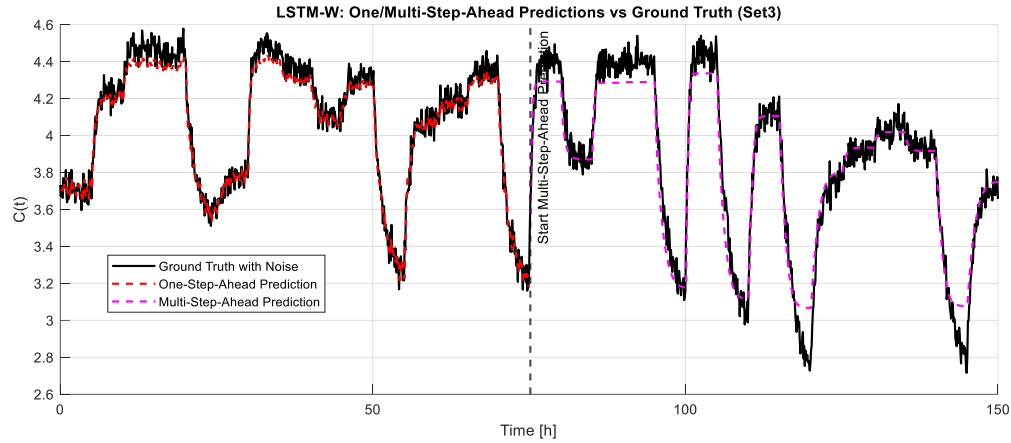


Fig. 30: LSTM-B predictions for Set 3 with multi-step training (Experiment B). Black: noisy ground truth; red: one-step; magenta: multi-step. Multiple excitations improve transition tracking, though some errors persist at extremes.

4.4.2.3 Experiment C: Sensitivity Analysis

Key observations :

- **Number of Steps:**

Errors and their dispersion are high with few excitations and stabilize at around 5 steps.

- **Step Duration Factor:**

Values between 6 and 8 yield stable errors; beyond this range, performance remains largely unchanged.

- **Sampling Time Factor:**

Faster sampling significantly increases errors in multi-step forecasts, particularly affecting extrapolation performance.

- **Hidden Units and Layers:**

Increasing the number of hidden units shows inconsistent effects, and additional layers generally do not improve performance.

Conclusions:

A moderate number of excitations, well-chosen step durations, and meticulously tuned delays and network complexity might lead to near-optimal performance. *Further details on the effect of different parameters (sensitivity analysis) are presented in Appendix G.*

4.4.2.4 Experiment D: Selection of the Minimal Configuration

Chosen Configuration:

- **Training Dataset (Set 1):**
 - 6 step excitations
 - Step duration factor: 6
 - Sampling-time factor: 10
- **LSTM-B Specific Parameters:**
 - Window size: 8
 - Hidden neurons: 3
 - Layers: 1

Observations:

- The minimal configuration performs well across all test sets: it accurately captures dynamics within the training range (see Fig. 31), effectively extrapolates (see Fig. 32), and handles abrupt step changes (Sets 4 and 5).

Conclusions:

This minimal configuration for LSTM-B provides a strong balance between simplicity and predictive performance, ensuring stable one-step-ahead and multi-step-ahead forecasts. *Further details on the other test sets and training outcomes are provided in Appendix G.*

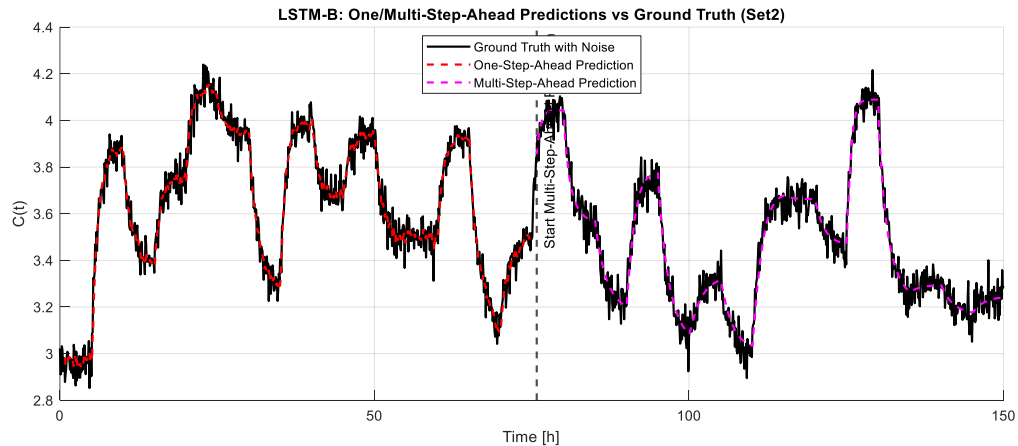


Fig. 31: LSTM-B predictions for Set 2 using minimal config (Experiment D). Black: noisy ground truth; red: one-step; magenta: multi-step. Optimized setup maintains stable and accurate multi-step forecasts.

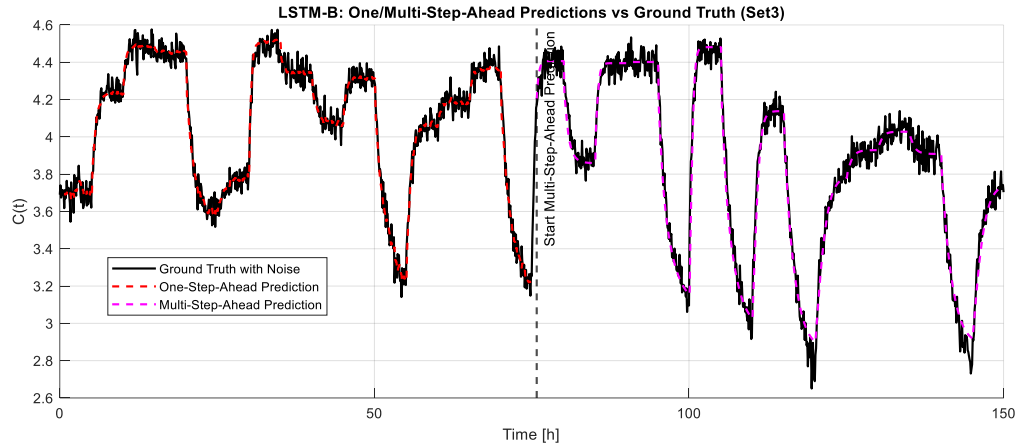


Fig. 32: LSTM-B predictions for Set 3 using minimal config (Experiment D). Black: noisy ground truth; red: one-step; magenta: multi-step. Model remains robust even under challenging conditions.

4.4.3 Summary of LSTM-B Analysis

The evaluation of LSTM-B reveals that:

- Training with a single excitation results in inadequate one-step-ahead predictions and poor multi-step-ahead performance.
- Incorporating 15 excitations substantially improves forecasting accuracy.
- Sensitivity analysis confirms that around 5-6 excitations, moderate step durations, an appropriate sampling rate, and a carefully chosen window size are critical for robust performance.
- A minimal configuration (5 step excitations, step duration factor 6, sampling-time factor 10, window size 8, 3 hidden neurons, and 8 delays) strikes an effective balance, delivering stable predictions across all test sets.

Overall, while LSTM-B is capable of capturing short-range dynamics, its ability to generalize over multiple steps improves significantly with richer training data and optimized parameters.

4.5 Overall Comparative Analysis

Sections 4.2, 4.3, and 4.4 evaluated NARX, LSTM-A, and LSTM-B independently, focusing on single-step vs. multi-step excitations, sensitivity analyses, and minimal configurations. This section consolidates those results to underscore cross-architecture trends and highlight any noteworthy differences.



4.5.1 Training Data Diversity: Single vs. Multiple Excitations

- **Single Excitation**

Although occasional attempts produced acceptable one-step-ahead predictions, it was uncommon. Multi-step forecasts were typically poor—not always diverging outright but often drifting toward incorrect steady values or “locked” patterns that did not match the real dynamics. In practice, one might observe a short initial window of decent performance, followed by a clear mismatch.

- **Multiple Excitations**

Introducing 5 to 15 step changes during training improved each model’s ability to capture the system’s dynamics. Indeed, with multiple excitations, the networks effectively learned the relationship between the dilution rate D and the future concentration C without needing continuous feedback of C . Predictions displayed only minor deviations, and no persistent drifts were observed, indicating a robust understanding of the process.

4.5.2 Extrapolation (Inside vs. Outside the Training Range)

- **Slight but Noticeable Increase in Error**

When encountering step amplitudes outside the training range (e.g., Sets 3 and 5), all three architectures saw a small but consistent rise in error. Nonetheless, networks trained with multiple, diverse excitations still maintained relatively stable multi-step forecasts.

- **Remarkably Capable of Unseen Conditions**

Despite not having explicit data for certain large or small steps, the models often managed to generalize well, suggesting they had indeed learned the underlying system dynamics.

4.5.3 Influence of Step Duration and Sampling Rate

- **Step Duration**

Ensuring each step reached (or nearly reached) steady state was crucial. If the step was truncated too early, the network had limited information on the full transient, harming one and multi-step accuracy.

- **Sampling Rate**

Faster sampling reduces the effective historical span when delays (or window sizes) remain fixed. For example, if the network only considers two delays and the interval between samples is very short, it only captures a limited data history—potentially insufficient to distinguish genuine transients from noise. Thus, there is an optimal sampling rate (with a corresponding adjustment in delays or window size) that ensures adequate capture of the temporal evolution while mitigating measurement noise.

4.6 Observations on MSE and MAE

Fig. 33 illustrates the evolution of one-step-ahead (top row) and multi-step-ahead (bottom row) MAE from Experiment A (a single excitation) through Experiment B (15 excitations) to Experiment D (minimal architecture). In Experiment A, the MAE varies considerably—especially for multi-step predictions—but with multiple excitations (Experiment B) and optimal tuning (Experiment D), errors drop significantly and remain consistently low, indicating that all three architectures can achieve similar performance with diverse training data. In addition, Fig. 34 compares the MSE (left) and MAE (right) for one-step-ahead (top row) and multi-step-ahead (bottom row) forecasts in Experiment D, evaluated on Sets 1–5, confirming that differences among NARX, LSTM-A, and LSTM-B are modest. Furthermore, Appendix I provides MAE histograms and CDFs encompassing all configurations tested in Experiment C, offering a broader statistical perspective that supports the overall similarity in performance.

4.6.1 Multi-Step Performance

- When trained on multiple excitations, all three architectures learn the mapping sufficiently to produce **long-horizon predictions** that stay near the real trajectory, albeit with somewhat larger error than the immediate one-step-ahead case.

4.6.2 Overall Accuracy

- As seen in Fig. 34, once the networks are properly tuned (**Experiment D**), differences among NARX, LSTM-A, and LSTM-B are relatively small. The **CDFs** in **Appendix I** confirm that, although one model may outperform another in certain scenarios, these margins are generally modest.

4.6.3 Extrapolation Errors

- All three models can handle **untrained conditions** to a degree, although **small increases** in MSE and MAE are visible—particularly in **Sets 3 and 5** of Fig. 34, which lie outside

the training range. This underscores both the **feasibility** of extrapolation and the **challenges** of predicting beyond the observed domain.

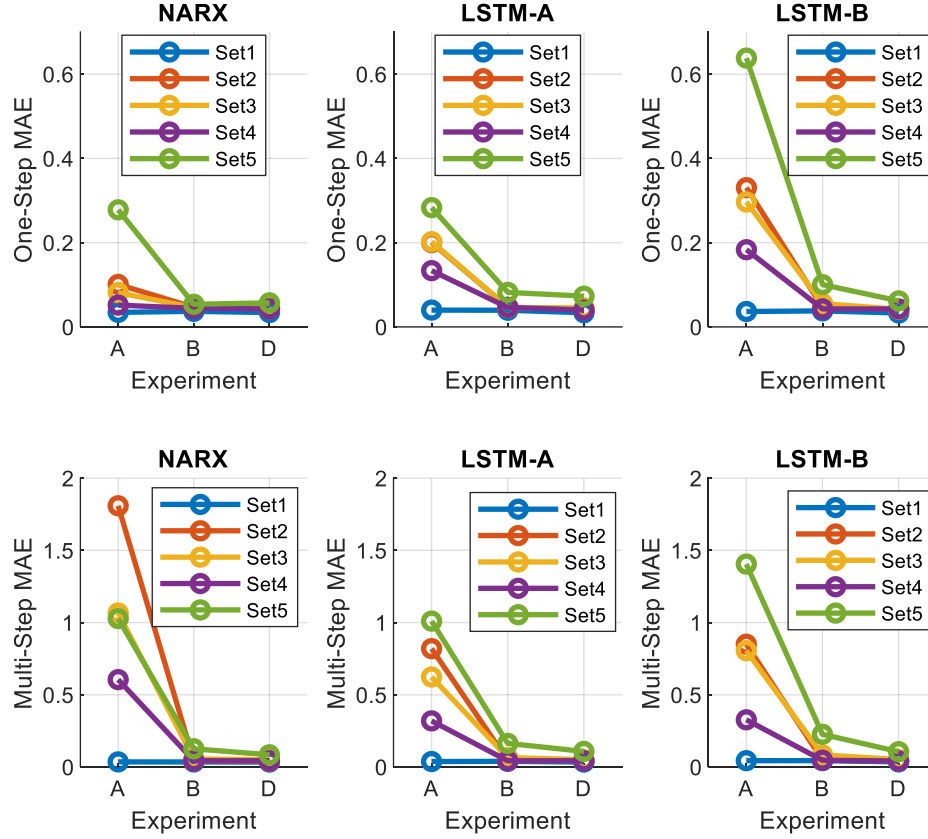


Fig. 33: One-step (top row) and multi-step (bottom row) MAE for NARX, LSTM-A, and LSTM-B across Experiments A, B, and D, evaluated on five different sets (Set1–Set5).

Comparison of NARX, LSTM-A, and LSTM-B by Set

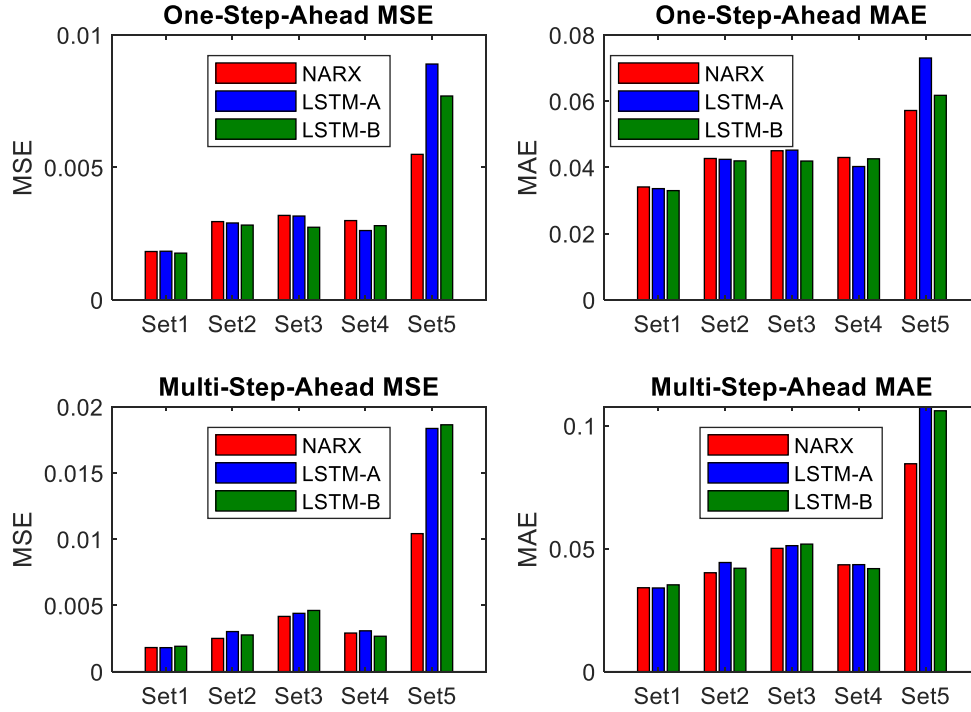


Fig. 34: One-step-ahead (top row) and multi-step-ahead (bottom row) MSE (left) and MAE (right) for NARX, LSTM-A, and LSTM-B across Sets 1–5 in the Experiment D.

4.7 Iterations and Training Efficiency

Table 1 compares the number of iterations required to reach the best validation score (“Selected Iterations”) and the total iterations performed before stopping. These values are based on the extensive set of experiments conducted in Experiment C, where many configurations were run and then computed statistical distributions of the iteration counts. The resulting distributions—presented in detail in Appendix H—support the following observations: Despite slight numerical differences, **all three architectures exhibit broadly similar training behaviors.**

- Despite slight numerical differences, all three architectures exhibit broadly similar training behaviors.
- NARX sometimes continues refining for more epochs, whereas LSTM-based models tend to converge or plateau earlier. However, the overall variance is large, and the practical differences between them are often minor.

Table 1: Iteration Performance Metrics for NARX, LSTM-A, and LSTM-B

Metric	Approach	Mean $\pm$ STD	P25	P50	P75
Selected Iterations	NARX	55.34 $\pm$ 32.50	34	50	68
	LSTM-A	49.31 $\pm$ 34.43	28	44	64
	LSTM-B	51.50 $\pm$ 37.58	24	44	66
Total Iterations	NARX	92.22 $\pm$ 31.39	72	86	104
	LSTM-A	79.65 $\pm$ 32.89	62	76	92
	LSTM-B	81.73 $\pm$ 37.86	58	76	96

4.8 Simulation and Training Times

Table 2 presents the simulation and training times obtained from Experiment C. The simulation time corresponds to the total time required to run all predictions (Sets 1–5 for both one-step-ahead and multi-step-ahead forecasts), while the training time indicates the duration needed to train the models. These metrics were calculated based on the extensive set of experiments conducted in Experiment C, and the corresponding statistical distributions are detailed in Appendix H. The results show that:

- NARX generally requires less training and simulation time than the two LSTM variants.
- LSTM-A and LSTM-B incur somewhat higher computational costs, though the difference is not dramatic—on the order of a few extra seconds.
- In practice, if computational resources are limited or real-time constraints exist, the slight speed advantage of NARX may be a deciding factor.

Table 2: Simulation and Training Times for NARX, LSTM-A, and LSTM-B

Metric	Approach	Mean $\pm$ STD	P25	P50	P75
Total Simulation Time (s)	NARX	5.86 $\pm$ 2.69	4.29	5.29	6.32
	LSTM-A	7.77 $\pm$ 3.53	5.59	7.24	8.56
	LSTM-B	7.91 $\pm$ 3.77	5.64	7.18	8.68
Training Time (s)	NARX	9.50 $\pm$ 3.52	7.28	8.73	10.67
	LSTM-A	11.04 $\pm$ 5.86	8.37	10.22	12.43
	LSTM-B	10.80 $\pm$ 5.11	7.71	9.8	12.47



4.9 Discussion and Conclusions for the Simple System

The analysis of the simple second-order reaction model demonstrates that all three neural network architectures—NARX, LSTM-A, and LSTM-B—can achieve reliable performance when trained on diverse datasets and tuned appropriately. The key findings from the experiments are as follows:

- **Training Data Diversity and Excitation Configuration:**

Experiment A, which used a single excitation, sometimes produced acceptable one-step-ahead predictions but failed to provide stable multi-step forecasts. In contrast, incorporating multiple excitations (Experiment B) markedly improved the ability of all models to capture the underlying system dynamics. The minimal configuration obtained in Experiment D further confirmed that a training set with around 6 excitations, along with properly adjusted parameters, is sufficient to yield consistently low errors across both one-step-ahead and multi-step-ahead forecasts.

- **Error Metrics and Extrapolation Performance:**

As shown in Fig. 33, the evolution of the MAE from Experiment A to D indicates a significant reduction in error—especially for multi-step predictions—when training data diversity is increased. Fig. 34 confirms that, in the optimal (minimal) configuration, differences in MSE and MAE among NARX, LSTM-A, and LSTM-B are modest. Although small increases in error are observed in extrapolation scenarios (Sets 3 and 5), all three architectures demonstrate a remarkable ability to generalize beyond the training range. The broader statistical perspective provided by the MAE histograms and CDFs in Appendix I (from Experiment C) further reinforces the overall similarity in performance.

- **Training Efficiency and Computational Cost:**

While NARX tends to require slightly more iterations before reaching the best validation state, its overall training and simulation times are generally lower compared to the LSTM-based architectures. Although LSTM-A and LSTM-B incur a higher computational cost, the difference is only a few seconds on average—a factor that may become relevant under real-time constraints or limited computational resources.

In summary, the results suggest that the choice among NARX, LSTM-A, and LSTM-B for modeling the second-order reaction system should be guided by the specific needs of the application. If computational speed and simplicity are prioritized, NARX is a solid candidate. Although the experimental results indicate that LSTM-A and LSTM-B achieved similar final prediction accuracies under the evaluated conditions, it is important to note that LSTM-A processes each input window independently by re-initializing states for every window. In contrast, LSTM-B retains an internal state that functions as memory within each window—



although this state is reset for each new window. This difference might be advantageous for capturing longer-term temporal dependencies in processes with more complex dynamics, although such benefits were not directly observed in the current study.

It is also worth mentioning that attempts were made in both LSTM-A and LSTM-B to update the state at each prediction; however, this approach led to an accumulation of errors, likely because the networks were not trained to handle continuous state updates.

In any case, the careful selection of training data—particularly through the use of multiple excitations—emerges as a critical factor for achieving reliable one-step-ahead and multi-step-ahead predictions.



5. Analysis of the Anaerobic Digestion System

5.1 Introduction

This chapter applies insights from the simple system analysis to more sophisticated anaerobic digestion models, using the Hill model (see Section 2.3.2) and the AM2 model (see Section 2.3.3) as emulators of the real process. The Hill model, with its simplified representation of anaerobic digestion dynamics, serves as a bridge between the simple reaction system and more detailed process descriptions, while the AM2 model incorporates additional complexities such as electrochemical equilibrium and mass transfer effects. The objective of this chapter is to evaluate the performance of LSTM-B architecture when applied to these sophisticated models.

LSTM-B was chosen for its flexibility and tunable parameters—such as the adjustable window size—and its inherent memory capability, which may be beneficial when dealing with more complex systems. Although previous experiments indicated that all architectures performed similarly, the additional adaptability and memory features of LSTM-B justify its selection for these experiments, despite some trade-offs in simplicity and speed. The following sections will assess its short-term and long-term prediction accuracy, thereby determining whether LSTM-B is well-suited for capturing the dynamic behavior of sophisticated anaerobic digestion processes.

It is noteworthy that, based on the parameter values obtained from the literature and employed in this study, the Hill model exhibits a slower dynamic response compared to the AM2 model. While this difference was not intentionally designed, it provides an opportunity to analyze process behavior over a broader range of time scales.

5.2 Experimental Setup

For the anaerobic digestion models, simulation data are generated separately for the Hill model and the AM2 model. In both cases, step changes in the dilution rate (D) are applied, and the corresponding methane flow rate is recorded. Gaussian noise is added to the methane flow rate measurements to mimic real-world variability. The range of D values is chosen to be near the washout region—without reaching it—since this region exhibits highly nonlinear behavior and is particularly challenging to model. This setup is used to evaluate the network's ability to capture the dynamic response of the system under demanding conditions.

5.2.1 Dataset Division and Training Details

The datasets for each digestion model are generated independently.

For the Hill Model:

- **Set 1 (Training and Validation):**

Training and validation data for the Hill model are generated using controlled step excitations. This dataset is used exclusively to train the LSTM-B network in a one-step-ahead manner. An example of this data is shown on the left of Fig. 35.

- **Set 2 (Testing):**

The test dataset for the Hill model is generated using controlled step excitations within the same operating range as the training set. This dataset is used to evaluate both one-step-ahead and multi-step-ahead prediction performance. An example of the test data is shown on the right side of Fig. 35.

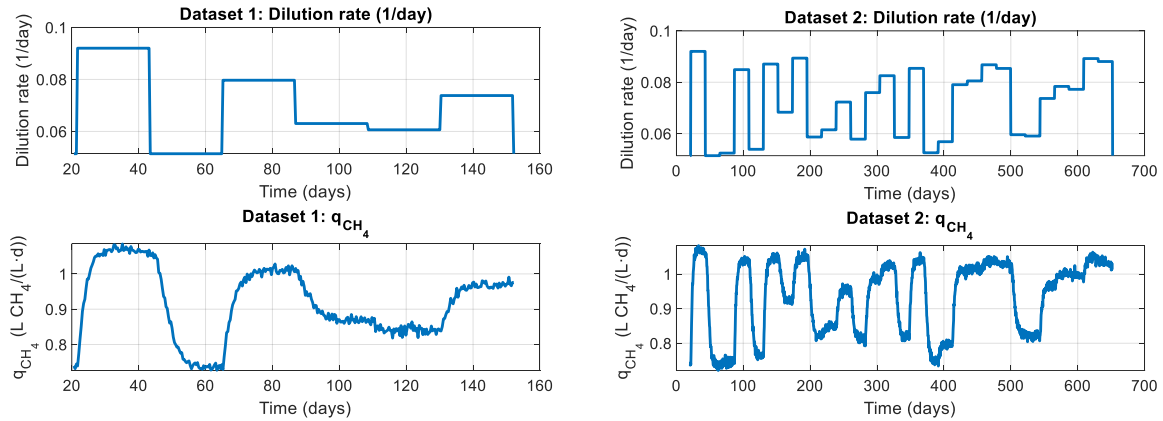


Fig. 35: Example training/validation dataset (left) and testing dataset (right) for the Hill model. The top row shows the dilution rate profiles, while the bottom row illustrates the corresponding methane flow rate (q_{CH_4} in $L CH_4/L \cdot d$)

For the AM2 Model:

- **Set 1 (Training and Validation):**

An independently generated training and validation dataset is prepared for the AM2 model under defined operating conditions. Controlled step excitations are used to expose the network to the essential dynamics of this more complex process. An example of the training/validation data for the AM2 model is provided on the left side of Fig. 36.

- **Set 2 (Testing):**

The test dataset for the AM2 model is designed to assess the network's ability to accurately predict both one-step-ahead and multi-step-ahead outputs over extended prediction horizons. These test cases challenge the network by covering a broader range of operating conditions than those used during training. An example of the corresponding test data is shown on the right side of Fig. 36.

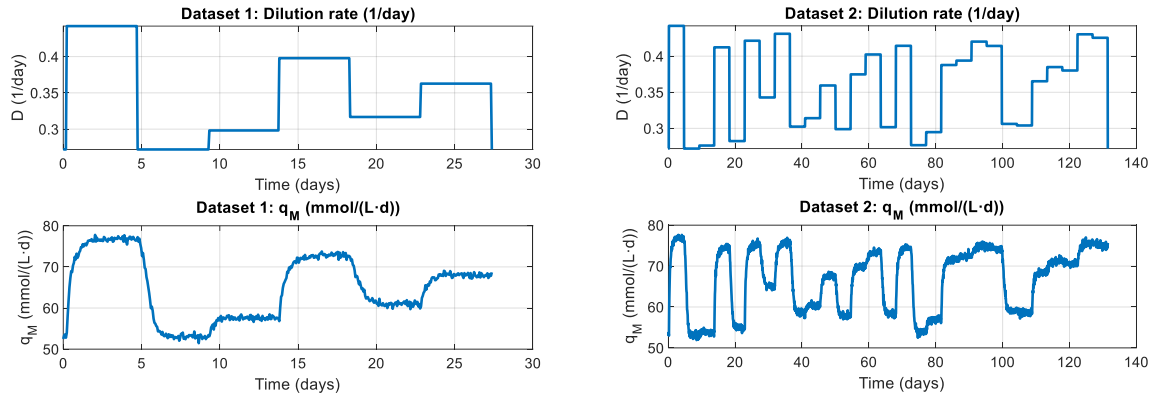


Fig. 36: Example AM2 datasets: training/validation (left) and testing (right). Top: dilution rate D ; bottom: methane flow q_M . Training spans 30 days with step changes; testing covers 140 days.

For both digestion models, the LSTM-B network is trained using the respective Set 1 datasets in a one-step-ahead mode. During testing, multi-step predictions are obtained by recursively feeding the network's outputs back as inputs. Network parameters were adjusted from the simple system experiments to better accommodate the increased complexity of each digestion model.

5.3 Evaluation of LSTM-W on Anaerobic Digestion Models

5.3.1 Analysis on the Hill Model

This experiment evaluates the LSTM-B network's performance on the Hill model under two different training configurations: one using **2 step excitations** and another using **6 step excitations**. Anaerobic digestion is inherently slow, making it challenging and costly to obtain large datasets. Thus, assessing whether the network can accurately predict system dynamics with limited training data is crucial.

- **2 Step Excitation:**

This configuration tests the network's ability to learn from minimal data. Although a single step was found to be insufficient, using 2 step excitations offers a potential compromise, allowing to investigate if this minimal input can still capture the essential system dynamics for both one-step-ahead and multi-step-ahead predictions.



- **6 Step Excitation:**

In this setup, the network is trained with 6 successive step excitations, providing a richer representation of the dynamic behavior. The extended excitation configuration is expected to improve long-term prediction accuracy, as it offers more comprehensive information on the process dynamics, leading to more stable multi-step forecasts.

Given the slow nature of anaerobic digestion, which makes extensive data collection challenging, comparing these two configurations will reveal how effectively the LSTM-B model can adapt to limited versus more extensive training data in capturing the Hill model's dynamics.

5.3.1.1 Experiment A: 2-Step Excitation Configuration

Objective:

Evaluate whether training the LSTM-B network on **only two excitation steps** is sufficient to capture the Hill model's dynamics for both one-step-ahead and multi-step-ahead predictions.

Observations:

- The **training vs. validation loss** shows that the **validation loss** exhibits noticeable oscillations, whereas the **training loss** steadily decreases toward a near-perfect fit. This discrepancy likely stems from the **limited variety** in the training set: the network can memorize those few samples quite well but lacks sufficient diversity to **generalize** robustly to the validation segment. Consequently, small adjustments in the network weights can lead to significant fluctuations in validation performance, producing the observed oscillations in the validation loss curve.
- **One-step-ahead** predictions on the test set are generally accurate, even with only two training steps. (see Fig. 37)
- **Multi-step-ahead** forecasts are reasonable overall but tend to **overestimate** at medium-to-high methane flow rates, suggesting the model does not fully capture the more nonlinear behavior in that range. In contrast, predictions at lower flow rates remain more reliable. (see Fig. 37)
- A **segmented multi-step prediction** A segmented approach was also tested, where the model is periodically re-fed with fresh real data after every 10-step forecast. Although this method may slightly reduce the overestimate compared to continuous multi-step prediction, the overestimation at higher methane flow rates remains notable. (see Fig. 38)

Conclusions:

These results confirm that the LSTM-B network can capture system dynamics with very limited data (two excitation steps). However, the inconsistent reliability observed across different trials and the oscillatory pattern in the validation loss indicate that the model's performance may vary.

Selecting the model at the point of best validation consistently yields better test performance than using the final training iteration, highlighting the need for further optimization to ensure robust and reliable predictions. *For further information on training progress and details regarding the training and validation sets, refer to Appendix J.*

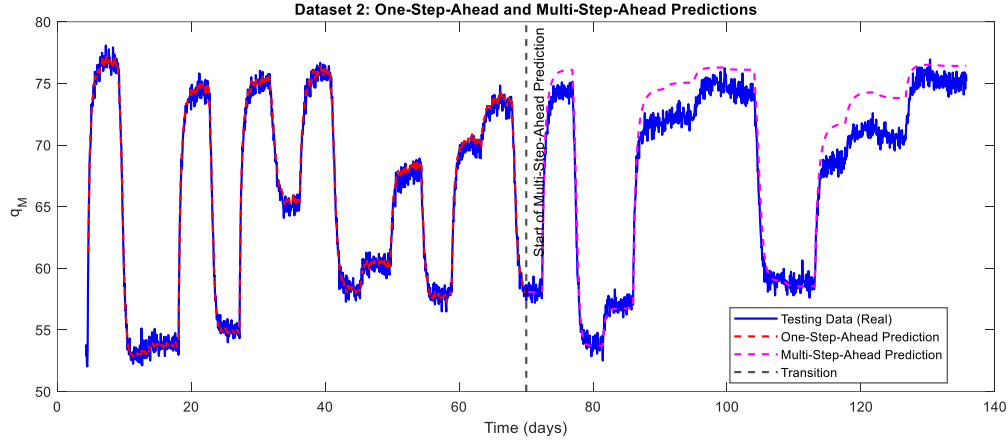


Fig. 37: LSTM-B on Hill model with two-step training (Experiment A). Blue: ground truth; red: one-step; magenta: multi-step. Overestimation at mid-to-high methane flow suggests limited training fails to capture nonlinear dynamics.

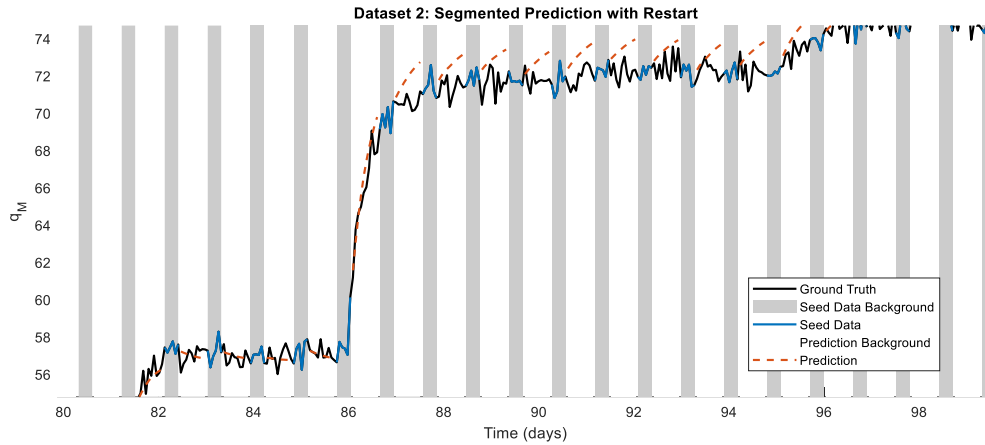


Fig. 38: Segmented multi-step prediction with reinitialization every 10 steps (gray: seed data). Slightly reduces overestimation vs. continuous mode, but high methane flow remains challenging due to limited training diversity.

5.3.1.2 Experiment B: 6-Step Excitation Configuration

Objective:

Determine whether increasing the number of excitation steps from 2 to 6 yields more stable training, improved one-step-ahead and multi-step-ahead predictions, and reduced overestimation in the Hill model.

Observations:

- **Training vs. Validation Loss:** Compared to the 2-step experiment, the validation loss in the 6-step configuration decreases more consistently, indicating improved generalization. Minor oscillations are still present, but they are much less pronounced.
- **One-Step-Ahead Predictions:** The model continues to accurately track the methane flow rate, with only small deviations observed on the test set. (see Fig. 39)
- **Multi-Step-Ahead Forecasts:** Although some overestimation is still observed at medium-to-high methane flow rates, its magnitude is noticeably lower than that observed in the 2-step configuration, suggesting that the additional excitations help the network better capture the system's nonlinear behavior. (see Fig. 39)
- **Segmented Multi-Step Predictions:** When using a segmented approach—where the network is periodically re-fed with fresh real data after every 10-step forecast—the overestimation remains but is further reduced compared to the 2-step configuration and is slight relative to the actual measured data. (see Fig. 40)

Conclusions:

Increasing the excitation steps to six leads to more stable training and improved predictive performance for the Hill model. The network accurately captures one-step-ahead dynamics, and while multi-step-ahead forecasts still exhibit some slight overestimation at medium-to-high methane flow rates, the magnitude of this overestimation is noticeably lower than that observed with only two excitation steps. These findings suggest that a richer training dataset with six excitations enhances the network's ability to capture the system's nonlinear behavior, although further optimization may be needed to fully minimize the residual overestimation at higher flow rates. *For further information on training progress and details regarding the training and validation sets, refer to Appendix J.*

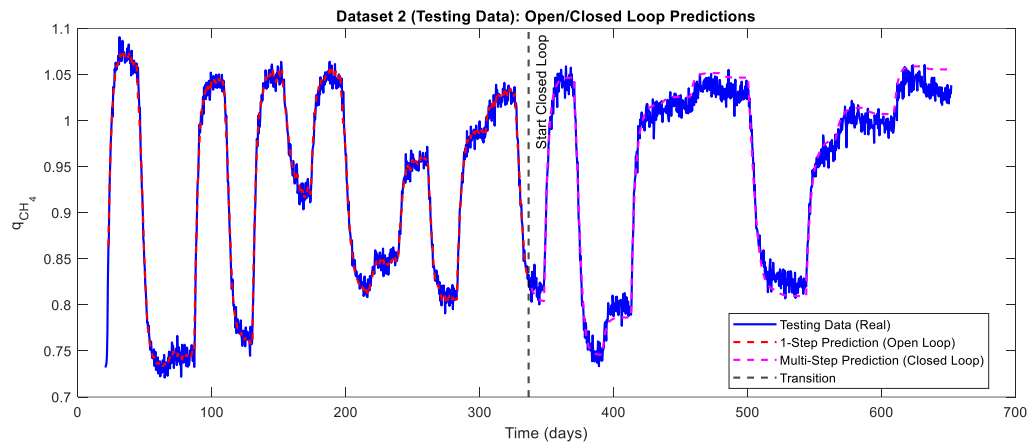


Fig. 39: Hill model on Dataset 2 with six-step training. Black: ground truth; red: one-step; magenta: multi-step. Overestimation at high methane flow is reduced vs. two-step case.

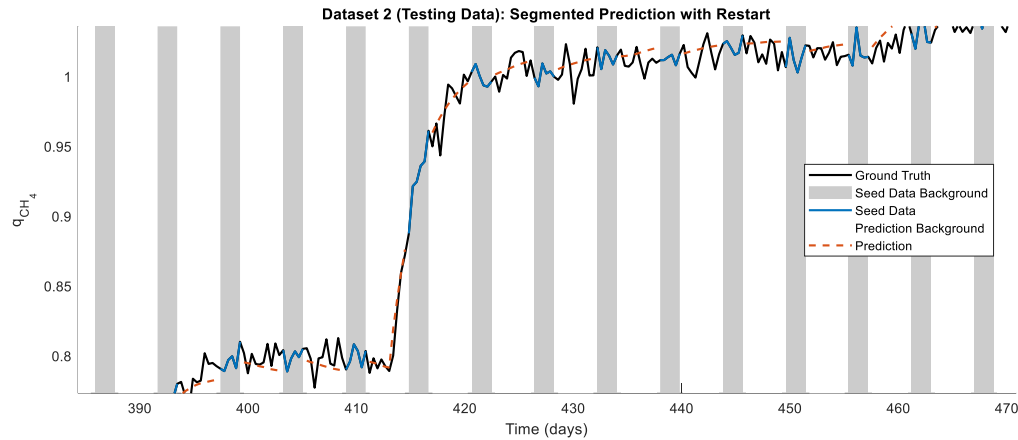


Fig. 40: Segmented multi-step prediction with restarts. Black: ground truth; orange: forecasts; shaded: seed data. Further reduces overestimation, but errors remain at high methane flow. Analysis on the AM2 Model

Following a similar approach to the Hill model, the LSTM-B network was evaluated on the AM2 model under **two** experimental configurations: one using **2 step excitations** and another using **6 step excitations**. This approach aims to determine how well the network can learn from **minimal data** versus a **richer training set**, and whether the improvements observed with the Hill model extend to the more complex AM2 model.

5.3.1.3 Experiment A: 2-Step Excitation Configuration

Objective:

Examine whether two excitation steps are sufficient for the LSTM-B network to learn the AM2 model's dynamics and deliver accurate one-step-ahead and multi-step-ahead forecasts.

Observations:

- **Training vs. Validation Loss:** The training loss decreases steadily, while the validation loss—based on data from the lower methane flow rate region—shows moderate fluctuations. This suggests that, although the model fits the limited training data well, its ability to generalize to conditions outside that range (i.e., medium-to-high methane flow rates) may be compromised, as was observed in the Hill model example.
- **One-Step-Ahead Performance:** Short-range forecasts generally align well with the actual data, indicating that the network can learn key dynamics even with minimal training. (see Fig. 41)
- **Overestimation at Medium-to-High Values:** Multi-step predictions exhibit notable overestimation in the medium-to-high methane flow rate range, suggesting that with only two excitation steps the model does not fully capture the system's nonlinear behavior. (see Fig. 41)

- **Segmented Multi-Step Forecasting:** A segmented approach, where the network is periodically re-fed with fresh real data after every 10-step forecast, was also tested. Although this strategy slightly reduces the overestimation compared to continuous multi-step prediction, the overestimation at higher flow rates remains significant. (see Fig. 42)

Conclusions:

While the LSTM-B network can produce acceptable one-step-ahead predictions with only two excitation steps, the multi-step forecasts reveal significant overestimation at medium-to-high methane flow rates. This outcome indicates that the limited training data—primarily covering the lower flow rate region—do not provide sufficient variety to capture the AM2 model's more complex, nonlinear dynamics. Consequently, a richer training dataset incorporating additional excitation steps is likely necessary for more robust performance. *For further information on training progress and details regarding the training and validation sets, refer to Appendix K.*

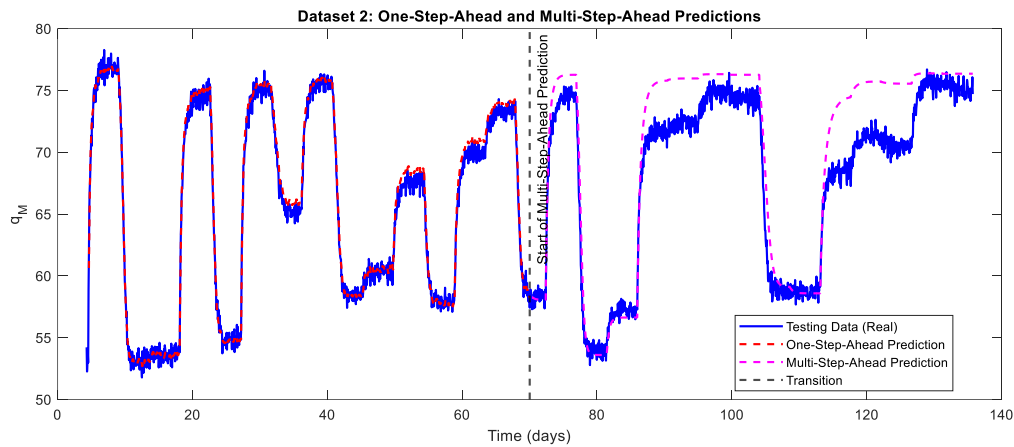


Fig. 41: AM2 model on Dataset 2 with two-step training (Experiment A). Blue: ground truth; red: one-step; magenta: multi-step. One-step tracks well, but multi-step overestimates at mid-to-high methane flow.

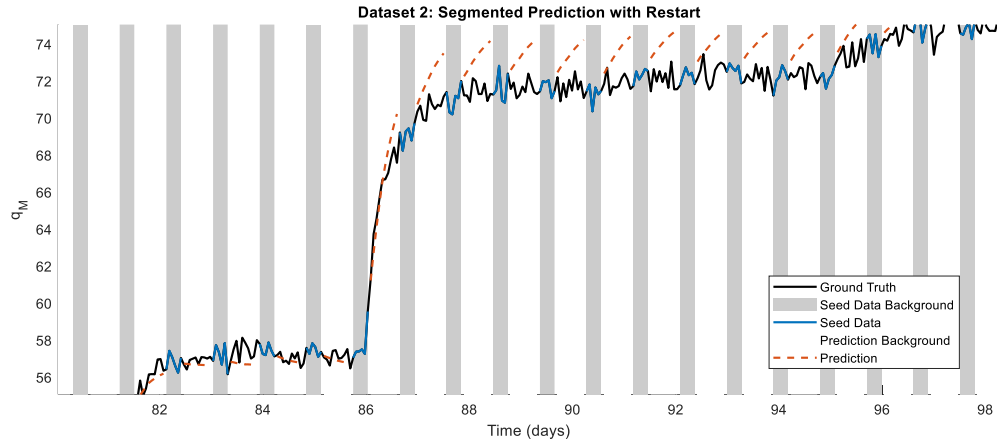


Fig. 42: Segmented multi-step forecasting with restarts. Black: ground truth; orange: short-window forecasts; shaded: seed data. Reduces overestimation slightly, but large errors persist at high flow.

5.3.1.4 Experiment B: 6-Step Excitation Configuration

Objective

Determine whether increasing the number of excitation steps from 2 to 6 improves training stability and predictive performance in the AM2 model, particularly regarding overestimation at higher methane flow rates.

Observations

- **Training vs. Validation Loss:** Compared to the 2-step experiment, both training and validation loss curves show more stable convergence (see figure above). The network generalizes better, with fewer fluctuations in validation loss.
- **One-Step-Ahead Performance:** Predictions closely track the true methane flow rate, showing minimal deviations even at higher flow rates (see Fig. 43).
- **Multi-Step-Ahead Forecasts:** Although a slight overestimation remains at medium-to-high flow rates, it is noticeably lower than in the 2-step case, indicating that the additional excitations help the model better capture the AM2 system's nonlinearities (see Fig. 43).
- **Segmented Multi-Step Prediction:** When periodically re-feeding the network with fresh real data (every 10-step forecast), overestimation is still present but appears smaller than in continuous multi-step mode, suggesting a modest additional benefit from segmentation (see Fig. 44).

Conclusions

Providing six excitation steps yields more stable training and improved overall predictions in the AM2 model. While multi-step forecasts still exhibit some overestimation at higher methane flow rates, the magnitude is reduced compared to the 2-step setup. This indicates that a richer

training dataset with six steps better equips the LSTM-B network to handle the nonlinear aspects of the AM2 model. *For further information on training progress and details regarding the training and validation sets, refer to Appendix K.*

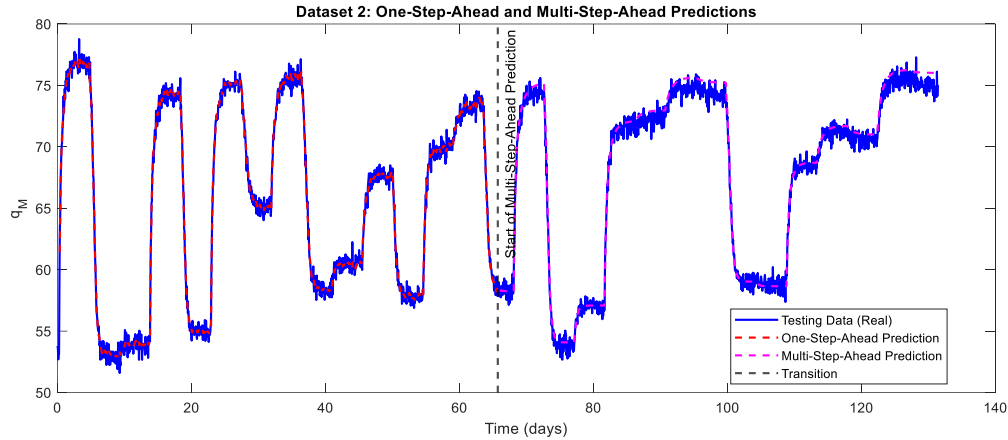


Fig. 43: AM2 model on Dataset 2 with six-step training (Experiment B). Black: ground truth; red: one-step; magenta: multi-step predictions. Overestimation at high methane flow is reduced vs. two-step setup, showing benefits of richer training.

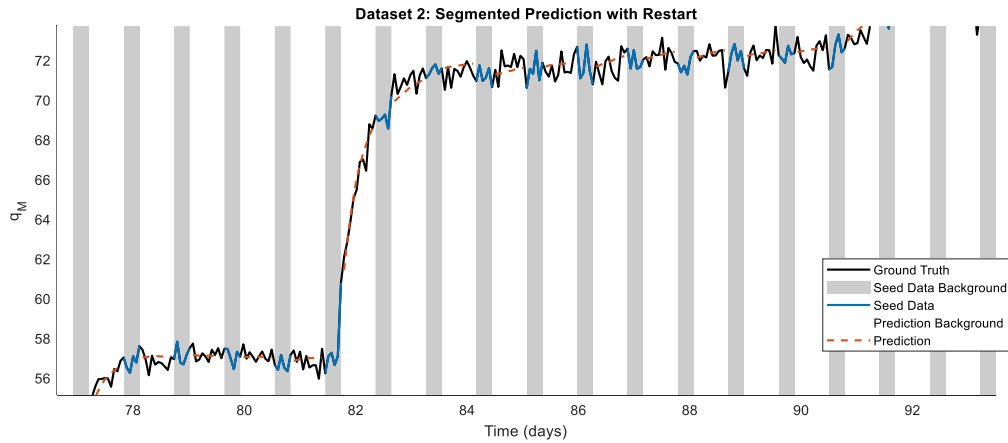


Fig. 44: Segmented multi-step forecasting on the same dataset. Black: ground truth, orange: forecast. Shaded areas indicate seed data. This approach reduces overestimation vs. continuous forecasting, though some errors remain at high flow.

5.3.2 Summary of MAE comparisons

Fig. 45 (Hill) and Fig. 46 (AM2) present the Mean Absolute Error (MAE) for one-step-ahead (left plots) and multi-step-ahead (right plots) predictions under Experiment A (blue bars) and Experiment B (orange bars). Each figure groups the MAE values by training, validation, and testing sets.

Hill Model (Figure Fig. 45):

- In one-step-ahead mode, Experiment B yields generally lower MAE across all sets.

- In multi-step-ahead mode, MAE rises overall—especially in testing—but remains lower in Experiment B, suggesting that adding more excitations (6 vs. 2) improves long-horizon predictions by reducing overestimation.

AM2 Model (Figure Fig. 46):

- For one-step-ahead predictions, the difference between Experiments A and B is most evident in validation and testing, where Experiment B tends to have lower MAE.
- Multi-step forecasts show higher errors than one-step but again favor Experiment B, indicating that additional excitations help reduce overestimation in the more complex AM2 system.

These bar charts align with the time-series observations in Sections 5.3.1, confirming that Experiment B typically improves multi-step accuracy and mitigates overestimation at higher flow rates. Full numeric tables (including MSE) can be found in Appendix J (Hill) and Appendix K (AM2).

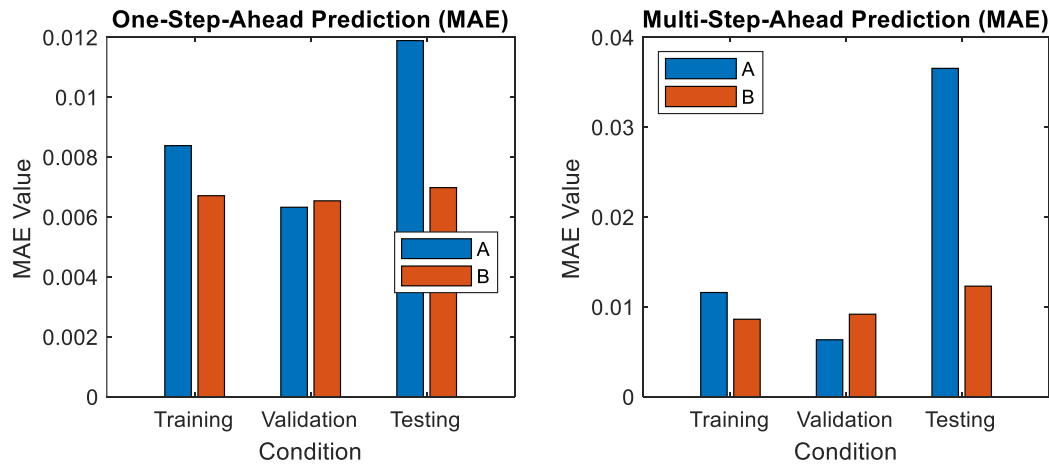


*Fig. 45: Mean Absolute Error (MAE) for the **Hill model** under one-step-ahead (left) and multi-step-ahead (right) predictions, comparing Experiment A (blue) and Experiment B (orange) across training, validation, and testing sets*

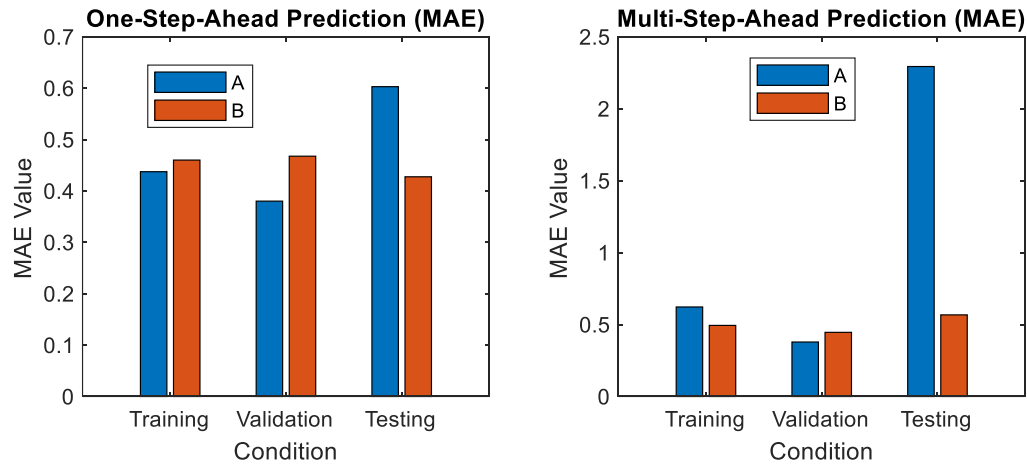


Fig. 46: Mean Absolute Error (MAE) for the **AM2 model** under one-step-ahead (left) and multi-step-ahead (right) predictions, comparing Experiment A (blue) and Experiment B (orange) across training, validation, and testing sets.

5.4 Discussion and Comparative Analysis

The experiments with the Hill and AM2 models provide valuable insights into the performance of the LSTM-B network in capturing anaerobic digestion dynamics under different training configurations. Although the current settings are not necessarily optimal, they offer a practical baseline and highlight key trade-offs:

Training Data Requirements and Configuration:

The 2-step excitation configuration can sometimes yield acceptable one-step-ahead predictions but struggles with multi-step forecasts, particularly when D is high within the chosen operating range—where the response becomes highly nonlinear. In contrast, the 6-step configuration improves training stability and overall prediction accuracy. While six excitations represent the minimum that produced consistent results in these experiments, this finding is qualitative and suggests that further fine-tuning of the training parameters is warranted.

Practical Considerations for Implementation:

In real-world applications, the controlled conditions used for data generation may not be maintained over extended periods. In these experiments, collecting six excitation steps required data over approximately 145 days for the process represented by the Hill model and about 30 days for the process represented by the AM2 model. Such lengthy data collection periods can be impractical in a pilot plant, where operating conditions may even change over time. Moreover, the ability to produce reliable multi-step predictions is essential for potential integration into Model Predictive Control (MPC) frameworks, which rely on accurate forecasts over a finite horizon. One potential strategy is to initially train the network with a smaller dataset (e.g., using two excitation steps) and then continuously update the model as new data become



available. This approach would allow the network to adapt to evolving conditions without the need to collect an extensive dataset before deployment.

Future Directions:

Further fine-tuning of the LSTM-B network—including a systematic exploration of the number of excitation steps, step durations, and other parameters—may help reduce the residual overestimation observed in multi-step forecasts. Additionally, investigating alternative training strategies or model architectures could lead to improved reliability and predictive accuracy across a broader range of operating conditions.

In summary, while the LSTM-B network shows promising performance in modeling anaerobic digestion processes for both the slow (Hill) and fast (AM2) systems, additional refinements and adaptive strategies are necessary before practical deployment in environments where operating conditions are subject to change.



6. Conclusions and Future Work

This final chapter summarizes the main findings and outlines potential directions for future work. The goal was to evaluate data-driven neural network architectures for predicting anaerobic digestion dynamics using process emulators (Hill and AM2 models), with a focus on short-term and multi-step forecasts and their potential application in control frameworks such as Model Predictive Control (MPC).

6.1 Key Findings

6.1.1 Performance on the Simple System

- **Model Performance:**

Three architectures—NARX, LSTM-A, and LSTM-B—were tested using a simple second-order reaction model. All architectures delivered accurate one-step-ahead predictions, while multi-step forecasts benefited significantly from training with multiple excitation steps (e.g., 6 steps). Although not essential, a balanced training configuration—one that maintains simplicity, uses limited data, and avoids excessive network complexity—contributed to obtaining reliable long-horizon predictions for the anaerobic digestion system.

- **Training Data Diversity:**

When training with a single excitation step, both one-step and multi-step forecasts were unreliable, with predictions deviating significantly from the true behavior. Incorporating multiple excitations (6 steps) enhanced training stability and resulted in more accurate long-horizon predictions.

- **Sensitivity Analysis:**

A systematic variation of key parameters (number of steps, sampling rate, step duration, delays, and network size) demonstrated that it is possible to identify a balanced configuration that effectively captures the system dynamics while maintaining simplicity.

6.2 Application to Anaerobic Digestion Models

- **Hill Model (Slow Process):**

For the Hill model, increasing the excitation steps from 2 to 6 improved training stability and reduced overestimation in multi-step forecasts at higher dilution rates. One-step predictions were generally reliable, while multi-step forecasts tended to overestimate

methane flow rates in regions where the process becomes highly nonlinear. These findings indicate that richer training data improves the network's ability to capture the nonlinear dynamics more effectively.

- **AM2 Model (Faster Process):**

Similar trends were observed for the AM2 model. Minimal training (2-step excitations) produced acceptable one-step predictions, but multi-step forecasts revealed significant overestimation at medium-to-high methane flow rates. Increasing the excitation steps to 6 reduced this overestimation and resulted in more stable forecasts.

6.3 General Insights

- **Feasibility of Data-Driven Models:**

The results show that data-driven neural network models, particularly those based on LSTM architectures, can capture complex nonlinear dynamics in anaerobic digestion processes. While one-step predictions are robust with limited training data, reliable multi-step forecasts require a richer dataset to accurately represent the system's behavior.

- **Implications for Control Applications:**

Accurate multi-step forecasting is essential for integration into MPC frameworks, which depend on reliable short-term predictions to optimize control actions. Although the current models perform well under controlled conditions, the extended data collection periods required for training (approximately 145 days for the Hill model and 30 days for the AM2 model) may be impractical in pilot-scale systems, where operating conditions are more variable.

6.4 Limitations

- **Training Data Diversity:**

Using only a few excitation steps may not capture the full range of system dynamics, particularly under higher dilution rates where nonlinear effects are more pronounced.

- **Data Collection Duration:**

The required training data collection periods (approximately 145 days for the Hill model and 30 days for the AM2 model) may be impractical in a dynamic pilot plant environment.



6.5 Future Work

- **Further Parameter Exploration:**

A continued examination of key parameters—such as the number of excitation steps, step duration, delays, and network configuration—will help identify settings that minimize forecasting errors and better capture anaerobic digestion dynamics.

- **Adaptive Model Updating:**

Methods for updating the model with newly available data will be explored to allow for adjustments to changing process conditions without requiring an extensive initial data collection period.

- **Integration with Control Strategies:**

The focus will be on evaluating the performance of the data-driven model as part of a control system under realistic operating conditions.

- **Development and Validation of an Anaerobic Digestion predictor:**

Further work will refine the data-driven approach for anaerobic digestion, potentially developing a predictor based on a detailed model such as ADM1. The suitability of the current models as emulators for control purposes will be evaluated, along with any necessary improvements.

- **Exploration of Alternative Architectures:**

In addition to the NARX and LSTM-based models examined in this study, other architectures may be considered to determine if they offer advantages in modeling the complex dynamics of anaerobic digestion.

6.6 Final Remarks

The findings indicate that data-driven neural network models are a promising method for predicting the nonlinear dynamics of anaerobic digestion. While the current configurations provide reliable one-step predictions, achieving consistent multi-step forecasts requires richer datasets. The long data collection periods observed (ranging from 30 to 145 days) present practical challenges for pilot-scale applications. The planned master's thesis will build on these results by refining the data-driven approach and assessing its integration into control frameworks such as MPC for anaerobic digestion processes.



7. References

- Bernard, O., Hadj-Sadok, Z., Dochain, D., Genovesi, A., & Steyer, J.-P. (2001). Dynamical Model Development and Parameter Identification for an Anaerobic Wastewater Treatment Process. *Biotechnology and Bioengineering*, 75(4), 424–438.
- Brunton, S. L., & Kutz, J. N. (2021). *Data-Driven Science and Engineering Machine Learning, Dynamical Systems, and Control*.
- Cruz, I. A., Andrade, L. R. S., Bharagava, R. N., Nadda, A. K., Bilal, M., Figueiredo, R. T., & Ferreira, L. F. R. (2021). An overview of process monitoring for anaerobic digestion. In *Biosystems Engineering* (Vol. 207, pp. 106–119). Academic Press. <https://doi.org/10.1016/j.biosystemseng.2021.04.008>
- Emebu, S., Pecha, J., & Janáčová, D. (2022). Review on anaerobic digestion models: Model classification & elaboration of process phenomena. *Renewable and Sustainable Energy Reviews*, 160. <https://doi.org/10.1016/j.rser.2022.112288>
- Haugen, F., Bakke, R., & Lie, B. (2013). Adapting dynamic mathematical models to a pilot anaerobic digestion reactor. *Modeling, Identification and Control*, 34(2), 35–54. <https://doi.org/10.4173/mic.2013.2.1>
- Hill, D. T. (1983). Simplified Monod Kinetics of Methane Fermentation of Animal Wastes. In *Agricultural Wastes* (Vol. 5).
- Husain, A. (1998). *MATHEMATICAL MODELS OF THE KINETICS OF ANAEROBIC DIGESTION A SELECTED REVIEW*.
- IWA. (2005). *Anaerobic Digestion Model No.1 (ADM1)*. IWA Publishing. <https://doi.org/10.2166/9781780403052>
- John D. Hedengren. (2024, March 22). *LSTM Networks*. <https://apmonitor.com/pds/index.php/Main/LongShortTermMemory>
- Kil, H., Li, D., Xi, Y., & Li, J. (2017). Model predictive control with on-line model identification for anaerobic digestion processes. *Biochemical Engineering Journal*, 128, 63–75. <https://doi.org/10.1016/j.bej.2017.08.004>
- Ljung. (1999). *System Identification: Theory for the User* (2nd ed.). Prentice-Hall PTR.
- M. Norgaard, O. Ravn, N.K. Poulsen, & L.K. Hansen. (2000). *Neural Networks for Modelling and Control of Dynamic Systems*. Springer London.
- MathWorks. (2024). *Long Short-Term Memory Neural Networks*. <https://nl.mathworks.com/help/deeplearning/ug/long-short-term-memory-networks.html>



- Piadeh, F., Offie, I., Behzadian, K., Rizzuto, J. P., Bywater, A., Córdoba-Pachón, J. R., & Walker, M. (2024). A critical review for the impact of anaerobic digestion on the sustainable development goals. In *Journal of Environmental Management* (Vol. 349). Academic Press. <https://doi.org/10.1016/j.jenvman.2023.119458>
- Sepp Hochreiter, & Jürgen Schmidhuber. (1997). LONG SHORT-TERM MEMORY. *Neural Computation*, 9(8), 1735–1780. <https://doi.org/https://doi.org/10.1162/neco.1997.9.8.1735>
- Wang, H., & Song, G. (2014). Innovative NARX recurrent neural network model for ultra-thin shape memory alloy wire. *Neurocomputing*, 134, 289–295. <https://doi.org/10.1016/j.neucom.2013.09.050>

8. Appendix

A. Symple System Parameters

Table 3: Parameters of the Simple Reaction System

Parameter	Value	Units	Description
k	0.05	L/(g·h)	Reaction rate constant for the second-order reaction.
C_{in}	5	g/L	Inlet concentration of the reactant.

B. Hill Model Parameters

Table 4: Hill Model Parameter Equations

Parameter	Equation / Definition	Units	Description
T	35	°C	Operating temperature.
μ_m	$0.013 \times T - 0.129$	1/d	Maximum specific growth rate of acidogenic bacteria.
μ_{mc}	μ_m	1/d	Maximum specific growth rate of methanogenic bacteria (set equal to μ_m).
K_{dm}	μ_m	1/d	Decay rate for acidogenic bacteria (set equal to μ_m).
K_{dmc}	μ_{mc}	1/d	Decay rate for methanogenic bacteria (set equal to μ_{mc}).
K_{id}	16	g VFA/L	Inhibition constant for acidogenesis (Husain, 1998).
K_{idc}	16	g VFA/L	Inhibition constant for methanogenesis (Husain, 1998).
Y	0.1	g organism/g BVS	Yield coefficient for acidogenic bacteria (Husain, 1998).
Y_c	0.0315	g organism/g VFA	Yield coefficient for methanogenic bacteria (Husain, 1998).
K_s	9	g BVS/L	Half-saturation constant for substrate (BVS) (Husain, 1998).

K_{sc}	3	g VFA/L	Half-saturation constant for volatile fatty acids (VFA) (Husain, 1998).
K_i	9	g VFA/L	Inhibition constant for VFA on acidogenic bacteria (Husain, 1998).
K_{ic}	11	g VFA/L	Inhibition constant for VFA on methanogenic bacteria (Husain, 1998).
S_{in}	30.2	g BVS/L	Influent concentration of biodegradable volatile solids (BVS) (Haugen, 2013).
VFA_{in}	5.21	g VFA/L	Influent concentration of volatile fatty acids (Haugen, 2013).
D_0	1/14 ($\approx$ 0.0714)	1/d	Nominal dilution rate. Note: D is expressed in 1/d, ensuring unit consistency in the mass balance equations.
S_0	4.6919	g BVS/L	Initial substrate concentration in the reactor.
X_0	1.8594	g/L	Initial concentration of acidogenic bacteria.
X_{c0}	0.6114	g/L	Initial concentration of methanogenic bacteria.
VFA_0	1.539	g VFA/L	Initial concentration of volatile fatty acids.

C. Hill Model Parameters

Table 5: AM2 Model Parameter Values and Definitions (Bernard et al., 2001)

Parameter	Value	Units	Description
k_1	42.14	mmol/g	Yield for substrate degradation.
k_2	116.5	mmol/g	Yield for volatile fatty acids (VFA) production.
k_3	268	mmol/g	Yield for VFA consumption.
k_4	50.6	mmol/g	Yield for CO ₂ production.
k_5	343.6	mmol/g	Yield for CO ₂ production.
k_6	457.73	mmol/g	Yield for CH ₄ production.
k_b	6.5×10^{-7}	mol/L	Equilibrium constant.
k_H	16	mmol/(L·atm)	Henry's constant.
k_{La}	19.8	d ⁻¹	Liquid–gas transfer constant.
k_{I2}	383.977	mmol/L	Inhibition constant.
k_{S1}	7.1	g/L (gCOD/L)	Half-saturation constant for substrate (S1).
k_{S2}	2.9524	mmol/L	Half-saturation constant for VFA (S2).
$\mu_{1,max}$	1.2	d ⁻¹	Maximum specific growth rate of acidogens.

$\mu_{2,max}$	0.2688	d ⁻¹	Maximum specific growth rate of methanogens.
α	0.5	–	Fraction of bacteria in the liquid phase.
P_T	1	atm	Total pressure.
D_0	0.34	d ⁻¹	Nominal dilution rate.
$X_{1,0}$	0.8	g/L	Initial concentration of acidogens.
$X_{2,0}$	0.8	g/L	Initial concentration of methanogens.
$S_{1,0}$	1.8	gCOD/L	Initial substrate concentration.
$S_{2,0}$	3	mmol/L	Initial VFA concentration.
Z_0	60	mmol/L	Initial total alkalinity.
C_0	65	mmol/L	Initial total inorganic carbon concentration.
S_{1in}	9.5	gCOD/L	Influent substrate concentration.
S_{2in}	93.6	mmol/L	Influent VFA concentration.
Z_{in}	62.5	mmol/L	Influent total alkalinity.
C_{in}	60	mmol/L	Influent total inorganic carbon concentration.

D. Detailed Experimental Configurations

This appendix summarizes the experimental configurations used in the study, including the base configuration, parameter variation ranges (Experiment C), dilution rate (D) settings, noise levels, and advanced model-specific settings for the Hill and AM2 models.

Table 6: Base Configuration for the Simple System (Sensitivity Experiments)

Parameter	Base Value	Units / Definition
Number of Steps (Multi-Step)	15	Number of consecutive excitations applied (starting from steady state)
Step Duration Factor	5	Step Duration = Factor / D_{max}
Sampling Rate Factor	10	Sampling Rate = 1 / ($D_{max} \times$ Factor)
Window Size / Delays	2	Number of previous time steps (delays) fed into the network
Hidden Neurons / Hidden Units	4 (NARX) ; 2 (LSTM-A and LSTM-B)	For NARX: number of neurons per hidden layer; for LSTMs: dimension of the hidden state vectors h and c
Number of Layers	1	Number of hidden layers (assumed equal size for simplicity)
Noise Level	0.05	Standard deviation added to the output C (g/L)



Table 7: Parameter Variation Ranges for the Simple System (Experiment C)

Parameter	Variation Range	Units / Notes
Number of Steps (Multi-Step)	1 to 20	
Duration Factor	1 to 15	Used in Step Duration = Factor / $D_{\max}$
Sampling Rate Factor	2 to 30	Used in Sampling Rate = $1 / (D_{\max} \times \text{Factor})$
Window Size / Delays	1 to 15	
Hidden Neurons / Units	1 to 20	(For NARX, neurons per hidden layer; for LSTMs, units per layer)
Number of Layers	1 to 4	

Table 8: Dilution Rate Settings and Noise Levels

System/Model	Dilution Rate (D) Range	Noise Level	Units
Simple System (Sets 1, 2, 4)	0.2 to 1	0.05	1/h for D; noise in C in g/L
Simple System (Sets 2, 3, 5)	0.1 to 2 (extrapolation)	0.05	1/h for D; noise in C in g/L
Hill Model	0.0514 to 0.0920	0.008	1/d for D; noise in q_M in $LCH_4/(L \cdot d)$
AM2 Model	0.2720 to 0.4420	0.5	1/d for D; noise in q_M in $mmol/(L \cdot d)$

Table 9: Anaerobic Digestion Model Base Configurations (Hill and AM2)

Parameter	Hill Model Base	AM2 Model Base	Units / Notes
Step Duration Factor	2	2	Used in Step Duration = Factor / $D_{\max}$
Sampling Rate Factor	30	40	Sampling Rate = $1 / (D_{\max} \times \text{Factor})$
Window Size	6	6	For the LSTM-B architecture
Hidden Units (for LSTM)	3	3	Dimension of the hidden state vectors h and c

E. NARX Experiment Figures

a. Experiment A – Single-Step Excitation

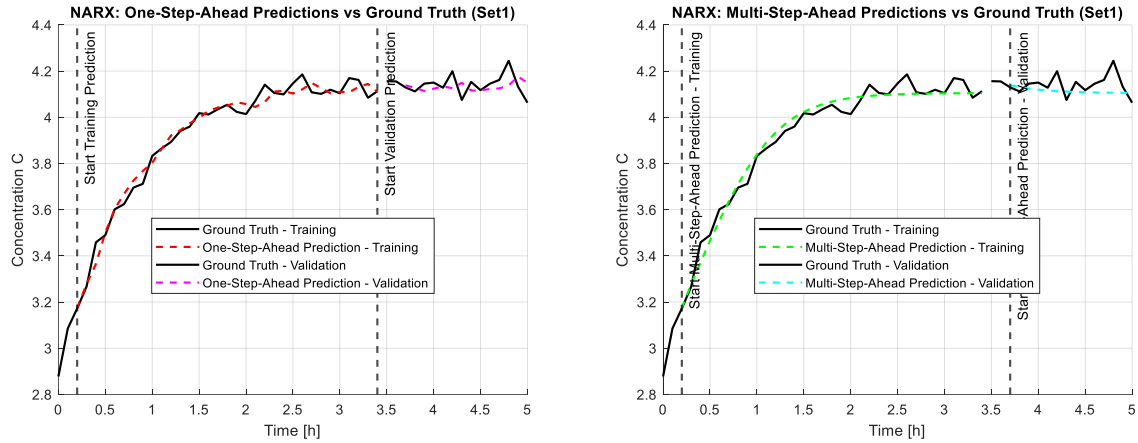


Fig. 47: Predictions for Set 1 (One-Step: left, Multi-Step: right) – Single-Step Training

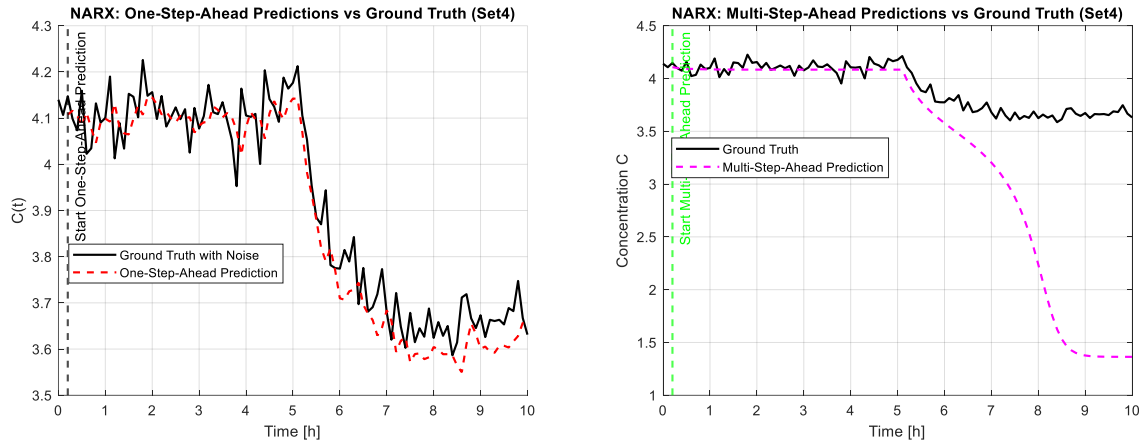


Fig. 48: Predictions for Set 4 (One-Step: left, Multi-Step: right) – Single-Step Training

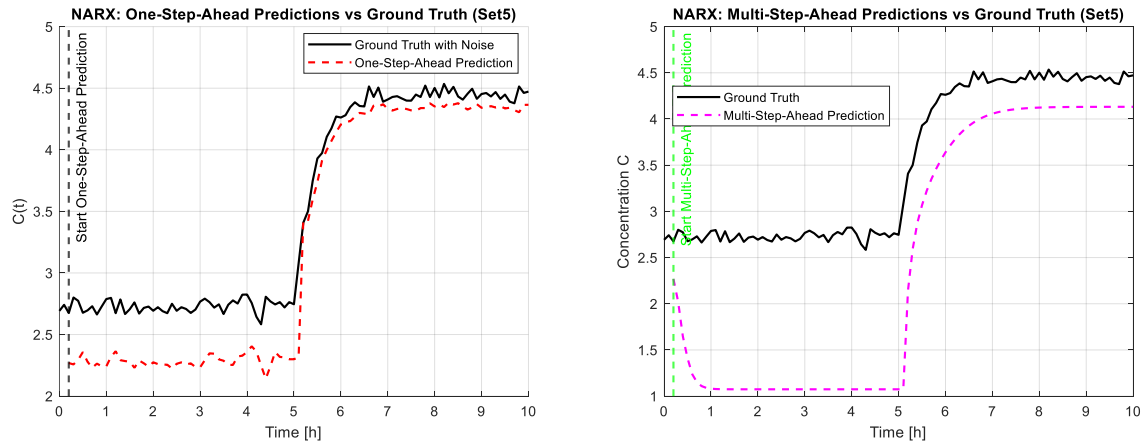


Fig. 49: Predictions for Set 5 (One-Step: left, Multi-Step: right) – Single-Step Training

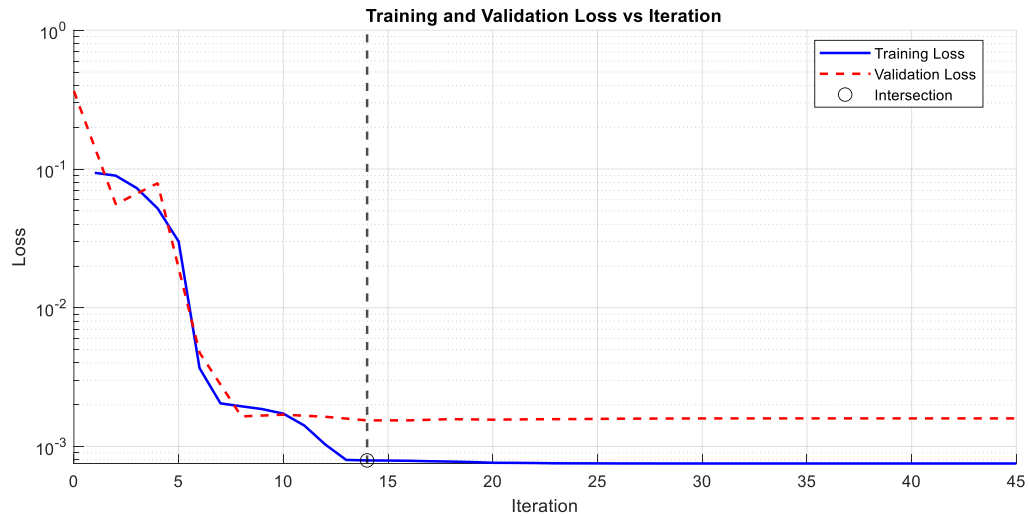


Fig. 50: Training and validation loss curves over 45 iterations – Single-Step Training

b. Experiment B – Multiple-Step Excitation (15-step configuration)

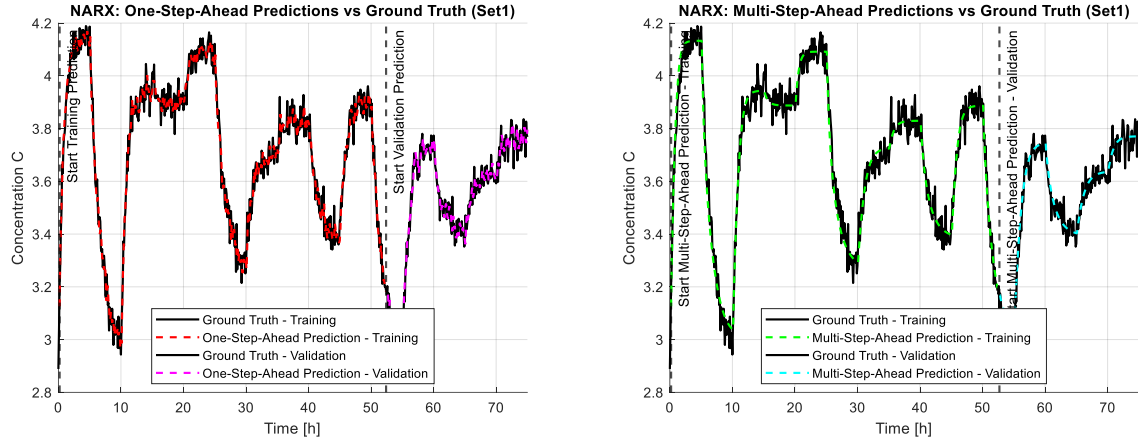


Fig. 51: Predictions for Set 1 (One-Step: left, Multi-Step: right) – 15-Step Training

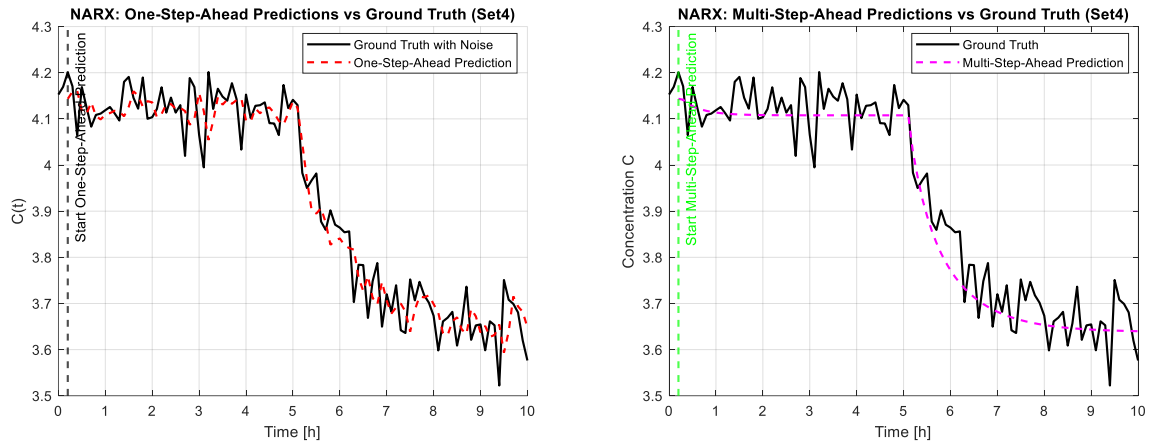


Fig. 52: Predictions for Set 4 (One-Step: left, Multi-Step: right) – 15-Step Training

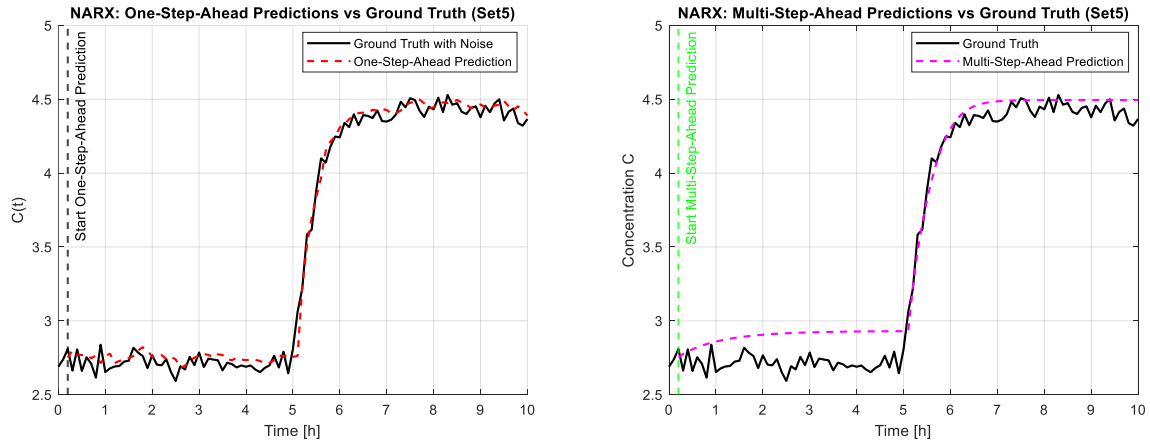


Fig. 53: Predictions for Set 5 (One-Step: left, Multi-Step: right) – 15-Step Training



Fig. 54: Training and validation loss curves over 80 iterations – 15-Step Training

c. Experiment C – Sensitivity Analysis

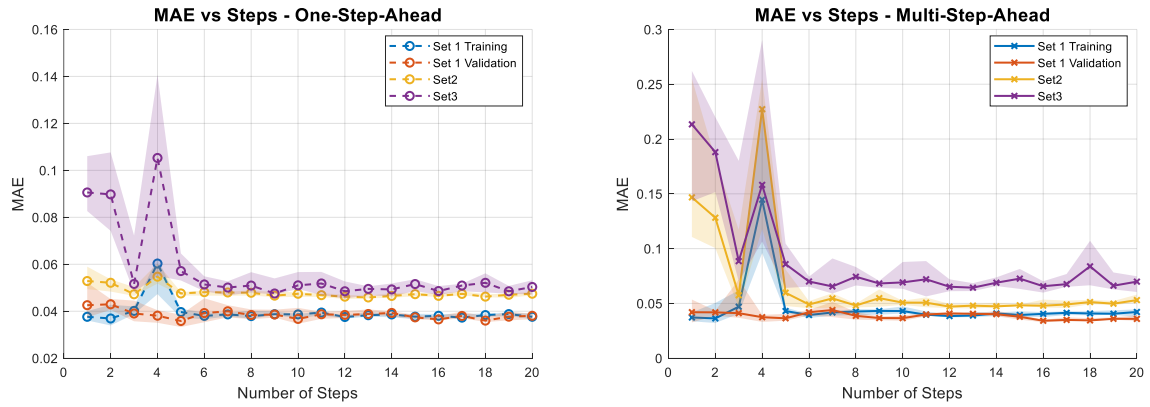


Fig. 55: MAE vs. Forecast Steps (One-Step: left, Multi-Step: right)

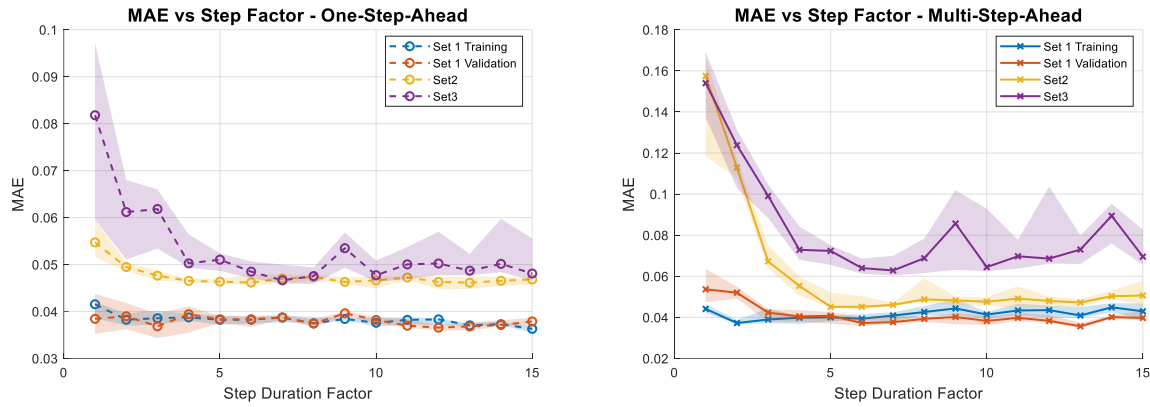


Fig. 56: MAE vs. Step Duration Factor (One-Step: left, Multi-Step: right).

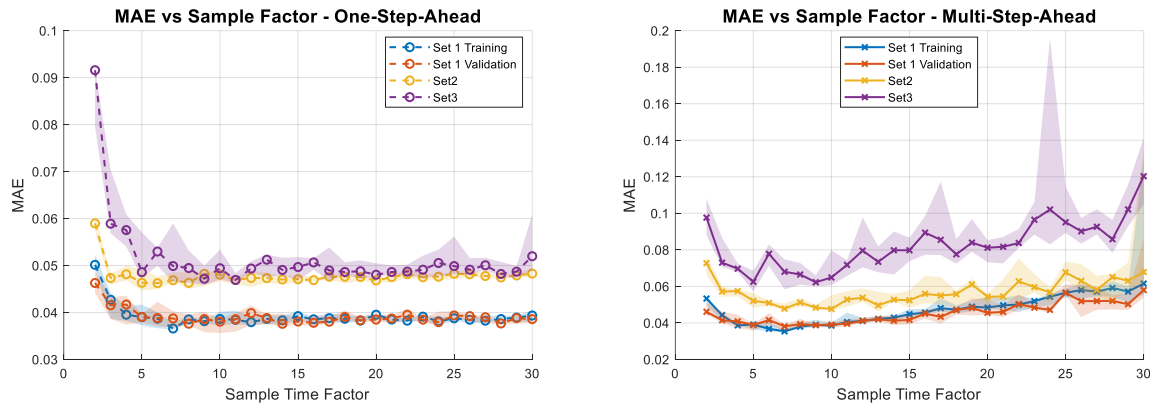


Fig. 57: MAE vs. Sampling Rate Factor (One-Step: left, Multi-Step: right).

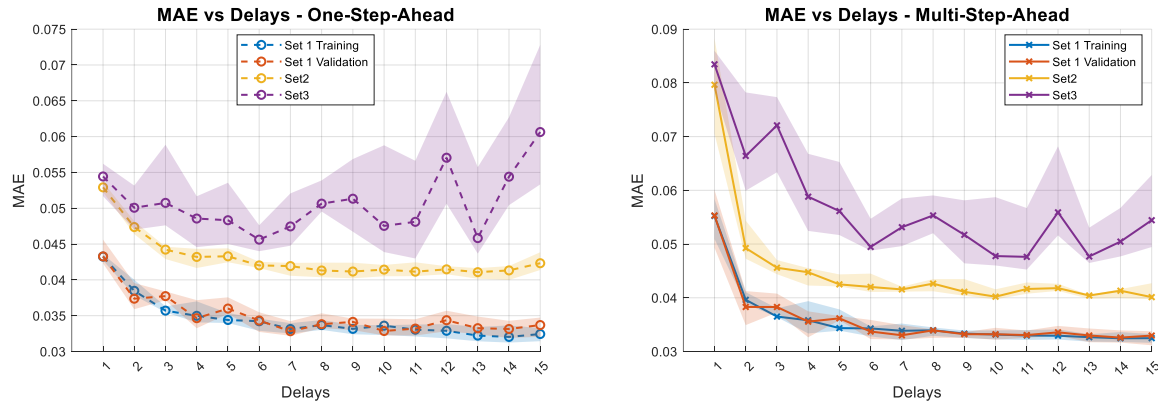


Fig. 58: MAE vs. Number of Delays (One-Step: left, Multi-Step: right)

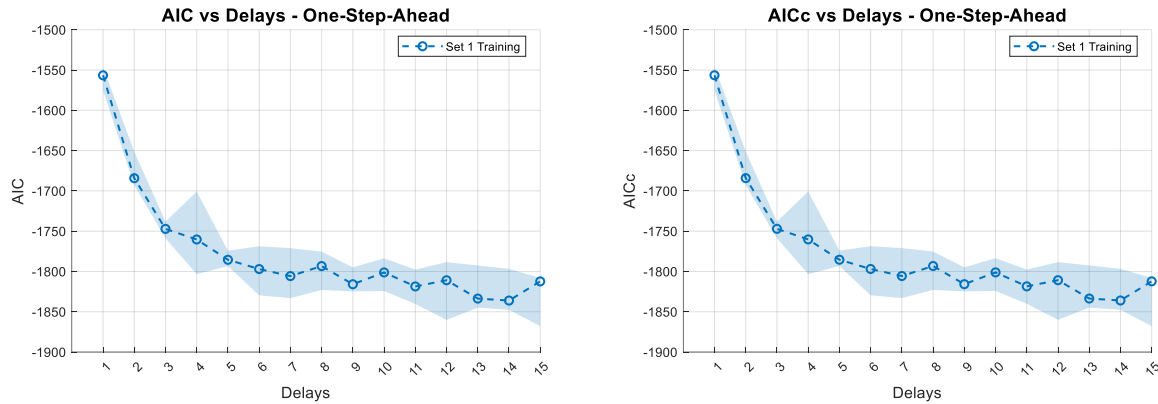


Fig. 59: AIC vs. Delays (left), AICc vs. Delays (right) – One-Step

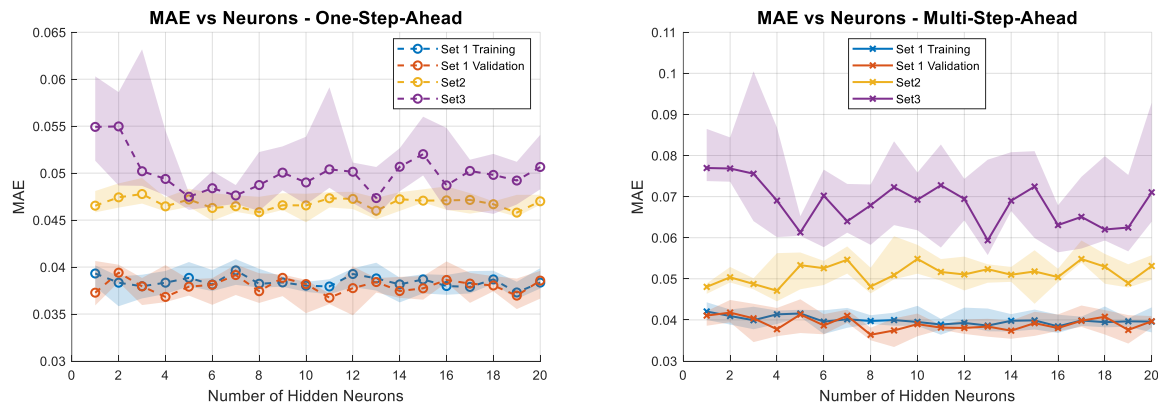


Fig. 60: MAE vs. Hidden Units (One-Step: left, Multi-Step: right)

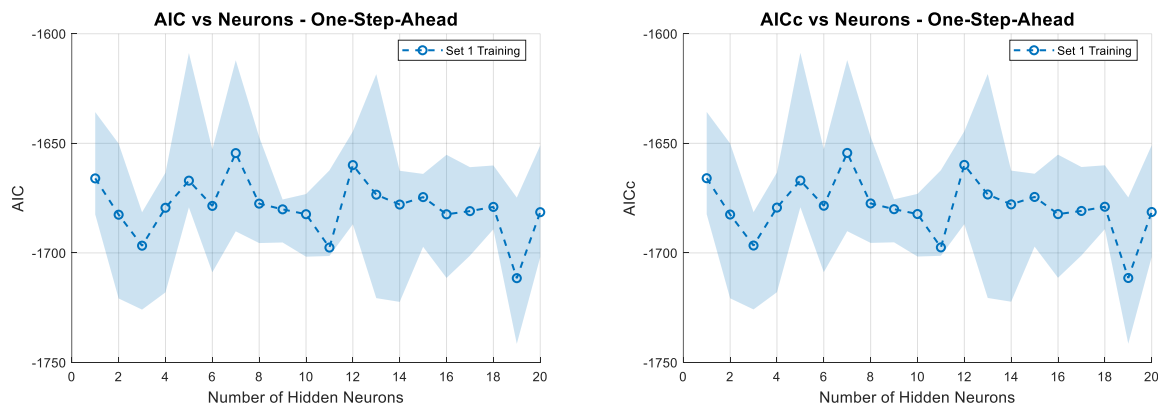


Fig. 61: AIC vs. Hidden Units (left), AICc vs. Hidden Units (right) – One-Step

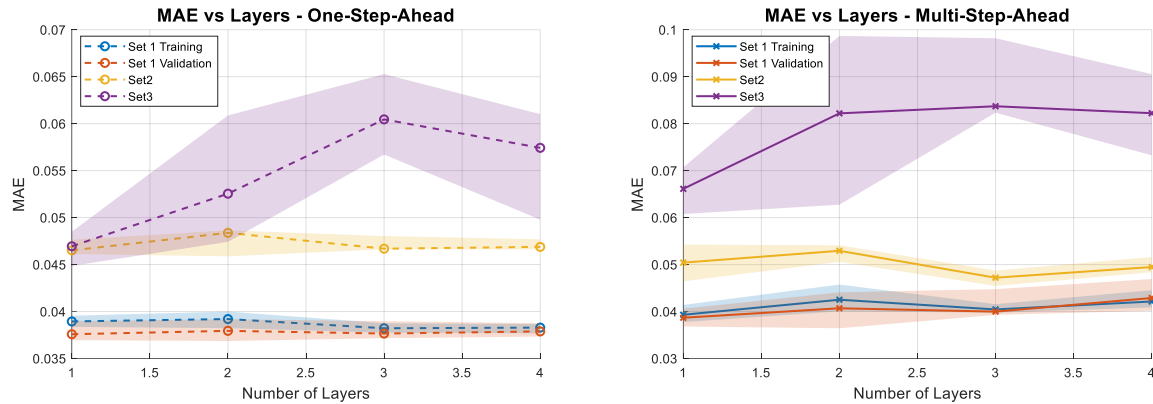


Fig. 62: MAE vs. Number of Layers (One-Step: left, Multi-Step: right)

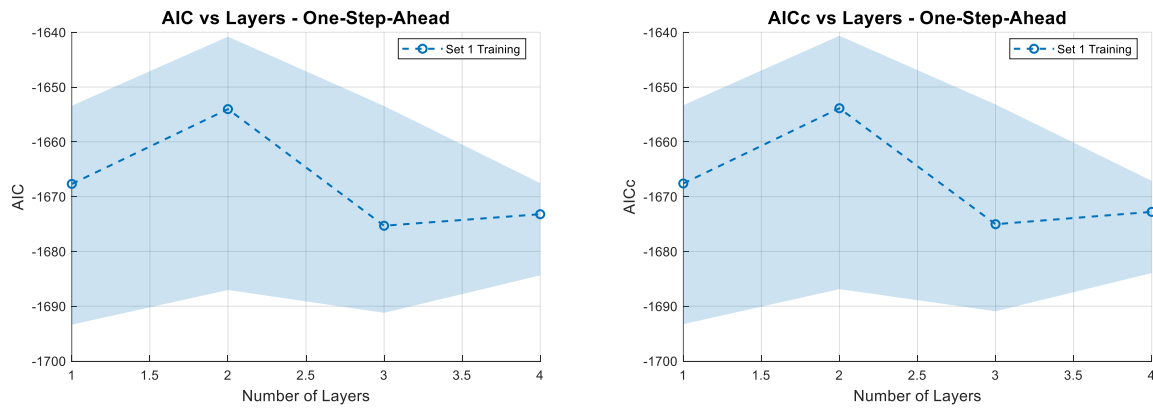


Fig. 63: AIC vs. Layers (left), AICc vs. Layers (right) – One-Step

d. Experiment D– Minimal Configuration Selection

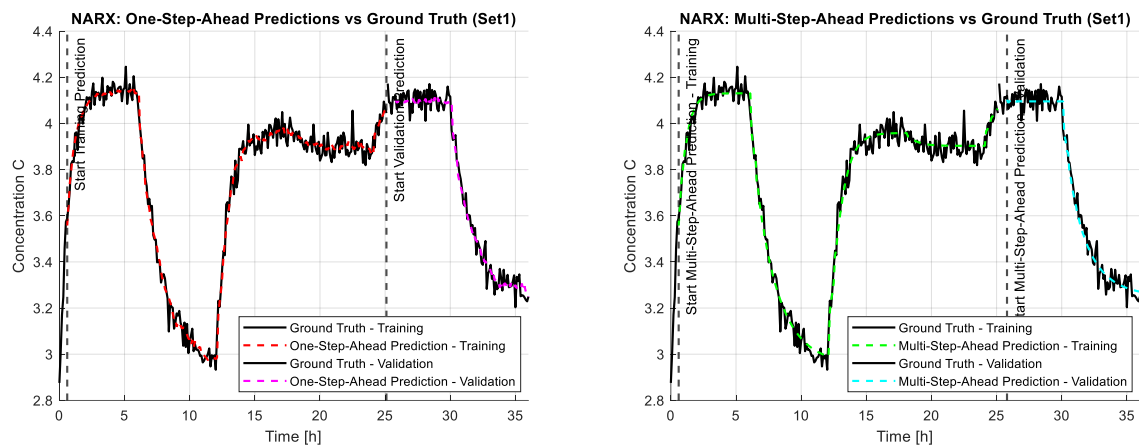


Fig. 64: Predictions for Set 1 (One-Step: left, Multi-Step: right) – Minimal Config

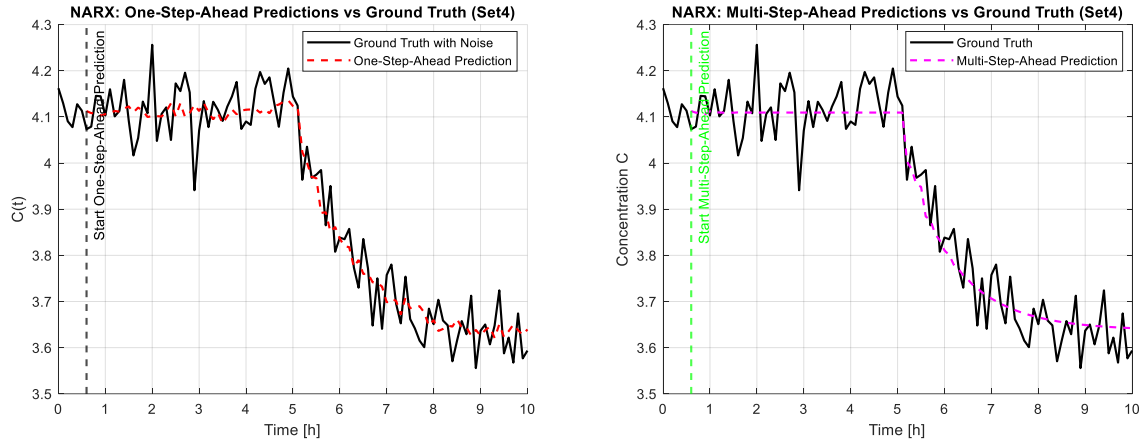


Fig. 65: Predictions for Set 4 (One-Step: left, Multi-Step: right) – Minimal Config

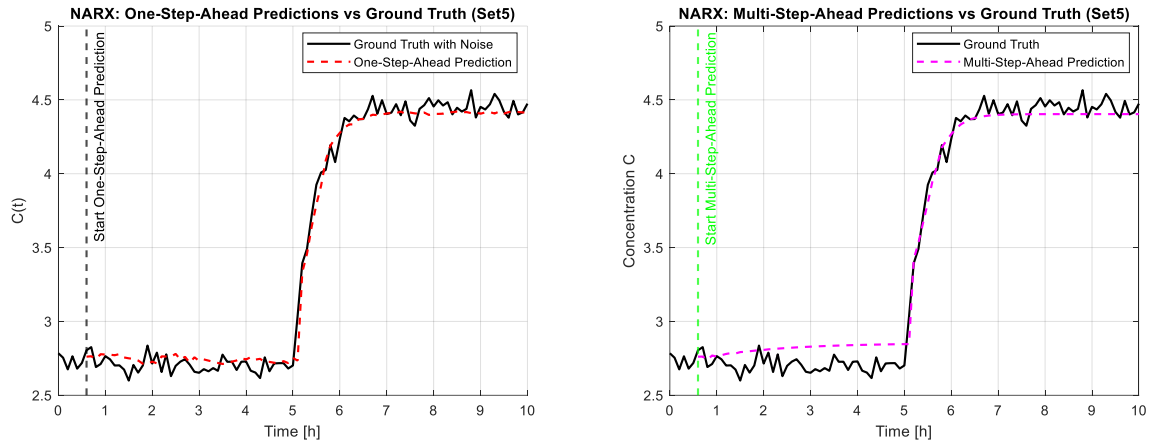


Fig. 66: Predictions for Set 5 (One-Step: left, Multi-Step: right) – Minimal Config

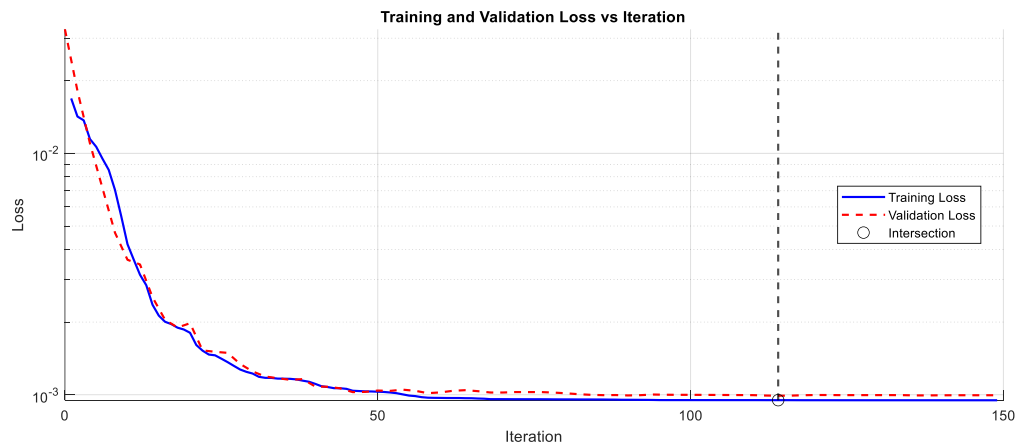


Fig. 67: Training and validation loss curves over 150 iterations – Minimal Config

F. LSTM-A Experiment Figures

e. Experiment A – Single-Step Excitation

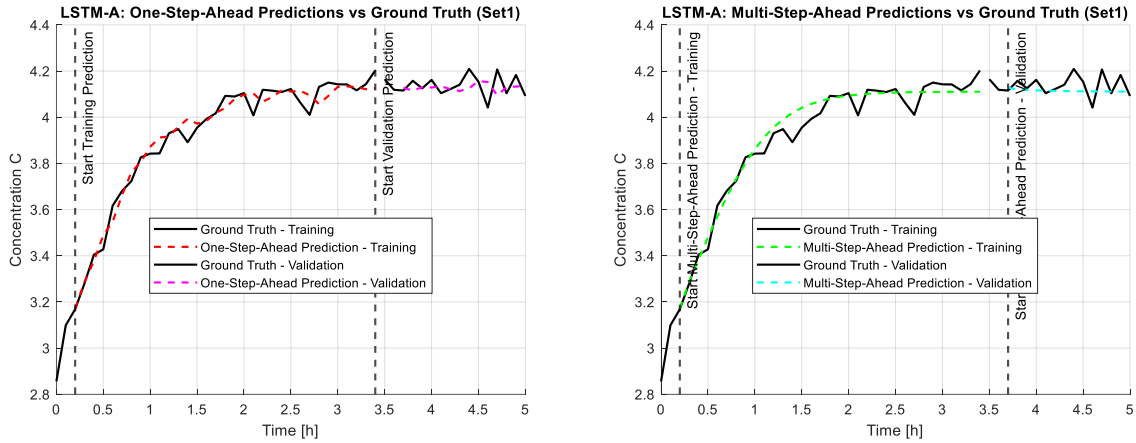


Fig. 68: Predictions for Set 1 (One-Step: left, Multi-Step: right) – Single-Step

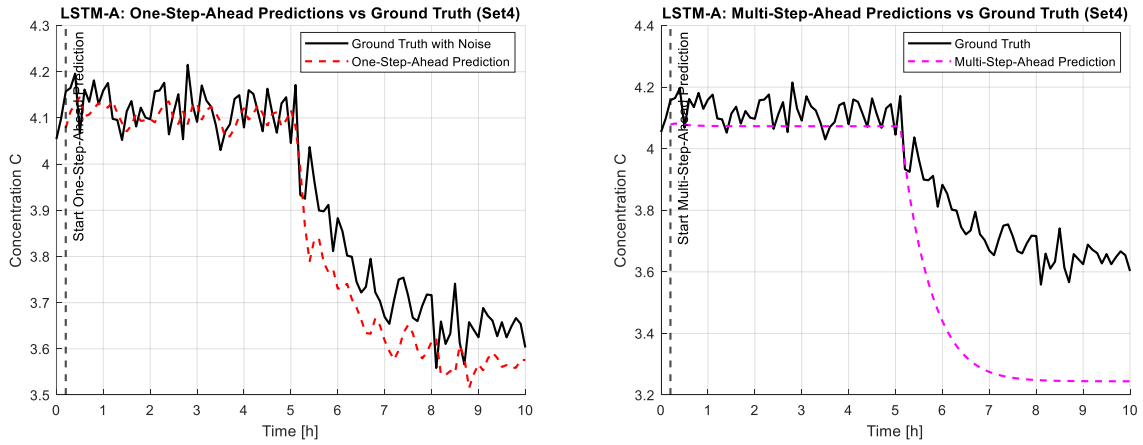


Fig. 69: Predictions for Set 4 (One-Step: left, Multi-Step: right) – Single-Step

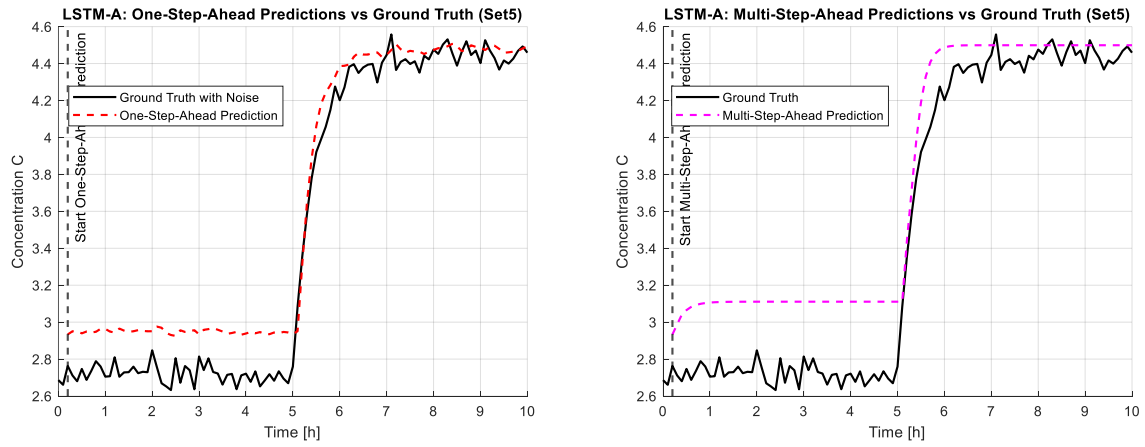


Fig. 70: Predictions for Set 5 (One-Step: left, Multi-Step: right) – Single-Step

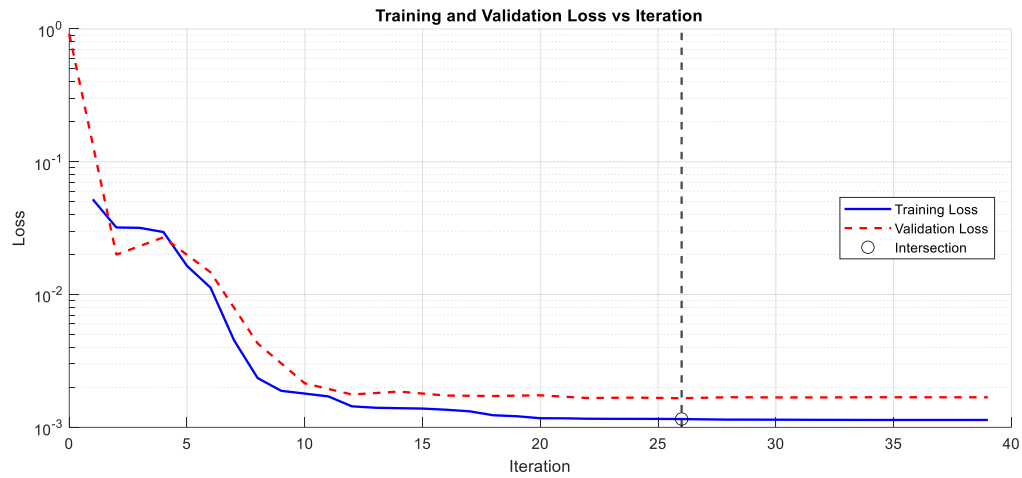


Fig. 71: Training and validation loss curves over 60 iterations – Single-Step

f. Experiment B – Multiple-Step Excitation (15-step configuration)

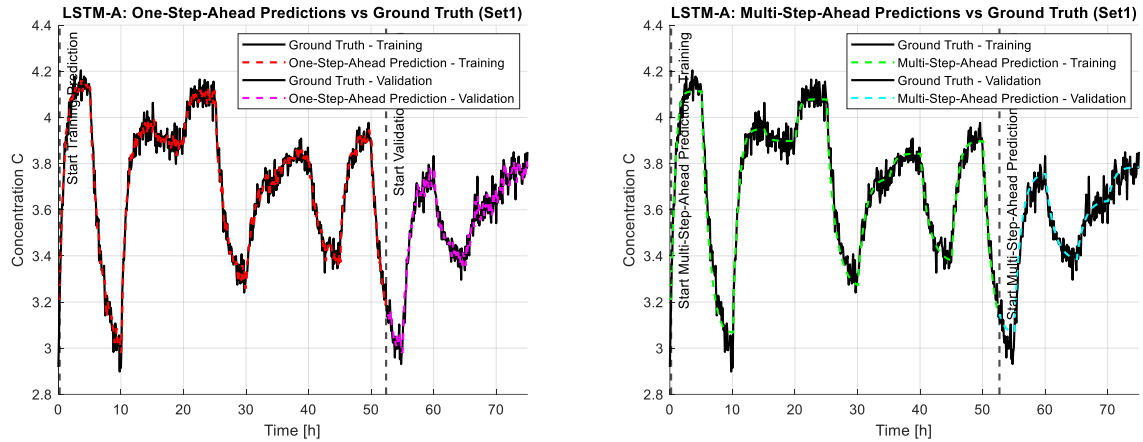


Fig. 72: Predictions for Set 1 (One-Step: left, Multi-Step: right) – 15-Step

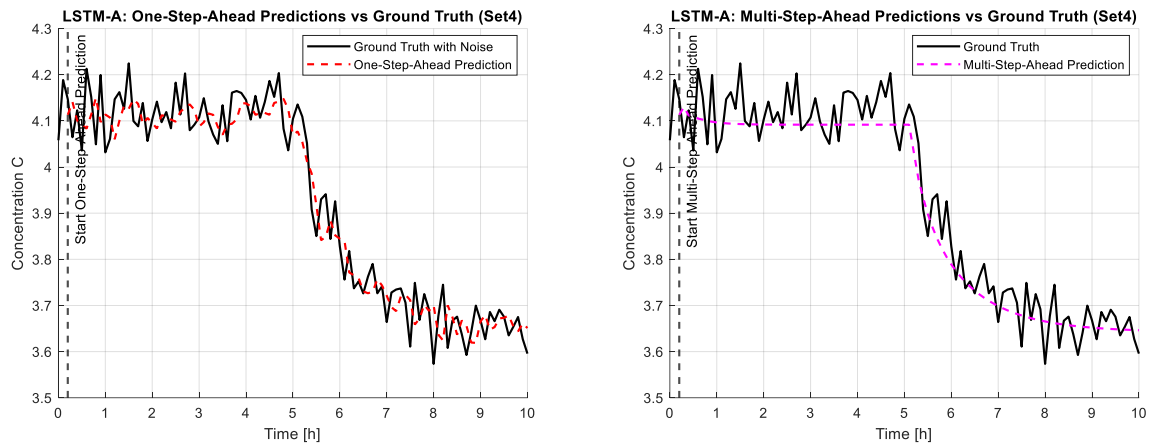


Fig. 73: Predictions for Set 4 (One-Step: left, Multi-Step: right) – 15-Step

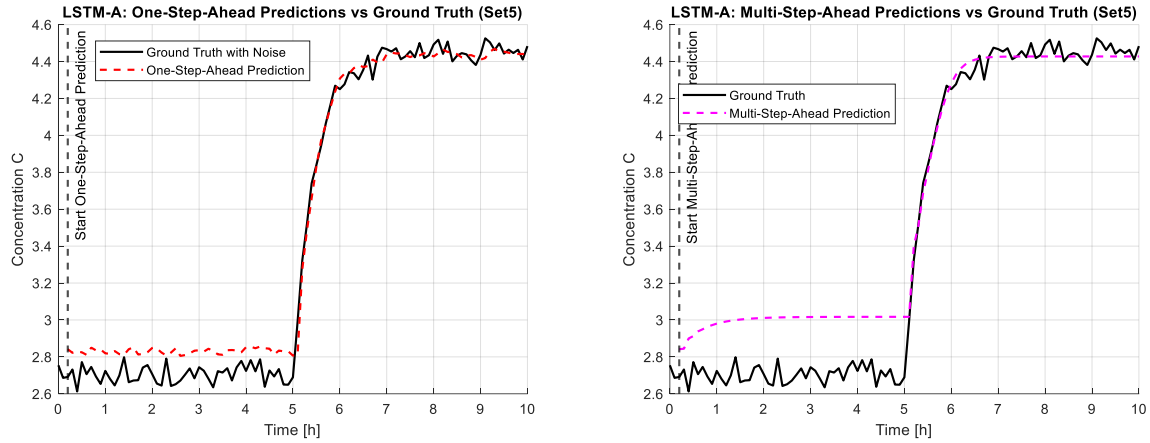


Fig. 74: Predictions for Set 5 (One-Step: left, Multi-Step: right) – 15-Step

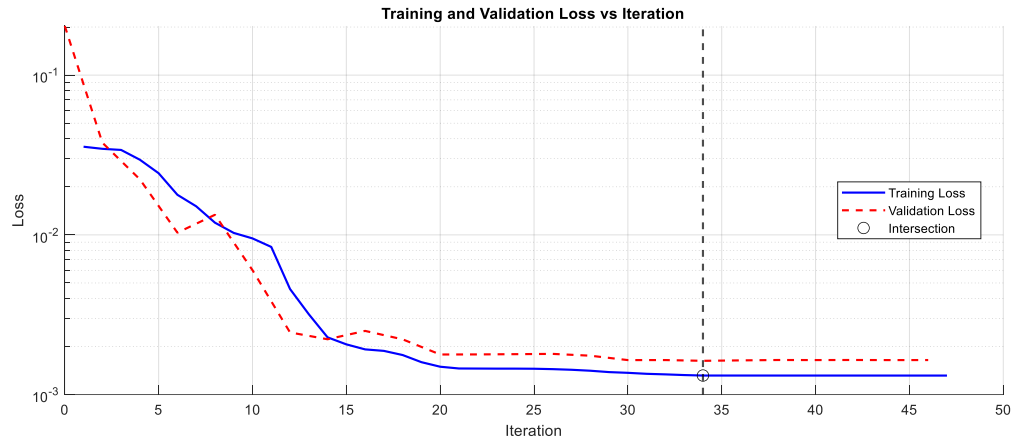


Fig. 75: Training and Validation Loss curves over 50 iterations – 15-Step

g. Experiment C – Sensitivity Analysis

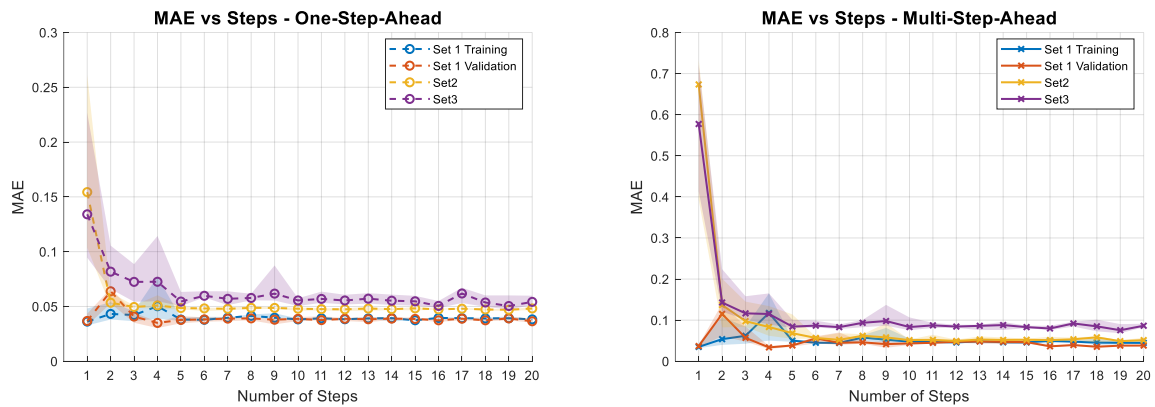


Fig. 76: MAE vs. Forecast Steps (One-Step: left, Multi-Step: right)

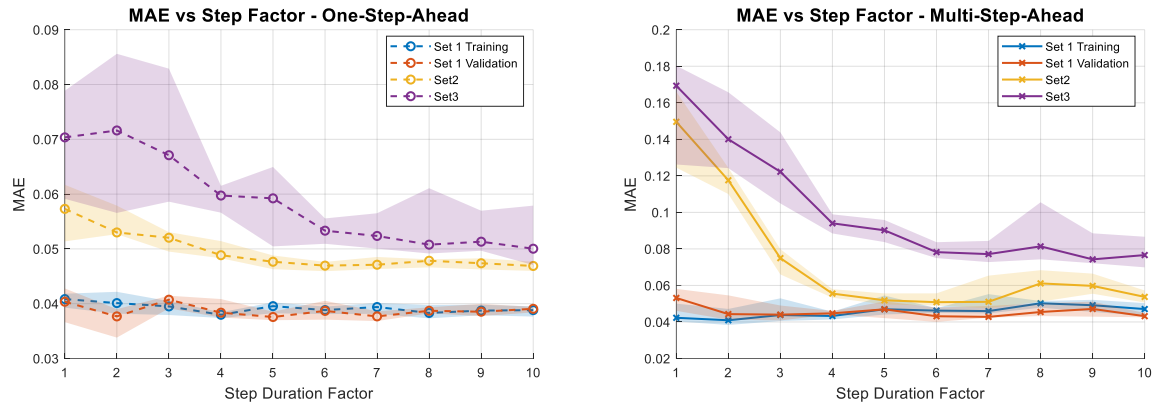


Fig. 77: MAE vs. Step Duration Factor (One-Step: left, Multi-Step: right)

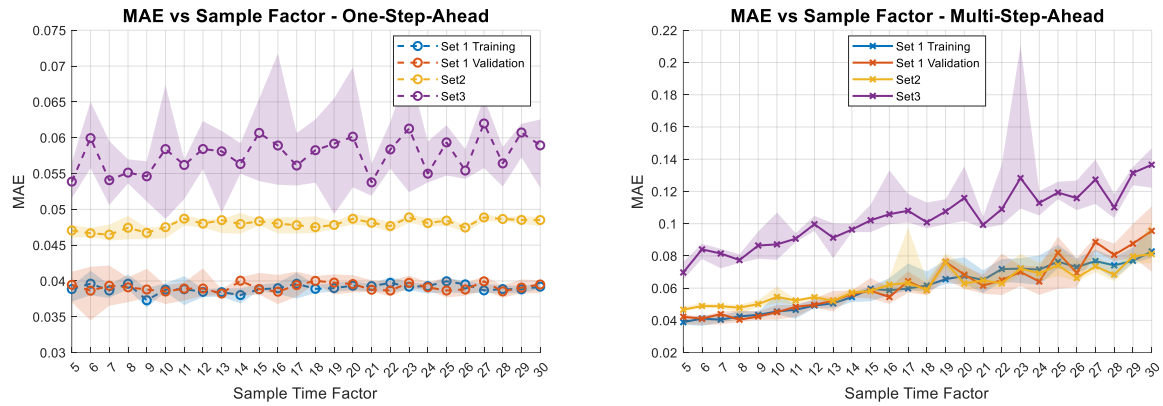


Fig. 78: MAE vs. Sampling Rate Factor (One-Step: left, Multi-Step: right)

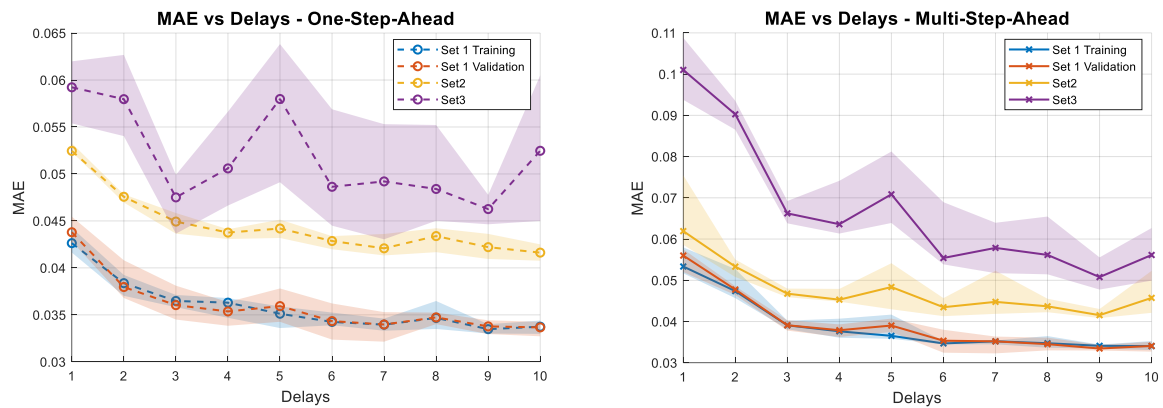


Fig. 79: MAE vs. Number of Delays (One-Step: left, Multi-Step: right)

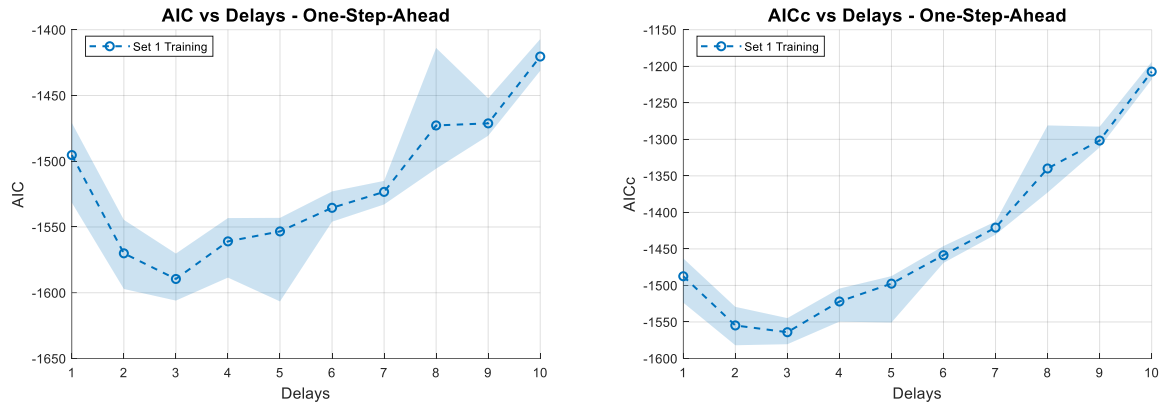


Fig. 80: AIC vs. Delays (left), AICc vs. Delays (right) – One-Step

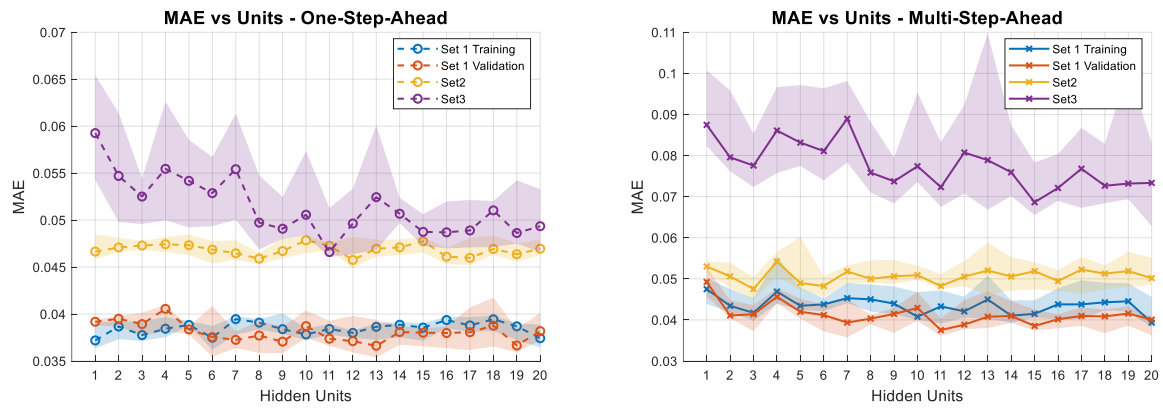


Fig. 81: MAE vs. Hidden Units (One-Step: left, Multi-Step: right)

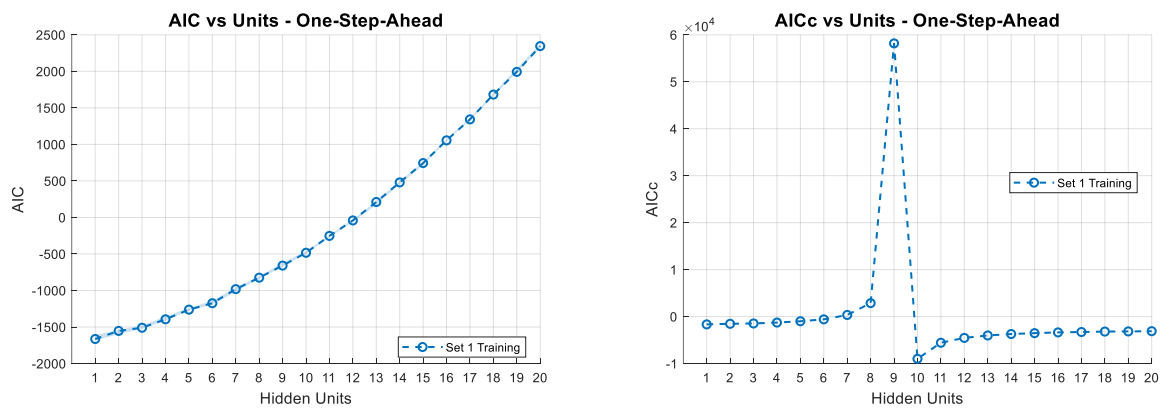


Fig. 82: AIC vs. Hidden Units (left), AICc vs. Hidden Units (right) – One-Step

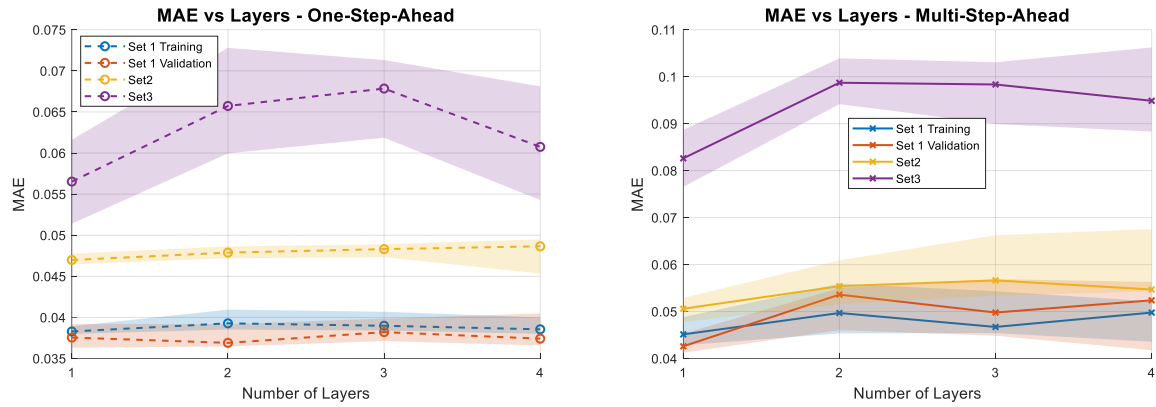


Fig. 83: MAE vs. Layers (One-Step: left, Multi-Step: right)

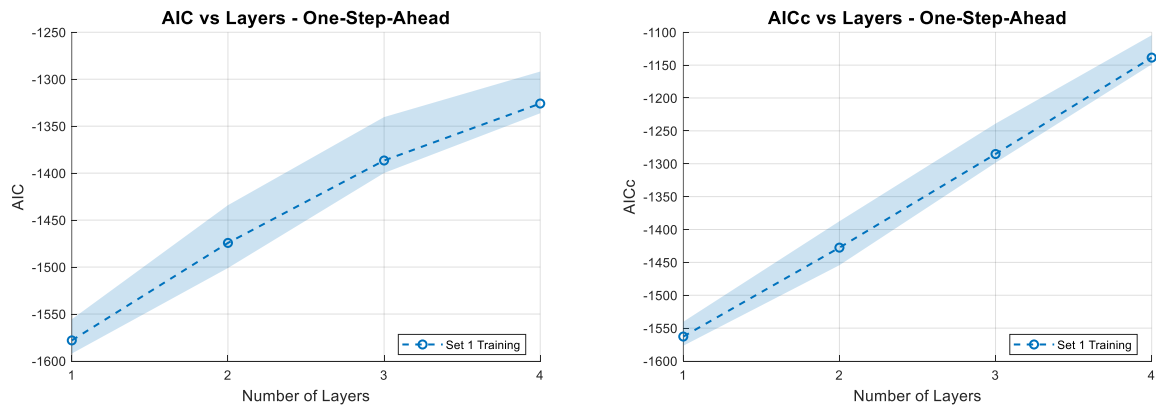


Fig. 84: AIC vs. Layers (left), AICc vs. Layers (right) – One-Step

h. Experiment D– Minimal Configuration Selection

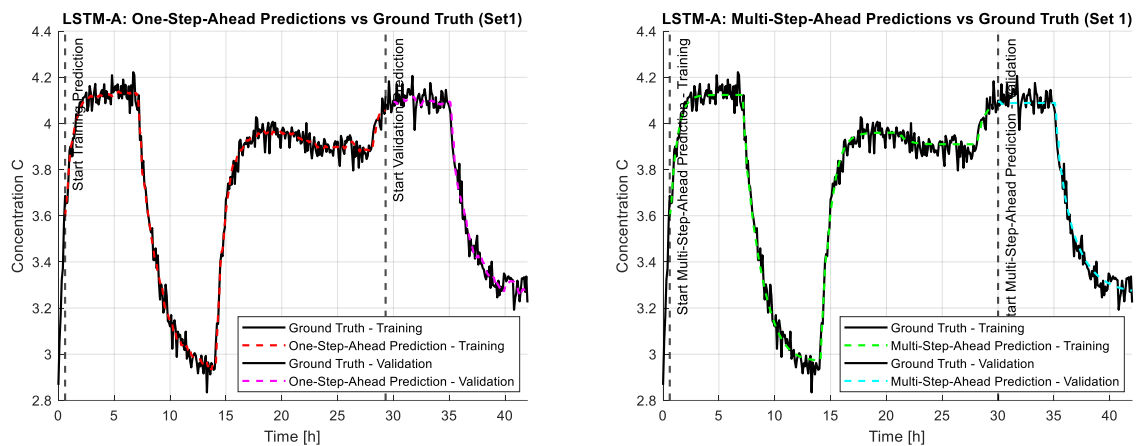


Fig. 85: One-step-ahead (left) and multi-step-ahead (right) predictions for Set 1 (LSTM-A).

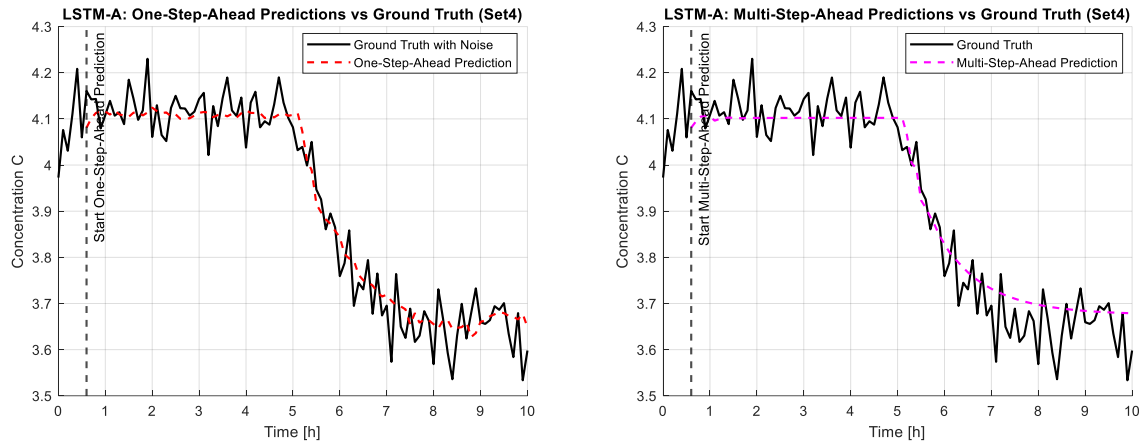


Fig. 86: One-step-ahead (left) and multi-step-ahead (right) predictions for Set 4 (LSTM-A).

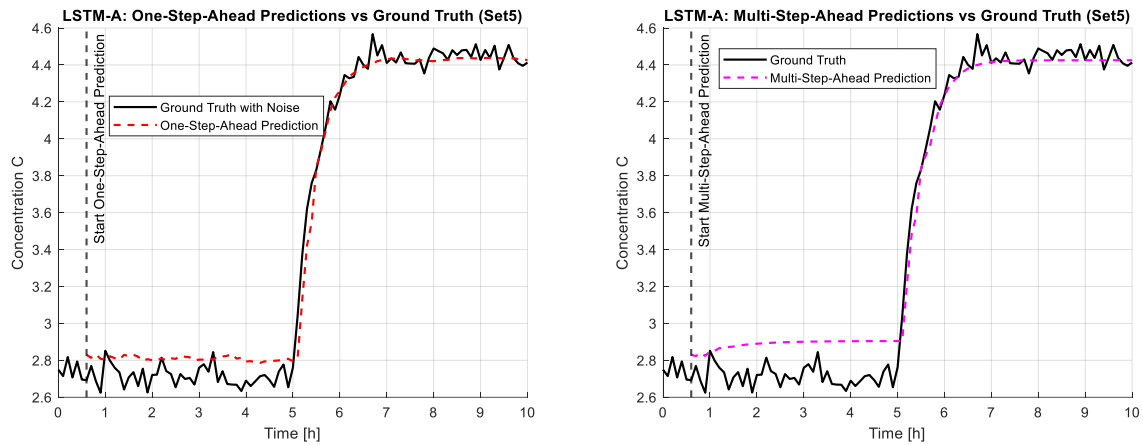


Fig. 87: One-step-ahead (left) and multi-step-ahead (right) predictions for Set 5 (LSTM-A).

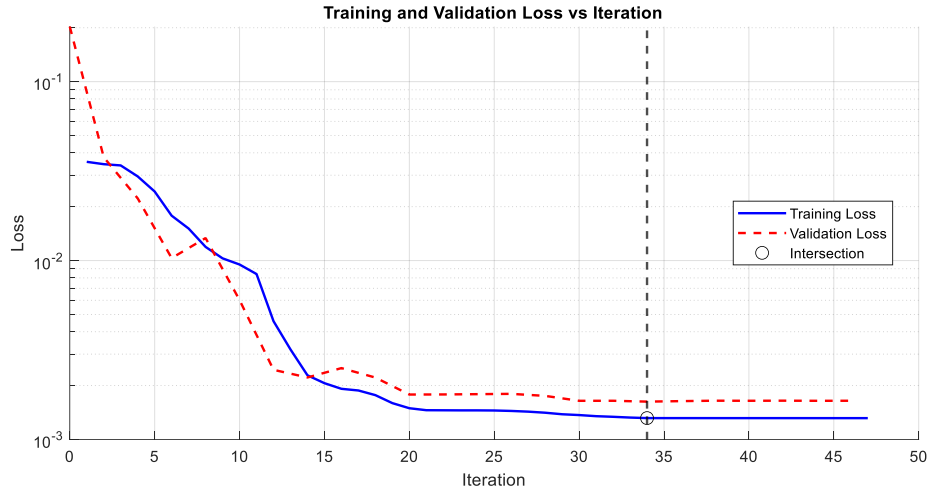


Fig. 88: Training and validation loss curves over 50 iterations (LSTM-A).

G. LSTM-B Experiment Figures

i. Experiment A – Single-Step Excitation

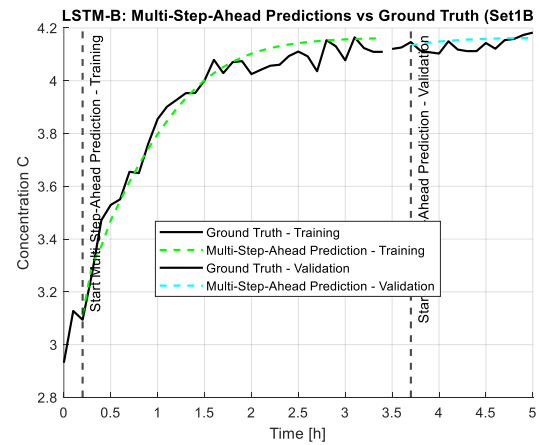
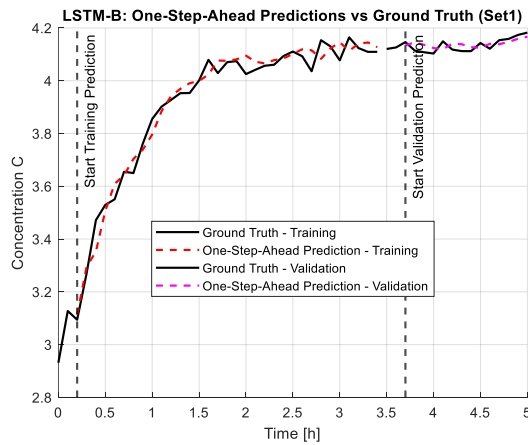


Fig. 89: One-step-ahead (left) and multi-step-ahead (right) predictions for Set 1 (LSTM-B).

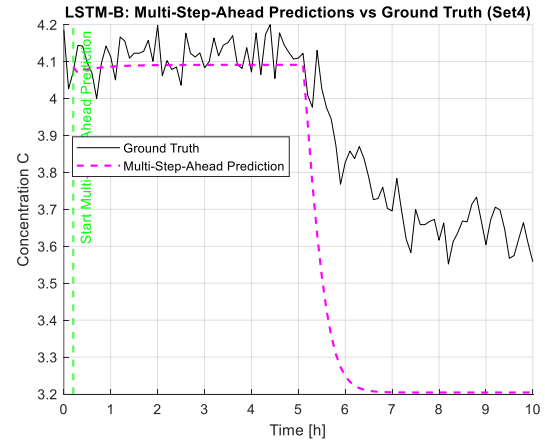
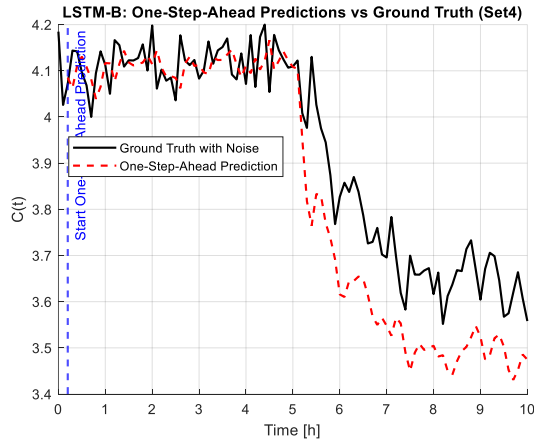


Fig. 90: One-step-ahead (left) and multi-step-ahead (right) predictions for Set 4 (LSTM-B).

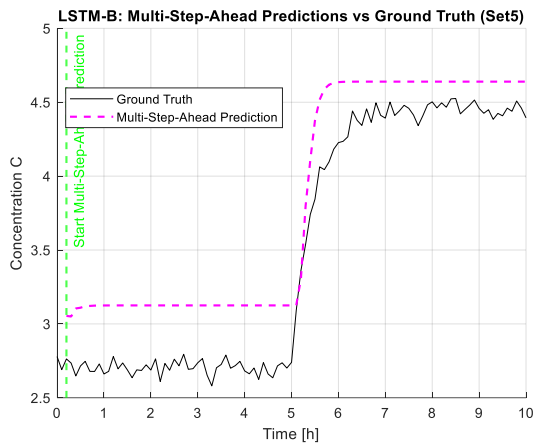
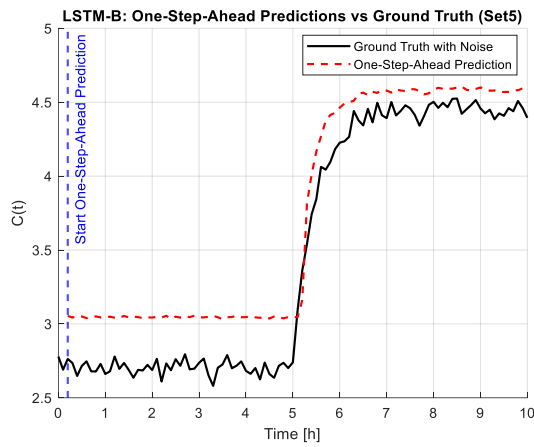


Fig. 91: One-step-ahead (left) and multi-step-ahead (right) predictions for Set 5 (LSTM-B).

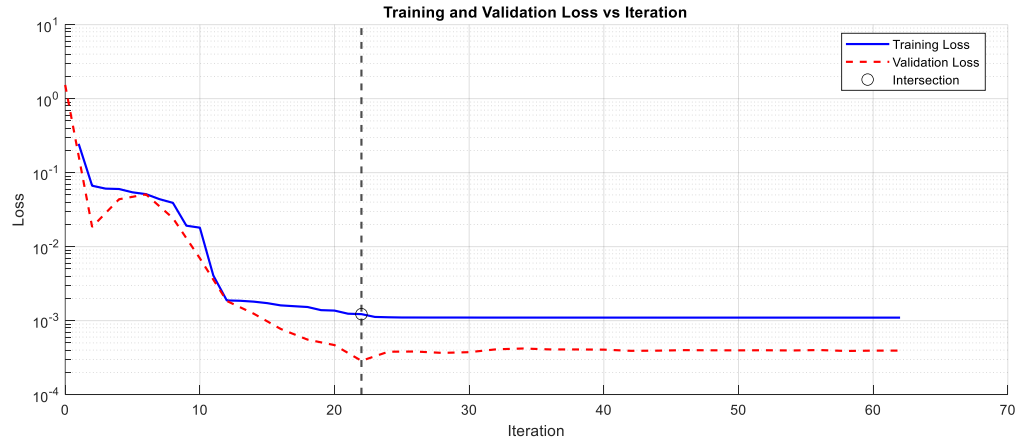


Fig. 92: Training and validation loss curves over 35 iterations (LSTM-B).

j. Experiment B – Multiple-Step Excitation (15-step configuration)

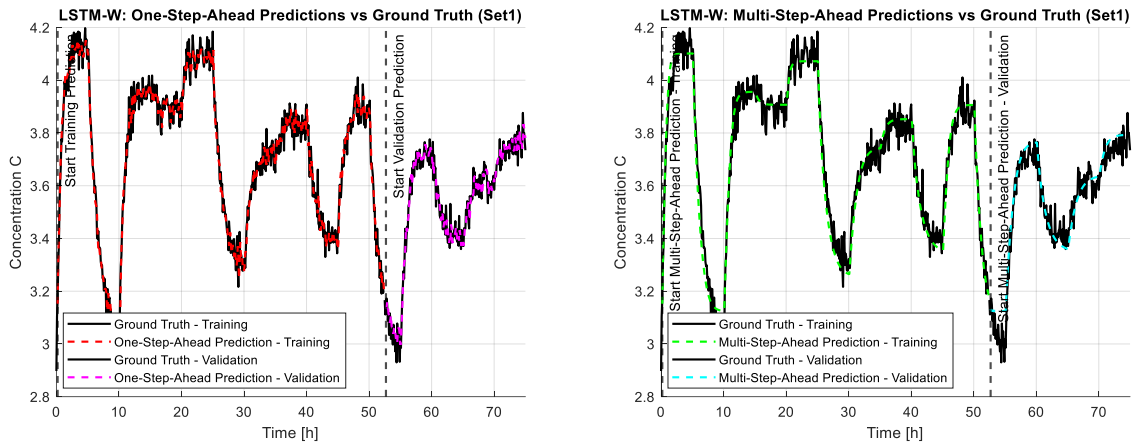


Fig. 93: One-step-ahead (left) and multi-step-ahead (right) predictions for Set 1 (AM2 Model, Single-Step Training).

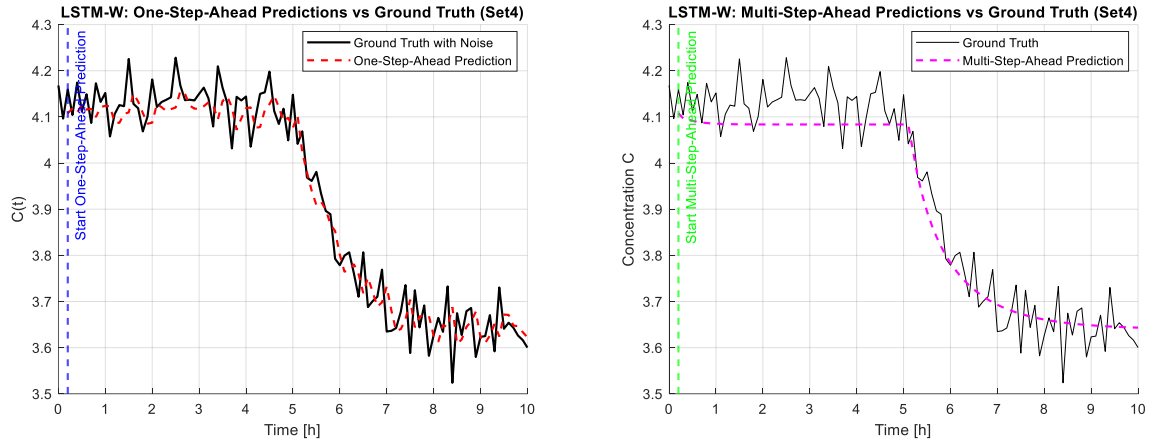


Fig. 94: One-step-ahead (left) and multi-step-ahead (right) predictions for Set 4 (AM2 Model, Single-Step Training).

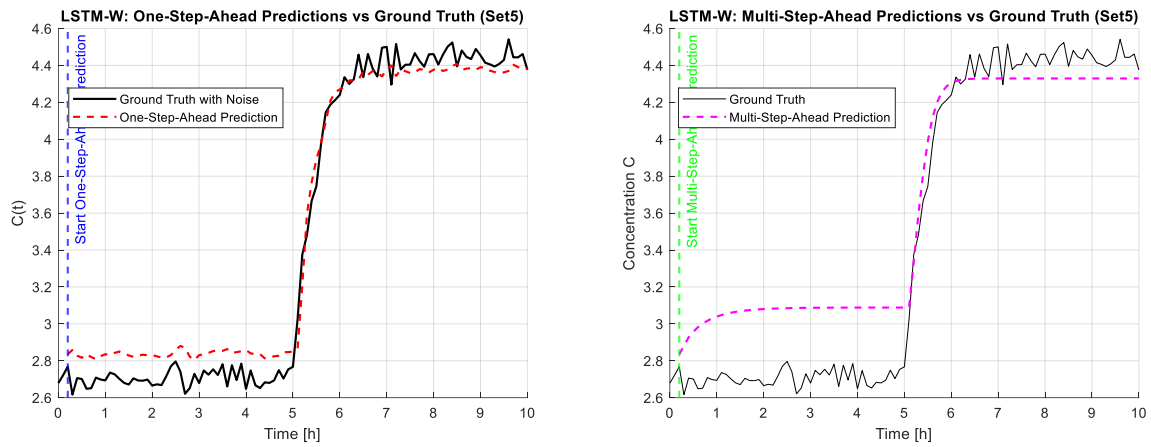


Fig. 95: One-step-ahead (left) and multi-step-ahead (right) predictions for Set 5 (AM2 Model, Single-Step Training).

k. Experiment C – Sensitivity Analysis

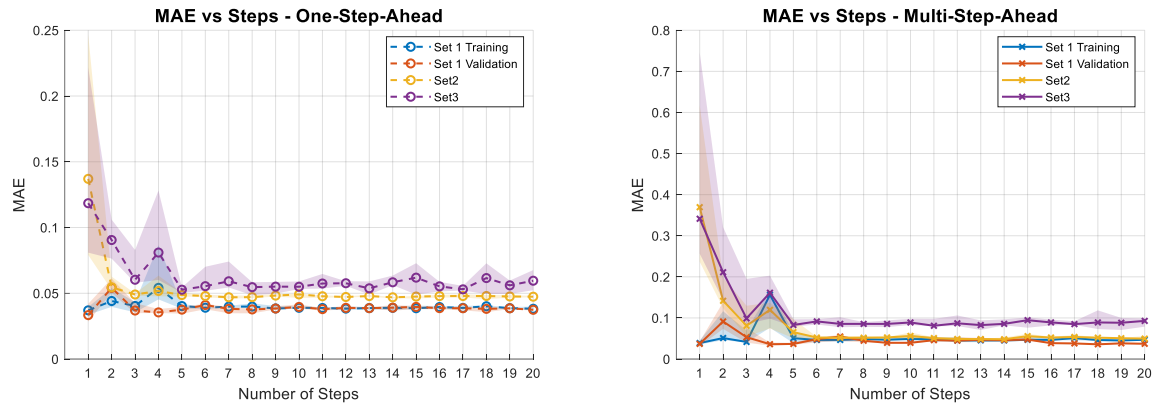


Fig. 96: MAE vs Steps (One-Step-Ahead – left, Multi-Step-Ahead – right)

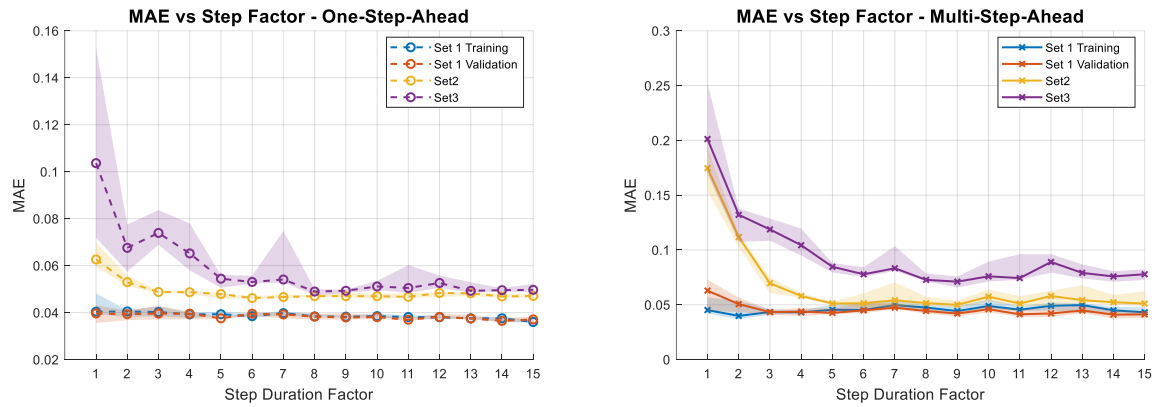


Fig. 97: MAE vs Step Factor (One-Step-Ahead – left, Multi-Step-Ahead – right)

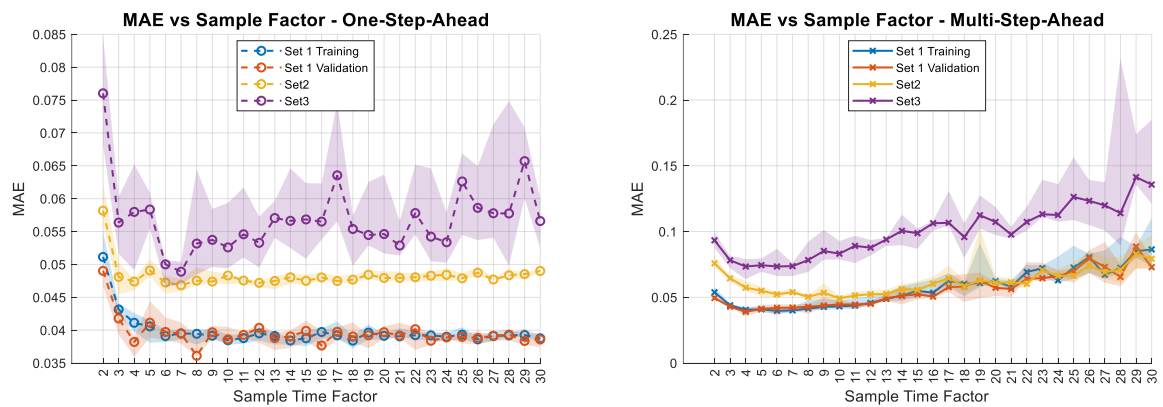


Fig. 98: MAE vs Sample Time Factor (One-Step-Ahead – left, Multi-Step-Ahead – right)

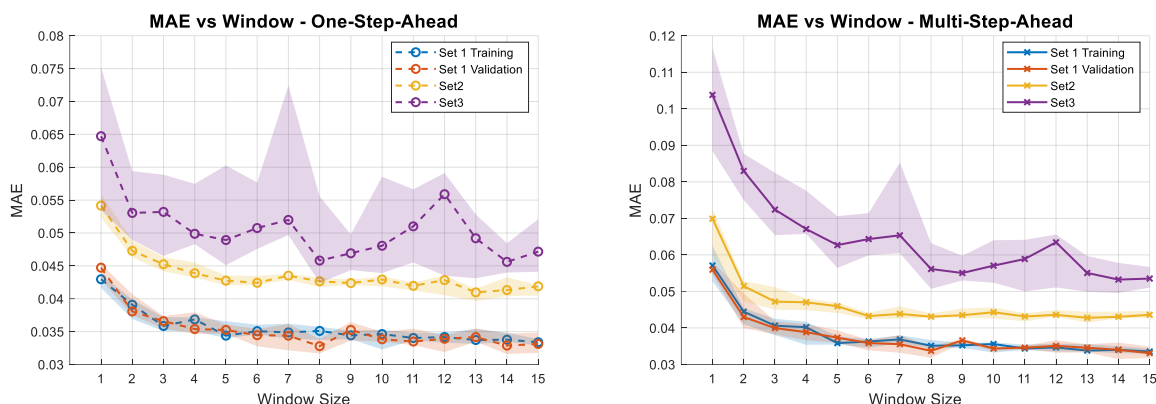


Fig. 99: MAE vs Window Size (One-Step-Ahead – left, Multi-Step-Ahead – right)

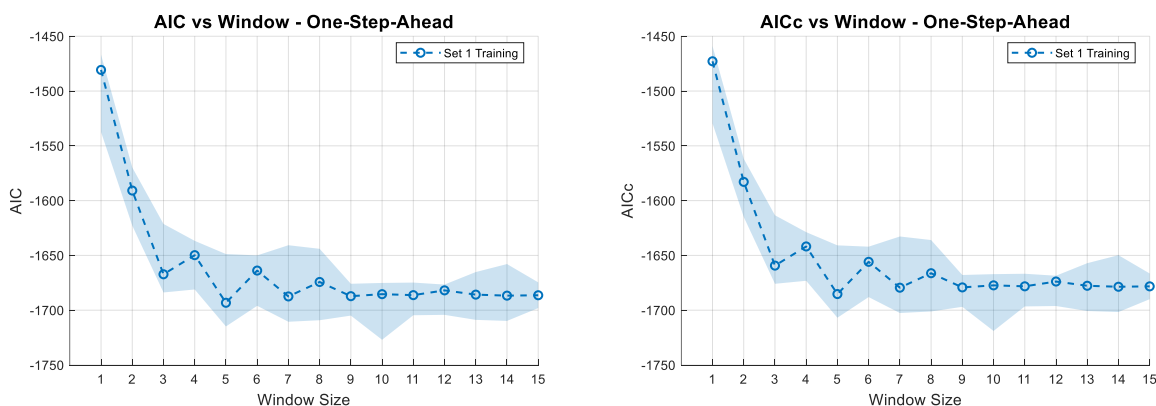


Fig. 100: AIC (left) and AICc (right) vs Window Size.

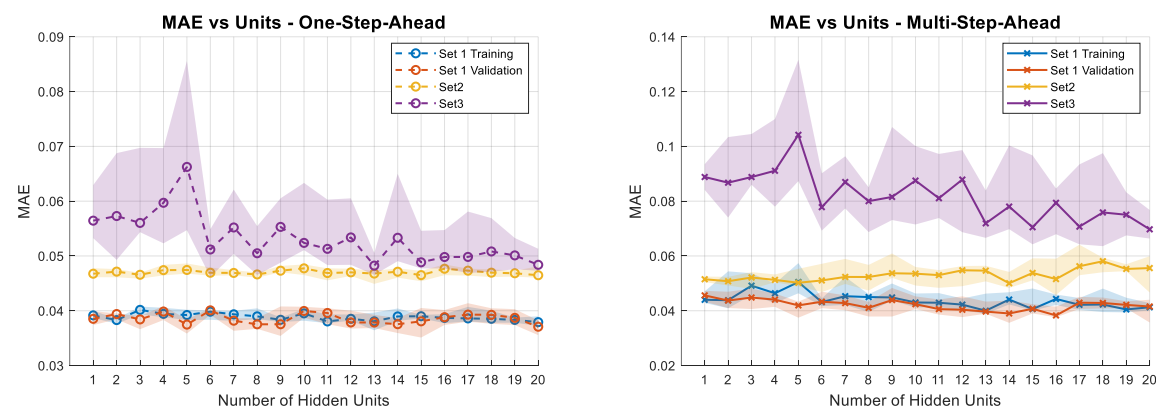


Fig. 101: MAE vs Hidden Units (One-Step-Ahead – left, Multi-Step-Ahead – right)

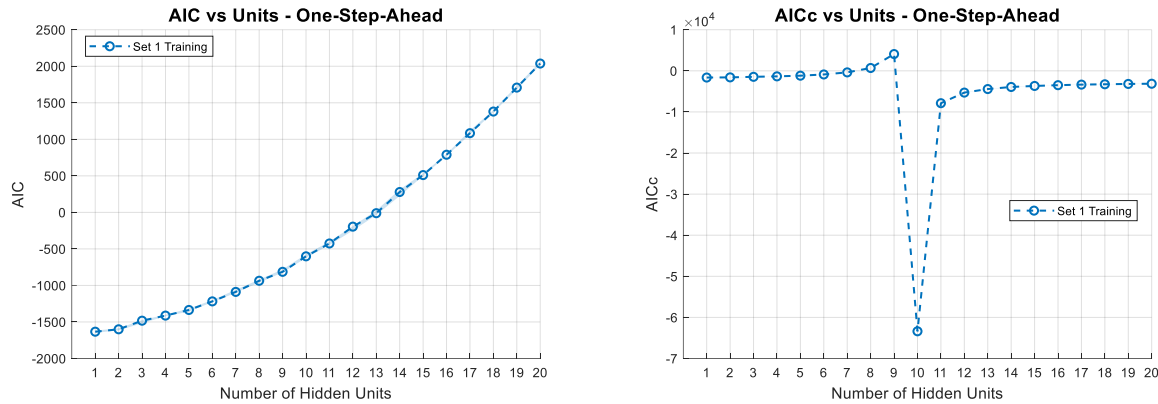


Fig. 102: AIC (left) and AICc (right) vs. number of hidden units for one-step-ahead predictions.

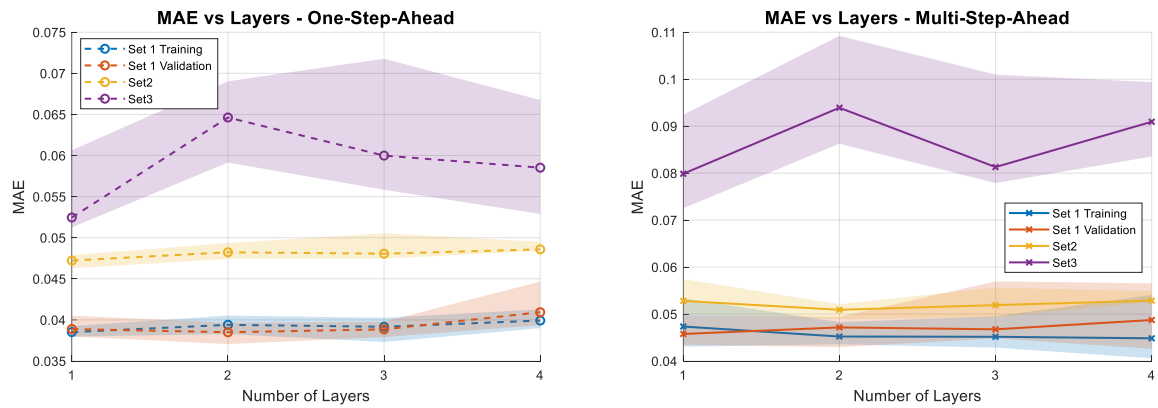


Fig. 103: MAE vs. number of layers for one-step-ahead (left) and multi-step-ahead (right) predictions.

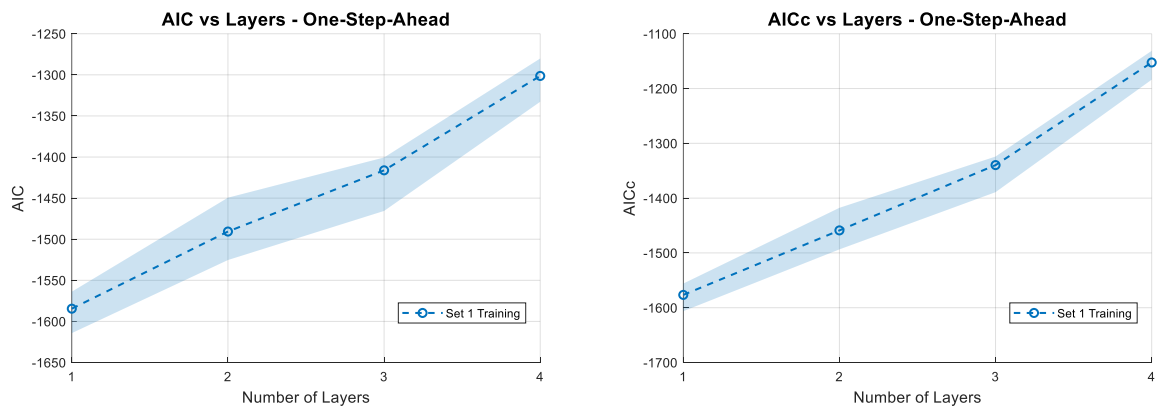


Fig. 104: AIC (left) and AICc (right) vs. number of layers for one-step-ahead predictions.

1. Experiment D– Minimal Configuration Selection

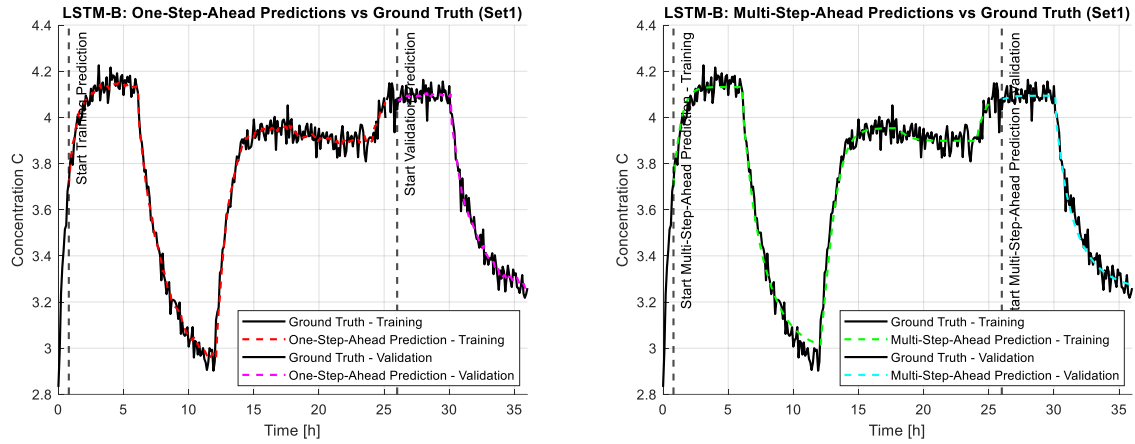


Fig. 105: One-step-ahead (left) and multi-step-ahead (right) predictions for Set 1 (LSTM-B, minimal configuration).

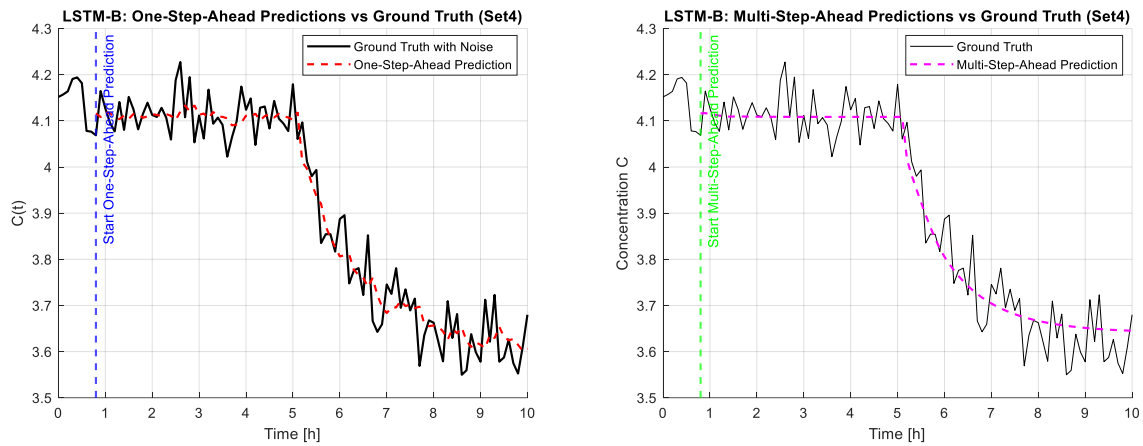


Fig. 106: One-step-ahead (left) and multi-step-ahead (right) predictions for Set 4 (LSTM-B, minimal configuration).

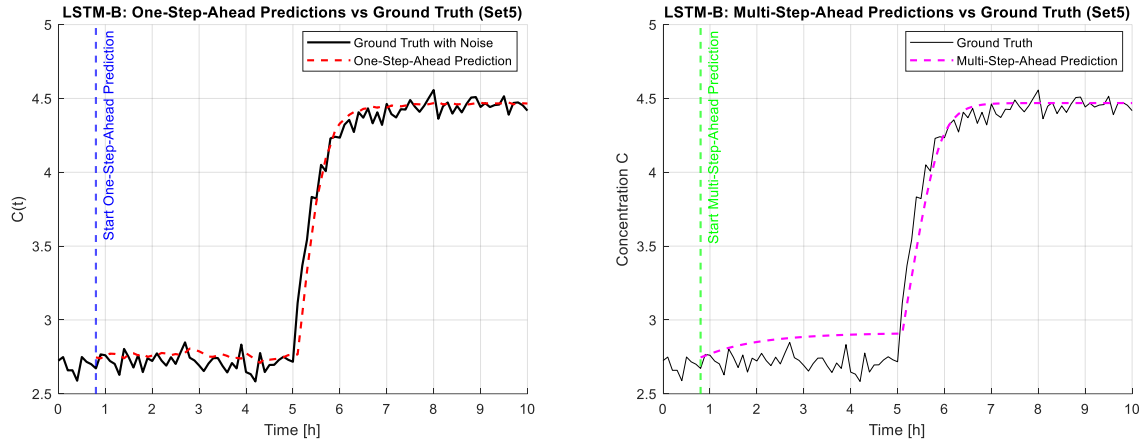


Fig. 107: One-step-ahead (left) and multi-step-ahead (right) predictions for Set 5 (LSTM-B, minimal configuration).



Fig. 108: Training and validation loss vs. iteration for the minimal LSTM-B configuration.

H. Benchmark Comparison

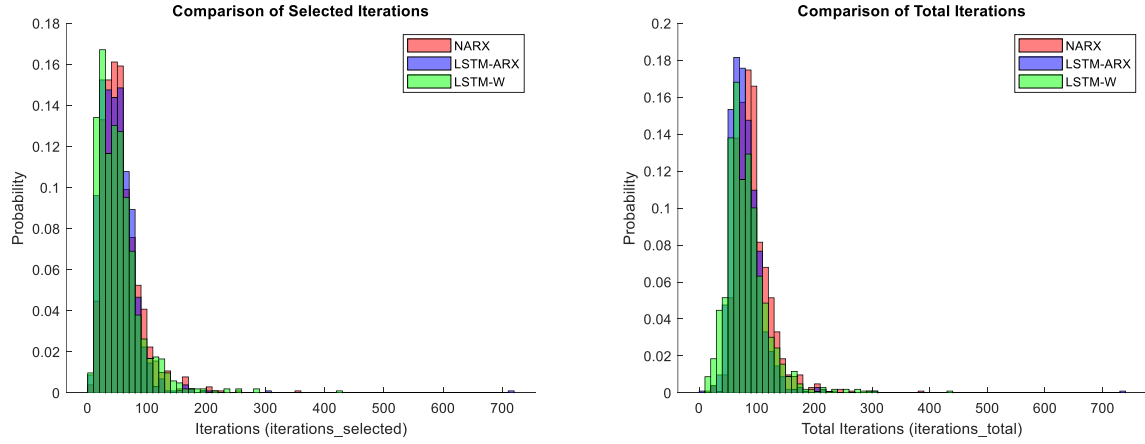


Fig. 109: Comparison of selected iterations (left) and total iterations (right).

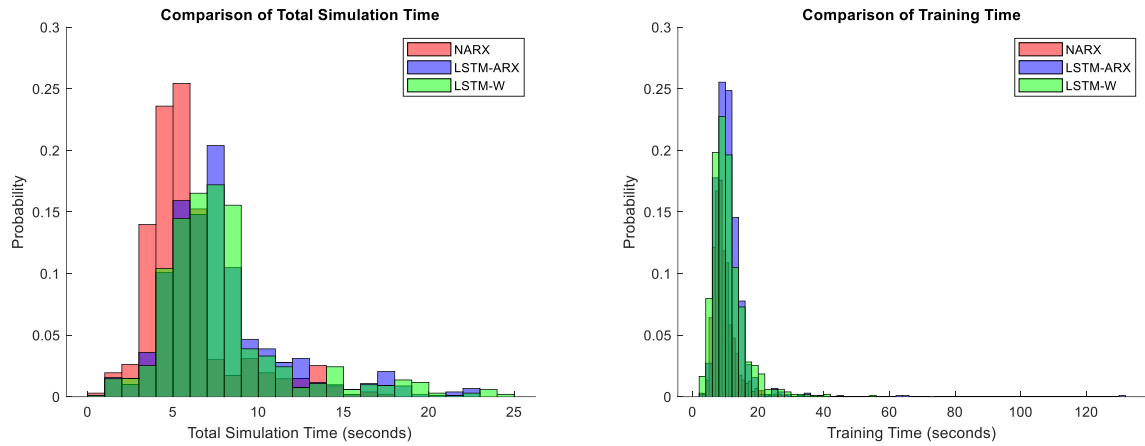


Fig. 110: Comparison of total simulation time (left) and training time (right).

I. Metrics Distribution

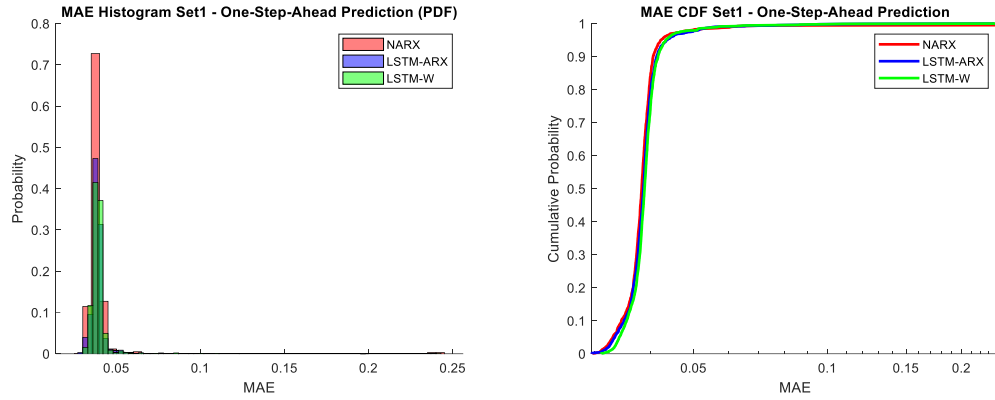


Fig. 111: MAE distributions for Set 1 – One-Step-Ahead Predictions.

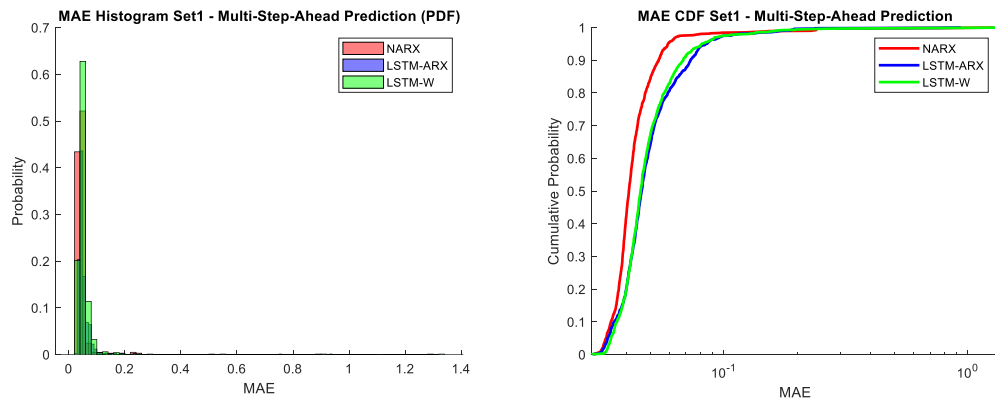


Fig. 112: MAE distributions for Set 1 – Multi-Step-Ahead Predictions.

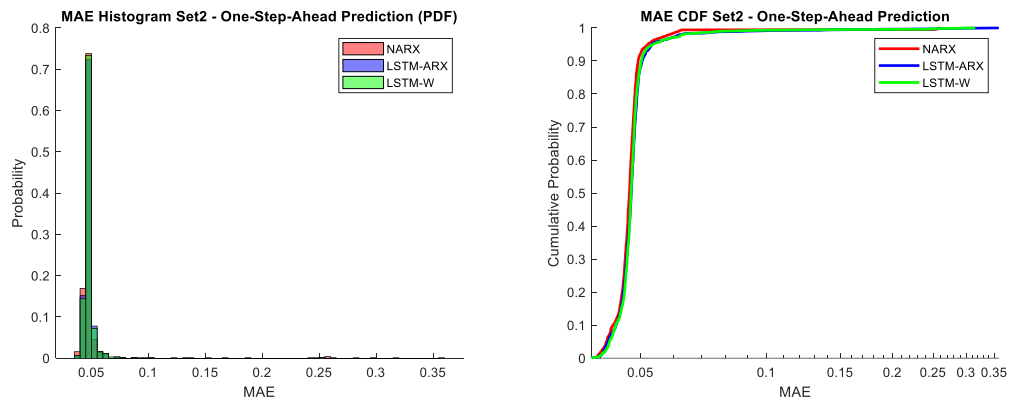


Fig. 113: MAE distributions for Set 2 – One-Step-Ahead Predictions.

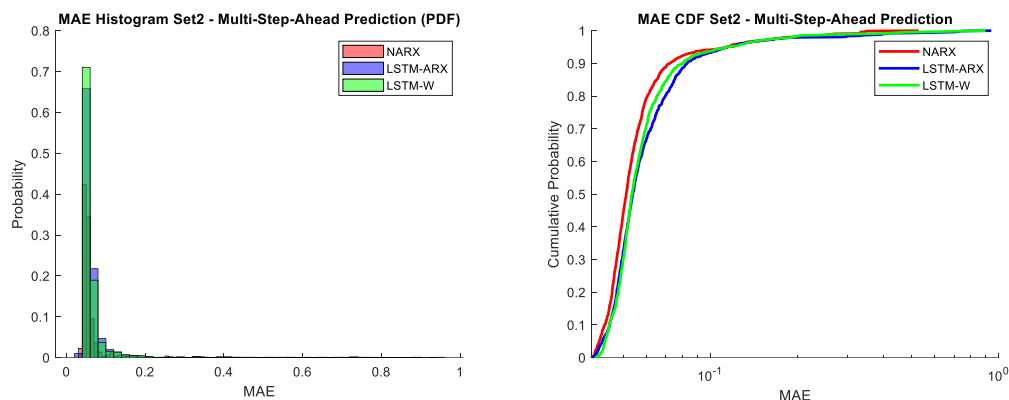


Fig. 114: MAE distributions for Set 2 – Multi-Step-Ahead Predictions.

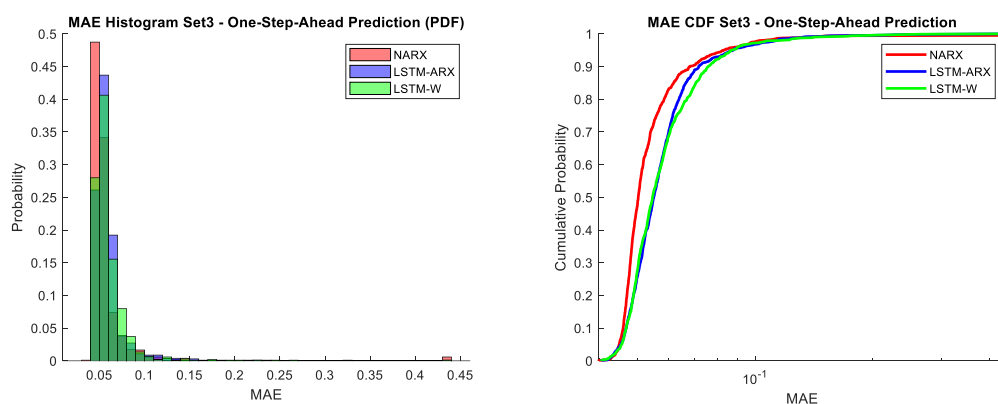


Fig. 115: MAE distributions for Set 3 – One-Step-Ahead Predictions.

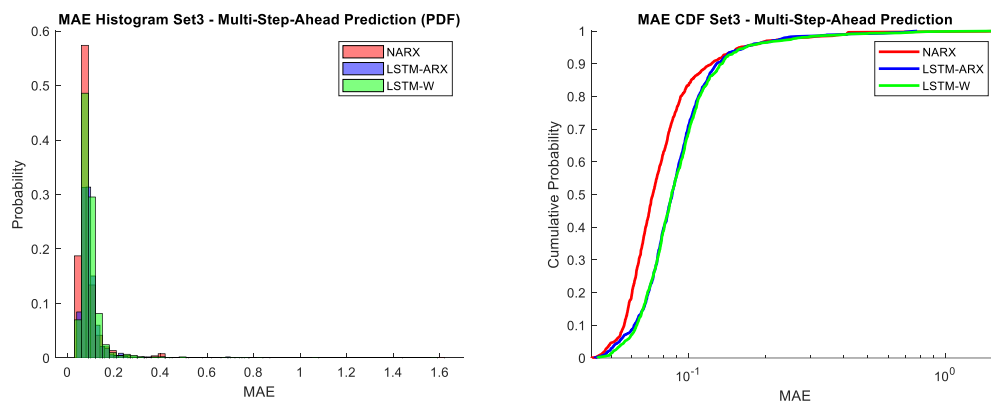


Fig. 116: MAE distributions for Set 3 – Multi-Step-Ahead Predictions.

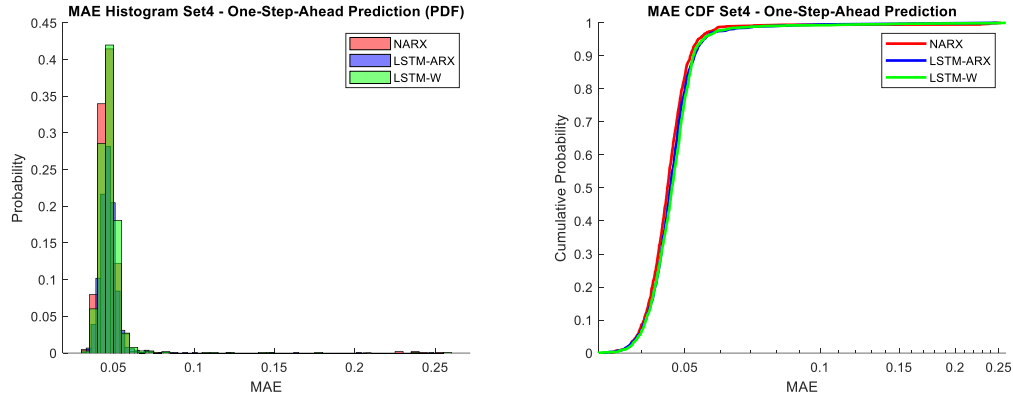


Fig. 117: MAE distributions for Set 4 – One-Step-Ahead Predictions.

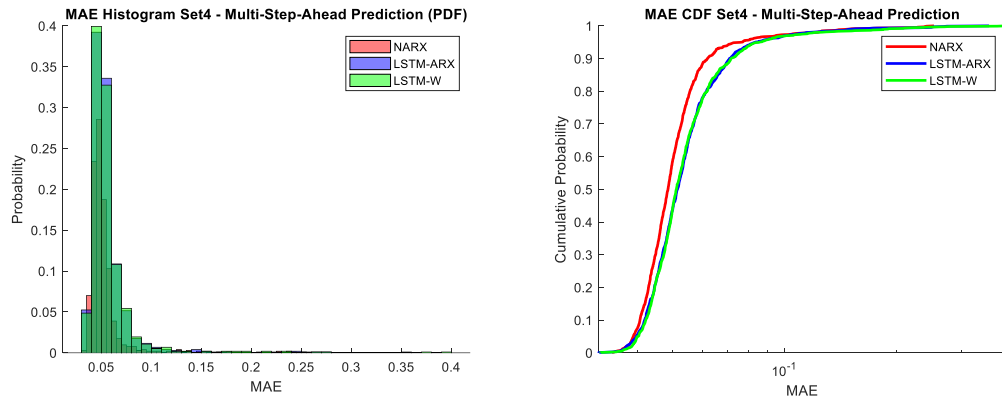


Fig. 118: MAE distributions for Set 4 – Multi-Step-Ahead Predictions.

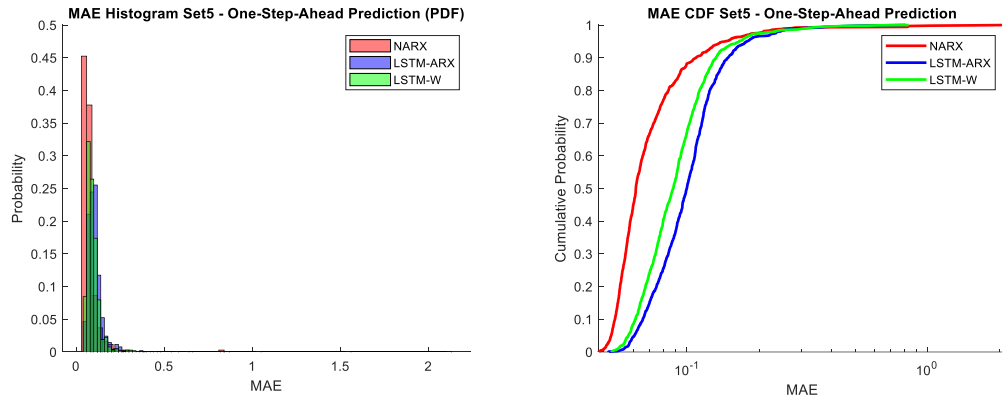


Fig. 119: MAE distributions for Set 5 – One-Step-Ahead Predictions.

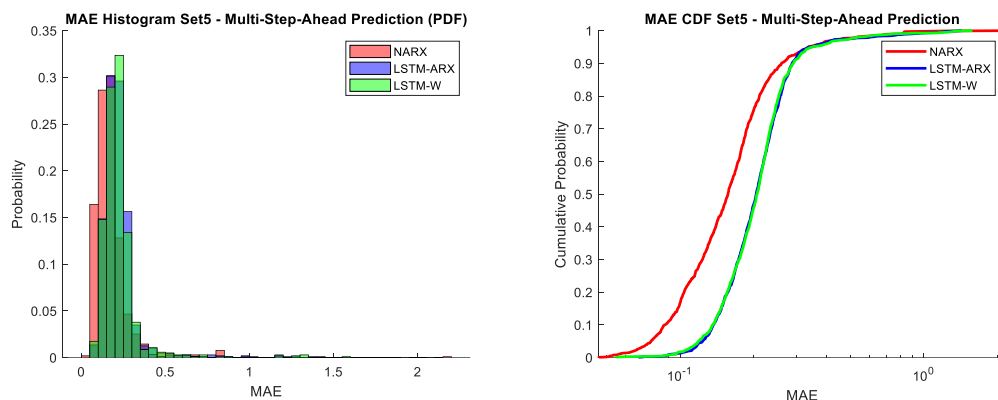


Fig. 120: MAE distributions for Set 5 – Multi-Step-Ahead Predictions.

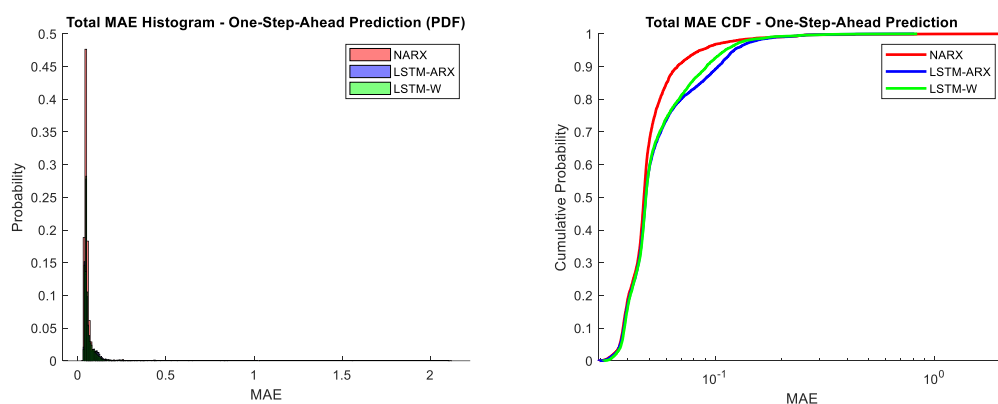


Fig. 121: Total MAE distributions – One-Step-Ahead Predictions (All Sets).

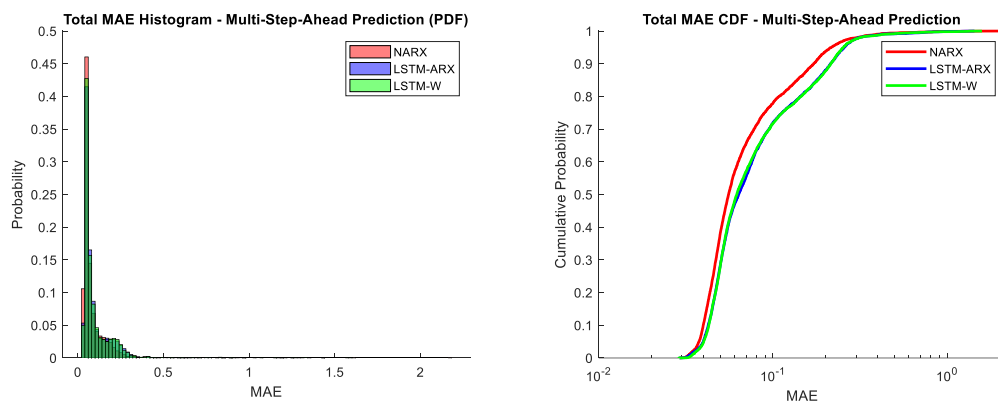


Fig. 122: Total MAE distributions – Multi-Step-Ahead Predictions (All Sets).

J. Hill Model Experiments

a. Experiment A – Two-Step Excitation

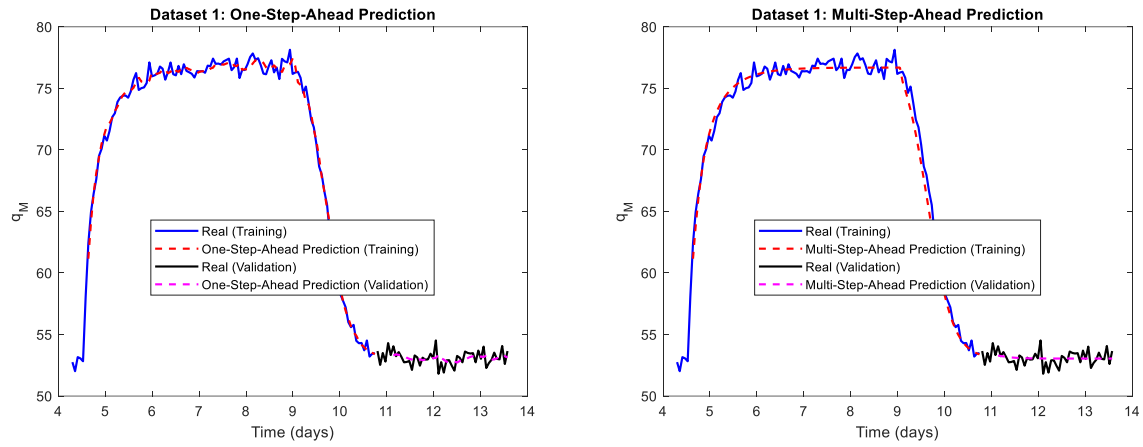


Fig. 123: One-step-ahead (left) and multi-step-ahead (right) predictions for Dataset 1 under two-step excitation (Hill Model).



Fig. 124: Training and validation loss vs. iteration for the Hill Model (two-step excitation).

Table 10: MSE and MAE for one-step-ahead and multi-step-ahead predictions (Hill Model, two-step excitation).

Set	One-Step-Ahead		Multi-Step-Ahead	
	MSE	MAE	MSE	MAE
Training	0.000120	0.008389	0.000234	0.011618
Validation	0.000059	0.006332	0.000061	0.006359
Testing	0.000215	0.011895	0.001632	0.036561

b. Experiment B – Six-Step Excitation

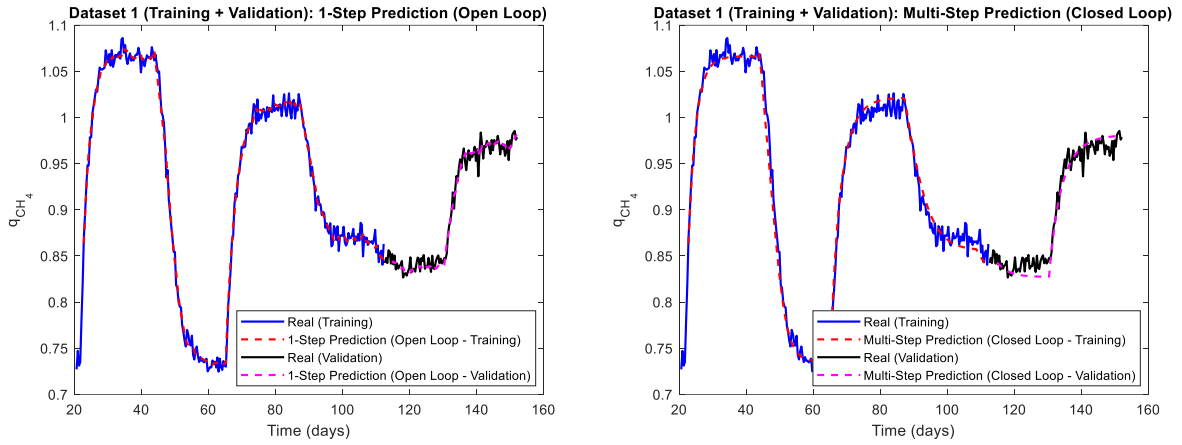


Fig. 125: One-step-ahead (left) and multi-step-ahead (right) predictions for Dataset 1 (Hill Model, six-step excitation).

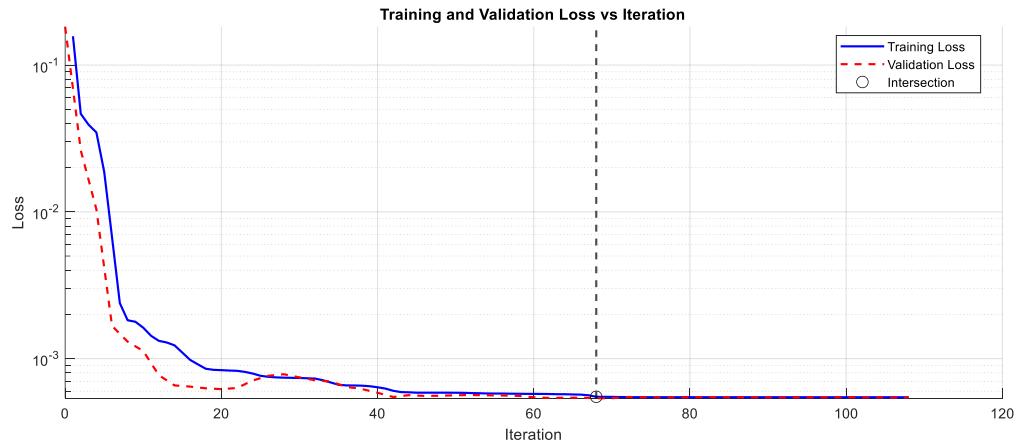


Fig. 126: Training and validation loss vs. iteration (Hill Model, six-step excitation).

Table 11: MSE and MAE for one-step-ahead and multi-step-ahead predictions (Hill Model, six-step excitation).

Set	One-Step-Ahead		Multi-Step-Ahead	
	MSE	MAE	MSE	MAE
Training	0.000072	0.006717	0.000125	0.008644
Validation	0.000070	0.006546	0.000134	0.009204
Training	0.000078	0.006986	0.000218	0.012327

K. AM2 Model Experiments

c. Experiment A – Two-Step Excitation

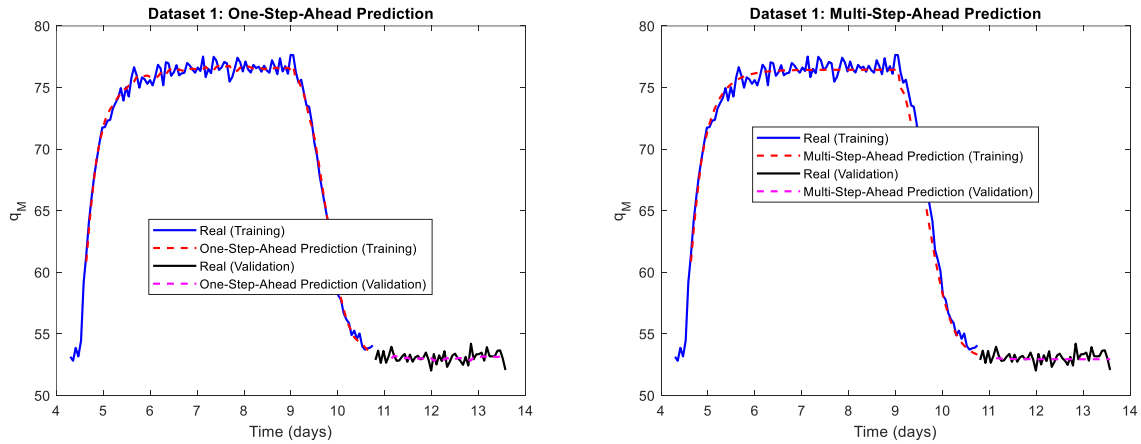


Fig. 127: One-step-ahead (left) and multi-step-ahead (right) predictions for Dataset 1 (AM2 Model, two-step excitation).



Fig. 128: Training and validation loss vs. iteration (AM2 Model, two-step excitation).

Table 12: MSE and MAE for one-step-ahead and multi-step-ahead predictions (AM2 Model, two-step excitation).

Set	One-Step-Ahead		Multi-Step-Ahead	
	MSE	MAE	MSE	MAE
Training	0.30	0.44	0.67	0.62
Validation	0.23	0.38	0.23	0.38
Testing	0.56	0.60	8.36	2.29

d. Experiment B – Six-Step Excitation

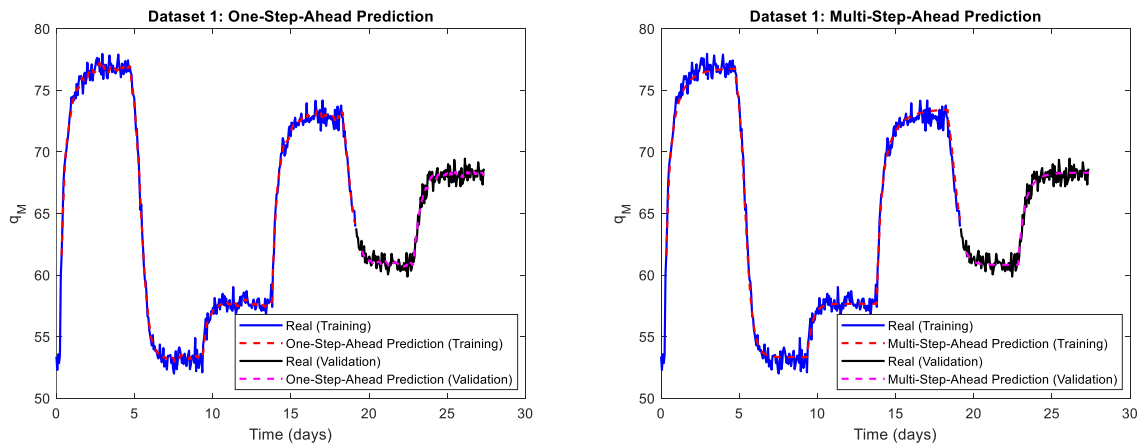


Fig. 129: One-step-ahead (left) and multi-step-ahead (right) predictions for Dataset 1 (AM2 Model, six-step excitation).

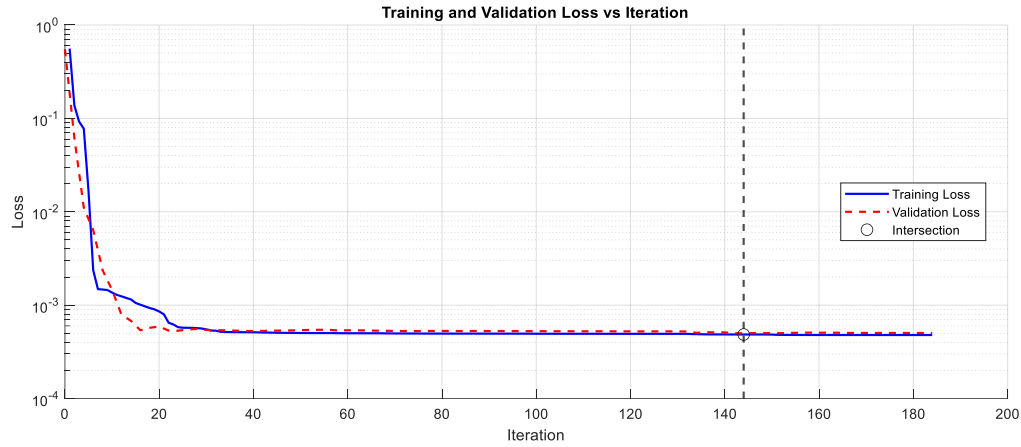


Fig. 130: Training and validation loss vs. iteration (AM2 Model, six-step excitation).

Table 13: MSE and MAE for one-step-ahead and multi-step-ahead predictions (AM2 Model, six-step excitation).

Set	One-Step-Ahead		Multi-Step-Ahead	
	MSE	MAE	MSE	MAE
Training	0.327	0.460	0.375	0.494
Validation	0.337	0.468	0.291	0.446
Testing	0.295	0.428	0.506	0.567

L. LSTM-A Snippet of Code

```
% Convert data to darray in CTB format
numFeatures = size(XTrain{1},1);
numTrainSamples = numel(XTrain);
XTrainMat = zeros(numFeatures, 1, numTrainSamples);
for i = 1:numTrainSamples
    XTrainMat(:,:,i) = XTrain{i};
end
XTrainDl = darray(XTrainMat, 'CTB');

numValSamples = numel(XVal);
XValMat = zeros(numFeatures, 1, numValSamples);
for i = 1:numValSamples
    XValMat(:,:,i) = XVal{i};
end
XValDl = darray(XValMat, 'CTB');

%% PART 6: Define the LSTM-ARX Network Architecture
```



```
disp('Defining the LSTM-ARX network architecture...');
numResponses = 1;
numHiddenUnits = num_hidden_neurons;
layers = [ ...
    sequenceInputLayer(numFeatures, 'Name', 'input')
    lstmLayer(numHiddenUnits, 'OutputMode', 'last', 'Name', 'lstm1')
    fullyConnectedLayer(numResponses, 'Name', 'fc2')];

%% PART 7: Configure Training Options
trainingType = 'lbfgs';

if strcmpi(trainingType, 'adam')
    options = trainingOptions('adam', ...
        'MaxEpochs', 1000, ...
        'MiniBatchSize', 32, ...
        'GradientThreshold', 5, ...
        'InitialLearnRate', 0.01, ...
        'LearnRateSchedule', 'piecewise', ...
        'LearnRateDropPeriod', 100, ...
        'LearnRateDropFactor', 0.01, ...
        'ValidationData', {XValDL, YVal}, ...
        'ValidationFrequency', 20, ...
        'ValidationPatience', 5, ...
        'InputDataFormats', "CTB", ...
        'Verbose', true, ...
        'OutputNetwork', "best-validation", ...
        'Plots', 'training-progress', ...
        'ExecutionEnvironment', 'gpu');
elseif strcmpi(trainingType, 'lbfgs')
    options = trainingOptions('lbfgs', ...
        "ExecutionEnvironment", "gpu", ...
        "ValidationData", {XValDL, YVal}, ...
        "ValidationFrequency", 2, ...
        "ValidationPatience", 20, ...
        "InputDataFormats", "CTB", ...
        "Plots", "training-progress", ...
        "OutputNetwork", "best-validation", ...
        "Verbose", true);
else
    error("Unrecognized training type. Use 'adam' or 'lbfgs'.");
end

%% PART 8: Train the LSTM-ARX Network using trainnet
disp('Training the LSTM-ARX network with trainnet...');
net_initial = dlnetwork(layerGraph(layers));

% Train using XTrainDL (dlarray) and YTrain (numeric vector)
[net_trained, info] = trainnet(XTrainDL, YTrain, net_initial, "mse", options);
disp(info)
net_lstm_arx = net_trained;
fprintf('Network trained (trainnet) and saved as dlnetwork.\n');
```

M. LSTM-B Snippet of Code

```
% Convert sequences to dldarray in CTB format
numFeatures = size(XTrain{1},1);
sequenceLength = size(XTrain{1},2);
numTrainSamples = numel(XTrain);
XTrainMat = zeros(numFeatures, sequenceLength, numTrainSamples);
for i = 1:numTrainSamples
    XTrainMat(:,:,i) = XTrain{i};
end
XTrainDl = dldarray(XTrainMat, 'CTB');
numValSamples = numel(XVal);
XValMat = zeros(numFeatures, sequenceLength, numValSamples);
for i = 1:numValSamples
    XValMat(:,:,i) = XVal{i};
end
XValDl = dldarray(XValMat, 'CTB');

%% PART 6: Define the LSTM-W Network Architecture
disp('Defining the LSTM-W network architecture...');
numResponses = 1;
numHiddenUnits = num_hidden_neurons;
layers = [ ...
    sequenceInputLayer(numFeatures, 'Name', 'input')
    lstmLayer(numHiddenUnits, 'OutputMode', 'last', 'Name', 'lstm1')
    fullyConnectedLayer(numResponses, 'Name', 'fc2')];

%% PART 7: Configure Training Options
trainingType = 'lbfgs';
if strcmpi(trainingType, 'adam')
    options = trainingOptions('adam', ...
        'MaxEpochs', 1000, ...
        'MiniBatchSize', 32, ...
        'GradientThreshold', 5, ...
        'InitialLearnRate', 0.01, ...
        'LearnRateSchedule', 'piecewise', ...
        'LearnRateDropPeriod', 100, ...
        'LearnRateDropFactor', 0.01, ...
        'ValidationData', {XValDl, YVal}, ...
        'ValidationFrequency', 20, ...
        'ValidationPatience', 5, ...
        'Verbose', true, ...
        'OutputNetwork', "best-validation", ...
        'Plots', 'training-progress', ...
        'ExecutionEnvironment', 'gpu');
elseif strcmpi(trainingType, 'lbfgs')
    options = trainingOptions('lbfgs', ...
        'ExecutionEnvironment', 'gpu', ...
        'ValidationData', {XValDl, YVal}, ...
```



```
"ValidationFrequency", 2, ...
"ValidationPatience", 20, ...
"InputDataFormats", "CTB", ...
"Plots", "training-progress", ...
'OutputNetwork', "best-validation", ...
"Verbose", true);
else
    error("Unrecognized training type. Use 'adam' or 'lbfgs'.");
end

%% PART 8: Train the LSTM-W Network using trainnet
disp('Training the LSTM-W network with trainnet...');
net_initial = dlnetwork(layerGraph(layers));
[net_trained, info] = trainnet(XTrainDl, YTrain, net_initial, "mse", options);
disp(info)
net_lstm_w = net_trained;
fprintf('Network trained (trainnet) and saved as dlnetwork.\n');
```