Machine Learning for Asset Management in Inland Waterway Networks

by

Yichao Zeng

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(College of Engineering and Computer Science)
in The University of Michigan-Dearborn
2026

Doctoral Committee:

Associate Professor Zhen Hu, Chair
Associate Professor Abdallah Chehade, Co-Chair
Professor Shan Bao
Professor Jian Hu
Associate Professor Hugo Casquero
Dr. Kai Wu, Ford Motor Company

Yichao Zeng
zyichao@umich.edu
ORCID iD: 0009-0001-8498-1744

© Yichao Zeng 2026

DEDICATION

To my family, my advisor Dr. Zhen Hu, and my girlfriend, Shuying Yin, for their unwavering love, guidance, encouragement, and support throughout this journey.

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my advisor, Dr. Zhen Hu, for his invaluable guidance, encouragement, and mentorship throughout my doctoral studies. His insight, patience, and support have shaped not only the development of this dissertation but also my growth as a researcher. I am also sincerely grateful to my co-advisor, Dr. Abdallah Chehade, for his guidance and thoughtful discussions during the early stages of my doctoral research.

I would like to extend my sincere thanks to the members of my doctoral committee, Dr. Shan Bao, Dr. Jian Hu, Dr. Hugo Casquero, and Dr. Kai Wu, for their time, insightful comments, and constructive suggestions. Their diverse perspectives and expertise have significantly strengthened the quality and depth of this dissertation.

I am also grateful to my colleagues and collaborators, including Jice Zeng, Zhao Zhao, Zifei Wang, Xudong Wang, Zhiwei Wang, and Enjian Cai, for their support, collaboration, and valuable discussions on the research projects related to this dissertation. Working with them has been an important part of my doctoral experience, and their contributions and encouragement have helped advance the work presented here.

I am profoundly grateful to my family, whose unwavering love and support have been the foundation of my academic journey. Their belief in me has given me the strength and motivation to persevere through challenges and to celebrate each achievement along the way.

I also wish to express my heartfelt thanks to my girlfriend, Shuying Yin, for her constant encouragement, understanding, and companionship. Her support has been a source of balance, comfort, and inspiration throughout the demanding years of this work.

Without the collective support of my advisor, co-advisor, doctoral committee, colleagues, family, and Shuying, this dissertation would not have been possible.

TABLE OF CONTENTS

DEDICATION ii

ACKNOWLEDGEMENTS iii

LIST OF FIGURES viii

LIST OF TABLES xiii

ABSTRACT xv

CHAPTER

1 Introduction 1

1.1 Background and Motivation 1
1.2 Research Objectives 4
1.3 Contributions 5
1.4 Dissertation Organization 7

2 Data Augmentation Based on Image Translation for Bayesian Inference-Based Damage Diagnostics of Miter Gates 9

2.1 Introduction 9
2.2 Background 13
2.2.1 Gap damage of miter gates in lock systems 13
2.2.2 Bayesian inference-based damage detection 14
2.2.3 Synthetic data generation using Generative Adversarial Networks (GANs) 16
2.3 Proposed Method 18
2.3.1 Overview 18
2.3.2 Data pre-processing 19
2.3.3 CycleGAN for Synthetic Data Generation 21
2.3.4 Accuracy Verification of Synthetically Generated Data 25
2.3.5 Bayesian Model Updating Based on Data Augmentation 29
2.4 Case Study 31
2.4.1 Problem description 31
2.4.2 Training of CycleGAN and performance evaluation of the training . 33
2.4.3 Accuracy assessment of synthetically generated new observations . . 35
2.4.4 Damage detection using Bayesian inference 46

2.5 Conclusions and Discussion . . . 48

3 Enhancing Bayesian Inference-Based Damage Diagnostics Through Domain Translation with Application to Miter Gates . . . 51

3.1 Introduction . . . 51
3.2 Background . . . 55
3.2.1 Probabilistic damage detection using Bayesian inference . . . 55
3.2.2 Structural health monitoring of miter gates in inland waterway networks . . . 56
3.2.3 Augmenting observation data in Bayesian inference using Generative Adversarial Networks (GANs) . . . 56
3.2.4 Limitations of existing methods . . . 58
3.3 Proposed Method . . . 59
3.3.1 Problem formulation . . . 59
3.3.2 Overview of the proposed BiEDT framework . . . 60
3.3.3 Domain translation of observations . . . 63
3.3.4 Bayesian inference-based damage diagnostics in the translated domain 69
3.3.5 Bayesian model averaging for the integration of different estimations 73
3.4 Case Study . . . 77
3.4.1 Problem description . . . 77
3.4.2 Domain translation of the observations . . . 80
3.4.3 Bayesian damage diagnostics in translated domain . . . 85
3.4.4 Bayesian model averaging for damage diagnostics of miter gates . . . 87
3.5 Conclusions and Discussion . . . 90

4 Hybrid Modeling for Streamflow Prediction in Ungauged Rivers with Uncertainty Quantification: A Comparative Analysis . . . 95

4.1 Introduction . . . 95
4.2 Overview of the proposed framework . . . 98
4.3 Hybrid modeling for streamflow prediction of ungauged rivers . . . 100
4.3.1 Physics-based RAPID . . . 100
4.3.2 Feature selection for ML-based river discharge prediction . . . 101
4.3.3 ML models for time-series river discharge prediction of un-gauged rivers . . . 105
4.3.4 Hybrid model architecture . . . 109
4.4 Uncertainty quantification of river discharge prediction of un-gauged rivers . 110
4.4.1 Methods for UQ of ML models . . . 110
4.4.2 Evaluation method for river discharge prediction uncertainty . . . 114
4.5 Case Study . . . 115
4.5.1 Raw data extraction and preprocessing . . . 116
4.5.2 Hybrid model setup . . . 118
4.5.3 Uncertainty quantification methods . . . 119
4.6 Results and Discussion . . . 120
4.6.1 Feature selection analysis results . . . 120

4.6.2 Model performance and the results of uncertainty quantification analysis . . . 121
4.6.3 Comparison of uncertainty quantification quality using calibration curves . . . 126
4.6.4 Comparison of computational time . . . 127
4.7 Conclusions . . . 128

5 A Data-Driven Channel Infilling Rate Forecasting Framework with Quantified Uncertainty for Southwest Pass and Houston Ship Channel . . . 130

5.1 Introduction . . . 130
5.2 Background . . . 133
5.2.1 The process of shoaling . . . 133
5.2.2 Study sites: HSC and SWP . . . 134
5.2.3 Data sources and existing tools . . . 136
5.2.4 Challenges and needs of data-driven models for shoaling forecasting 141
5.3 Methodology . . . 142
5.3.1 Overview . . . 142
5.3.2 Data Preprocessing . . . 142
5.3.3 Feature selection . . . 148
5.3.4 CIR Forecasting Problem Description and Solution Strategies . . . 150
5.3.5 Forecasting framework . . . 157
5.3.6 Uncertainty quantification of CIR forecasting . . . 159
5.3.7 Error Metrics . . . 160
5.4 Case Study . . . 161
5.4.1 Houston Ship Channel (HSC) . . . 161
5.4.2 The Southwest Pass (SWP) . . . 172
5.5 Conclusion . . . 183

6 A Recursive Shoaling Forecasting Framework with Shifted Delays and Its Application to the Southwest Pass in the United States . . . 185

6.1 Introduction . . . 185
6.2 Background . . . 190
6.2.1 Factors Impacting Shoaling Forecasting . . . 190
6.2.2 Machine Learning Approaches for Modeling Sediment Transport and Dredging Operations . . . 194
6.3 Methodology . . . 196
6.3.1 Overview of the Proposed Framework . . . 196
6.3.2 Introduction and Pre-processing of the Target Variable for the ML model . . . 198
6.3.3 Introduction and Pre-processing of the Input Variables for the ML Model . . . 200
6.3.4 A MIMO LSTM-NARX Model with Shifted Delays for CIR Prediction 203
6.3.5 Modified Loss Function for Training of the LSTM Model . . . 213
6.3.6 A Recursive Prediction Scheme to Enable Monthly CIR Prediction 214
6.4 Case Study . . . 218

6.4.1 The Southwest Pass (SWP) in the United States 218
6.4.2 Proposed Framework Setup 220
6.4.3 Evaluation Metrics 222
6.5 Results and Discussion 223
6.5.1 GSA Results 223
6.5.2 LSTM-NARX Model Prediction Results Comparison 227
6.6 Conclusions and Future Work 237
7 Conclusions and Future Work 239
7.1 Conclusions 239
7.2 Future Work 241
BIBLIOGRAPHY 243

LIST OF FIGURES

FIGURE

1.1 Three aspects of inland waterway network management discussed in this study and their relationships: sediment dynamics, structural health monitoring, and hydrologic forecasting. . . . 3

2.1 Lock system in inland waterway transportation . . . 9
2.2 Gap-caused water leakage on miter gate . . . 13
2.3 The basic structure of GANs . . . 17
2.4 Overview of the proposed framework . . . 19
2.5 The process of converting raw data to image data . . . 21
2.6 The structure of CycleGAN . . . 22
2.7 The structure of Convolutional U-net generator and discriminator of CycleGAN 24
2.8 The structure of Self-ONN of CycleGAN . . . 26
2.9 Flowchart of accuracy verification of synthetically generated observations . . . 28
2.10 Bayesian Model Updating Augmentation with synthetic data . . . 30
2.11 Miter Gate and physical-based FE model . . . 31
2.12 A finite element model for the miter gate. . . . 32
2.13 Strain observations from sensors comparison between miter gate 1 and 2 . . . 33
2.14 Data used for training and validation . . . 33
2.15 Normalized strain measurement value each sensor at different time step for miter gate 1 . . . 36
2.16 Normalized strain measurement value each sensor at different time step for miter gate 2 . . . 37
2.17 Normalized strain measurement image format for miter gate 1 (Checking training accuracy) . . . 38
2.18 Normalized strain measurement image format for miter gate 2 (Checking training accuracy) . . . 39
2.19 Low water level of miter gate 2 for months 21 to 40 . . . 40
2.20 High water level of miter gate 2 for months 21 to 40 . . . 40
2.21 Real (left) and synthetic (right) strain measurement image format of miter gate 2 for months 21 to 40 . . . 41
2.22 Error confidence interval comparison at each sensor (CycleGAN) . . . 42
2.23 Error confidence interval comparison at each sensor (Self-ONN CycleGAN) . . . 42
2.24 Error confidence interval comparison at each sensor (Self-generated CycleGAN) 43
2.25 Comparison of error distributions for sensor 1 . . . 43
2.26 Comparison of error distributions for sensor 2 . . . 44

2.27 Comparison of error distributions for sensor 3 44
2.28 Comparison of error distributions for sensor 4 45
2.29 Comparison of error distributions for sensor 5 45
2.30 PDF of the posterior samples with 5 months data 48
2.31 PDF of the posterior samples with 10 months data 49
2.32 PDF of the posterior samples with 15 months data 49
2.33 PDF of the posterior samples with 20 months data 50

3.1 Miter gates in inland waterway navigation 56
3.2 Illustration of "gap" on miter gate structures 57
3.3 A framework proposed in our previous research [1, 2] 58
3.4 Overview of the proposed BiEDT framework 61
3.5 The structure of CycleGAN for translation of miter gate observations. 64
3.6 The structure of DANN for translation of miter gate observations. 67
3.7 Transformation of distributions in generative models [3] 70
3.8 Illustration of a cINN model as a regression model for Bayesian inference 71
3.9 An example of FEA model of miter gates 74
3.10 Illustration of source and target miter gates on a river. 78
3.11 Water level comparison of source and target miter gates 80
3.12 A comparison of monitoring data of source and target miter gates (Data of only five sensors are plotted for illustration purpose) 81
3.13 Scatter plot comparison of monitoring data of the source and target miter gates 82
3.14 Scatter plot comparison of monitoring data of source and target miter gates after observation translation using CycleGAN 84
3.15 Comparison of DANN features of the source and target miter gates 86
3.16 Posterior distribution of gap length obtained using cINN and CycleGAN-based domain translation 87
3.17 Errorbar plot of the posterior distribution of different gap length estimates (CycleGAN-based domain translation) 88
3.18 Posterior distribution of gap length obtained using cINN and DANN-based domain translation 89
3.19 Errorbar plot of the posterior distribution of different gap length estimates (DANN-based domain translation) 90
3.20 Posterior distribution of gap length obtained using physics-based Bayesian inference 91
3.21 Errorbar plot of the posterior distribution of different gap length estimates (Physics-based Bayesian inference) 92
3.22 Bayesian model averaging of posterior distributions (Case 1) 92
3.23 Bayesian model averaging of posterior distributions (Case 2) 93
3.24 Errorbar plot of the posterior distribution of different gap lengths after model selection using Bayes factor 93

4.1 Overview of the proposed framework. 99
4.2 The diagram of SNGP Resnet-based model structure. 107
4.3 The structure of the NNE LSTM-based model. 113
4.4 The map of the six gauged river locations of interest. 116

4.5 Variance decomposition-based feature selection. 122
4.6 Divided time-series prediction plot with 95% confidence interval and true values (MC-dropout) between purely data-driven and hybrid model. 123
4.7 Violin plots of absolute error between models and observation discharge using MC-dropout (a), NNE (b), and SNGP (c). 124
4.8 Divided time-series prediction plot with 95% confidence interval and true values (NNE) between purely data-driven and hybrid model. 125
4.9 Divided time-series prediction plot with 95% confidence interval and true values (SNGP) between purely data-driven and hybrid model. 126
4.10 Calibration curve comparisons between hybrid model and purely data-driven model using (a) MC-dropout, (b) NNE, and (c) SNGP. 127

5.1 Schematic diagram showing progressive Morphological Changes in Channel Geometry 134
5.2 Houston Shipping Channel (HSC). (a) Buffalo Bayou and San Jacinto River Basins showing data collection stations considered for sensitivity analysis to select inputs for building the forecasting model for the HSC region. (b) Zoomed-in view of the HSC. (c) Bathymetric survey for a portion of channel in HSC. 135
5.3 The Southwest Pass (SWP). (a) Mississippi River and its tributaries showing discharge and the locations of data collection stations considered for sensitivity analysis to select inputs for building the forecasting model for the SWP region. (b) Zoomed-in view of SWP. (c) Bathymetric survey for a portion of channel in SWP. 137
5.4 Annual hydrographic survey entries for the Southwest Pass (SWP) and Houston Ship Channel (HSC) from 2012 to 2024. 140
5.5 Proposed framework of the solution approaches. 143
5.6 Pre-processing of historic bathymetry survey data extracted from *CSAT input* . 144
5.7 Calculating the sediment volume for a fixed survey grid and time instance . . . 145
5.8 IQR-based extreme event filter for HSC region considering aggregated sedimentation data 147
5.9 IQR-based extreme event filter for SWP region considering aggregated sedimentation data 148
5.10 The overall structure of the LSTM model. 153
5.11 The overall structure of the CNN-LSTM model for CIR forecasting. 154
5.12 The overall structure of the TCN-LSTM model for CIR forecasting. 156
5.13 Sedimentation volume and channel infilling rate (CIR) for representative reaches in the HSC region (The legend for all subplots is identical to that of the top-left figure). 162
5.14 HSC region: Forecasted channel infilling rate (CIR) and sediment volume at a representative individual reach under the disaggregated strategy, compared across three modeling approaches. The top row shows the overall sediment volume time series together with the forecast windows. The second row presents the CIR forecasts for four selected forecast start dates, and the third row shows the corresponding sediment volume forecasts. Each column corresponds to one forecast window. 164

5.15 HSC region: Forecasting of cumulative sedimentation volume aggregated over all reaches using two modeling strategies, illustrated for four selected forecast start dates using the MLR model with the three most sensitive gauges. The top row shows the overall sediment volume time series together with the forecast windows. The second row presents the CIR forecasts for the four selected start dates, while the third row shows the corresponding sediment volume forecasts. Each column corresponds to one forecast window. 166
5.16 HSC region: Forecasting of cumulative sedimentation volume under the aggregated modeling strategy, illustrated for four selected forecast start dates and compared across different modeling approaches. The top row shows the overall sediment volume time series together with the forecast windows. The second row presents the CIR forecasts for the four selected start dates, while the third row shows the corresponding sediment volume forecasts. Each column corresponds to one forecast window. 169
5.17 HSC region: Cumulative sedimentation volume forecasting error growth with prediction horizon using different approaches. 169
5.18 HSC region: Cumulative sedimentation volume forecasting uncertainty quantification. HSC region: Comparison of CIR forecasts and corresponding region-scale cumulative sedimentation volume predictions with uncertainty bounds, shown for four different time instances and compared across different modeling approaches. 171
5.19 HSC region: Cumulative sedimentation volume forecasting calibration curve . . 172
5.20 Sedimentation volume and CIR for representative reaches in the SWP region (The legend for all subplots is identical to that of the top-left figure). 173
5.21 SWP region: Forecasts of CIR and sediment volume at a representative individual reach under the disaggregated modeling strategy, compared across three modeling approaches. The top row shows the overall sediment volume time series together with the forecast windows. The second row presents the CIR forecasts for four selected forecast start dates, while the third row shows the corresponding sediment volume forecasts. Each column corresponds to one forecast window. . 174
5.22 SWP region: Forecasting of cumulative sedimentation volume aggregated over all reaches using two modeling strategies, illustrated for four selected forecast start dates using the LSTM model with the eight most sensitive gauges. The top row shows the overall sediment volume time series together with the forecast windows. The second row presents the CIR forecasts for the four selected start dates, while the third row shows the corresponding sediment volume forecasts. Each column corresponds to one forecast window. 176
5.23 SWP region: Forecasting of cumulative sedimentation volume under the aggregated modeling strategy, illustrated for four selected forecast start dates and compared across different modeling approaches. The top row shows the overall sediment volume time series together with the forecast windows. The second row presents the CIR forecasts for the four selected start dates, while the third row shows the corresponding sediment volume forecasts. Each column corresponds to one forecast window. 179
5.24 SWP region: Cumulative sedimentation volume forecasting error growth with prediction horizon using different approaches. 179

5.25 SWP region: Cumulative sedimentation volume forecasting uncertainty quantification. HSC region: Comparison of CIR forecasts and corresponding region-scale cumulative sedimentation volume predictions with uncertainty bounds, shown for four different time instances and compared across different modeling approaches. 182
5.26 SWP region: Cumulative sedimentation volume forecasting calibration curve . . 183

6.1 Factors affecting sedimentation. 191
6.2 Overview of the proposed framework. 197
6.3 Structure of LSTM cell. 205
6.4 Structure of LSTM-NARX model. 206
6.5 Conceptual sketch of sediment dynamics within a river system. 211
6.6 Illustration of low and high weighting scenarios. 213
6.7 Illustration of direct and recursive prediction schemes for multi-step LSTM CIR forecasting. 215
6.8 Geographical background of SWP and the selected gauge station locations (Station IDs in Table 6.3 are highlighted in green). 219
6.9 Impact of IQR-based outlier removal on CIR (top) and sediment volume (bottom) trends for the aggregated SWP data. 220
6.10 Samples of the aggregated GSA index values and corresponding optimal shifts. . 225
6.11 Map of the most influential station locations along the Mississippi River, annotated with their corresponding optimal number of shifted days as determined by GSA (Station IDs in Table 6.4 are highlighted in green). 226
6.12 Comparison of CIR and back-calculated sediment volume predictions using recursive and direct predictions at selected periods. 228
6.13 Comparison of CIR and back-calculated sediment volume predictions using different input combinations under the 3×10-day recursive prediction scheme at selected periods. 231
6.14 Comparison of CIR and back-calculated sediment volume predictions using standard and weighted MSE loss functions under the 3×10-day recursive prediction scheme with shifted gauge-only input variables at selected periods. 234
6.15 Comparison of CIR and back-calculated sediment volume predictions using weighted MSE loss function with $w_{\text{high}} = 5, 20, 50$. 236

LIST OF TABLES

TABLE

2.1 Model parameters of two CycleGAN models 34
2.2 Accuracy evaluation metrics for training 35

3.1 Model parameters of CycleGAN 83
3.2 Comparison of metrics for different objects after using CycleGAN 83
3.3 Model parameters of DANN 85

4.1 An overview of features in this case study 118
4.2 Top 5 features using gradient-based feature selection 121
4.3 LSTM-based model validation performance using different input feature combinations 123

5.1 HSC region: Evaluation metrics for reach-level predictions at a representative reach under the disaggregated modeling strategy, compared across different modeling approaches. Columns correspond to four selected forecast start dates and the full test set. For the full test set, the best value for each metric is highlighted in bold. 163
5.2 HSC region: Evaluation metrics for reach-level predictions at a representative reach under the disaggregated modeling strategy, using different sets of sensitive gauges (MLR). Columns correspond to four selected forecast start dates and the full test set. Best values for the full test set are highlighted in bold. 165
5.3 HSC region: Evaluation metrics for cumulative sediment volume prediction under different modeling strategies using the three most sensitive gauges (MLR). Columns correspond to four selected forecast start dates and the full test set. Best values for the full test set are highlighted in bold. 167
5.4 HSC region: Evaluation metrics for cumulative sediment volume prediction under the aggregated modeling strategy, compared across different modeling approaches. Columns correspond to four selected forecast start dates and the full test set. For the full test set, the best value for each metric is highlighted in bold. 170
5.5 SWP region: Evaluation metrics for reach-level predictions at a representative reach under the disaggregated modeling strategy, compared across different modeling approaches. Columns correspond to four selected forecast start dates and the full test set. Best values for the full test set are highlighted in bold. 174

5.6 SWP region: Evaluation metrics for reach-level predictions at a representative reach under the disaggregated modeling strategy using different sets of sensitive gauges (LSTM). Columns correspond to four selected forecast start dates and the full test set. Best values for the full test set are highlighted in bold. 175
5.7 SWP region: Evaluation metrics for cumulative sediment volume predictions under two modeling strategies using the eight most sensitive gauges (LSTM). The eight forecast windows are shown in two blocks for readability, and the overall performance over the full test set is reported in the final column. 177
5.8 SWP region: Evaluation metrics for cumulative sediment volume prediction under the aggregated modeling strategy, compared across different modeling approaches. Columns correspond to four selected forecast start dates and the full test set. Best values for the full test set are highlighted in bold. 180

6.1 Categories of input variables for ML-based sediment prediction with corresponding references. 194
6.2 Categories of ML Models applied to sediment transport and dredging operations with corresponding references. 195
6.3 Top 10 input variables ranked in descending order by aggregated GSA index. . 225
6.4 Top 10 river discharge gauges ranked by aggregated GSA index, with corresponding optimal number of shifted days (ONSD). 227
6.5 Comparison of prediction metrics for recursive and direct prediction methods. . 230
6.6 Comparison of prediction metrics for different input configurations. 232
6.7 Comparison of prediction metrics for standard MSE loss and weighted MSE loss with different w_{high} values. 235

ABSTRACT

Inland waterway networks are essential to the economic security, freight mobility, and water resource management of the United States. Their reliable operation depends not only on the condition of navigation infrastructure, such as locks and miter gates, but also on the surrounding hydrologic and sedimentary processes that govern structural loading, channel depth, and navigation availability. These components are highly interdependent in practice. River discharge and water levels provide a common physical link across inland waterway management: they govern hydraulic loading on navigation structures, influence strain responses used for structural health monitoring, and drive sediment transport processes that gradually reshape navigation channels.

These coupled challenges create a need for predictive methods that can make effective use of sparse and heterogeneous data, incorporate physical knowledge, quantify uncertainty, and support maintenance and operational decisions. Following this motivation, this dissertation develops a progressive set of machine learning (ML) and hybrid modeling frameworks that move from structural data augmentation, to probabilistic miter gate damage diagnostics, to uncertainty-aware streamflow prediction, and finally to data-driven shoaling and sedimentation forecasting with enhanced predictive capability.

The work first addresses data scarcity in miter gate structural health monitoring (SHM) by developing a data augmentation method that transfers monitoring information across similar gate structures. It then extends this idea into a broader Bayesian diagnostic framework that combines translated observations, physics-based inference, and uncertainty quantification (UQ) for miter gate damage assessment. The scope is subsequently expanded from local structural diagnostics to river-scale hydrologic forecasting, where hybrid ML and physics-based models are used to predict streamflow with quantified uncertainty. Finally, the work moves toward sediment management by developing ML frameworks for shoaling and sedimentation forecasting, first through a general data-driven prediction framework and then through an enhanced recursive forecasting strategy that accounts for delayed hydrologic effects.

Through this sequence, the dissertation connects infrastructure monitoring, hydrologic prediction, and sediment forecasting as related components of data-driven inland water-

way asset management. The resulting frameworks improve the use of limited monitoring and environmental data, provide uncertainty-aware predictions, and support more informed decision-making for infrastructure maintenance, navigation planning, and sediment management.

CHAPTER 1

Introduction

1.1 Background and Motivation

The inland waterway network is a vital component of the United States transportation system and plays an essential role in supporting national economic growth, freight mobility, and water resource management. Spanning more than 25,000 miles of navigable rivers and canals, this system enables the movement of more than 630 million tons of freight each year, valued at more than $770 billion [4]. Waterway transportation is particularly important for bulk commodities such as coal, petroleum, grain, aggregates, and chemicals, offering lower transportation costs and higher energy efficiency than rail or truck transport [5]. Beyond freight movement, inland waterways contribute an estimated $7 billion annually in transportation cost savings and support domestic supply chains, exports, and regional economic development. Continued modernization of this system is expected to generate further economic benefits through job creation, improved reliability, and enhanced transportation efficiency [6].

Although the economic and logistical importance of the inland waterway system is well recognized, its continued reliability depends on the performance of both navigation infrastructure and the surrounding riverine environment. Among these assets, navigation locks and their miter gates play a central role by allowing vessels to move safely between river reaches with different water levels. However, many lock-and-gate systems are aging and are increasingly exposed to operational and environmental stresses, including fluctuating water levels, extreme flow events, and sediment deposition. As freight demand grows and hydrologic conditions become more uncertain, these stresses create increasing challenges for infrastructure maintenance, navigation planning, and long-term waterway management.

Reliable inland waterway operation therefore requires more than periodic inspection and maintenance of individual structures. It also requires predictive understanding of the hydrologic and sedimentary processes that affect those structures and the channels they serve.

River discharge and water levels provide a common physical link across these problems: they govern hydraulic loading on navigation structures, influence the strain responses used for structural health monitoring, and drive sediment transport processes that gradually reshape navigation channels. As sediment accumulates, channel geometry and available depth change, reducing navigation reliability and increasing dredging needs. At the same time, direct condition assessment of lock-and-gate components remains challenging because many critical components are submerged, and the resulting monitoring data are often sparse, incomplete, or heterogeneous.

These coupled challenges create a need for predictive methods that can make effective use of limited data, incorporate physical knowledge where available, quantify uncertainty, and support maintenance and operational decisions. Structural health monitoring, hydrologic forecasting, and sedimentation prediction are therefore not isolated problems. Rather, they represent interconnected components of inland waterway asset management. Structural health monitoring supports the assessment of critical navigation assets such as miter gates; hydrologic forecasting provides information about water levels and discharge conditions that affect both structural loading and sediment transport; and shoaling prediction supports dredging planning and channel maintenance by forecasting sediment accumulation and channel infilling. Together, these components form the basis for a data-driven and science-based approach to improving the reliability and resilience of inland waterway networks.

Figure 4.1 illustrates the three major research themes considered in this dissertation: structural health monitoring, hydrologic forecasting, and sediment dynamics. These themes are linked through shared physical variables, including water levels and discharge, as well as through their common role in supporting operational decision-making. At the infrastructure level, miter gates are critical components of navigation locks, and monitoring their condition is essential for maintaining safety and reliability. However, field data collected from these structures are often limited, which makes it difficult to detect deterioration and quantify damage with confidence. At the river-system level, streamflow forecasting provides essential information for navigation scheduling, flood management, and water resource operations. Streamflow also influences the loading conditions experienced by hydraulic structures and provides key forcing information for sediment transport. At the channel-management level, shoaling and sedimentation affect channel depth, vessel passage, and dredging requirements, making sedimentation forecasting essential for proactive maintenance planning.

Recent advances in machine learning (ML) provide new opportunities to address these challenges. In structural health monitoring, generative modeling and domain translation techniques can help enrich sparse monitoring datasets and improve damage diagnostics

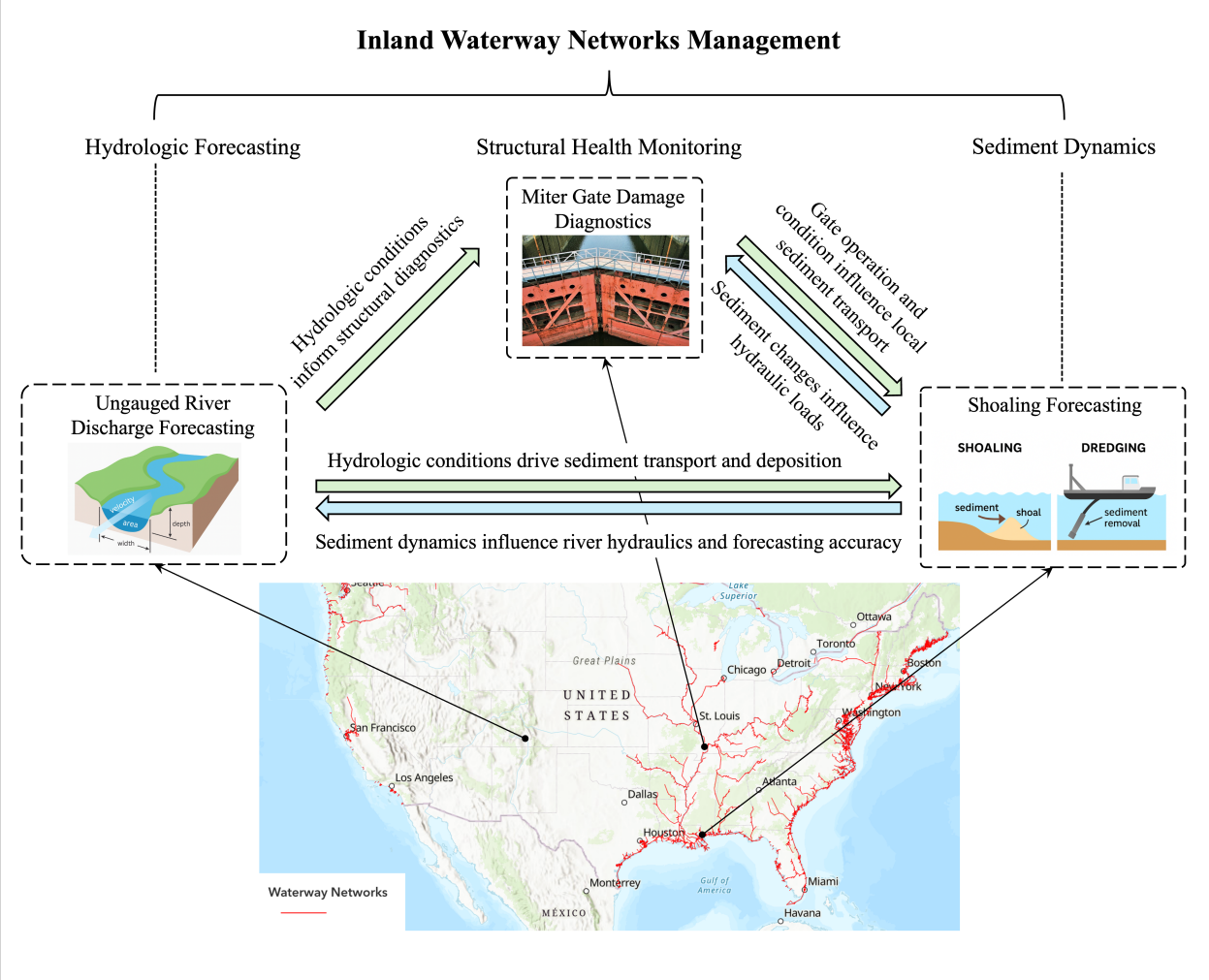


Figure 1.1: Three aspects of inland waterway network management discussed in this study and their relationships: sediment dynamics, structural health monitoring, and hydrologic forecasting.

when direct observations are limited. In hydrologic forecasting, hybrid models that combine physics-based river-routing information with ML can improve streamflow prediction in ungauged or sparsely gauged basins while providing uncertainty estimates. In sedimentation prediction, data-driven models can integrate hydrographic survey data, hydrologic variables, and delayed upstream effects to forecast channel infilling and sediment accumulation. However, for these methods to be useful in waterway asset management, they must be developed in ways that preserve physical meaning, quantify uncertainty, and support practical decision-making.

Motivated by these needs, this dissertation develops a progressive set of ML and hybrid modeling frameworks for inland waterway asset management. The research begins with data augmentation for miter gate structural health monitoring, where monitoring information is transferred across similar gate structures to address data scarcity. It then extends to a broader probabilistic diagnostic framework that combines domain translation, physics-based Bayesian inference, and uncertainty quantification (UQ) for miter gate damage assessment. The scope is subsequently expanded to river-scale streamflow prediction, where hybrid ML and physics-based models are developed with UQ. Finally, the dissertation addresses sediment management through two shoaling and sedimentation forecasting studies: the first develops a general data-driven forecasting framework for channel infilling rate and sediment volume prediction, and the second enhances forecasting performance through a recursive prediction strategy that accounts for shifted hydrologic delays.

Through this sequence, the dissertation connects infrastructure monitoring, hydrologic prediction, and sediment forecasting as related components of data-driven inland waterway asset management. The proposed methods aim to improve the use of sparse and heterogeneous data, provide uncertainty-aware predictions, and support more informed decision-making for infrastructure maintenance, navigation planning, and dredging management.

1.2 Research Objectives

The overall objective of this dissertation is to advance ML-based and hybrid modeling methods for asset management in inland waterway networks. The dissertation addresses this objective through five connected research tasks:

- **Miter Gate Data Augmentation for Structural Health Monitoring:** To develop a Cycle-Consistent Generative Adversarial Network (CycleGAN)-based data translation approach that transfers monitoring information across similar miter gate structures and generates synthetic observations for data-scarce structural health mon-

itoring (SHM) applications. The objective is to improve the accuracy and confidence of Bayesian inference-based damage diagnostics under limited monitoring data.

- **Bayesian Diagnostics with Domain Translation for Miter Gates:** To establish a Bayesian Inference-based Damage Diagnostics Enhanced through Domain Translation (BiEDT) framework that integrates domain translation, physics-based Bayesian inference, conditional invertible neural networks (cINNs), and Bayesian model averaging. The objective is to improve probabilistic miter gate damage assessment while explicitly quantifying diagnostic uncertainty.

- **Hybrid Modeling for Streamflow Prediction with UQ:** To develop a hybrid streamflow prediction framework that combines physics-based river-routing information with ML models for ungauged or sparsely monitored rivers. The objective is to improve forecast accuracy and provide uncertainty-aware streamflow predictions for risk-informed waterway operation and planning.

- **Data-Driven Shoaling and Sedimentation Forecasting:** To develop a data-driven forecasting framework for channel infilling rate (CIR) and cumulative sedimentation volume prediction using hydrographic survey data, hydrologic variables, and ML models. The objective is to support sediment management, dredging planning, and channel maintenance through reliable forecasting of shoaling behavior.

- **Enhanced Recursive Shoaling Forecasting with Shifted Hydrologic Delays:** To improve sedimentation forecasting performance by developing a recursive prediction framework that accounts for delayed hydrologic effects. The objective is to identify influential upstream discharge gauges and their time shifts, incorporate this information into a multi-input multi-output LSTM-NARX model, and enhance monthly CIR and sediment volume prediction for operational decision support.

Collectively, these objectives address data scarcity in structural monitoring, uncertainty in hydrologic forecasting, and predictive limitations in sedimentation management. They demonstrate how ML, physics-based modeling, UQ, and domain knowledge can be integrated to support more reliable and sustainable inland waterway asset management.

1.3 Contributions

This dissertation makes the following contributions to ML-enabled inland waterway asset management:

- **Data Translation and Augmentation for Miter Gate SHM.** A data augmentation method is developed to translate monitoring data between similar miter gate structures using CycleGAN. This method addresses the scarcity of monitoring observations by generating synthetic data that preserves meaningful relationships between hydraulic inputs and structural responses. In addition, an accuracy verification strategy is introduced to evaluate whether generated observations are suitable for Bayesian model updating. By incorporating verified synthetic observations into Bayesian inference-based damage diagnostics, the proposed approach improves the estimation of gap length in miter gates and demonstrates the value of data augmentation for probabilistic structural health monitoring.

- **Bayesian Diagnostics Enhanced through Domain Translation.** A broader Bayesian diagnostic framework is proposed to integrate information from source and target miter gate domains. The framework combines domain translation methods, including CycleGAN and Domain-Adversarial Neural Networks (DANN), with conditional invertible neural networks to provide probabilistic damage estimates. Physics-based Bayesian inference is also incorporated to retain physical interpretability. Furthermore, Bayesian model averaging and model selection are introduced to synthesize different probabilistic estimates of the damage state. This contribution provides a more comprehensive framework for uncertainty-aware miter gate damage diagnostics.

- **Hybrid Streamflow Prediction with UQ.** A hybrid modeling framework is developed for streamflow prediction in ungauged rivers by integrating physics-based river-routing information with ML models. The framework uses hydrologic features and physics-model outputs to guide data-driven prediction and improve model performance under data-scarce conditions. UQ methods are incorporated to evaluate forecast reliability and provide probabilistic information. This contribution supports risk-informed decision-making for waterway operation, navigation planning, and hydrologic management.

- **Data-Driven Forecasting of Shoaling and Channel Infilling.** A data-driven framework is developed to forecast channel infilling rate and cumulative sedimentation volume in navigation channels. The framework integrates hydrographic survey data, hydrologic variables, feature selection, ML-based forecasting models, and UQ. Applications to the Houston Ship Channel and the Southwest Pass demonstrate how data-driven models can support sediment management by predicting shoaling behavior and evaluating forecast uncertainty.

- **Enhanced Shoaling Forecasting through Shifted Delays and Recursive Prediction.** An enhanced recursive shoaling forecasting framework is developed to improve prediction performance for the Southwest Pass. The framework uses sensitivity analysis to identify influential upstream discharge gauges and their optimal shifted delays, reflecting the delayed influence of hydrologic conditions on downstream sedimentation. A multi-input multi-output LSTM-NARX model is then used to recursively predict CIR and back-calculate sediment volume over monthly horizons. This contribution improves the representation of delayed hydrologic effects and provides a more operationally useful approach for sedimentation forecasting.

Together, these contributions advance data-driven asset management for inland waterway networks by improving data utilization, incorporating physical understanding, quantifying uncertainty, and connecting structural monitoring, hydrologic forecasting, and sediment management within a unified dissertation framework.

1.4 Dissertation Organization

This dissertation is organized into seven chapters. Chapter 1 introduces the motivation, research objectives, contributions, and overall organization of the dissertation. Chapter 2 presents a data augmentation approach for Bayesian inference-based damage diagnostics of miter gates. In this chapter, CycleGAN is used to translate monitoring data between similar miter gate structures, and the generated synthetic observations are incorporated into Bayesian model updating for gap length estimation.

Chapter 3 extends the miter gate study by developing a Bayesian Inference-based Damage Diagnostics Enhanced through Domain Translation framework. This chapter integrates domain translation, conditional invertible neural networks, physics-based Bayesian inference, and Bayesian model averaging to improve probabilistic damage diagnostics and UQ.

Chapter 4 focuses on hydrologic forecasting by developing a hybrid modeling framework for streamflow prediction in ungauged rivers. Physics-based river-routing information is combined with ML models, and UQ methods are used to evaluate the reliability of streamflow predictions.

Chapter 5 presents a data-driven channel infilling rate forecasting framework with quantified uncertainty for the Houston Ship Channel and the Southwest Pass. This chapter develops ML-based models for predicting CIR and cumulative sedimentation volume and evaluates their performance for shoaling and sediment management.

Chapter 6 further enhances shoaling and sedimentation forecasting by proposing a recursive forecasting framework with shifted hydrologic delays for the Southwest Pass. This

chapter identifies influential upstream discharge gauges and their delayed effects, incorporates them into a multi-input multi-output LSTM-NARX model, and evaluates recursive CIR and sediment volume prediction performance.

Chapter 7 summarizes the main findings and contributions of the dissertation and discusses future research directions for ML-enabled inland waterway asset management.

CHAPTER 2

Data Augmentation Based on Image Translation for Bayesian Inference-Based Damage Diagnostics of Miter Gates

2.1 Introduction

The U.S. Army Corps of Engineers (USACE) operates 236 locks at 191 sites (see one such lock system in Fig. 2.1). More than half of these locks have surpassed their 50-year original design life. There are increasing concerns about their continued safe, reliable operation into the future, especially because routine maintenance, lock dewatering, and inspections sometimes occur at less than optimal intervals [7]. There is a pressing need for an effective damage detection and maintenance planning framework, especially for structures under the SMART Gate program, such as the Dalles Navigation Lock, Greenup Lock, and Meldahl Lock [8].

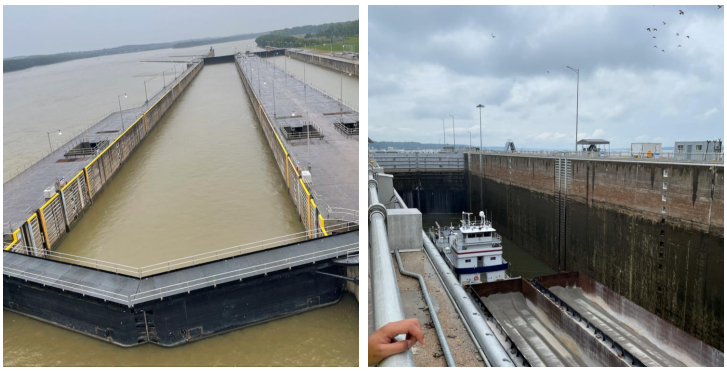


Figure 2.1: Lock system in inland waterway transportation

A critical aspect of this maintenance focuses on the quoin blocks within these structures, whose deterioration, often manifested as a loss-of-contact gap that compromises the struc-

tural integrity of miter gates, highlights the importance of optimized maintenance strategies to prevent failures and minimize downtime costs. Although critical repairs are prioritized, delayed maintenance increases the risk of failures that result in lock closures. One significant factor that contributes greatly to the difficulty of lock condition assessment is that much of the lock infrastructure typically remains underwater. When a lock is dewatered, it is common to find previously unidentified deterioration and damage. That is to say, it is difficult for the damage to be directly observed. To address such maintenance issues, Bayesian model updating-based damage detection is crucial and there is an increasing need to gather more accurate information on repair needs and to prioritize those repairs.

In the last twenty years, considerable research efforts within the discipline of structural health monitoring (SHM) have focused on developing methods for objective and quantitative evaluation of structural damage and integrity, primarily utilizing data collected from different sensors [9, 10, 11, 12, 13]. Finite element (FE) models, known for their significance in the realms of modeling and analytical methods for civil, geotechnical, and soil-structure systems under various load conditions such as static, quasi-static, and dynamic [14, 15, 16, 17, 18], represent one of the models that can be employed in SHM applications. For a given damage state, the FE model can be used as a forward model to predict the structural response. The predicted response can then be compared against field observations to inversely estimate the damage state. This process is usually referred to as Bayesian model updating. In other words, as structures change due to aging, damage, or environmental factors, FE models need to be updated.

FE model updating methods can be divided into deterministic or probabilistic approaches [3]. Probabilistic methods are favored here because they address the primary limitation of deterministic approaches, which provide only point estimates without confidence or uncertainty margins, by treating uncertain model parameters as distribution functions or intervals. This process accounts for sources of uncertainty within the updating method itself, including model form discrepancies and measurement errors [19]. Bayesian model updating stands as a commonly utilized probabilistic method for the FE model updating in SHM. This approach has been thoroughly investigated for its applications in probabilistic damage detection and system identification [20]. In Bayesian updating, the posterior distribution of uncertain model parameters is obtained by applying Bayes' theorem to combine prior distributions, which represent initial beliefs about the parameters, with likelihood functions derived from measurements, leading to an updated and probabilistically quantified determination of model parameters [21]. Bayesian updating is valuable for SHM because it allows engineers to systematically combine physical models with observational data and to make statistically informed decisions regarding the health of a structure. For example, the param-

eters of FE models for a structure can be refined by calibrating their theoretical and observed modal characteristics [22]. The Bayesian updating techniques for FE models have been successfully applied to large-scale structural systems [23, 24, 25]. Furthermore, to tackle the computational burden of performing FE Bayesian model updating, which requires repeated evaluations of physics-based simulation models like FE models during sequential damage estimation, researchers have introduced the use of surrogate models [14, 26]. Surrogate models are simplified representations that can be computed much more quickly and still approximate the functions for efficient analysis. The process of Bayesian model updating will be elaborated in Sec 3.2.2.

For some applications, the amount of monitoring data is limited due to a variety of factors [26], not only including but also extending beyond the availability of sensors, changing structural health states, and occurrences of missing data. Data required for updating may not always be available or may be costly to obtain. Such limitations may also stem from financial constraints that hinder the extensive deployment of sensor networks, logistical challenges associated with the installation and maintenance of sensors in hard-to-reach areas, or even environmental factors that may impair sensor functionality and compromise the data integrity over time [27]. Moreover, the challenge of interpreting data from structures subjected to varied operational and environmental conditions can further exacerbate the scarcity of useful data. This scarcity consequently introduces considerable uncertainty in SHM, potentially leading to biased estimates that may not optimally represent the actual condition of the structure. The resulting uncertainty can significantly influence decision-making processes, including setting maintenance and repair priorities, allocating resources, and evaluating structural lifespan. Such impacts can diminish the efficacy of SHM programs, affecting the safety and reliability of the structures being monitored.

To address the data scarcity challenge in SHM, this paper proposes a data augmentation method based on image translation for Bayesian inference-based damage diagnostics of miter gates. Data augmentation is a widely applicable, model-agnostic solution that works on the data side. It seeks to enhance the generalization capabilities of trained models by minimizing over-fitting and broadening the model's decision boundary [28]. The commonly used data augmentation methods for time-series data include random transformation, pattern mixing, generative models, decomposition, transfer learning, among others [29]. Among these methods, transfer learning is a powerful approach to address the data scarcity problem in machine learning projects and has been widely used. It allows us to leverage knowledge from previously trained models on large datasets to enhance performance on a new, similar task that may not have as much available data [30, 31]. However, transfer learning generally requires a certain degree of similarity between the source and target tasks or domains, and

often relies on labeled data in both domains. In the case study section, we will see that the lack of paired labeled data makes it unsuitable to employ traditional transfer learning techniques. Moreover, transfer learning techniques typically serve to enhance machine learning models when transferring knowledge across varied domains. This differs from our paper's goal of augmenting datasets specifically for Bayesian model updating. Consequently, this paper focuses on employing generative models as a means to expand the datasets utilized in probabilistic model updating for damage detection. Within the category of generative models, there are statistical models like Gaussian trees and posterior sampling, as well as Neural Networks such as Generative Adversarial Networks (GAN). In particular, CycleGAN, as one variant of GANs, will be studied in this paper considering its capability in creating synthetic data for target applications through unpaired training process.

This paper aims to solve multiple problems associated with generating synthetic data for Bayesian model updating: First, there is the issue of effectively leveraging data that may be limited in quantity or incomplete. This involves developing strategies to identify and use the most informative data from available observations, ensuring that the accuracy of the SHM is not compromised. Second, there is the complexity of integrating data coming from diverse sources, like distinct but possibly interrelated miter gates along a single waterway. Nonetheless, the challenge lies in translating data from one miter gate to another and learning the mapping between two distinct data domains to mitigate the issue of data scarcity. This requires the development of a robust method capable of identifying common features between two miter gates and adapting data from one structure to another. It is also important to ensure that the translation is physically meaningful and preserves the informational value of the original data. Third, applying Bayesian updating with synthetic data introduces an extra layer of complexity, which requires developing a sophisticated algorithm that can leverage both real and synthetically generated data to accurately estimate the posterior distributions of model parameters. The key contributions of this paper can be summarized as follows:

- Firstly, the paper introduces an effective method for data translation from one structural entity to another, specifically between different miter gates. This method significantly addresses the challenge of data scarcity by generating synthetic data that retains physical relevance. Moreover, it enables the application of insights gained from one structure to enhance maintenance and monitoring strategies for another, ultimately improving the understanding and health assessment of the targeted miter gate.

- Secondly, the paper proposes an intuitive approach to determine whether the generative model is performing effectively. It addresses challenges of measuring the accuracy of synthetically generated data, a task that is difficult due to: (1) the inherently unpaired

nature of the datasets and (2) the lack of real data for verification and validation.

- Lastly, and more critically, the paper demonstrates a pioneering use of synthetic data, generated through CycleGAN, to augment the robustness and accuracy of Bayesian updating processes for estimating gap lengths in miter gates. This novel application highlights the paper's primary contribution: leveraging high-quality synthetic data to improve the efficacy of Bayesian model updating-based damage diagnostics.

Next, the background of the gap damage, Bayesian model updating-based damage detection, and synthetic data generation using CycleGAN will be explained. Following that, the proposed methodology will be elaborated in detail.

2.2 Background

2.2.1 Gap damage of miter gates in lock systems

The damage of interest for the miter gates in this paper is the 'gap'—a term used to describe the loss of contact between the gate and its adjoining wall as illustrated in Fig. 2.2. This gap, unfortunately, cannot be inspected visually during operation, as the lower section of the miter gate remains submerged. The gate typically faces time-varying hydraulic loads during operation, characterized by varying upstream and downstream water levels. These loads are transmitted from the gate to the wall, from top to bottom, via quoin blocks. Our objective is to estimate the extent of this loss of contact line, also known as gap length, by employing a Bayesian model updating framework using a physics-based FE model and strain measurements.

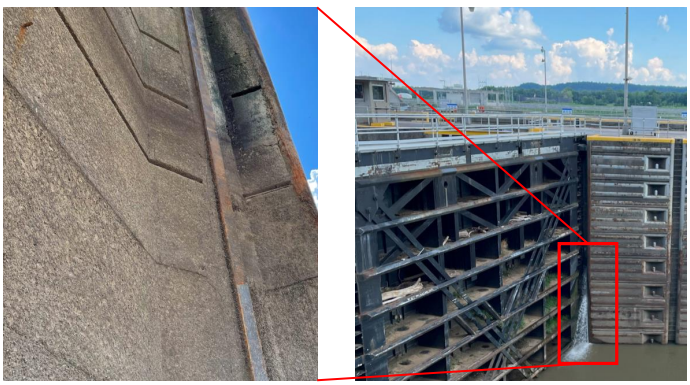


Figure 2.2: Gap-caused water leakage on miter gate

Defining the gap length as h and the downstream and upstream water levels as $\mathbf{v} = \{v_d, v_u\}$, the FE model that maps the water levels and gap length to the strain response of

the gate is given by

$$\mathbf{s}_{\mathrm{FE}} = \mathcal{F}(h, \mathbf{v}), \tag{2.1}$$

where $\mathcal{F}(\cdot)$ stands for the FE model and $\mathbf{s}_{\mathrm{FE}}$ are the predicted strain response of different nodes on the gate.

The FE model prediction is related to the strain measurements as

$$\mathbf{s} = \mathbf{s}_{\mathrm{FE}} + \varepsilon, \tag{2.2}$$

where $\varepsilon \sim N(\mathbf{0},\ \mathbf{\Sigma})$ is a vector of Gaussian noises with zero mean and covariance $\mathbf{\Sigma}$. If the model uncertainty of the FE model is not considered, the diagonal elements of the covariance matrix are assumed to be the variance σ_ε^2 of the measurement noise. Otherwise, both the mean vector and/or the covariance matrix can be assumed to be unknown [32, 33]. In this paper, we assume that the FE model has been validated and the model uncertainty is therefore negligible. Quantification of model form uncertainty is not the focus of this paper and is an important topic that is worth investigating in our future work.

2.2.2 Bayesian inference-based damage detection

In the context of SHM, particularly for infrastructure like miter gates in hydraulic facilities, understanding and predicting the evolution of damage over time is crucial to devising effective maintenance strategies, ensuring safety, and maintaining operational efficiency. These structures are exposed to deterioration from regular wear, environmental influences, and mechanical over-stress, all of which can cause them to degrade over time. For instance, the growth of gap lengths can threaten the structural integrity of these gates. However, the complex nature of these degradation mechanisms is often poorly understood in terms of precise physical degradation modeling, which poses significant challenges for accurate damage detection and prediction.

Using physics-based model analysis, Bayesian inference offers a powerful framework for addressing these challenges by updating prior knowledge using new observations, resulting in probability density function (PDF) of degradation model parameters. This approach is inherently probabilistic, allowing for direct uncertainty quantification in damage detection, which is critical for risk-informed decision-making.

Let the damage state of the miter gate at t_i be h_i, the strain measurements at t_i be $\mathbf{s}_i$, and the water levels of downstream and upstream at t_i be $\mathbf{v}_i = \{v_{d,i}, v_{u,i}\}$, we can estimate h_i using Bayesian model updating methods [32]. Since the damage state h_i changes with time, the recursive Bayesian model updating method is usually employed to dynamically estimate

and update the damage state as [26, 34]

$$f\left(h_i \mid \mathbf{s}_{1:i}, \mathbf{v}_{1:i}\right) = \frac{f\left(\mathbf{s}_i \mid h_i, \mathbf{v}_i\right) f\left(h_i \mid \mathbf{s}_{1:i-1}, \mathbf{v}_{1:i-1}\right)}{\int f\left(\mathbf{s}_i \mid h_i, \mathbf{v}_i\right) f\left(h_i \mid \mathbf{s}_{1:i-1}, \mathbf{v}_{1:i-1}\right) dh_i} \propto f\left(\mathbf{s}_i \mid h_i, \mathbf{v}_i\right) f\left(h_i \mid \mathbf{s}_{1:i-1}, \mathbf{v}_{1:i-1}\right), \tag{2.3}$$

where $\mathbf{s}_{1:i} \triangleq \{\mathbf{s}_1, \cdots, \mathbf{s}_i\}$, $\mathbf{v}_{1:i} \triangleq \{\mathbf{v}_1, \cdots, \mathbf{v}_i\}$, $f\left(\mathbf{s}_i \mid h_i\right)$ is a likelihood function which is usually derived using a physics-based FE model and considering various uncertainty sources [26], and $f\left(h_i \mid \mathbf{s}_{1:i-1}, \mathbf{v}_{1:i-1}\right)$ is given by

$$f\left(h_i \mid \mathbf{s}_{1:i-1}, \mathbf{v}_{1:i-1}\right) = \int f\left(h_i \mid h_{i-1}\right) f\left(h_{i-1} \mid \mathbf{s}_{1:i-1}, \mathbf{v}_{1:i-1}\right) dh_{i-1}. \tag{2.4}$$

Through the recursive execution of the above equations, the gap length of a miter gate can be estimated dynamically over time using strain measurement data. During a relatively short period (e.g., several weeks) and if the gap length is assumed to be an unknown constant, the gap length can be estimated using the batch Bayesian estimation method as

$$f(h|\mathbf{s}_{1:i}, \mathbf{v}_{1:i}) \propto f(\mathbf{s}_{1:i}|h, \mathbf{v}_{1:i}) f(h), \tag{2.5}$$

where $f(h)$ is the prior distribution of gap length and a non-informative uniform prior distribution can be assumed if there is no prior knowledge is available.

In this paper, it is assumed that the gap length is an unknown constant and the above batch Bayesian estimation method is employed for damage detection. Extending the proposed framework to the recursive Bayesian model updating case is a topic that is worth investigating in the future. The particle filtering (PF) method is utilized to solve the above Bayesian estimation problems by combining strain data with physics-based simulations [35]. It involves predicting the posterior distribution by resampling particles, each representing a potential state, based on calculated weights, thereby enhancing the accuracy of gap length estimations [36]. By providing a probabilistic estimate of the current state of the infrastructure, Bayesian model updating enables decision-makers to make risk-informed decisions that balance the need for the safety and longevity of critical infrastructure with the available resources.

As discussed above, observational data is a crucial component of Bayesian model updating, as it provides the primary input/information for estimating model parameters and quantifying uncertainty. However, the accuracy and effectiveness of Bayesian model updating are heavily reliant on both the quality and quantity of sensor-monitoring data. Unfortunately,

the persistent challenge of data scarcity can significantly hinder the efficacy of Bayesian model updating. To overcome this limitation, researchers have developed various techniques to enhance the datasets, including data fusion [37], data augmentation, and optimizing sensor networks for better data collection [27, 38], among others. This paper investigates the application of data augmentation techniques to improve the effectiveness of Bayesian model updating-based damage detection. Data augmentation involves generating synthetic data through the process of sampling and transformation of existing data, to increase the diversity and volume of an existing dataset.

2.2.3 Synthetic data generation using Generative Adversarial Networks (GANs)

For generating synthetic data, models based on GANs are the most widely used due to their proficiency in producing high-quality and accurate synthetic datasets [39, 40]. GAN is a class of artificial intelligence algorithms used in unsupervised machine learning, introduced by Ian Goodfellow and his colleagues in 2014 [41]. They consist of two neural networks, the generator and the discriminator, which are trained simultaneously through adversarial processes. Figure 2.3 shows the basic structure of GANs. The generator aims to produce synthetic data that is indistinguishable from real data, while the discriminator evaluates whether the data it receives is genuine or produced by the generator. Through this competition, the generator learns to produce increasingly realistic data. This framework enables the generator to create highly realistic samples as both networks improve through training. GANs have been used for various applications, including image generation, style transfer, and more due to their ability to generate realistic data samples without needing labeled data for training [40]. The minimax loss function is integral to the GANs, wherein the Generator aims to minimize this function, while the Discriminator endeavors to maximize it. The loss function is given by

$$\begin{aligned} \min_G \max_D \mathcal{L}(D,G) &= \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] \\ &+ \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1-D(G(\boldsymbol{z})))], \end{aligned} \tag{2.6}$$

where $\mathcal{L}(\cdot)$ is the loss value function, $\mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}(\cdot)$ is the expected value over all real samples, $D(\boldsymbol{x})$ represents the probability that $\boldsymbol{x}$ came from the real sample rather than the fake sample which is a single scalar, $D(G(\boldsymbol{z}))$ is the Discriminator's probability estimate if the fake generated sample is real, and $\mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}}(\cdot)$ is the expected value over inputs to the Generator.

Due to the significant potential of GANs for enhancing the detection and analysis of

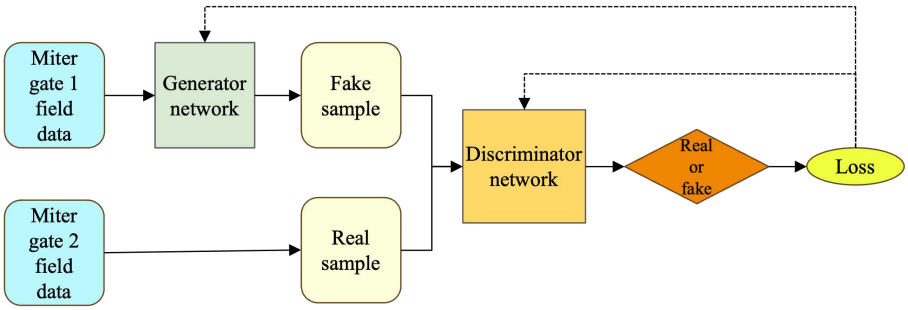


Figure 2.3: The basic structure of GANs

damage in infrastructure through the generation of synthetic sensor data, GANs are getting increased attention in the SHM domain. Leveraging historical sensor data from structures, such as miter gates, GANs are capable of synthesizing new, realistic datasets that accurately replicate a variety of operational states and damage conditions. The generation of realistic sensor data or imagery of damaged structures by GANs addresses the significant challenges associated with inadequate real-world data on damaged infrastructure [42]. For instance, Zhang *et al.* [43] utilized GAN to create synthetic data, which enhanced the training datasets for Convolutional Neural Networks (CNNs), leading to improved the CNN's ability to detect damage in laboratory truss bridges. Heesch *et al.* [44] implemented a style-based GAN to produce synthetic ultrasound wave data for a composite plate with an application of SHM. Luleci *et al.* [45] addressed the issue of data scarcity by employing a deep convolutional Wasserstein GAN with Gradient Penalty (WGAN-GP) to model the damage domain of a steel grandstand structure. The goal was to generate datasets that closely resemble actual damage, thereby enhancing the imbalanced datasets for more effective model training. However, these GAN-based data augmentation techniques are limited to single-domain applications—they generate synthetic data exclusively from their original datasets. Furthermore, the inherent limitation of GANs in requiring paired data makes them less suitable for tasks where the goal is to generate synthetic data that captures the complex, conditional relationships between different operational states, such as those found in miter gate sensor readings. Consequently, this method may yield synthetic data that lacks diversity in features.

CycleGAN, which is specifically designed for image-to-image translation tasks without requiring paired examples, thus providing a solution for scenarios where direct data pairs are unavailable. CycleGANs [46] can learn to translate between domains from unpaired datasets, making them ideal for tasks requiring the generation of synthetic data from different but related sources. For example, in a situation with limited sensor data from miter gate 1, CycleGAN can be employed to enrich this dataset by translating sensor data from miter gate 2, assuming the two miter gates share operational and environmental similarities but not iden-

tical conditions. CycleGAN and its derivatives have been employed to learn the mappings between distinct domains, accomplishing domain translation. This advancement has led to successful applications in data augmentation within the field of SHM [47]. Nevertheless, the majority of these methods are confined to generating synthetic data, which in turn only supports basic damage detection tasks such as classification or localization. To advance toward probabilistic damage quantification, there is a need for a more in-depth exploration of how synthetic data can be integrated with the Bayesian model updating framework.

2.3 Proposed Method

2.3.1 Overview

Figure 3.3 provides an overview of the proposed framework utilizing CycleGAN to enhance the Bayesian Inference process for estimating the damage status (i.e., gap length) of miter gates. As illustrated in Fig. 3.3, the quantities of interest of this paper are two similar but structurally different miter gates (i.e., referred to as "Miter gate 1" and "Miter gate 2"). We denote the strain measurements of the two gates as

$$\begin{aligned} \mathbf{s}^{(1)} &\triangleq \{\mathbf{s}_{11}, \cdots, \mathbf{s}_{1N}\} \in \mathbb{R}^{N \times N_s}, \\ \mathbf{s}^{(2)} &\triangleq \{\mathbf{s}_{21}, \cdots, \mathbf{s}_{2N}\} \in \mathbb{R}^{N \times N_s}, \end{aligned} \tag{2.7}$$

where $\mathbf{s}^{(i)}$ are all the measurements of the i-th gate, $\mathbf{s}_{ij} \equiv \{s_{ij1}, \cdots, s_{ijN_s}\} \in \mathbb{R}^{1 \times N_s},\ \forall i = 1, 2;\ j = 1, \cdots, N$ are the strain measurements of the i-th gate at time step j, N is the total number of time steps, and N_s represents the total number of strain gauges.

In addition to the strain measurements, we also have measurement data of water levels on the two gates as $\mathbf{v}^{(1)} \triangleq \{\mathbf{v}_{11}, \cdots, \mathbf{v}_{1N}\} \in \mathbb{R}^{N \times 2}$ and $\mathbf{v}^{(2)} \triangleq \{\mathbf{v}_{21}, \cdots, \mathbf{v}_{2N}\} \in \mathbb{R}^{N \times 2}$, where $\mathbf{v}_{ij} \triangleq \{v_{d,ij}, v_{u,ij}\} \in \mathbb{R}^{1 \times 2}$ are the downstream and upstream water levels of the i-th gate at time step j. Based on the above definitions of the measurements, the overall objective of the proposed method can be summarized as:

Using data $\mathbf{s}^{(1)}$ and $\mathbf{v}^{(1)}$ of "Miter gate 1" to augment data $\mathbf{s}^{(2)}$ and $\mathbf{v}^{(2)}$ of "Miter gate 2", and thus increase the confidence of Bayesian inference-based damage estimation (i.e., gap length) of "Miter gate 2" as described in Sec. 3.2.2.

As mentioned in Sec. 3.2.3 and illustrated in Fig. 3.3, the above objective is accomplished through CycleGAN. It consists of four primary modules, namely (1) data pre-processing, (2) synthetic data generation, (3) accuracy verification of synthetically generated data, and (4) augmenting Bayesian Inference with synthetic data. In what follows, we explain each

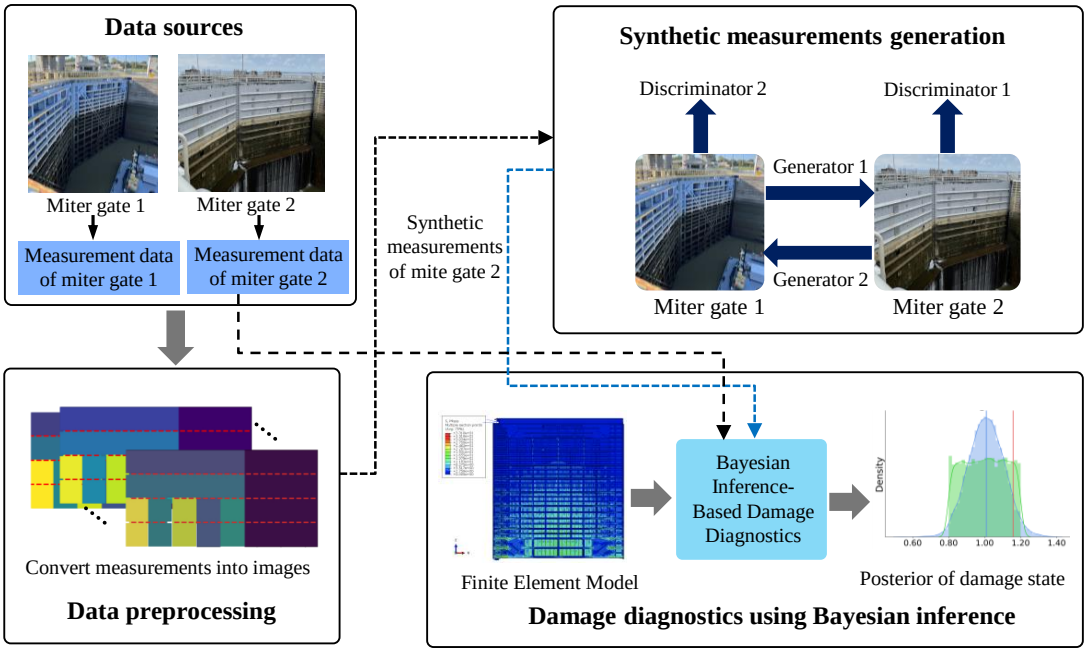


Figure 2.4: Overview of the proposed framework

module in detail.

2.3.2 Data pre-processing

Before we discuss the details of the two CycleGAN model architectures studied in this paper, we perform data pre-processing to convert measurement data into a format that is necessary for the training of CycleGAN. Figure 2.5 shows the process of data pre-processing. For a given time step j and taking the measurement data of the i-th gate as an example, we have strain and water-level measurements as $\mathbf{s}_{ij}$ and $\mathbf{v}_{ij}$.

2.3.2.1 Sorting

Before the data is transformed into an image format, it undergoes a sorting process. In this study, the data is sorted by the upstream value across the time steps. This arrangement helps establish a preliminary correspondence between datasets from the two miter gates, underlining the crucial relationships based on hydrostatic differentials at each time step. This aspect will be further developed in Sec. 6.4.

2.3.2.2 Converting to images

For any given time step, the sorted data then goes through another preprocessing technique, which converts both spatially constant variables (hydrostatic loads) and the spatially-varying variable (strain meansurement data) into images for CycleGAN input. This is a method supported by similar research endeavors. For example, Sood *et al.* present an approach by converting 1D numeric time series into 2D images for visual forecasting, using deep-learning techniques, which enhances the prediction accuracy for complex time series data [48]. By converting two different data sources into a unified data matrix, this method aims to exploit CycleGAN's ability to handle complex image data to reveal patterns and relationships within the data matrix. The initial input matrix of "Miter gate 1" data can be denoted as

$$(\mathbf{s}_{1j}, \mathbf{v}_{1j}) = \begin{bmatrix} v_{d,1j}, & v_{d,1j}, & \cdots, & v_{d,1j} \\ v_{u,1j}, & v_{u,1j}, & \cdots, & v_{u,1j} \\ s_{1j1}, & s_{1j2}, & \cdots, & s_{1jN_s} \end{bmatrix} \in \mathbb{R}^{3\times N_s}. \tag{2.8}$$

Respectively, the initial input matrix of "Miter gate 2" data is denoted as

$$(\mathbf{s}_{2j}, \mathbf{v}_{2j}) = \begin{bmatrix} v_{d,2j}, & v_{d,2j}, & \cdots, & v_{d,2j} \\ v_{u,2j}, & v_{u,2j}, & \cdots, & v_{u,2j} \\ s_{2j1}, & s_{2j2}, & \cdots, & s_{2jN_s} \end{bmatrix} \in \mathbb{R}^{3\times N_s}. \tag{2.9}$$

2.3.2.3 Padding

The initial input is adjusted and reformatted to be compatible with CycleGAN's training requirements. Padding is added to the input matrix to counteract the reduction of pixels at the image's edges due to convolutional layer application. This loss is minimal per convolution but accumulates across multiple layers. Padding effectively enlarges the image's boundary, ensuring peripheral pixels are utilized and preserving the image's dimensions through successive convolutions. Subsequently, the input undergoes generalization, transforming the generalized data into an image format conducive to CycleGAN's learning process.

Through the data pre-processing, we convert the measurement data at each time step into an image. For the sake of explanation, we represent the overall process as

$$\begin{aligned} \mathcal{X}_j &= \mathcal{Z}(\mathbf{s}_{1j}, \mathbf{v}_{1j}), j = 1, \cdots, N, \\ \mathcal{Y}_j &= \mathcal{Z}(\mathbf{s}_{2j}, \mathbf{v}_{2j}), j = 1, \cdots, N, \end{aligned} \tag{2.10}$$

where $\mathcal{X}_j$ and $\mathcal{Y}_j$ are respectively the image of miter gate 1 and miter gate 2 at time step j and $\mathcal{Z}(\cdot)$ stands for the overall data pre-processing step.

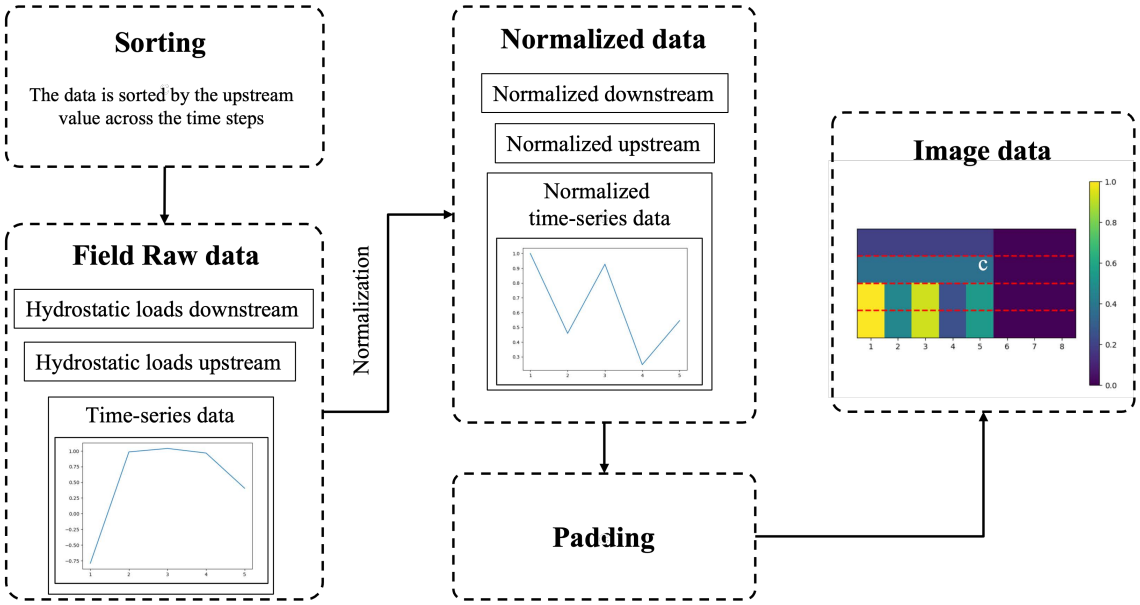


Figure 2.5: The process of converting raw data to image data

2.3.3 CycleGAN for Synthetic Data Generation

To address the challenges in synthetic data generation highlighted previously in Sec. 3.2.3, this paper employs CycleGAN. Zhu *et al.* first propose CycleGAN for solving unpaired image to image translation problems [46]. By learning to translate between two unpaired image domains, CycleGAN enables the generation of images in one style from images in another style, such as turning horses into zebras or converting summer scenes into winter. This is achieved through the use of two generator-discriminator pairs, where each pair learns to translate images from one domain to the other and vice versa, along with a cycle consistency loss to ensure that an image translated from one domain to the other can be translated back to the original domain. Figure 2.6 shows the basic structure of CycleGAN. The design of the Generator and Discriminator in CycleGAN can determine the performance of the model through several tests we have done. Subsequent sections will detail the implementation of two distinct sets of Generators and Discriminators utilized in this study. We design the two popular and powerful model architectures for CycleGAN, namely (1) convolutional U-net and (2) Self-organized Operational Neural Network U-net, to investigate the effect of structures on synthetic data generation and Bayesian model updating.

2.3.3.1 Convolutional U-net generator for cycleGAN

The Convolutional U-Net architecture, when adapted for a CycleGAN generator, embodies a powerful tool for handling complex image-to-image translation tasks, such as those

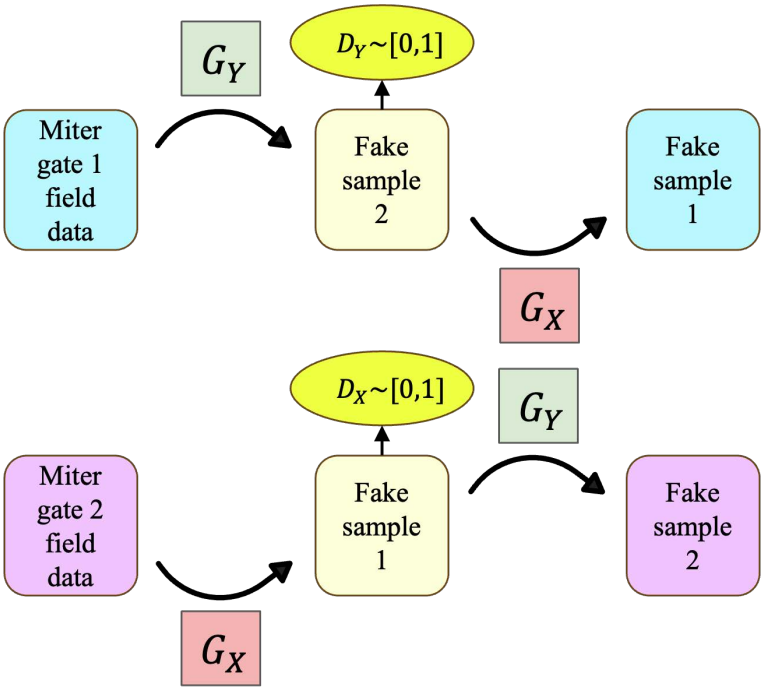


Figure 2.6: The structure of CycleGAN

encountered in synthetic data generation for damage detection or sensor data enhancement. The U-Net model, originally conceived for biomedical image segmentation, is characterized by its U-shaped structure [49]. It comprises a contracting path to capture context and a symmetric expanding path that enables precise localization. This design facilitates the detailed analysis of images, making it adept at understanding and generating nuanced features within data—attributes that are crucial for creating realistic synthetic images or sensor readings. In the context of CycleGAN, incorporating a U-Net architecture as the generator allows for more nuanced handling of spatial hierarchies and details within images. This specificity is achieved through the U-Net's ability to combine high-level, contextual information from the contracting path with local, detailed information from the expanding path, ensuring that the synthetic images it generates are both contextually coherent and rich in detail. Figure 2.7 shows the structure of the U-net layers used in this paper. The loss function of CycleGAN with a Convolutional U-net generator contains three types of terms: adversarial losses, cycle consistency loss, and identity loss. The adversarial losses are expressed as [46]

$$\begin{aligned}\mathcal{L}_{\text{GAN}}\left(G_Y, D_Y, X, Y\right) = &\mathbb{E}_{\mathcal{Y}\sim p_{\text{data}}\,(\mathcal{Y})}\left[\log D_Y(\mathcal{Y})\right] \\ &+ \mathbb{E}_{\mathcal{X}\sim p_{\text{data}}\,(\mathcal{X})}\left[\log\left(1 - D_Y(G_Y(\mathcal{X}))\right)\right],\end{aligned} \tag{2.11}$$

where $X = \{\mathcal{X}_1, \cdots, \mathcal{X}_N\}$, $Y = \{\mathcal{Y}_1, \cdots, \mathcal{Y}_N\}$, $G_Y(\cdot)$ is the generator that generates synthetic

data for miter gate 2. $D_Y(\cdot)$ is the discriminator that distinguishs between translated samples $\hat{\mathcal{Y}}_j = G_Y(\mathcal{X}_j)$ and real sample $\mathcal{Y}_j$. The data distributions are denoted as $\mathcal{X} \sim p_{\text{data}}(\mathcal{X})$ and $\mathcal{Y} \sim p_{\text{data}}(\mathcal{Y})$.

Similar to the GANs loss function described in Sec. 3.2.3, G_Y strives to minimize a specific goal in opposition to D_Y, which seeks to maximize the same goal, establishing an adversarial dynamic between them. The adversarial loss facilitates the transformation from domain X to Y, and there exists a comparable loss function for the inverse transformation from Y to X, ensuring bidirectional learning between the two domains. Another adversarial loss function is given as [46]

$$\begin{aligned}\mathcal{L}_{\text{GAN}}(G_X, D_X, Y, X) = \mathbb{E}_{\mathcal{X}\sim p_{\text{data}}(\mathcal{X})}[\log D_X(\mathcal{X})] \\ + \mathbb{E}_{\mathcal{Y}\sim p_{\text{data}}(\mathcal{Y})}[\log(1 - D_X(G_X(\mathcal{Y})))]\end{aligned}, \tag{2.12}$$

where $G_X(\cdot)$ is the generator that generates synthetic data for miter gate 1. $D_X(\cdot)$ is the discriminator that distinguishes between translated samples $\hat{\mathcal{X}}_j = G_X(\mathcal{Y}_j)$ and real sample $\mathcal{X}_j$.

The second part is the cycle consistency loss, which is given as:

$$\begin{aligned}\mathcal{L}_{\text{cyc}}(G_Y, G_X) = \mathbb{E}_{\mathcal{X}\sim p_{\text{data}}(\mathcal{X})}[\|G_X(G_Y(\mathcal{X})) - \mathcal{X}\|_1] \\ + \mathbb{E}_{\mathcal{Y}\sim p_{\text{data}}(\mathcal{Y})}[\|G_Y(G_X(\mathcal{Y})) - \mathcal{Y}\|_1]\end{aligned}, \tag{2.13}$$

where $\|\cdot\|_1$ denotes the L1 norm, which penalizes the absolute differences between the original and reconstructed images.

The last part is the identity mapping loss which is given by [46]

$$\mathcal{L}_{\text{id}}(G_Y, G_X) = \|G_Y(\mathcal{Y}) - \mathcal{Y}\|_1 + \|G_X(\mathcal{X}) - \mathcal{X}\|_1. \tag{2.14}$$

Therefore, the full objective is:

$$\begin{aligned}\mathcal{L}(G_X, G_Y, D_X, D_Y) = \mathcal{L}_{\text{GAN}}(G_Y, D_Y, X, Y) \\ + \mathcal{L}_{\text{GAN}}(G_X, D_X, Y, X), \\ + \mathcal{L}_{\text{cyc}}(G_Y, G_X) + \mathcal{L}_{\text{id}}(G_Y, G_X),\end{aligned} \tag{2.15}$$

and we aim to solve

$$G_X^*, G_Y^* = \arg\min_{G_X, G_Y} \max_{D_X, D_Y} \mathcal{L}(G_X, G_Y, D_X, D_Y). \tag{2.16}$$

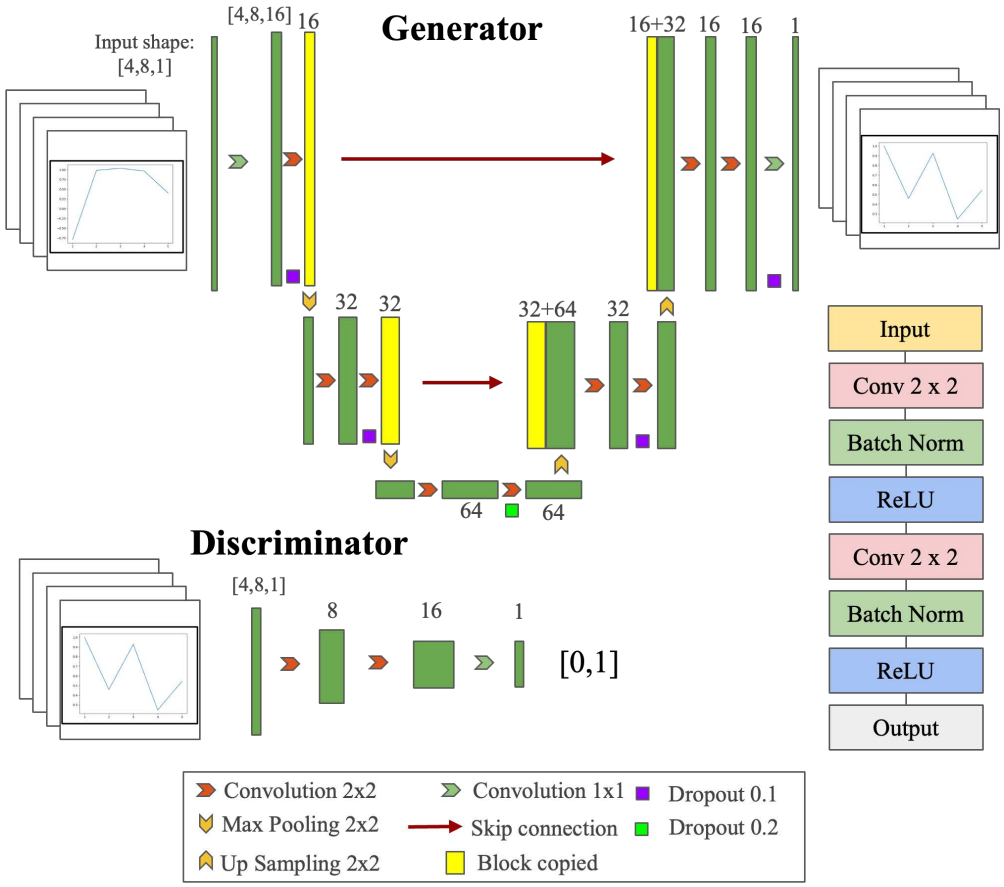


Figure 2.7: The structure of Convolutional U-net generator and discriminator of CycleGAN

2.3.3.2 Self-organized Operational Neural Network (Self-ONN) U-net generator for CycleGAN

In addition to the above CycleGAN architecture with Convolutional U-net generator, we also studied another CycelGAN architecture where the convolutional U-net generator is replaced with a self-organized operational Neural Network (Self-ONN).

ONN marks a significant advancement in addressing the limitations found in conventional CNNs, which typically utilize a uniform linear neuron model. Distinct from CNNs, ONNs offer a diverse network design that features a generalized neuron model capable of integrating a variety of non-linear operators. This flexibility significantly improves the network's ability in processing complex and multi-modal functions or domains, allowing for simpler network designs and less reliance on extensive datasets. To advance this innovation, Kiranyaz *et al.* [50] have developed Self-organized ONNs (Self-ONNs), a novel architecture that includes generative neurons. These neurons have the unique ability to independently refine the operator at each junction during training. This breakthrough permits Self-ONNs to reach an unmatched degree of diversity, precisely aligned with the requirements of the specific learning task. Furthermore, this self-adaptive method removes the need for a fixed library of operator sets and the initial search for the most suitable operators, thus making the learning process more efficient. By substituting the CNNs architecture with self-ONNs in CycleGAN, it has the potential to boost the model's capability in processing and analyzing time-series data.

Figure 2.8 shows the structure of Self-ONN of CycleGAN. In the CycleGAN model used in this paper, both the generators and discriminators adhere to an identical architecture design. The generator network is designed based on the U-Net configuration, which includes five 1D operational layers for the encoding phase and an additional five 1D transposed operational layers for the decoding phase. A kernel size of 5 is employed across all layers, with the exception of the final transposed convolutional layer with a kernel size of 6. Uniformly, for all ten layers, a stride of 2 and a padding of 1 are applied. The input data is then processed through a Tanh activation layer to finalize the transformation. We also define the coefficient q within the generator's operational layers, which controls the extent of non-linearity, as 3. The discriminator networks comprise six operational layers, each featuring a kernel size of 4. The stride across these layers follows a specific sequence: 4, 4, 2, 2, 1, and 2.

2.3.4 Accuracy Verification of Synthetically Generated Data

Prior to utilizing the synthetically generated observation data to enhance Bayesian model updating, it is essential to validate the accuracy/quality of the synthetic data. In the past decade, various metrics have been proposed to verify the accuracy/quality of images gen-

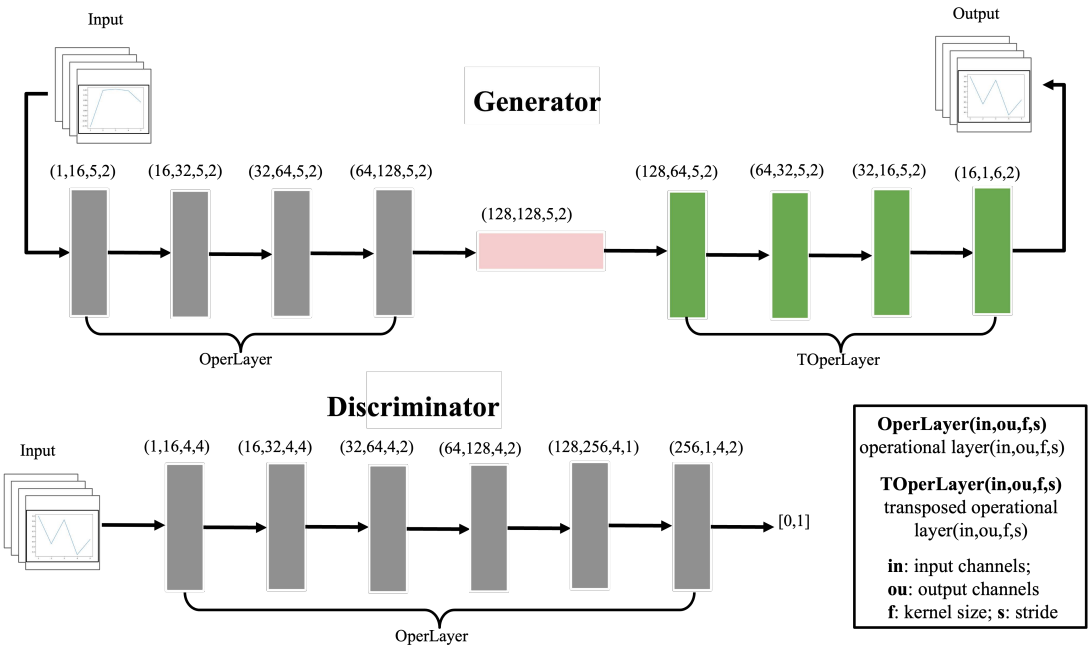


Figure 2.8: The structure of Self-ONN of CycleGAN

erated using generative models. The commonly used metrics include correlation coefficient (CC), Structural Similarity (SSIM) Index [51], and the Fréchet inception distance (FID) [52]. Although these metrics offer different insights into the quality of the synthetic images, they may not be able to quantify the capacity of the synthetically generated data to accurately capture the complex relationships between inputs and outputs, such as the connection between variables $\mathbf{v}$ and $\mathbf{s}$ in the case of the miter gate. For successful augmenting Bayesian model updating using synthetic data, it is critical to ensure that the synthetic data not only replicates but also preserves the functional correspondence between inputs (such as $\mathbf{v}$) and corresponding outputs (such as $\mathbf{s}$). To this end, this paper proposes an accuracy verification framework to verify the quality of the synthetically generated data, in addition to the commonly used metrics as mentioned above.

Figure 2.9 shows an overview of the implementation procedure of the proposed accuracy verification framework. It is inspired by the model verification and validation of computational simulation models [53]. As illustrated in this figure, the main idea is to examine the similarity between two error distributions. The first error distribution models the difference/discrepancy between the real observation data and its corresponding posterior predictions from a physics-based FE model, while the second one captures its counterpart for the synthetically generated strain data with its corresponding posterior predictions from the same FE model. If these two error distributions are similar, we can accept the synthetically

generated data for Bayesian model updating. The rational behind using the error distributions to access the accuracy/quality of synthetic data are twofold: (1) the error/discrepancy between the prediction and observation can reflect how well the relationship between the inputs and outputs is captured in the synthetic data; (2) Bayesian model updating essentially is to minimize the discrepancy/error between the prediction of the FE model and the real-world observations. If the error distribution of the synthetic data is very similar to that of the real data, it indicates that the data well-preserved the uncertainty/error of the model. Consequently, this high degree of similarity validates the suitability of synthetic data for Bayesian model updating. The remainder of this section will provide a more detailed elaboration of this approach.

Let us recall that we denote strain observations and water level data from t_1 to t_N for miter gate 2 as $\mathbf{s}^{(2)} \triangleq \{\mathbf{s}_{21}, \cdots, \mathbf{s}_{2N}\} \in \mathbb{R}^{N\times N_s}$ and $\mathbf{v}^{(2)} \triangleq \{\mathbf{v}_{21}, \cdots, \mathbf{v}_{2N}\} \in \mathbb{R}^{N\times N_s}$ respectively, as indicated in Fig. 2.9, the observation data is first used to update the estimation of the unknown gap length h as

$$f(h|\mathbf{s}^{(2)}, \mathbf{v}^{(2)}) \propto f(\mathbf{s}^{(2)}|h, \mathbf{v}^{(2)})f(h), \tag{2.17}$$

where $f(\mathbf{s}^{(2)}|h, \mathbf{v}^{(2)})$ is the likelihood of observing $\mathbf{s}^{(2)}$ for given water level data $\mathbf{v}^{(2)}$ (see Sec. 3.2.2 for detailed discussion of Bayesian model updating).

After that, we have the posterior samples of h and denote the samples as $h_{p,1}, \cdots, h_{p,N_B}$, where N_B is the number of posterior samples. For each $\mathbf{v}_{2j} \in \mathbf{v}^{(2)}$, $\forall j = 1, \cdots, N$, we plug $\mathbf{v}_{2j}$ and $h_{p,k}$ into Eq. (2.1) to obtain the posterior prediction of the strain responses corresponding to the observations as

$$\mathbf{s}_{\mathrm{FE},2jk} = \mathcal{F}_2(h_{p,k}, \mathbf{v}_{2j}), \forall j = 1, \cdots, N; k = 1, \cdots, N_B, \tag{2.18}$$

where $\mathcal{F}_2(\cdot)$ is the FE model of miter gate 2, $\mathbf{s}_{\mathrm{FE},2jk} \triangleq \{s_{\mathrm{FE},2jk1}, \cdots, s_{\mathrm{FE},2jkN_s}\} \in \mathbb{R}^{1\times N_s}$, and $s_{\mathrm{FE},2jkq}, q = 1, \cdots, N_s$ is the posterior prediction of the strain for the q-th strain gauge on miter gate 2 at t_j based on the k-th posterior sample of h.

The prediction error of the posterior prediction for the q-th strain gauge can then be computed as (see Fig. 2.9)

$$\begin{aligned} &\varepsilon_{2jq} = \{s_{2jq} - s_{FE,2jkq},\ \forall k = 1, \cdots, N_B\} \in \mathbb{R}^{1\times N_B}, \\ &j = 1, \cdots, N;\ q = 1, \cdots, N_s, \end{aligned} \tag{2.19}$$

where s_{2jq} is the strain measurements of miter gate 2 at time t_j from the q-th strain gauge (see definition given in Eq. (2.7)). Based on this, we can obtain the error distribution of

different strain gauges for posterior predictions corresponding to the real observations.

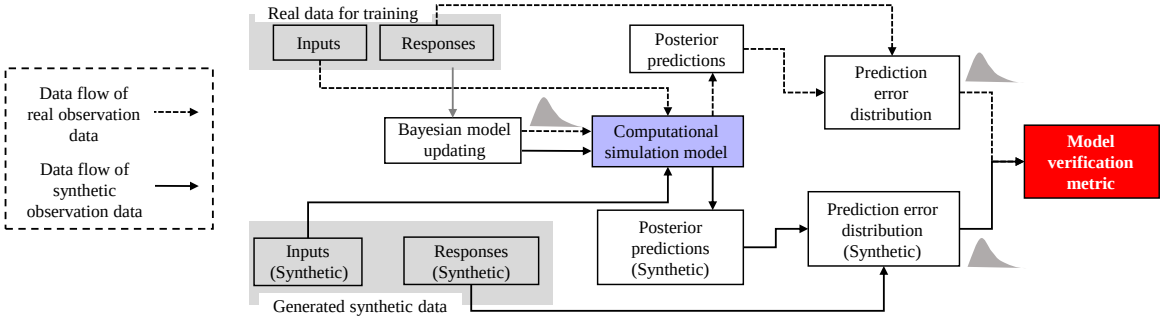


Figure 2.9: Flowchart of accuracy verification of synthetically generated observations

Assume that we have used $\mathbf{v}^{(2)}$ and $\mathbf{s}^{(2)}$ to train a CycleGAN model by following the procedure described in Secs. 2.3.2 through 2.3.3, the trained model can then be used to generate synthetic observations. Let the new real observation data from the two miter gates be

$$\begin{aligned} \mathbf{s}_{new}^{(1)} &\triangleq \{\mathbf{s}_{1(N+1)}, \cdots, \mathbf{s}_{1M}\}, \\ \mathbf{s}_{new}^{(2)} &\triangleq \{\mathbf{s}_{2(N+1)}, \cdots, \mathbf{s}_{2M}\}, \end{aligned} \tag{2.20}$$

where $M \geq (N+1)$.

We can then use the trained CycleGAN models to translate the observation data of miter gate 1 to that of miter gate 2 as follows

$$\hat{\mathcal{Y}}_j = G_Y(\mathcal{X}_j), \tag{2.21}$$

where $\mathcal{X}_j, \forall j = N+1, \cdots, M$ is the pre-processed observation data of miter gate 1.

The synthetically generated images $\hat{\mathcal{Y}}_j, \forall j = N+1, \cdots, M$ are then inverse transformed from the normalized domain into the original domain of strain and water level measurements. Let the final synthetic strain and water level measurements of miter gate 2 be respectively $\{\mathbf{s}_{\text{sy},(N+1)}, \cdots, \mathbf{s}_{\text{sy},M}\}$ and $\{\mathbf{v}_{\text{sy},(N+1)}, \cdots, \mathbf{v}_{\text{sy},M}\}$, we can then verify the accuracy/quality of the data by following the procedure as illustrated in Fig. 2.9, before applying it to Bayesian model updating for structural health monitoring.

Similar to how we compute prediction errors of the posterior predictions corresponding to the real observations (i.e., Eqs. (2.18) and (2.19)), we first plug the synthetic water levels $\mathbf{v}_{\text{sy},j}, \forall j = N+1, \cdots, M$ and posterior sample $h_{p,k}$ from Eq. (2.17) into Eq. (2.1) to obtain

the posterior prediction of strains corresponding to the synthetic observations as

$$\begin{aligned}&\mathbf{s}_{\text{FE-sy},jk} = \mathcal{F}_2(h_{p,k}, \mathbf{v}_{\text{sy},j}),\\&\forall j = N+1, \cdots, M; k = 1, \cdots, N_B,\end{aligned} \tag{2.22}$$

where $\mathbf{s}_{\text{FE-sy},jk} \triangleq \{s_{\text{FE-sy},jk1}, \cdots, s_{\text{FE-sy},jkN_s}\} \in \mathbb{R}^{1\times N_s}$, and $s_{\text{FE-sy},jkq}, q = 1, \cdots, N_s$ is the posterior prediction of the strain correspond to the synthetic observations for the q-th strain gauge on miter gate 2 at t_j based on the k-th posterior sample.

After that, the prediction error of the posterior prediction corresponding to the synthetic observations for the q-th strain gauge can be computed similar to Eq. (2.19) as

$$\begin{aligned}&\varepsilon_{\text{sy},jq} = \{s_{\text{sy},jq} - s_{\text{FE-sy},jkq},\ \forall k = 1, \cdots, N_B\} \in \mathbb{R}^{1\times N_B},\\&j = N+1, \cdots, M;\ q = 1, \cdots, N_s.\end{aligned} \tag{2.23}$$

Once we have samples of the error/discrepancy of the posterior predictions from Eqs. (2.19) and (2.23) respectively for the real observations and synthetic observations, the error distributions can be compared using certain metrics, such as the Kullback–Leibler (KL) divergence (i.e., following the procedure illustrated in Fig. 2.9). This metric can then be used as a measure to quantify the quality of the generated synthetic data for Bayesian model updating purpose. If the two error distributions are determined to be close enough based on the metric, the synthetic data can then be used for augmenting Bayesian model updating in Sec. 2.3.5. Otherwise, we can use only the real observations for model updating (as indicated in Fig. 2.10).

2.3.5 Bayesian Model Updating Based on Data Augmentation

As mentioned above and indicated in Fig. 2.10, we can use synthetic data to augment Bayesian model updating if the synthetic data can pass the accuracy verification step. This is particularly valuable in scenarios where actual data on certain types of damage or at specific stages of degradation are limited. The synthetic data can provide additional insights into how the structure might behave under various unobserved conditions, effectively expanding the dataset used for damage assessment.

In Bayesian inference, prior knowledge plays a critical role in shaping the posterior distributions of model parameters. Synthetic data can be used to refine these priors, especially when empirical data are insufficient to accurately represent the range of possible structural states. By incorporating synthetic observations, the priors can be adjusted to more accurately reflect the diversity and complexity of potential damage mechanisms. In this paper,

the Bayesian model updating based on data augmentation process is divided into two stages as shown in Fig. 2.10. Initially, the process begins by solely utilizing field data to perform Bayesian model updating. The posterior probability density function of the gap length h_M at time step M is given by

$$\begin{aligned} f(h_M|\mathbf{s}_o, \mathbf{v}_o) &= \frac{f(\mathbf{s}_o|\mathbf{v}_o, h_M) f(h_M)}{\int f(\mathbf{s}_o|\mathbf{v}_o, h_M) f(h_M) dh_M}, \\ &\propto f(\mathbf{s}_o, \mathbf{v}_o|h_M) f(h_M), \end{aligned} \tag{2.24}$$

where $\mathbf{s}_o = \left[\mathbf{s}^{(2)}, \mathbf{s}^{(2)}_{new}\right]$, $\mathbf{v}_o = \left[\mathbf{v}^{(2)}, \mathbf{v}^{(2)}_{new}\right]$, and $f(h_M)$ is the prior distribution of gap length. In this paper, a uniform prior distribution is used since all the observations are used for model updating in a batch.

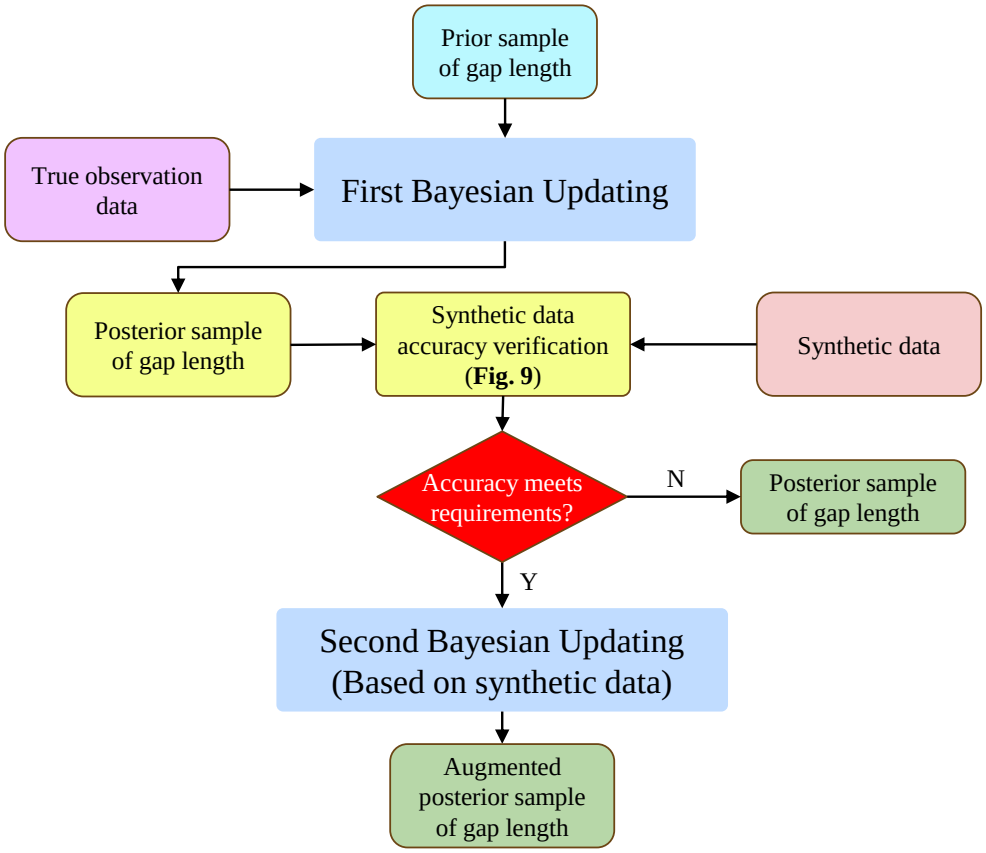


Figure 2.10: Bayesian Model Updating Augmentation with synthetic data

Following this, the method incorporates synthetic data, with the Bayesian model updating in this second phase starting from the posterior sample obtained in the initial stage. The

posterior probability density function of the second part is given by

$$f(h_M|\mathbf{s}_{\text{all}}, \mathbf{v}_{\text{all}}) = \frac{f(\mathbf{s}_{\text{sy}}|\mathbf{v}_{\text{sy}}, h_M) f(h_M|\mathbf{s}_o, \mathbf{v}_o)}{\int f(\mathbf{s}_{\text{sy}}|\mathbf{v}_{\text{sy}}, h_M) f(h_M|\mathbf{s}_o, \mathbf{v}_o) dh_M}, \\ \propto f(\mathbf{s}_{\text{sy}}|\mathbf{v}_{\text{sy}}, h_M) f(h_M|\mathbf{s}_o, \mathbf{v}_o), \tag{2.25}$$

in which $\mathbf{s}_{\text{all}} = [\mathbf{s}_o, \mathbf{s}_{\text{sy}}]$, $\mathbf{v}_{\text{all}} = [\mathbf{v}_o, \mathbf{v}_{\text{sy}}]$, $\mathbf{s}_{\text{sy}} \triangleq \{\mathbf{s}_{\text{sy},(N+1)}, \cdots, \mathbf{s}_{\text{sy},M}\}$, and $\mathbf{v}_{\text{sy}} \triangleq \{\mathbf{v}_{\text{sy},(N+1)}, \cdots, \mathbf{v}_{\text{sy},M}\}$.

In this paper, the particle filtering method described in Sec. 3.2.2 is employed to complete the two-stage Bayesian model updating process. Upon completing the modeling updating using field data and synthetic data, we obtained the posterior samples of gap length specifically tailored for miter gate 2. This dual-phase updating enhances the estimation process by leveraging both actual field data and generated synthetic data.

2.4 Case Study

2.4.1 Problem description

In this case study, two miter gates are selected to demonstrate the effectiveness of the proposed method. Figure 2.11 shows the physical structure of the first miter gate and its finite element model. Following that, Fig. 2.12 presents the stress response of the first miter gate (i.e., miter gate 1) obtained from its FE model for a given water level.

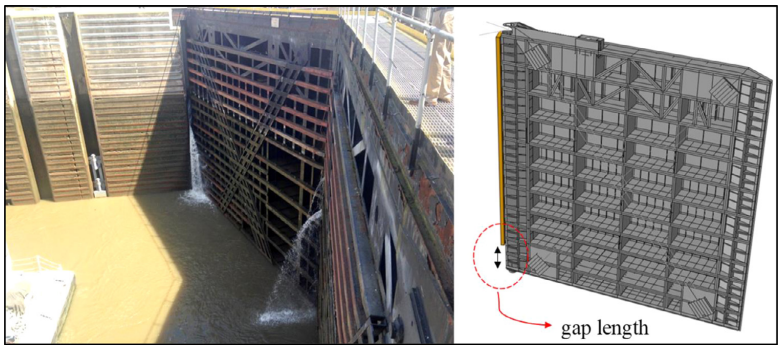


Figure 2.11: Miter Gate and physical-based FE model

The strain response of the second miter gate is created hypothetically by modifying that of the first miter gate so that the two miter gates will share some similarities, but be different. The strain response of the second miter gate is given by

$$s_{2jk} = s_{1jk}\lambda_{jk}, \tag{2.26}$$

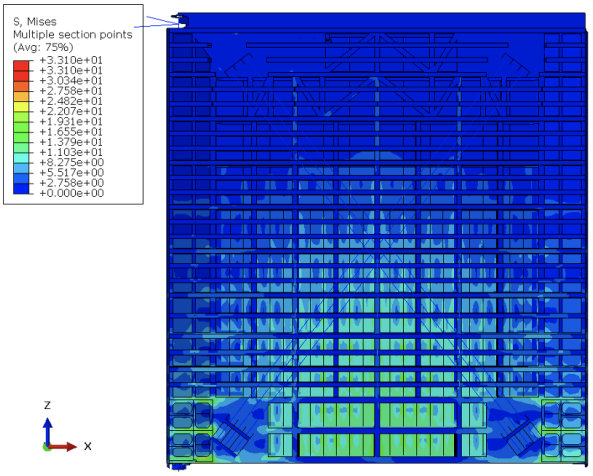


Figure 2.12: A finite element model for the miter gate.

where $k = 1, \cdots, 5$ (i.e., $N_s = 5$), λ_{jk} is a coefficient which is a nonlinear function of water levels of the second miter gate and sensor index k as follows

$$\lambda_{jk} = \left| \frac{\sin(k/5)}{\cos(2k/(v_{u,2j} - v_{d,2j})) + 1.5} \right| + 1.4 \left| \cos\left(\frac{30k}{v_{u,2j} - v_{d,2j}} \right) \right|^2. \tag{2.27}$$

The strain responses of the two miter gates (i.e., miter gate 1 and miter gate 2) are governed by damage states (i.e., gap length) and water levels. In this case study, it is assumed that the gap length (i.e., h) of miter gate 1 is 50 inches and that of miter gate 2 is 130 inches, but the gap lengths of both miter gates are unknown and only used to generate observations to verify the proposed method. Additionally, the water levels of the two miter gates are different and are generated using an Autoregressive Moving Average (ARMA) model. It is also assumed that we only have five strain gauges on each miter gate (i.e., $N_s = 5$). The measurement noises of the strain gauges on both gates are assumed to follow Gaussian distributions with zero mean and standard deviation of 4×10^{-6}. Figure 2.13 depicts the strain observations of the two miter gates after standardization using the same mean and standard deviation. This figure allows us to analyze the variance in data between these two miter gates. By selecting the data from a sensor at a random time step, we observe that both the shape and the value of the sensor data differ, each displaying a unique pattern. Even though the strain responses are quite different between these two gates, they share some similarities.

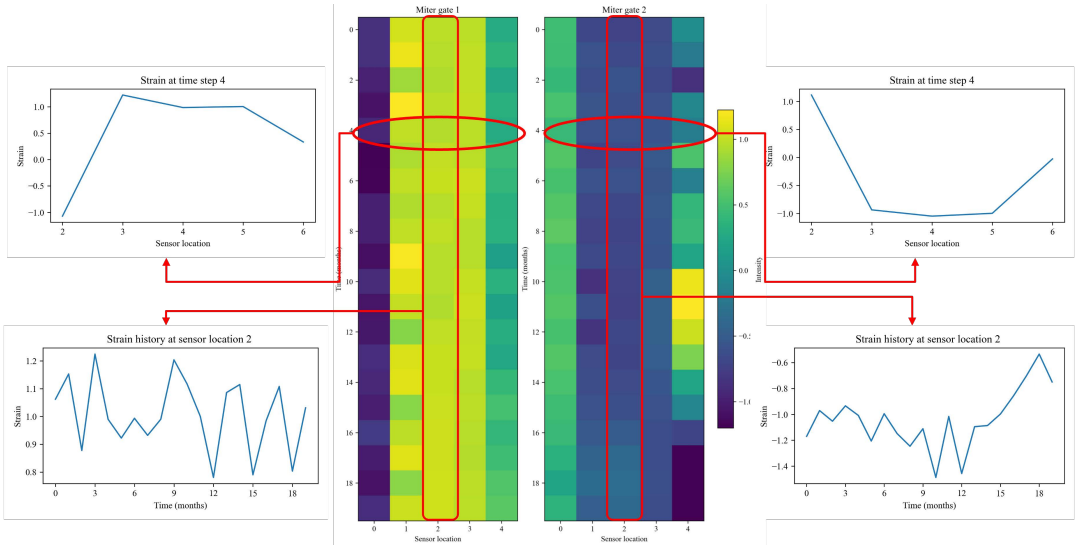


Figure 2.13: Strain observations from sensors comparison between miter gate 1 and 2

2.4.2 Training of CycleGAN and performance evaluation of the training

Figure 2.14 illustrates the inputs and outputs used for training and test stages in this case study. In the training phase, the model learns the relationship between data from two different miter gates by using data from miter gate 1 as input and data from miter gate 2 as output. However, during the test/validation phase, only new data from miter gate 1 will be used to generate synthetic data of miter gate 2. This approach ensures that the synthetic data produced by the model reflects the translated information from miter gate 1, without any direct knowledge of the actual data of miter gate 2.

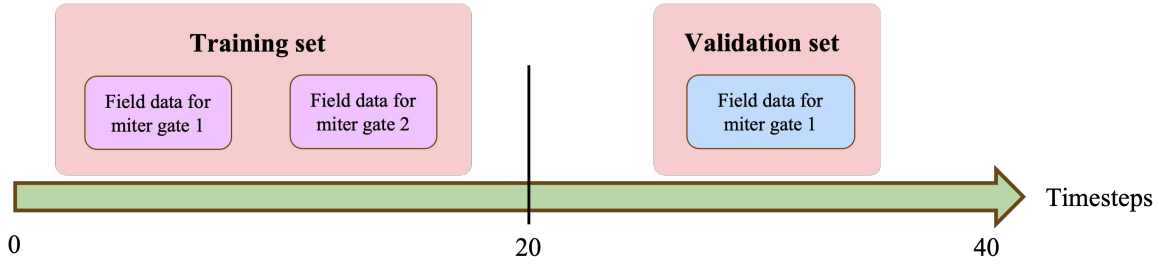


Figure 2.14: Data used for training and validation

As described in Sec. 2.3.3, two types of CycleGAN architectures are studied in this paper. Table 3.1 presents the hyper-parameters for the training of these two architectures.

After training the models, we first perform visual comparisons between the actual and synthetic datasets for the two miter gates to assess the quality of the CycleGAN model

Table 2.1: Model parameters of two CycleGAN models

Variable Name	**Conv U-net**	**Self-ONN**
Number of epochs	2000	2000
Batch size	5	9
Learning rate for generators	0.0001	0.0001
Learning rate for discriminators	0.0001	0.0001
Dropout	0.5	0.5
$\mathcal{L}_{\text{cyc}}$ (cycle consistency loss)	20	10
$\mathcal{L}_{\text{id}}$ (identity loss)	10	5
q for generators	/	3
q for discriminators	/	5
Optimizer	Adam	Adam

training. Initial analyses, which are shown in Figs. 2.15 and 2.16, indicate that synthetic datasets closely mirror the original data, capturing essential information. Figs. 2.17 and 2.18 show the image format of both real data and synthetic data. As shown in these figures, CycleGAN can successfully translate the image from one miter gate to another, and thus provide a potential way of generating synthetic observations for a miter gate of interest (i.e., miter gate 2 in this paper) by leveraging observations from not only the gate itself but also another similar miter gate (i.e., miter gate 1).

To quantitatively evaluate the quality of the trained CycleGAN models, we employ four evaluation metrics to compare the synthetic data with the real data. The first metric is the Fréchet inception distance (FID). It is a metric used to evaluate the quality of images generated by machine learning models, particularly GANs. It measures the similarity between the distribution of generated images and that of real images. The lower the FID score, the more similar the two distributions are, which typically indicates higher quality in the generated images. The second evaluation metric is the structural similarity (SSIM), which is employed to assess the similarity of the overall structures between the synthetic and real image data. The SSIM value ranges from -1 to 1, with higher values indicating better performance. The third evaluation metric is the root mean square error (RMSE), which is utilized to compare the deviation between the real and synthetic values. The last evaluation metric is the correlation coefficient (CC), which measures the amount of linear correlation between the data. The higher the value of the CC, the more closely the synthetic data resembles the real data. A value of 1 indicates a linear relationship between the synthetic and the real data. Table 2.2 shows the results of these evaluation metrics for the three models mentioned in this case study. 'Gate1' indicates the metrics score for the specific model that generates synthetic data of miter gate 1 from real data of miter gate 2, and the similar explanation applies to

Table 2.2: Accuracy evaluation metrics for training

	Conv U-net		Self-ONN		Self-generated
	Gate1	Gate2	Gate1	Gate2	Gate2
FID	0.0345	0.0213	0.0673	0.0605	**0.0171**
SSIM	0.7973	**0.8578**	0.7145	0.7691	0.8240
RMSE	0.2808	0.2698	0.3556	0.3226	**0.2465**
CC	0.7284	**0.7758**	0.7150	0.7215	0.7561

'Gate2'. Overall, three methods are accurate in generating synthetic data. Convolutional U-net CycleGAN can get the best score in SSIM and CC, and Self-generated CycleGAN has the best score in FID and RMSE. Self-ONN doesn't have the same high score but still does a good job in generating synthetic data for both Gate1 and Gate2. It implies that the three models have achieved good performance in the training.

2.4.3 Accuracy assessment of synthetically generated new observations

Since all models demonstrate their capability to produce synthetic data effectively in the training, it is crucial to also check whether the synthetically generated new observations for months 21 to 40 can accurately replicate the pattern of miter gate 2, for which we do not have the real observations corresponding to the synthetic data to check the accuracy. Figs. 2.19, and 2.20 present comparisons of water level values between the real and synthetic data for months 21 to 40. Fig. 2.21 shows the comparison between the real and synthetic observations of miter gate 2 for months 21 to 40. Note that the "real data" here refers to the newly collected real observations which have not been used for training the CycleGAN model. In addition, there is no correspondence between "real data" and "synthetic data" for months 21 to 40, and therefore they cannot be compared for accuracy verification. As indicated in these figures, the synthetic observations follow the overall trends of the real observations over time but are quite different. These similarities and differences allow the synthetic data to augment the real observations and thus overcome the challenge in Bayesian inference due to limited observations.

To verify the quality of the newly generated synthetic observations for Bayesian model updating, the data is evaluated using the accuracy verification method described in Sec. 3.3.4. Three different models are compared to the real observation data and the results are shown in Figs. 2.22, 2.23, and 2.24. Only the last sensor has a large error confidence interval for real observation data. The error confidence interval in Fig. 2.22 shows that Convolutional

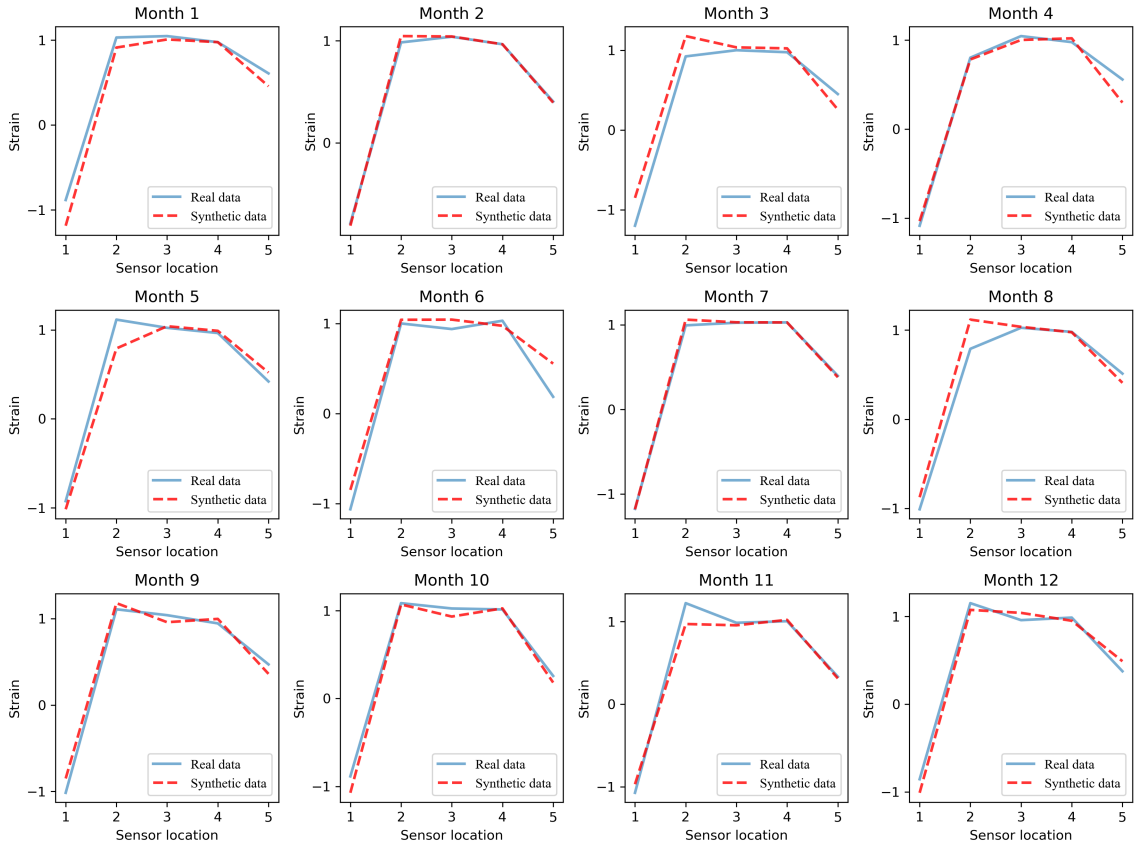


Figure 2.15: Normalized strain measurement value each sensor at different time step for miter gate 1

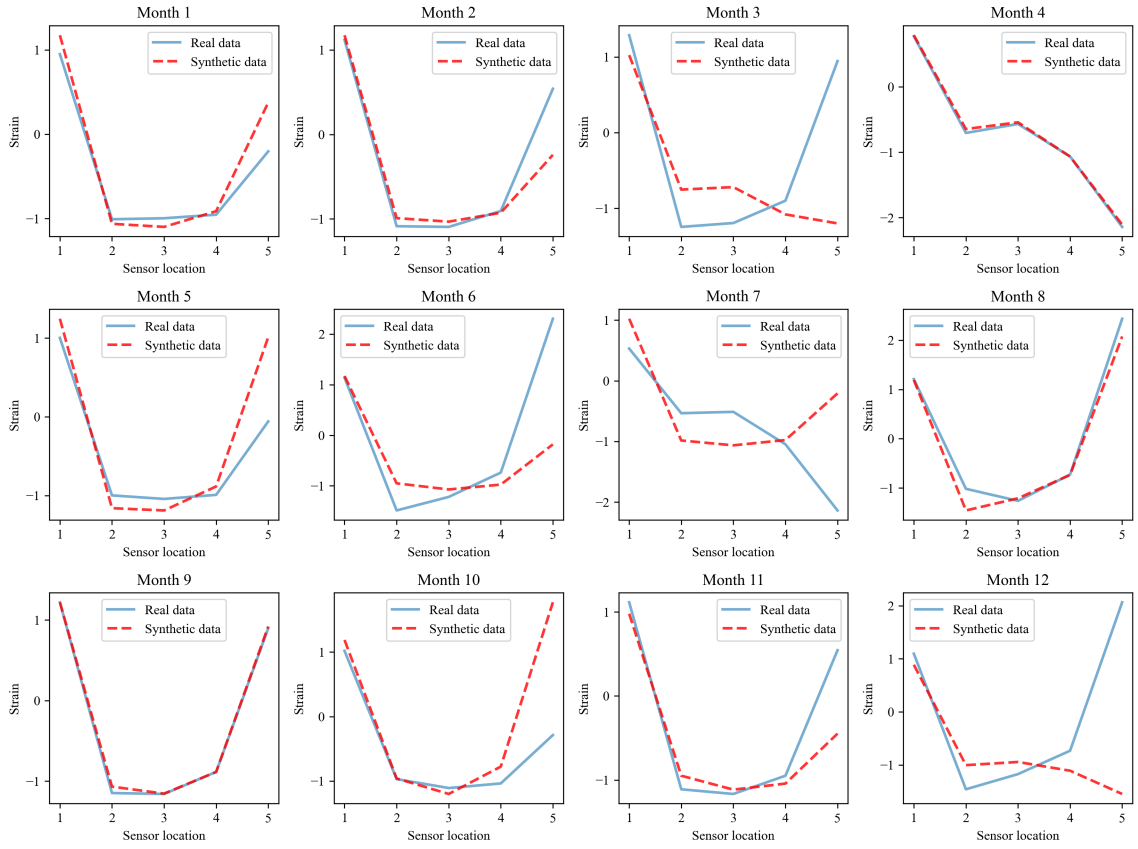


Figure 2.16: Normalized strain measurement value each sensor at different time step for miter gate 2

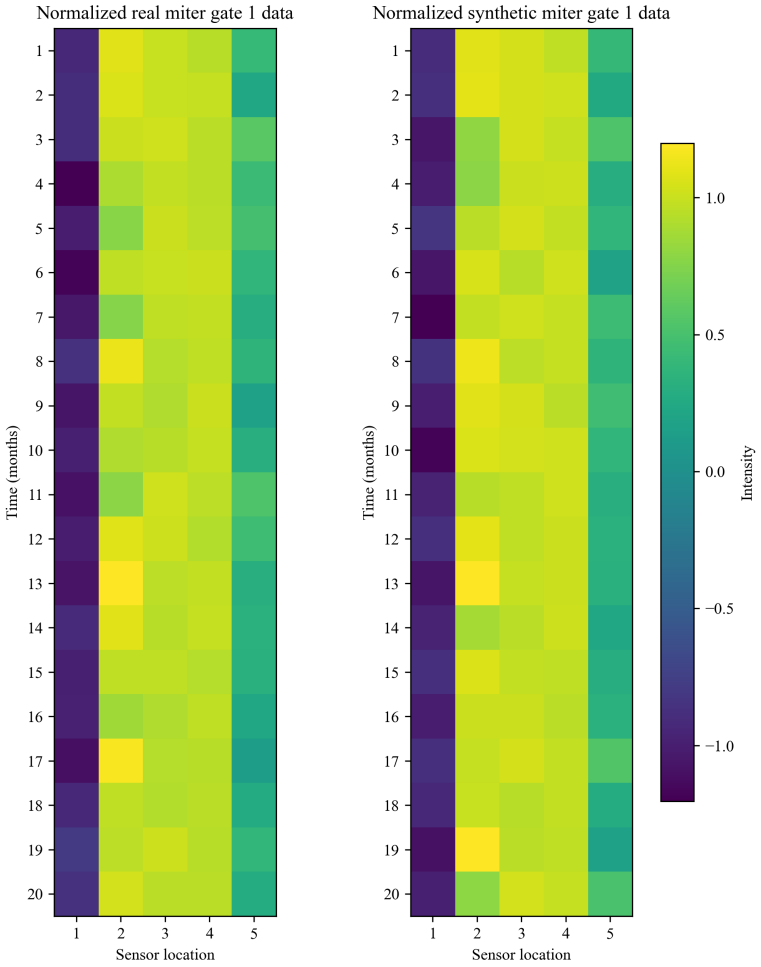


Figure 2.17: Normalized strain measurement image format for miter gate 1 (Checking training accuracy)

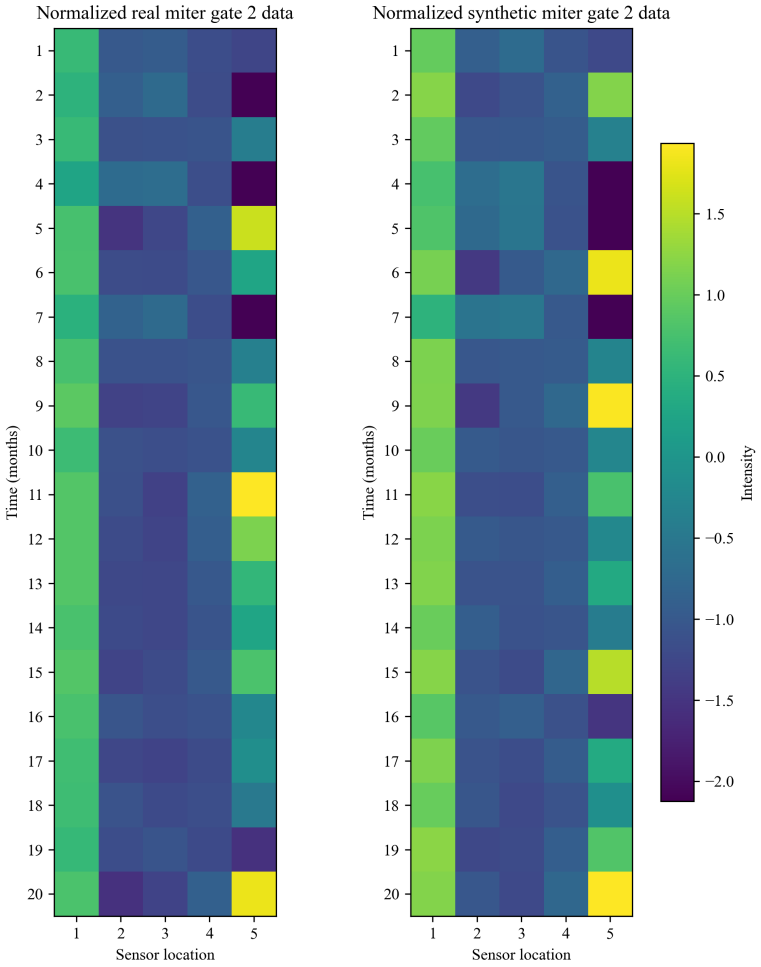


Figure 2.18: Normalized strain measurement image format for miter gate 2 (Checking training accuracy)

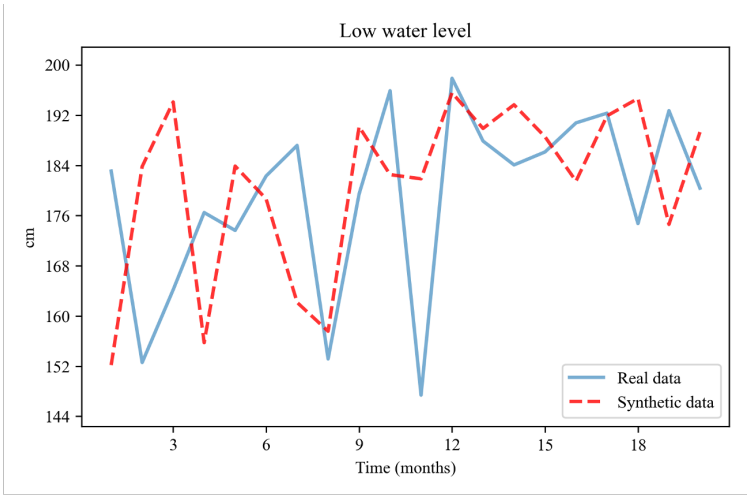


Figure 2.19: Low water level of miter gate 2 for months 21 to 40

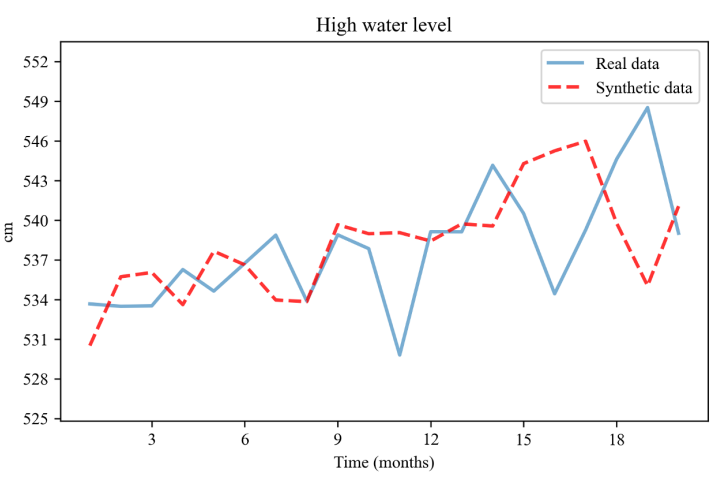


Figure 2.20: High water level of miter gate 2 for months 21 to 40

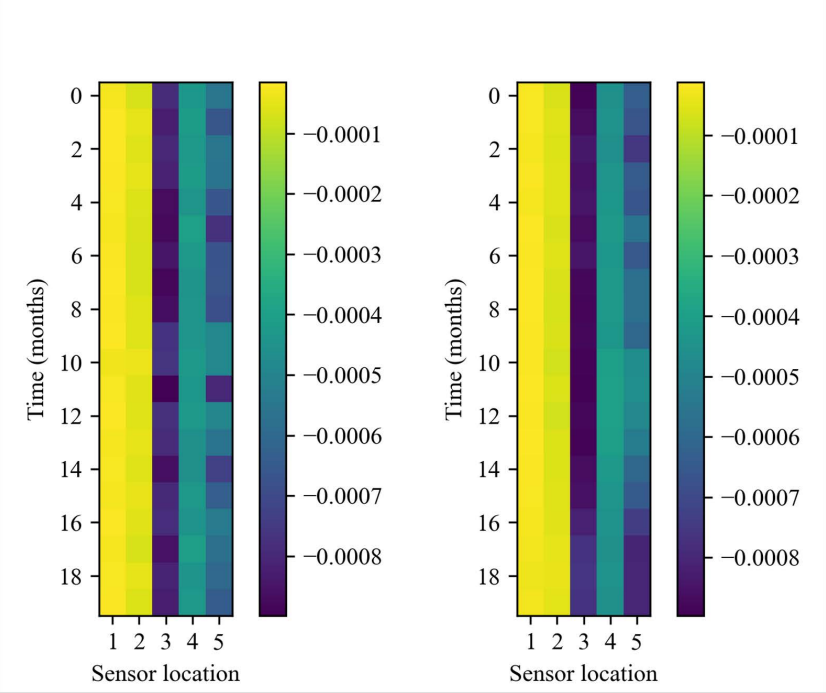


Figure 2.21: Real (left) and synthetic (right) strain measurement image format of miter gate 2 for months 21 to 40

U-net CycleGAN can capture the pattern of real observation and generate its synthetic data with relatively high accuracy. However, Fig. 2.23 shows that Self-ONN CycleGAN has a larger error confidence interval than real observation, which leads a relatively low-quality synthetic data. It is worth noting that in Fig. 2.24, the self-generated CycleGAN demonstrates the closest match to the real observational data, which is logical considering that the synthetic data is produced from the real observational data itself. This method can make sure the synthetic data mimics the real data but no additional information is learned from the unpaired image translation process. The results of error distribution comparison for sensors 1 to 5 are shown in Figs. 2.25, 2.26, 2.27, 2.28, and 2.29. Similar to the results above, self-generated CycleGAN has the best ability to mimic the real observation data. The Convolutional U-net CycleGAN performs well, particularly with sensors 1, 4, and 5. The Self-ONN CycleGAN performs the worst in all sensors for this case study. Overall, the above analysis shows that CycleGAN has promising capability in generating high quality synthetic data. Note that only visual comparisons are conducted between the error distributions for accuracy verification in this paper. We can also compute the similarity between different error distributions using KL divergence as mentioned in Sec. 3.3.4. After that, a threshold can be used to decide when to accept or reject the synthetically generated data.

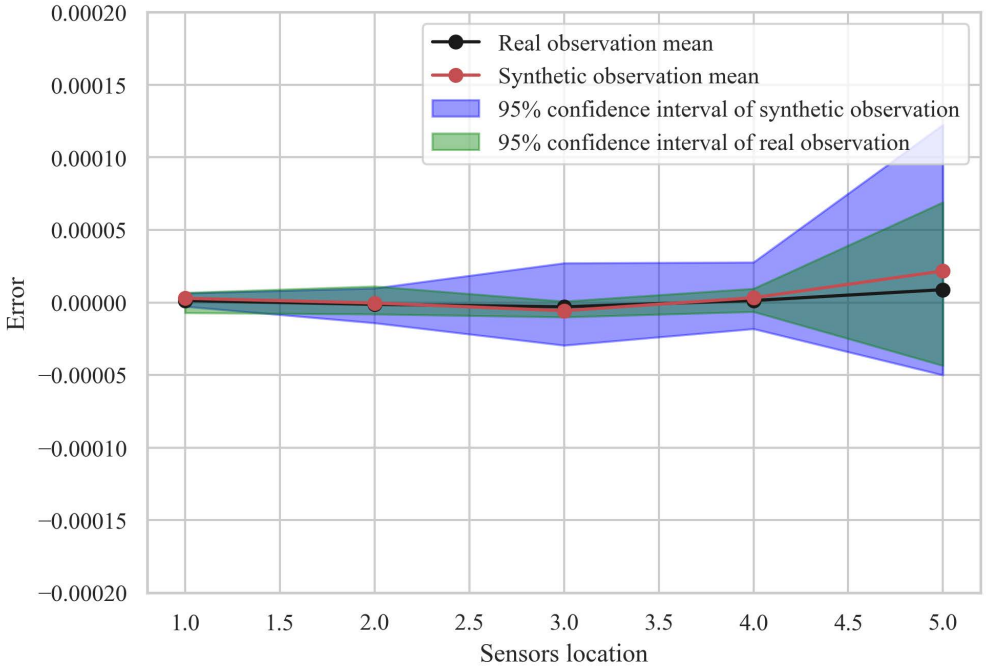


Figure 2.22: Error confidence interval comparison at each sensor (CycleGAN)

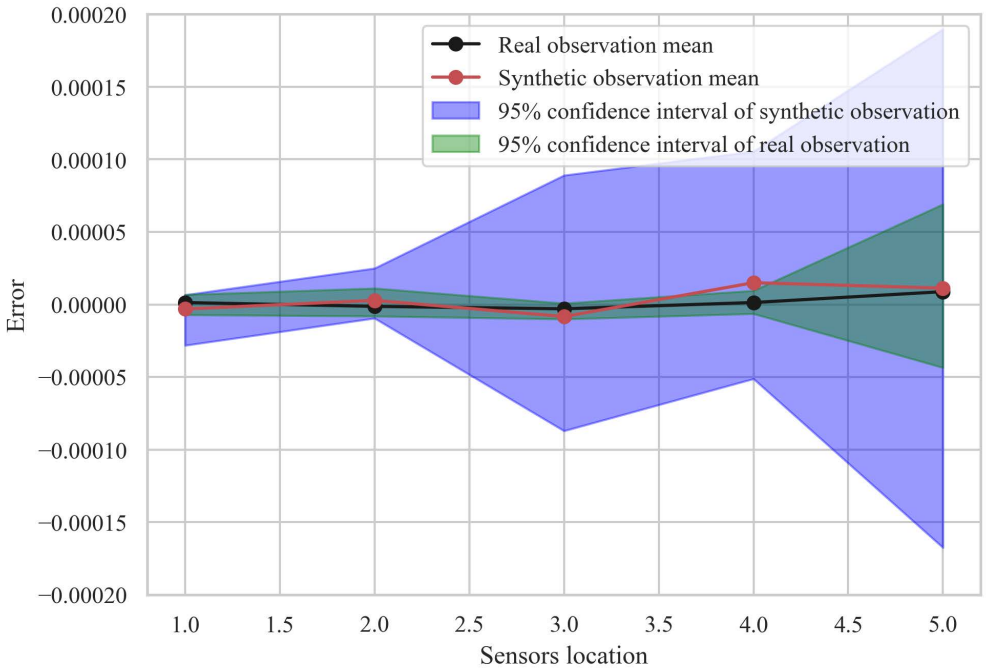


Figure 2.23: Error confidence interval comparison at each sensor (Self-ONN CycleGAN)

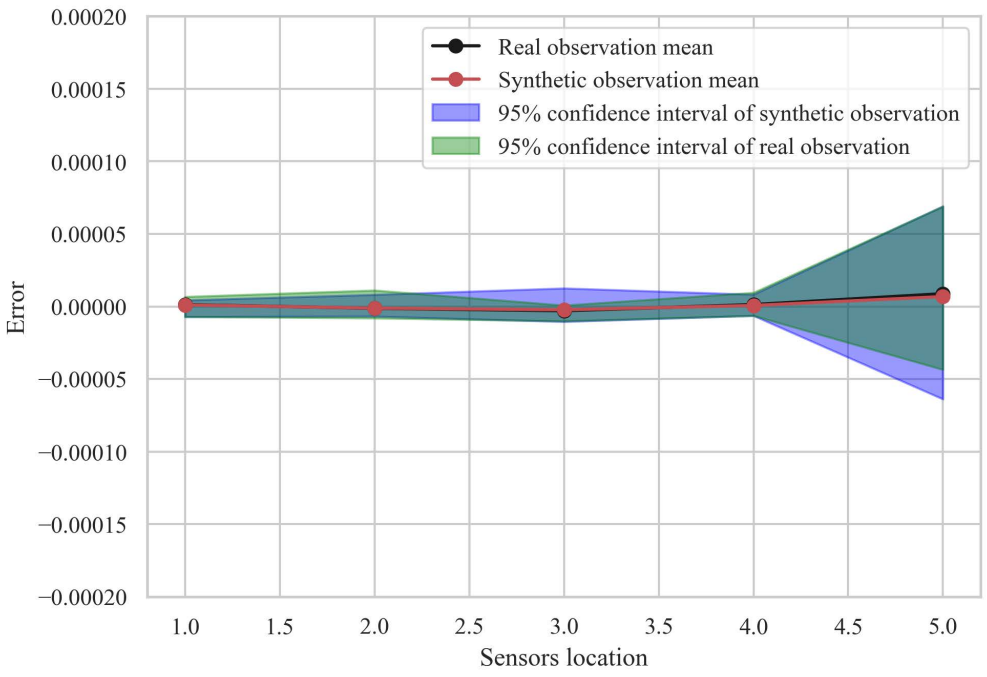


Figure 2.24: Error confidence interval comparison at each sensor (Self-generated CycleGAN)

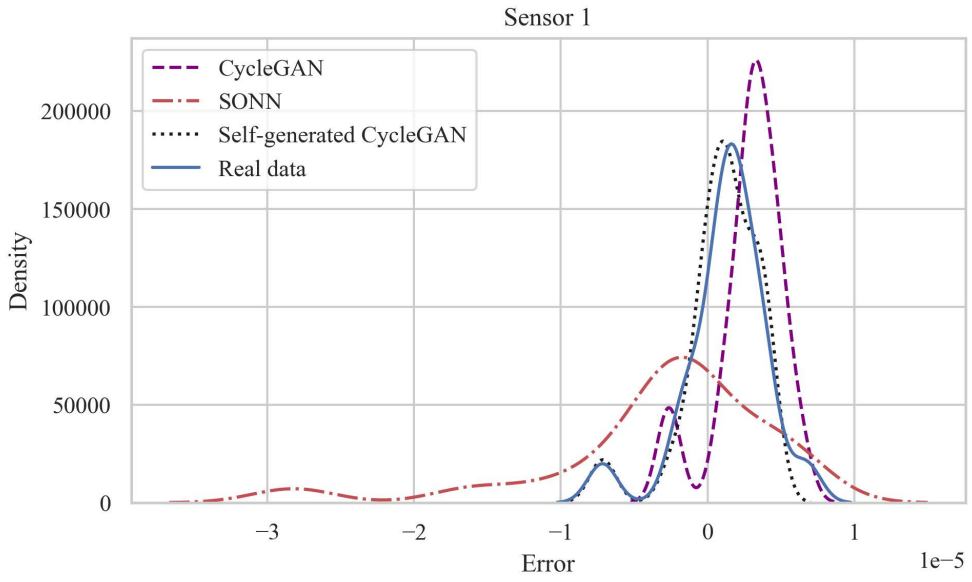


Figure 2.25: Comparison of error distributions for sensor 1

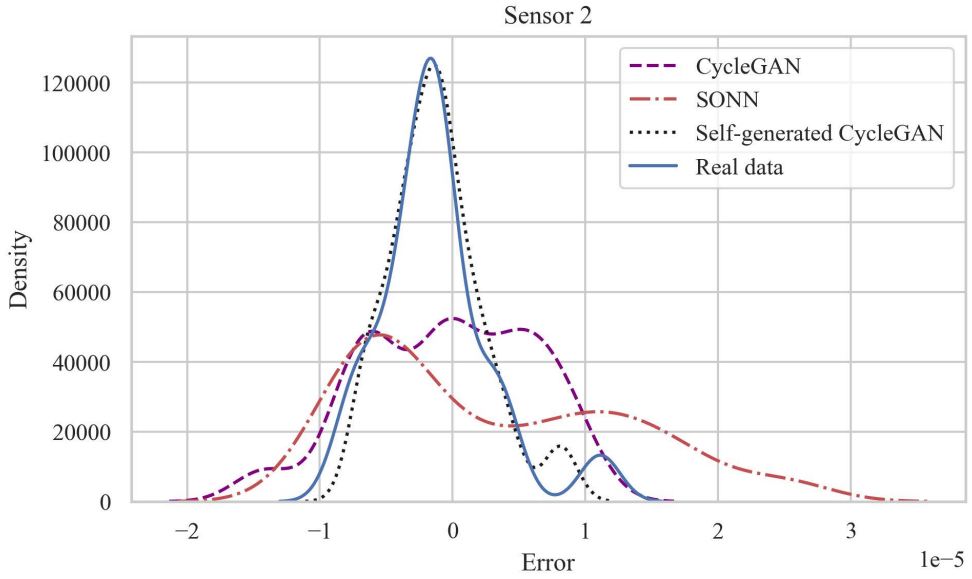


Figure 2.26: Comparison of error distributions for sensor 2

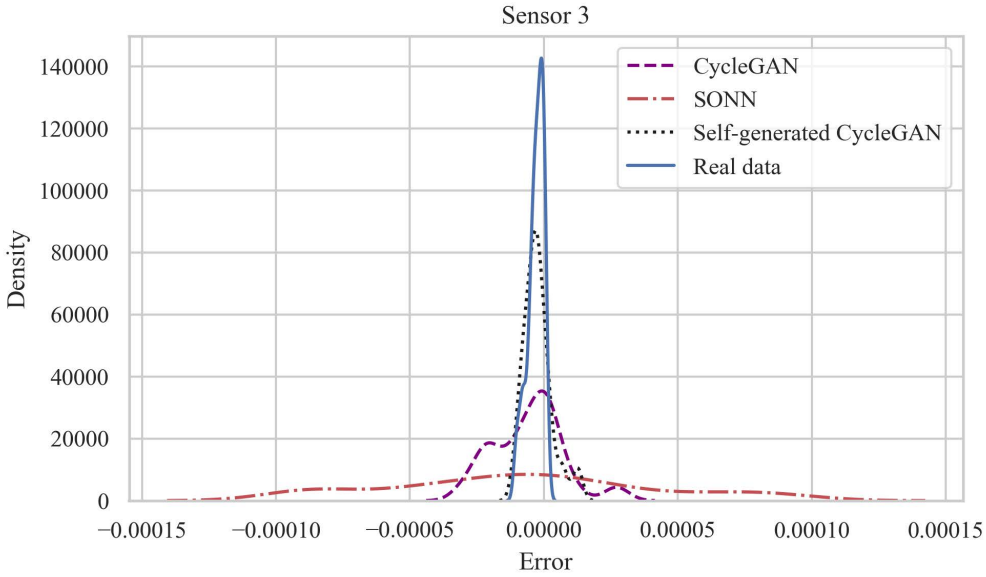


Figure 2.27: Comparison of error distributions for sensor 3

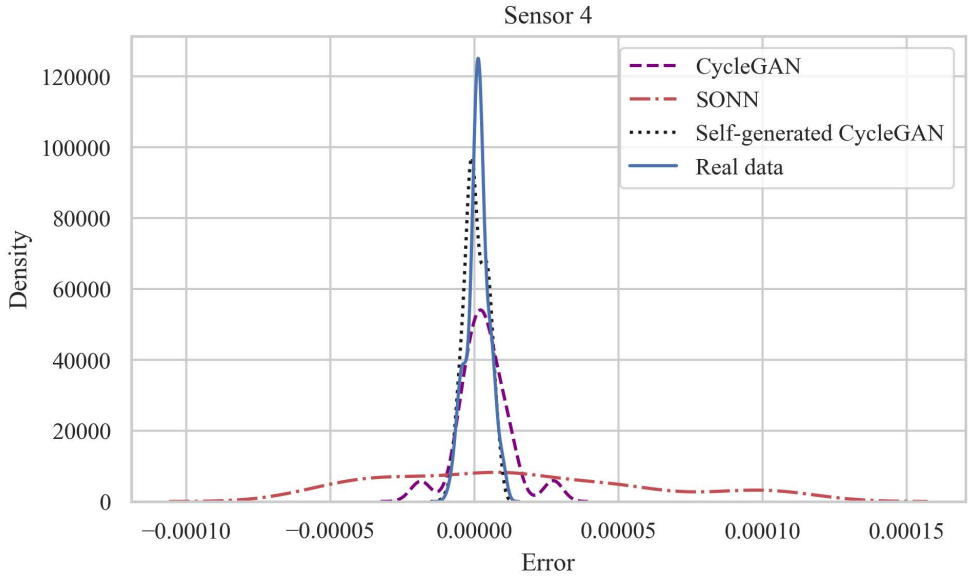


Figure 2.28: Comparison of error distributions for sensor 4

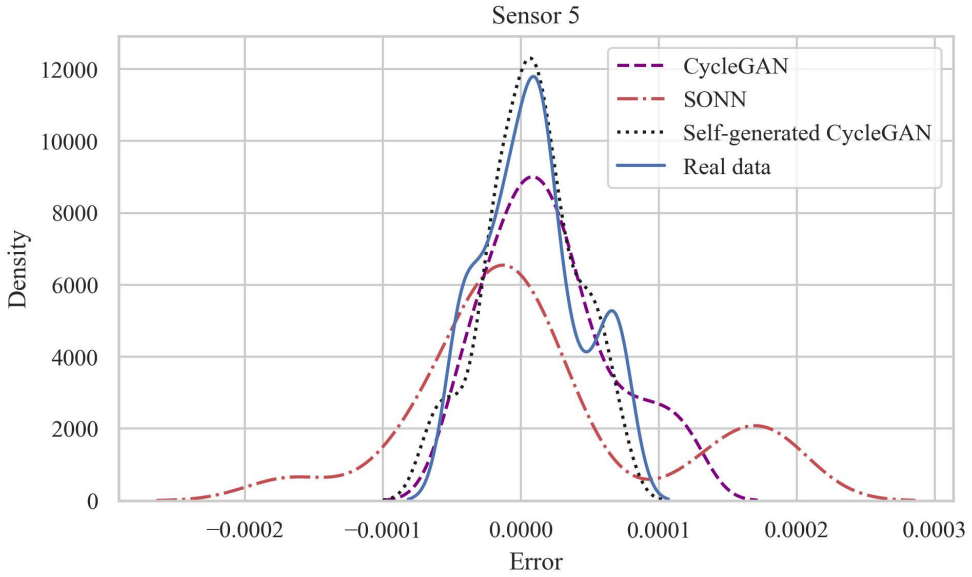


Figure 2.29: Comparison of error distributions for sensor 5

2.4.4 Damage detection using Bayesian inference

After the accuracy verification of the synthetic observations, we employ Bayesian inference for damage detection using the method described in Sec. 2.3.5. To evaluate the performance and robustness of our method, we conducted a comprehensive comparative study that involves: (1) varying the length of training data (i.e., 5, 10, 15, and 20 data points) and (2) four different approaches. The first approach (i.e., Only real observations) corresponds to the first Bayesian model updating in Fig. 2.10. The other three approaches correspond to the second Bayesian model updating as illustrated in Fig. 2.10. The goal of the comparison is to determine the effectiveness of different approaches in improving the accuracy of damage detection using Bayesian inference (i.e., Bayesian model updating). The four approaches are summarized as

- *Only real observations*: Bayesian inference using only real observations.
- *Real and synthetic observations (CGAN)*: Bayesian inference using (1) real observations and (2) synthetic observations generated using **training data of two different but similar miter gates** and the CycleGAN model architecture described in Sec. 2.3.3.1.
- *Real and synthetic observations (self-generated CGAN)*: It refers to Bayesian inference using (1) real observations and (2) synthetic observations generated using training data of **only the miter gate of interest** and the CycleGAN model architecture given in Sec. 2.3.3.1 and Fig. 2.7.
- *Real and synthetic observations (SONN)*: It means Bayesian inference using (1) real observations and (2) synthetic observations generated using **training data of two different but similar miter gates** and the self-ONN model architecture described in Sec. 2.3.3.2 and Fig. 2.8.

Figs. 2.30, 2.31, 2.32, and 2.33 show the comparison of posterior distributions with the true damage state for different methods and different time duration of data for training. The results in these figures show that:

1. When the amount of training data is very low (i.e., 5 points scenario given in Fig. 2.30), synthetic observations generated using CycleGAN model architecture given in Sec. 2.3.3.1 can hardly bring in additional information that is different from those already contained in the original real observations. This is manifested as the posterior distributions from methods: *Real and synthetic observations (CGAN)* and *Real and synthetic observations (self-generated CGAN)* are very close to that of method: *Only real observations* in Fig. 2.30.

2. Also in Fig. 2.30, even though method: *Real and synthetic observations (SONN)* can reduce the uncertainty of the posterior distribution significantly, the posterior distribution deviates from the true damage state. Such phenomena get more and more apparent when the duration of the observations increases as shown in Figs. 2.31, 2.32, and 2.33. It implies that for this particular case study, even though the self-ONN model architecture (i.e., Sec. 2.3.3.2) can bring in additional information that is different from those contained in the real observations, the additional information from the synthetic observations is misleading. The misleading or wrong information leads to the uncertainty of the posterior distribution being incorrectly reduced. As shown in Figs. 2.31, 2.32, and 2.33, the posterior distributions of method: *Real and synthetic observations (SONN)* becomes a straight line and deviate significantly from the true damage state.

3. For method: *Real and synthetic observations (self-generated CGAN)*, it can hardly bring in any additional information since the synthetic observations are generated using data of the miter gate of interest itself. As shown in Figs. 2.30, 2.31, 2.32, and 2.33, the posterior distributions from method: *Real and synthetic observations (self-generated CGAN)* are very similar to those from method: *Only real observations*.

4. Most importantly, the proposed method: *Real and synthetic observations (CGAN)* shows robustness and advantages over the other methods. As depicted in Figs. 2.30, 2.31, 2.32, and 2.33, as the length of training data increases, the posterior distribution of the method not only has less uncertainty compared to that from the method: *Only real observations*, but also gets closer to the true damage state. This demonstrates the efficacy of the proposed framework as illustrated in Fig. 3.3.

5. The effectiveness of synthetic data can also be shown when comparing Fig. 2.31 and 2.33. Under the scenario of the same length of data (10+10 months data for *Real and synthetic observations (CGAN)* in Fig. 2.31, 20 months data for *Only real observations* in Fig. 2.33), *Real and synthetic observations (CGAN)* still outperforms the latter set by having smaller confidence interval and getting closer to the true damage state. This indicates that the primary factor in reducing the standard deviation is not the addition of more data, but the additional information from synthetic data.

Even though SONN has shown improved performance in generating synthetic observations in other applications in our other research problems, it does not perform better than the convolutional U-Net architecture in this paper. A possible reason is that we have very limited training data in this case study (i.e., the maximum length of the data studied in different

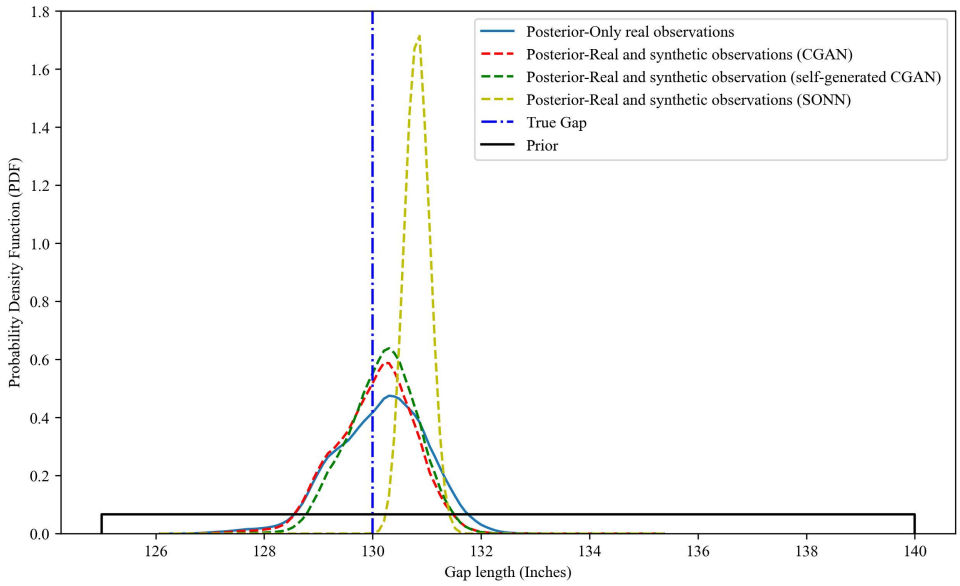


Figure 2.30: PDF of the posterior samples with 5 months data

scenarios is 20). As the architecture of SONN is more complicated than that of U-Net, the required amount of training data is also higher. The above findings highlight the critical need for careful selection and application of data augmentation and analysis techniques, especially in contexts where the available real-world data is limited. The findings validate the promising potential of leveraging CGAN-generated synthetic data to improve damage detection using Bayesian inference in structural health monitoring. In general, the results show that leveraging information from two different but similar gates (i.e., method: *Real and synthetic observations (CGAN)*) can improve the efficacy of damage detection using Bayesian inference in comparison to using information from only a single miter gate of interest (i.e., methods: *Real and synthetic observations (self-generated CGAN)* and *Only real observations*).

2.5 Conclusions and Discussion

This paper addresses the critical challenge of limited monitoring data in the SHM of large civil infrastructure, with a focus on miter gates. The scarcity of data, attributed to factors such as sensor availability, dynamic structural health states, and missing information, introduces significant uncertainty in evaluating structural integrity. To mitigate these issues, the study introduces an innovative data augmentation method leveraging image translation techniques within a Bayesian inference framework for damage diagnostics of miter gates. By translating monitoring data between different miter gates using CycleGAN, this approach

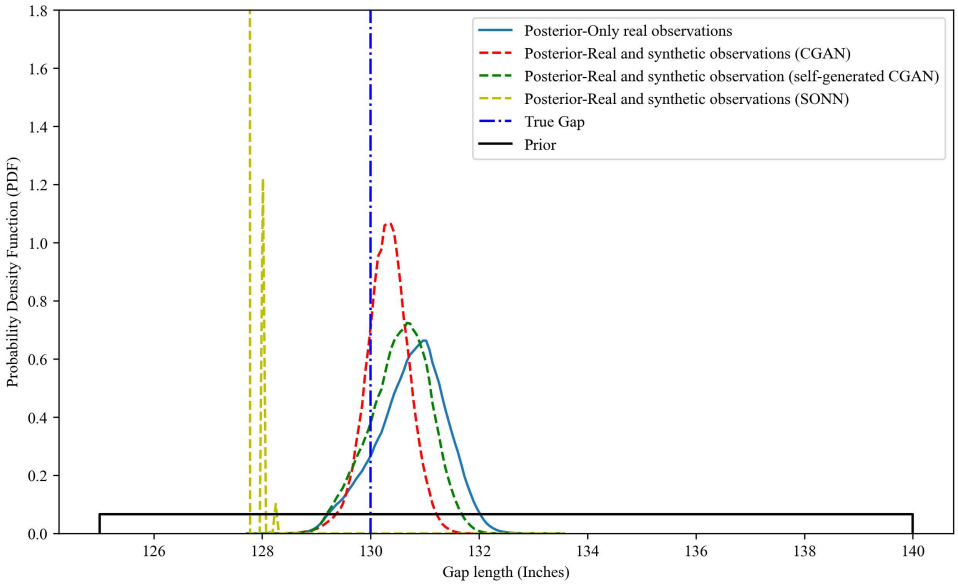


Figure 2.31: PDF of the posterior samples with 10 months data

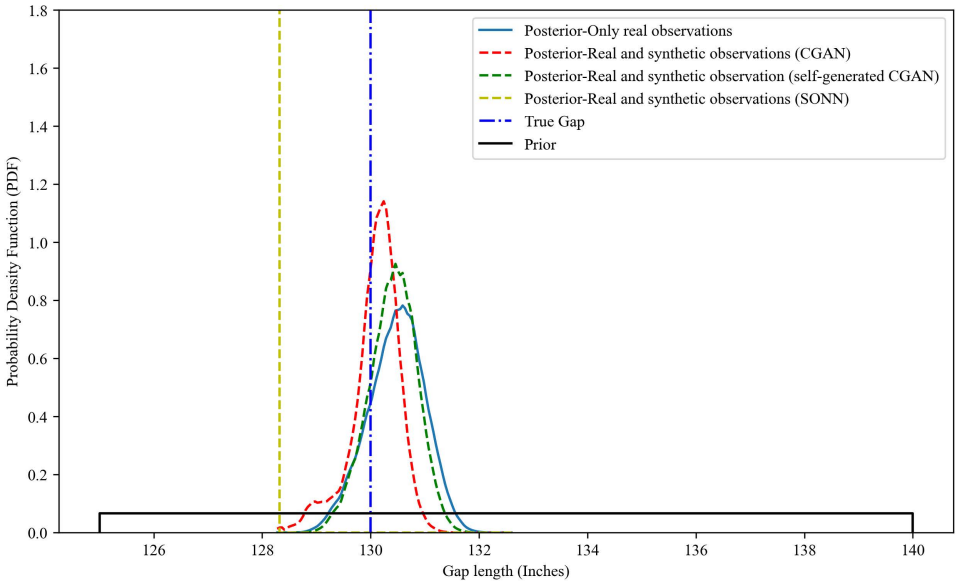


Figure 2.32: PDF of the posterior samples with 15 months data

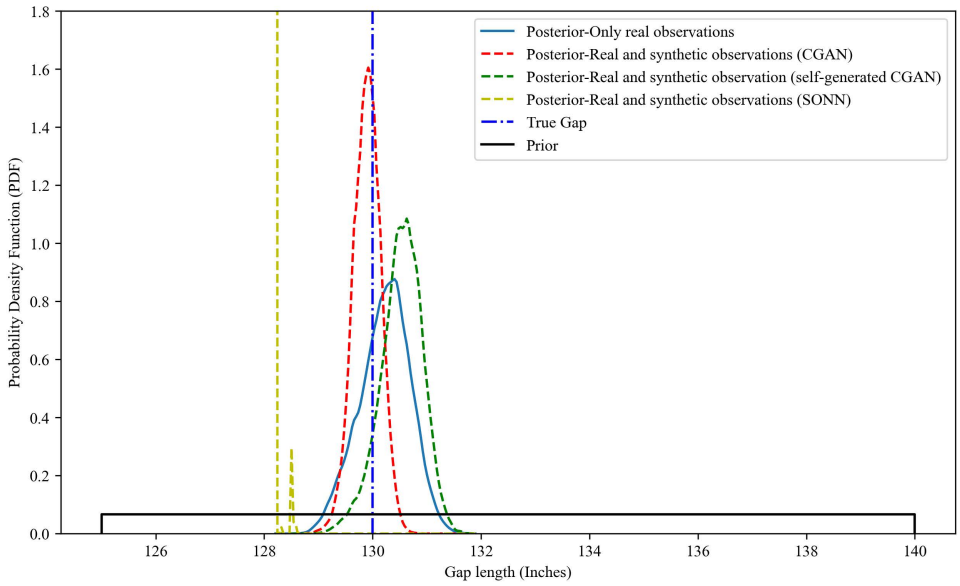


Figure 2.33: PDF of the posterior samples with 20 months data

effectively increases the volume of available monitoring data for a target miter gate by supplementing actual data with synthetic, yet plausible, monitoring data derived from other miter gates.

The application of this novel method is demonstrated through a case study, in which the results demonstrate the ability of the proposed method to enhance the accuracy and reliability of damage diagnostics. The use of translated images, serving as synthetic monitoring data, alongside real monitoring data, represents an innovative approach in model-based SHM. It not only addresses the challenges posed by data scarcity but also introduces a new paradigm in the integration of synthetic data generation with probabilistic model updating. This approach holds significant promise for the future of infrastructure maintenance and management, suggesting a broader application potential across various types of civil engineering structures beyond miter gates.

The current study is limited to data transformation between two specific domains. However, the potential to incorporate data from multiple sources, such as various miter gates, exists. Leveraging information from these diverse sources to augment data could significantly enhance the quality of data, thereby improving damage detection through Bayesian inference. Furthermore, integrating fundamental physics principles into CycleGAN training could markedly boost the accuracy and robustness of the synthetic data generated. These promising avenues will be explored in future research endeavors.

CHAPTER 3

Enhancing Bayesian Inference-Based Damage Diagnostics Through Domain Translation with Application to Miter Gates

3.1 Introduction

The US Army Corps of Engineers (USACE) operates numerous inland waterway infrastructure systems, such as navigation locks, at 236 sites across 191 locations. These locks are essential assets for facilitating the transportation of barges and goods from coastal regions to inland regions, which is critical for the national economic security of the United States [7]. Among the miter gates managed by USACE, many of them have exceeded their 50-year design life, which raises concerns about their safety and reliability [7]. As USACE continues to address the challenges of its aging infrastructure, routine evaluations of structural integrity become essential to preemptively identify potential failures and implement timely interventions [54]. This is of paramount importance for reducing unexpected shutdowns and confirming that planned maintenance and repairs are both needed and effective [7, 55]. The typical approach to inspecting and monitoring lock components involved shutting down the lock for visual assessments by expert elicitation (trained inspectors). This process is not only labor-intensive but also economically expensive, inefficient, and inconsistent [56].

As a structural integrity assessment and health management tool, structural health monitoring (SHM) and management methods have continued to gain wider acceptance and significance globally in the past decades [57, 58]. A well-designed SHM system offers significant advantages, including reduced ownership costs through data-driven maintenance decisions, enhanced operational performance, and minimized risks of catastrophic failures, thereby increasing overall safety [59, 60, 54]. To address reliability concerns and maintain operational integrity and safety, USACE started to implement SHM technologies, such as SMART Gate, on inland waterway civil infrastructures [54]. For instance, at the Dalles Navigation Lock,

Oregon, the miter gate downstream and the tainter gate upstream have been used for over 50 years. In 2007, inspections identified significant issues like cracking and misalignment, leading to the adoption of SMART Gate technology to monitor and analyze these conditions [8]. By 2009, detected anomalies through SHM system using SMART Gate prompted emergency interventions, including inspection dives and remote operations, culminating in crucial repairs that averted a potential disaster [8].

While SHM plays a vital role in ensuring structure integrity and safety, it requires a rigorous design and implementation approach to fully materialize its benefits [61]. Various approaches have been developed to enable effective damage diagnostics and failure prognostics in SHM. Amongst all available approaches, one of the most commonly used SHM approaches is damage diagnostics based on Bayesian inference, which leverages Bayesian techniques and physics-based modeling to address the inverse problem by aligning physics-based predictions with sensor monitoring observations [34]. In the past several years, numerous approaches have been developed using Bayesian methods for damage diagnostics of inland waterway infrastructure, such as miter gates. For instance, Ramancha *et al.*compared the finite element model with a surrogate model for Bayesian inference-based damage diagnostics of miter gate structural systems [14]. Levine *et al.*investigated the application of a Bayesian model updating technique in the diagnostics of miter gate anchorages and studied its implication on the design of such structures [62]. Vega *et al.*introduced a hybrid damage prognosis framework using the Bayesian method for miter gate components in navigational locks by minimizing human errors in evaluations and integrating abstract inspection data with data from SHM systems [56]. Qian *et al.* outlined a framework that combines machine learning with simulation updates for pitting corrosion detection in large infrastructures, utilizing an approximate Bayesian computation method to enhance model accuracy for future prognostics, demonstrated its efficacy on miter gates [63]. In another study, Qian *et al.* proposed a conditional invertible neural network-based Bayesian model updating method to enhance physics-based multi-scale corrosion simulations for large civil structures like miter gates. These simulations are crucial for predicting pitting corrosion and are integral to risk-based asset health management strategies [64].

Although Bayesian inference is a powerful technique for damage diagnostics in SHM, leveraging both physics-based simulations and sensor data, its effectiveness can be compromised by the quality and/or scarcity of monitoring data for a miter gate of interest [1]. This limitation is caused by the limited data sources available for SHM. Additionally, the accuracy of Bayesian inference-based diagnostics heavily relies on the fidelity of the simulation model. Various sources of uncertainty in the physics-based simulation—stemming from model simplifications, assumptions, or an incomplete understanding of the underlying

physics—can lead to biased estimations of the structural damage state. On the other hand, numerous similar, yet different, miter gates have been inspected in the past, generating valuable inspection and monitoring data. Leveraging knowledge from these "nominal classes" of miter gates could provide an alternative approach to overcoming these limitations. This concept, known as population-based SHM, involves using knowledge from groups of similar structures to estimate the damage condition of a given structure. For a thorough discussion on various aspects of population-based SHM, see the trio of references from the University of Sheffield [65, 66, 67]. Among the numerous techniques available, one widely adopted method for leveraging data from different assets to diagnose the damage of a specific asset is domain adaptation or translation within the context of transfer learning. [68, 31]. Domain adaptation is a technique where a model developed for a specific domain (the source domain) is adapted to work in a different but related domain (the target domain). This approach is particularly useful when the target domain has limited or no labeled data (such as the unknown damage state of a miter gate of interest). It involves adjusting the model so it can effectively transfer knowledge from one domain to another despite differences in data distribution [69].

In recent years, domain adaptation has increasingly been applied to the field of damage diagnostics. For example, Chun *et al.* introduced a self-training framework for unsupervised domain adaptation in concrete wall crack segmentation [70]. Chen *et al.* proposed a structural damage detection method that enhances detection performance by integrating one-dimensional and two-dimensional deep convolutional neural networks (CNN) for detailed spatiotemporal feature extraction from vibration data [71]. Huang *et al.* proposed a method that involves a Bayesian deep dual network that utilizes domain adaptation to transfer diagnostic insights from experimental setups to real-world machines with varying configurations and operational conditions [72]. Li *et al.* introduced a method using a deep convolutional neural network (DCNN) combined with Bayesian optimization and adaptive batch normalization (AdaBN) for predicting the remaining useful life of mechanical equipment [73]. Recently, adversarial-based domain adaptation includes methods like Generative Adversarial Networks (GANs), have gained popularity for their ability to create robust feature representations consistent across various domains. This characteristic makes GANs highly effective for domain adaptation challenges. Kwak et al. presented a diagnostic design solution that integrates Designable Data Augmentation (DDA) with Designable GAN and Prognosis and Health Management (PHM) technology, incorporating domain adaptation. This approach utilizes Bayesian transfer learning to transfer diagnostic knowledge from an established system to a new target system [74]. Despite the significant potential of domain adaptation in damage diagnostics, it also has its drawbacks, such as the risk of negative transfer. Addi-

tionally, it fails to leverage the rich information contained in physics-based simulations of a specific miter gate regarding the relationship between damage state and system response. Principled uncertainty quantification of the damage estimation is also challenging for such purely data-driven approaches.

Given that both physics-based Bayesian inference and domain adaptation in damage diagnostics have their own advantages and disadvantages, and their strengths and weaknesses complement each other, it is beneficial to determine when to use each method and how to integrate them for optimal damage state estimation. To this end, this paper proposes a novel Bayesian inference-based damage diagnostics approach Enhanced through Domain Translation (BiEDT), aiming to make more informed decisions regarding the damage state of a miter gate. The proposed method generally consists of three phases: The first phase converts observations from both the target and source miter gates into a common domain through domain translation; the second phase constructs a probabilistic model to determine the damage state of the target miter gate, using observations from the translated domain with known damage states of the source miter gates; finally, the method combines the results from the earlier phases with physics-based Bayesian inference through Bayesian model averaging. The main contributions of this paper can be summarized as follows:

- First, this study presents a novel framework that seamlessly integrates domain translation of observations with Bayesian inference techniques for damage diagnostics.
- Second, this paper synthesizes domain translation techniques with a conditional invertible neural network for the first time to enable damage diagnostics with quantified uncertainty.
- Third, a comparison of domain translation using Cycle-Consistent GAN (CycleGAN) and Domain-Adversarial Neural Network (DANN) in the proposed framework.
- Fourth, this paper proposes a Bayesian model averaging and selection method to synthesize different probabilistic estimates of the damage state.
- Finally, the proposed frameworks are demonstrated and compared using a practical application example of miter gate structural systems.

It is worth noting that the novelty of this paper lies in its comprehensive and innovative framework that seamlessly integrates domain translation techniques with Bayesian inference for damage diagnostics in miter gates. This holistic approach results in a robust and well-informed decision-making process concerning the damage state of miter gates, marking a significant advancement in the field of SHM.

The remainder of this paper is structured as follows: Sec. 3.2 provides a detailed background overview including Bayesian inference for probabilistic damage detection, health monitoring of miter gates, data augmentation with GANs in Bayesian models, and limitations of the existing methods. Sec. 3.3 formulates the problem and presents details of the proposed method. Sec. 6.4 uses a miter gate application example to demonstrate the proposed method. Finally, Sec. 3.5 presents the conclusion remarks of this paper.

3.2 Background

3.2.1 Probabilistic damage detection using Bayesian inference

In Bayesian inference-based probabilistic damage detection, the observation data from a physical system are utilized to update its simulation model and estimate the structural damage characterized by uncertain model parameters $\boldsymbol{\theta}$. Let $\mathbf{y}_k = g(\mathbf{u}_{1:k}, \boldsymbol{\theta})$ be the concerned simulation model, where $\mathbf{y}_k \in \mathbb{R}^{N_y \times 1}$ are the model outputs at the kth time step, N_y is the number of model outputs, $\mathbf{u}_{1:k} \in \mathbb{R}^{(N_u \times k) \times 1}$ are the model inputs at the past k time steps, and N_u is the number of model inputs. The mathematical relation between the model prediction $\mathbf{y}_k$ and the observation data $\mathbf{y}_{o,k}$ can be expressed as follows [75]

$$\begin{aligned} \mathbf{y}_{o,k} &= g(\mathbf{u}_{1:k}, \boldsymbol{\theta}) + \delta(\mathbf{u}_{1:k}) + \boldsymbol{\varepsilon}_k, \\ &= \mathbf{y}_k + \delta(\mathbf{u}_{1:k}) + \boldsymbol{\varepsilon}_k, \end{aligned} \tag{3.1}$$

where $\delta(\mathbf{u}_{1:k})$ is the model discrepancy term, $\boldsymbol{\varepsilon}_k \sim N(\mathbf{0}, \boldsymbol{\Sigma}_k)$ is the Gaussian noise term with zero mean and the following covariance matrix:

$$\boldsymbol{\Sigma}_k = \begin{pmatrix} \sigma_1^2 & 0 & \cdots & 0 \\ 0 & \sigma_2^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_{N_y}^2 \end{pmatrix} \in \mathbb{R}^{N_y \times N_y}. \tag{3.2}$$

Given the observations of the input and output, the joint posterior distribution of the uncertainty parameters $\boldsymbol{\theta}$ can be obtained by applying Bayes' rule as

$$f(\boldsymbol{\theta}|\mathbf{u}_{1:k}, \mathbf{y}_{o,1:k}) = \frac{L(\mathbf{y}_{o,1:k}|\mathbf{u}_{1:k}, \boldsymbol{\theta}) f(\boldsymbol{\theta})}{\int L(\mathbf{y}_{o,1:k}|\mathbf{u}_{1:k}, \boldsymbol{\theta}) f(\boldsymbol{\theta}) d\boldsymbol{\theta}}, \tag{3.3}$$

where $f(\boldsymbol{\theta})$ and $f(\boldsymbol{\theta}|\mathbf{u}_{1:k}, \mathbf{y}_{o,1:k})$ are the prior and posterior distributions of $\boldsymbol{\theta}$, respectively, and $L(\mathbf{y}_{o,1:k}|\mathbf{u}_{1:k}, \boldsymbol{\theta})$ is the likelihood function, which measures the agreement between the model prediction $\mathbf{y}_k$ and the observation value $\mathbf{y}_{o,k}$.

3.2.2 Structural health monitoring of miter gates in inland waterway networks

In the United States, the USACE maintains and operates 236 miter gate structures, as shown in Fig. 3.1, at 191 sites. These miter gates constitute a network that plays a crucial role in the inland waterway transportation system. More than half of these structures have been beyond their 50-year expected service life. Maintenance and upkeep plans are critical for the optimal operation of these miter gates, as unexpected closures can prevent shippers from completing their scheduled transportation tasks, leading to unpredictable economic losses.

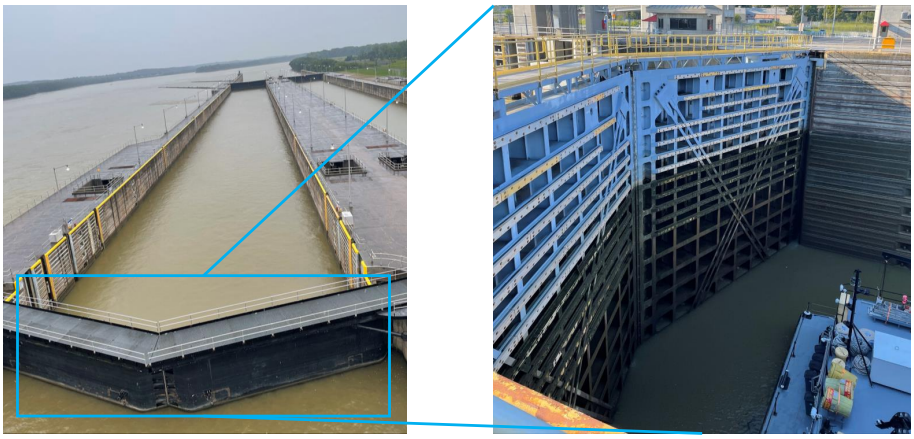


Figure 3.1: Miter gates in inland waterway navigation

One of the most common damage modes in miter gate structure is a gap that arises from the loss of contact between the gate quoin block and the wall quoin block the bottom of the gate, as shown in Fig. 3.2. This gap will further lead to the stress redistribution in the gate structure stress, forming a high stress zone, which may exceed the acceptable limit state and lead to failure. Because such gaps are often underwater, they cannot be directly observed. Instead, one can infer the gaps from the measurement data collected by the sensor installed in the high stress zone.

3.2.3 Augmenting observation data in Bayesian inference using Generative Adversarial Networks (GANs)

In Bayesian model updating, the observation data is of paramount importance as it provides insightful information for estimating model parameters and quantifying uncertainty. Thus, the accuracy and effectiveness of Bayesian model updating highly depend on both

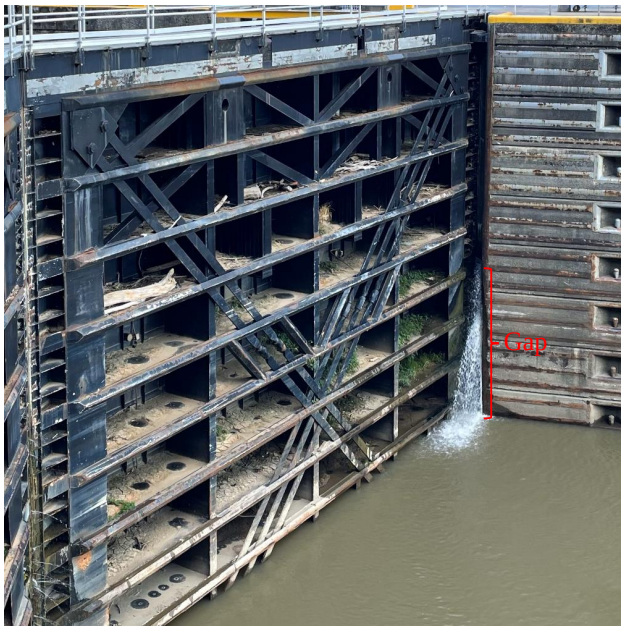


Figure 3.2: Illustration of "gap" on miter gate structures

the quality and quantity of sensor-monitoring data. However, the available data in practical engineering is often limited, which seriously hinders the effectiveness of Bayesian model updating. To overcome this limitation, in our previous work, we applied a data augmentation technique, which generates synthetic data through the process of sampling and transformation of existing data to increase the diversity and volume of an existing dataset, and thereby improving the effectiveness of the Bayesian model updating-based damage detection. This data augmentation technique has been demonstrated in our previous work in Refs. [1, 2].

Fig. 3.3 provides an overview of our previous approach using a technique called CycleGAN to enhance Bayesian inference for estimating the damage status (i.e., gap length) of miter gates. Assuming the two miter gates share operational and environmental similarities but not identical conditions, we employed CycleGAN to enrich the dataset of "Miter gate 2" by translating sensor data from "Miter gate 1", thus increasing the confidence of Bayesian inference-based damage estimation (i.e., gap length) of "Miter gate 2". The whole technical route consists of four parts: (1) data preprocessing, (2) synthetic data generation, (3) accuracy verification of synthetically generated data, and (4) augmenting Bayesian Inference with synthetic data.

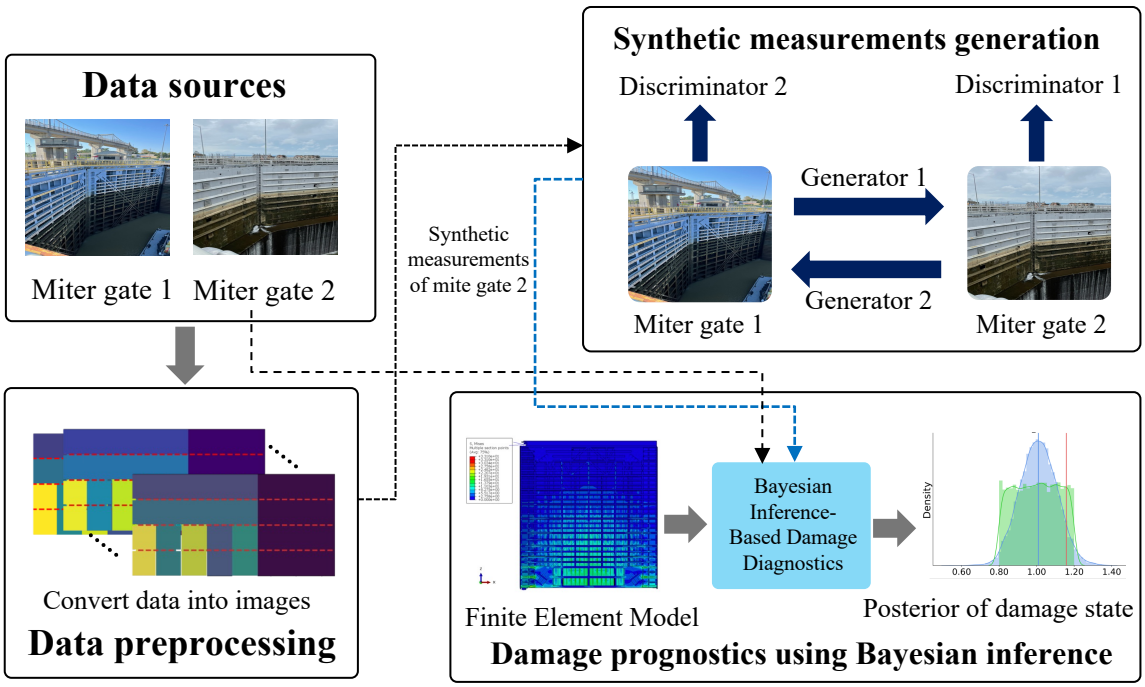


Figure 3.3: A framework proposed in our previous research [1, 2]

3.2.4 Limitations of existing methods

While the results of our previous research have shown the promise of data augmentation in enhancing the performance of Bayesian inference-based damage diagnostics, it also has several limitations. *First*, the method presented in Refs. [1, 2] and as shown in Fig. 3.3 assumes that the damage states of the two miter gates of interest remain unchanged for the time duration of data collection. This assumption greatly limits the application of the method since the damage states of the inspected miter gates are different most of the time. *Second*, the trained inference model only applies to one unknown damage state due to the first limitation. It needs to be retrained whenever the damage state of the miter gate of interest changes. *Third*, the method presented in Refs. [1, 2] mainly focuses on augmenting the observation data, which will subsequently be used in Bayesian inference using physics-based simulations. The damage estimation in this case highly dependent on the fidelity of the simulation model. It, therefore, cannot overcome the aforementioned limitations of Bayesian inference-based damage diagnostics caused by the imperfection of the physics-based simulations.

Motivated by overcoming the limitations of our previous method and developing a versatile approach for enhancing Bayesian inference-based damage diagnostics, in Sec. 3.3, we propose a novel framework called Bayesian inference-based damage diagnostics approach Enhanced

through Domain Translation (BiEDT). Even though BiEDT is developed for miter gate applications, it is very generalized and applicable to other engineering systems.

3.3 Proposed Method

In this section, we first formulate the problem that we aim to solve. After that, we will present the proposed method in detail.

3.3.1 Problem formulation

The objective of this paper is to estimate the damage state (i.e., gap) of a miter gate of interest based on three main data sources, namely (1) structural monitoring data of the miter gate itself, which has an unknown damage state; (2) monitoring data from similar, but different, miter gates whose damage states have already been inspected; and (3) a physics-based computational simulation model of the miter gate of interest.

For the sake of explanation, we refer to the miter gate of interest, which has an unknown damage state, as the *target miter gate*. Similarly, we refer to the miter gates whose damage states have already been inspected as *source miter gates*. This terminology is analogous to definitions commonly used in transfer learning within the machine learning community. We define the following variables to represent the data sources and models.

- $\mathbf{y}_{o,\mathrm{tg},i} \in \mathbb{R}^{N_t \times N_s}, \forall i = 1, \cdots, N_{ot}$: the i-th group of monitoring data of the target miter gate over N_t time steps from N_s strain gauges on the gate, where N_{ot} is the total number of available monitoring datasets.

- $\mathbf{y}_{o,\mathrm{sr},i} \in \mathbb{R}^{N_t \times N_s}, \forall i = 1, \cdots, N_{os}$: the i-th group of monitoring data of the source miter gates over N_t time steps from N_s strain gauges on the gates, where N_{os} is the total number of available monitoring datasets. Note that here the value of N_{os} could be different from that of N_{ot}, and the N_{os} datasets could come from different source miter gates.

- $\boldsymbol{\theta}_{\mathrm{sr},i}, i = 1, \cdots, N_{os}$: inspected damage state $\boldsymbol{\theta}_{\mathrm{sr}}$ of the source miter gates corresponding to the i-th group of monitoring data $\mathbf{y}_{o,\mathrm{sr},i}$ of the source miter gates.

- $\boldsymbol{\theta}_{\mathrm{tg},i}, i = 1, \cdots, N_{ot}$: unknown damage state $\boldsymbol{\theta}_{\mathrm{tg}}$ of the target miter gate corresponding to the i-th group of monitoring data $\mathbf{y}_{o,\mathrm{tg},i}$.

- $\mathbf{y}_{\text{tg}} = G(\boldsymbol{\theta}, \mathbf{u})$: computational simulation model (i.e., finite element analysis model) of the target miter gate, which can predict the strain response for given damage state $\boldsymbol{\theta}$ and load condition $\mathbf{u}$ (i.e., water level for the miter gate case).

It is worth mentioning that even though $\{\boldsymbol{\theta}_{\text{tg},i}, i = 1, \cdots, N_{ot}\}$ and $\{\boldsymbol{\theta}_{\text{sr},i}, i = 1, \cdots, N_{os}\}$ represent the damage states of different miter gates, they are domain-invariant quantities. For instance, all the gap lengths of the miter gates in the case study of this paper are bounded between 0 to 130 inches. Based on the above definitions, the problem that needs to be solved is to find the following posterior distribution of the unknown damage state, $\boldsymbol{\theta}_{\text{tg},i}$,

$$f(\boldsymbol{\theta}_{\text{tg},i} | (\tilde{\mathbf{y}}_{o,\text{sr}}, \tilde{\boldsymbol{\theta}}_{\text{sr}}, \mathbf{y}_{o,\text{tg},i}; \mathbb{G})), \ \forall i = 1, \cdots, N_{ot}, \tag{3.4}$$

where $\tilde{\mathbf{y}}_{o,\text{sr}} \doteq \{\mathbf{y}_{o,\text{sr},i}, i = 1, \cdots, N_{os}\}$, $\tilde{\boldsymbol{\theta}}_{\text{sr}} \doteq \{\boldsymbol{\theta}_{\text{sr},i}, i = 1, \cdots, N_{os}\}$, and $\mathbb{G}$ represents the simulation model $\mathbf{y}_{\text{tg}} = G(\boldsymbol{\theta}, \mathbf{u})$.

If we only have monitoring data from the target miter gate, the posterior distribution $f(\boldsymbol{\theta}_{\text{tg},i} | (\mathbf{y}_{o,\text{tg},i}; \mathbb{G}))$ can be solved using conventional Bayesian model updating methods [14, 33]. However, the limited monitoring data $\mathbf{y}_{o,\text{tg},i}$ from the target gate and various uncertainty sources in the simulation model, $\mathbf{y}_{\text{tg}} = G(\boldsymbol{\theta}, \mathbf{u})$, could result in large uncertainty or bias in the estimated damage state $\boldsymbol{\theta}_{\text{tg},i}$. Although damage inspection and monitoring data, $\{\boldsymbol{\theta}_{\text{sr},i}, \mathbf{y}_{o,\text{sr},i}\}, i = 1, \cdots, N_{os}$, from similar but different miter gates (referred to as source miter gates) can potentially mitigate the impact of limited monitoring data of the target gate and model uncertainty of $\mathbf{y}_{\text{tg}} = G(\boldsymbol{\theta}, \mathbf{u})$ on damage state estimation, effectively leveraging these data sources for Bayesian damage estimation of a target miter gate (i.e., solving Eq. (3.4)) remains a formidable challenge. This challenge is mainly caused by the disconnect between the damage state, $\boldsymbol{\theta}_{\text{tg},i}$, of the target miter gate and the monitoring and inspection data (i.e., $\tilde{\mathbf{y}}_{o,\text{sr}}$) of similar, yet different, miter gates. Motivated by overcoming this challenge, in this paper, we propose a novel framework called BiEDT that integrates domain adaptation/translation techniques with Bayesian inference-based damage diagnostics. In the subsequent sections, we will present the proposed BiEDT framework in detail.

3.3.2 Overview of the proposed BiEDT framework

Figure 3.4 provides an overview of the proposed method. The basic idea is to leverage the inspection and monitoring data from similar, yet different, miter gates to develop an alternative method for estimating the damage state of the target miter gate. This method will complement the commonly employed Bayesian model updating technique, which relies on computational simulation models (see descriptions in Sec. 3.2.1). Once the alternative

estimation method is established, we will integrate the damage state estimations from various approaches using a Bayesian framework.

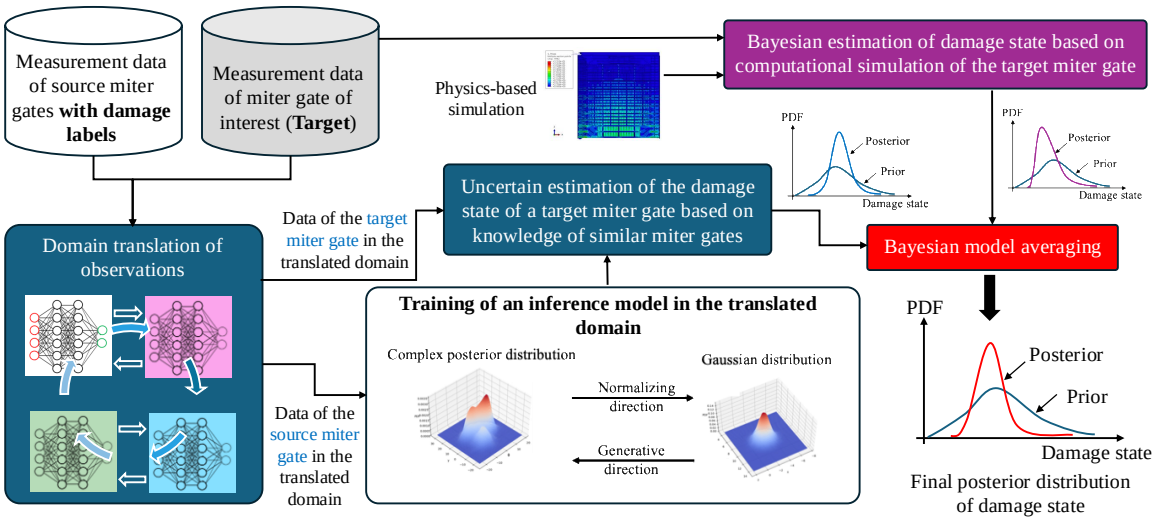


Figure 3.4: Overview of the proposed BiEDT framework

More specifically, in order to overcome the challenge that $\boldsymbol{\theta}_{\mathrm{tg},i}$ and $\tilde{\mathbf{y}}_{o,\mathrm{sr}}$ are disconnected, we first employ domain translation techniques to translate measurements of the source and target miter gates into the same analysis domain as follows

$$\begin{aligned} \boldsymbol{\eta}_{o,\mathrm{tg},i} &= \varphi(\mathbf{y}_{o,\mathrm{tg},i}),\ i = 1, \cdots, N_{ot}, \\ \boldsymbol{\eta}_{o,\mathrm{sr},i} &= \varphi(\mathbf{y}_{o,\mathrm{sr},i}),\ i = 1, \cdots, N_{os}, \end{aligned} \tag{3.5}$$

where $\boldsymbol{\eta}_{o,\mathrm{tg},i}, \boldsymbol{\eta}_{o,\mathrm{sr},i} \in \Omega_\eta$, which is a domain common to both the source and target miter gates, and $\varphi(\cdot)$ is a domain translator which translates measurements from one domain to a domain of interest. As detailed in Sec. 3.3.3, two types of domain translation/adaptation approaches will be investigated to achieve the purpose of measurement translation.

Since there is a unknown nonlinear relationship between $\{\boldsymbol{\theta}_{\mathrm{sr},i}, i = 1, \cdots, N_{os}\}$ and $\{\boldsymbol{\eta}_{o,\mathrm{sr},i}, i = 1, \cdots, N_{os}\}$, the same unknown relationship also exists between $\boldsymbol{\theta}_{\mathrm{tg},i}$ and $\boldsymbol{\eta}_{o,\mathrm{tg},i}$ in the translated domain, Ω_η. Given that $\boldsymbol{\eta}_{o,\mathrm{sr},i} \in \Omega_\eta$ and $\boldsymbol{\eta}_{o,\mathrm{tg},i} \in \Omega_\eta$ are in the same domain, we, therefore, can build a model to learn the unknown nonlinear relationship between the damage state $\boldsymbol{\theta}$ and the translated observations $\boldsymbol{\eta}$ of the miter gates using data $\{\boldsymbol{\theta}_{\mathrm{sr},i}, i = 1, \cdots, N_{os}\}$ and $\{\boldsymbol{\eta}_{o,\mathrm{sr},i}, i = 1, \cdots, N_{os}\}$ as follows

$$\boldsymbol{\theta} = H(\boldsymbol{\eta}; (\tilde{\boldsymbol{\eta}}_{o,\mathrm{sr}}, \tilde{\boldsymbol{\theta}}_{\mathrm{sr}})), \tag{3.6}$$

where $H(\cdot)$ is a probabilistic function that maps $\boldsymbol{\eta}$ to $\boldsymbol{\theta}$, which is trained using data of

$\{\tilde{\boldsymbol{\eta}}_{o,\text{sr}}, \tilde{\boldsymbol{\theta}}_{\text{sr}}\}$, and $\tilde{\boldsymbol{\eta}}_{o,\text{sr}} \doteq \{\boldsymbol{\eta}_{o,\text{sr},i}, i = 1, \cdots, N_{os}\}$. In this paper, a conditional invertible neural network is employed to build such a model for the mapping. More details are provided in Sec. 3.3.4.

By using the translated measurements of the target miter gate and the model learned in Eq. (3.6), we can then estimate the damage state of the target miter gate as

$$\hat{\boldsymbol{\theta}}_{\text{tg},i} = H(\boldsymbol{\eta}_{o,\text{tg},i}; (\tilde{\boldsymbol{\eta}}_{o,\text{sr}}, \tilde{\boldsymbol{\theta}}_{\text{sr}})), \tag{3.7}$$

where $\boldsymbol{\eta}_{o,\text{tg},i} = \varphi(\mathbf{y}_{o,\text{tg},i})$ as given in Eq. (3.5).

Until now, we established an approach to probabilistically estimate the damage state of the target miter gate by leveraging inspection and monitoring data of source miter gates. This proposed approach integrates domain translation techniques with probabilistic damage diagnostics to fill the gap between two types of similar but different miter gates. In addition to this new way of estimating the damage state, as mentioned above, we also have another way of estimating the damage state of the target miter gate by using the physics-based computational simulation model (i.e., $\mathbf{y}_{\text{tg}} = G(\boldsymbol{\theta}, \mathbf{u})$) of the target miter gate. Using Bayesian method, the damage state can be estimated using only the measurements of the target miter gate as

$$f(\boldsymbol{\theta}_{\text{tg},i}|(\mathbf{y}_{o,\text{tg},i}; \mathbb{G})) \propto L(\mathbf{y}_{o,\text{tg},i}|(\boldsymbol{\theta}_{\text{tg},i}; \mathbb{G}))f_{\boldsymbol{\theta}}(\boldsymbol{\theta}_{\text{tg},i}), \tag{3.8}$$

where $L(\mathbf{y}_{o,\text{tg},i}|(\boldsymbol{\theta}_{\text{tg},i}; \mathbb{G}))$ is a likelihood function and $f_{\boldsymbol{\theta}}(\boldsymbol{\theta}_{\text{tg},i})$ is the prior distribution of the damage state of the target miter gate.

Both Eqs. (3.7) and (3.8) can provide an estimation of the damage state of the target gate using different approaches. The next question that needs to be answered now is how to combine these two types of estimations together to get a final estimated posterior distribution of the damage state. An intuitive combination of these two types of estimations is to use the estimation given in Eq. (3.7) as the prior distribution of Eq. (3.8). Such an approach, however, risks double-counting the measurement data of the target miter gate in the Bayesian damage diagnostics, as both Eqs. (3.7) and (3.8) use $\mathbf{y}_{o,\text{tg},i}$ as inputs, despite Eq. (3.7) only employing the translated information from $\mathbf{y}_{o,\text{tg},i}$.

To mitigate the risk of using the same information twice, a Bayesian model averaging approach is developed by combining estimations from the two different types of approaches (as illustrated in Fig. 3.4) [76]. The posterior distribution given in Eq. (3.4) is thus approximated as

$$\begin{aligned} &f(\boldsymbol{\theta}_{\text{tg},i}|(\tilde{\mathbf{y}}_{o,\text{sr}}, \tilde{\boldsymbol{\theta}}_{\text{sr}}, \mathbf{y}_{o,\text{tg},i}; \mathbb{G})) \\ &\approx f_{DT}(\boldsymbol{\theta}_{\text{tg},i})p(\mathbb{H}|\mathbf{D}) + f(\boldsymbol{\theta}_{\text{tg},i}|(\mathbf{y}_{o,\text{tg},i}; \mathbb{G}))p(\mathbb{G}|\mathbf{D}), \end{aligned} \tag{3.9}$$

where $f_{DT}(\boldsymbol{\theta}_{\text{tg},i})$ represents the posterior distribution obtained from domain translation (i.e., Eq. (3.6)), $\mathbb{H}$ represents the overall domain translation model for estimating the damage state of the target miter gate, $\mathbf{D} \doteq \{\tilde{\mathbf{y}}_{o,\text{sr}}, \tilde{\boldsymbol{\theta}}_{\text{sr}}, \mathbf{y}_{o,\text{tg},i}\}$, and $p(\mathbb{H}|\mathbf{D})$ and $p(\mathbb{G}|\mathbf{D})$ represent respectively the weights of different approaches or models in the model averaging. In addition to the above model averaging, we can also select which model or poster distribution to use in decision-making through Bayesian hypothesis testing. More details of such a Bayesian model averaging or model selection are presented in Sec. 3.3.5.

In summary, as shown in Fig. 3.4, the proposed framework consists of three main modules:

1. *Domain translation of observations*: This module concentrates on translating observations of both the target gate and the source gates into a common domain. This translation is essential for building a connection between the damage state of the target miter gate and the inspection and measurements of the source miter gates.

2. *Bayesian inference-based damage diagnostics in the translated domain*: This module focuses on building a model for inferring the damage state of the target miter gate based on observation data with labeled damage states of the source miter gates in the translated domain.

3. *Bayesian model averaging for the integration of probabilistic estimations*: Building upon results and models from the previous two modules, as illustrated in Fig. 3.4, this module combines both posterior distribution from computational simulation (i.e., Eq. (3.7)) and inference results in translated domain (i.e., Eq. (3.8)) by solving Eq. (3.9), thereby obtain the final posterior distribution of the damage state of the target miter gate.

In the following subsections, we explain each of the above modules in detail.

3.3.3 Domain translation of observations

In this paper, two types of domain-translation approaches are investigated to perform domain translation of miter gate observations. The two approaches, namely the CycleGAN-based method and Domain-Adversarial Neural Network (DANN)-based method, translate observations of a target miter gate to different domains. In particular, the CycleGAN-based method translates observations of the target miter gate directly into observations of source miter gates while the DANN-based method translates observations of both the target and source miter gates into a common/shared domain that is different from the original domains of both types of miter gates. In what follows, we explain the main idea of the studied two approaches in detail.

3.3.3.1 CycleGAN-based domain translation

CycleGAN is a method introduced by Zhu *et al.*for addressing unpaired image-to-image translation challenges [46]. It facilitates the transformation of images from one style into another, for example, converting images of horses into zebras or summer scenes into winter [46]. It is an effective technique for discovering and learning the relationships between different unpaired domains.

Fig. 3.5 illustrates the architecture of CycleGAN as it is applied to domain translation for miter gates. The architecture consists of two main components: two generator networks and two discriminator networks. Each generator-discriminator pair is responsible for one direction of image translation between two distinct domains, Domain A (source miter gate) and Domain B (target miter gate). The first generator, $\mathbb{K}(\cdot)$, translates images from Domain A to Domain B, while the second generator, $\mathbb{F}(\cdot)$, performs the inverse, translating images from Domain B to Domain A. Correspondingly, each discriminator, $\mathbb{D}_1(\cdot)$ and $\mathbb{D}_2(\cdot)$, aims to distinguish between the translated images and real images in their respective domains. A novel and critical aspect of CycleGAN is its use of a cycle consistency loss. This loss function ensures that an image from one domain can be translated to the other domain and then back again to the original domain, with the goal of retaining the original image's content and structure [46]. Mathematically, for an image $\mathcal{X}$ from Domain A, the cycle consistency loss ensures that $\mathbb{F}(\mathbb{K}(\mathcal{X})) \approx \mathcal{X}$, and similarly, for an image $\mathcal{Y}$ from Domain B, it ensures that $\mathbb{K}(\mathbb{F}(\mathcal{Y})) \approx \mathcal{Y}$. Next, we will delve into each component of the CycleGAN and provide detailed explanations.

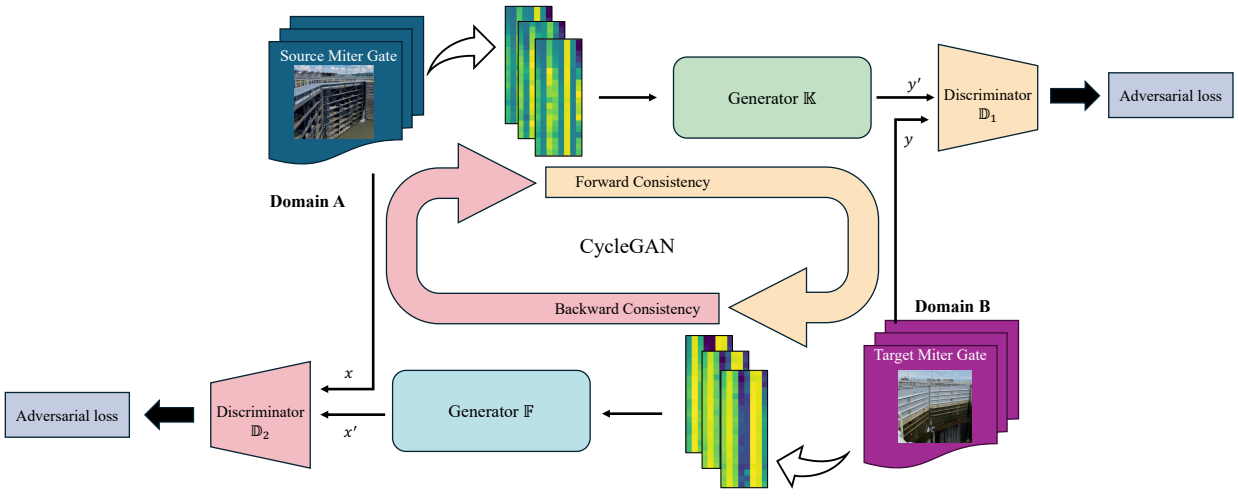


Figure 3.5: The structure of CycleGAN for translation of miter gate observations.

CycleGAN in our study utilizes a modified U-Net architecture for its generators $\mathbb{K}(\cdot)$ and

$\mathbb{F}(\cdot)$. The U-Net architecture is particularly effective due to its encoder-decoder structure with skip connections that help preserve fine-grained details throughout the network:

- *Encoder*: The encoder part typically consists of convolutional layers that successively downsample the input image, capturing increasingly abstract representations of the input.
- *Decoder*: The decoder uses transposed convolutions (or upsampling layers followed by convolutions) to progressively upsample the encoded representations back to the original image size. The upsampling process reintegrates finer details into the synthesized image.
- *Skip Connections*: These connections pass information directly from the encoder to the corresponding layers in the decoder, helping in recovering fine details that might be lost during downsampling.

This architecture allows CycleGAN's generators to efficiently translate the input image from one domain to the other while maintaining critical information. The discriminators in CycleGAN, often referred to as PatchGAN, classify whether small patches of the image are real or fake. This approach differs from traditional full-image discriminators by focusing on the realism of local image patches [46]. PatchGAN consists of a few convolutional layers that downsample the image to a smaller set of patches. Each patch in the output map is independently classified as real or fake, and the discriminator's final output is the average of these classifications. This method is computationally less intensive and focuses on getting high-frequency details right, which is often enough to make the whole image appear convincingly real or accurately translated.

The loss functions in CycleGAN play a crucial role in enabling the effective translation of images between two unpaired domains. Adversarial losses, cycle consistency loss, and identity loss are components in the loss function of CycleGAN. The first part of the adversarial losses is shown as follows [46]

$$\begin{aligned}\mathcal{L}_{\text{GAN}}\left(\mathbb{K}, \mathbb{D}_1, X, Y\right) = & \mathbb{E}_{\mathcal{Y} \sim p_{\text{data}}(\mathcal{Y})}\left[\log \mathbb{D}_1(\mathcal{Y})\right] \\ & + \mathbb{E}_{\mathcal{X} \sim p_{\text{data}}(\mathcal{X})}\left[\log\left(1 - \mathbb{D}_1(\mathbb{K}(\mathcal{X}))\right)\right],\end{aligned} \tag{3.10}$$

where the image input for domain A and B are $X = \{\mathcal{X}_1, \cdots, \mathcal{X}_N\}$, $Y = \{\mathcal{Y}_1, \cdots, \mathcal{Y}_N\}$ respectively. $\mathbb{K}(\cdot)$ is the generator that generates synthetic data for domain B. $\mathbb{D}_1(\cdot)$ is the discriminator that distinguishes between translated samples $\hat{\mathcal{Y}}_j = \mathbb{K}(\mathcal{X}_j)$ and real sample $\mathcal{Y}_j$. The data distributions are denoted as $\mathcal{X} \sim p_{\text{data}}(\mathcal{X})$ and $\mathcal{Y} \sim p_{\text{data}}(\mathcal{Y})$.

Another adversarial loss function is given as [46]

$$\begin{aligned}\mathcal{L}_{\text{GAN}}\left(\mathbb{F},\mathbb{D}_2,Y,X\right)=&\mathbb{E}_{\mathcal{X}\sim p_{\text{data}}\left(\mathcal{X}\right)}\left[\log\mathbb{D}_2(\mathcal{X})\right]\\&+\mathbb{E}_{\mathcal{Y}\sim p_{\text{data}}\left(\mathcal{Y}\right)}\left[\log\left(1-\mathbb{D}_2(\mathbb{F}(\mathcal{Y}))\right)\right]\end{aligned},\tag{3.11}$$

where $\mathbb{F}(\cdot)$ is the generator that generates synthetic data for domain A. $\mathbb{D}_2(\cdot)$ is the discriminator that distinguishes between translated samples $\hat{\mathcal{X}}_j=\mathbb{F}(\mathcal{Y}_j)$ and real sample $\mathcal{X}_j$.

The cycle consistency loss is given as [46]

$$\begin{aligned}\mathcal{L}_{\text{cyc}}(\mathbb{K},\mathbb{F})=&\mathbb{E}_{\mathcal{X}\sim p_{\text{data}}\left(\mathcal{X}\right)}\left[\|\mathbb{F}(\mathbb{K}(\mathcal{X}))-\mathcal{X}\|_1\right]\\&+\mathbb{E}_{\mathcal{Y}\sim p_{\text{data}}\left(\mathcal{Y}\right)}\left[\|\mathbb{K}(\mathbb{F}(\mathcal{Y}))-\mathcal{Y}\|_1\right]\end{aligned},\tag{3.12}$$

where $\|\cdot\|_1$ denotes the L1 norm, which penalizes the absolute differences between the original and reconstructed images.

The identity mapping loss is given by [46]

$$\mathcal{L}_{\text{id}}(\mathbb{K},\mathbb{F})=\|\mathbb{K}(\mathcal{Y})-\mathcal{Y}\|_1+\|\mathbb{F}(\mathcal{X})-\mathcal{X}\|_1.\tag{3.13}$$

Thus, the overall loss function of CycleGAN is denoted as

$$\begin{aligned}\mathcal{L}(\mathbb{K},\mathbb{F},\mathbb{D}_1,\mathbb{D}_2)=&\mathcal{L}_{\text{GAN}}\left(\mathbb{K},\mathbb{D}_1,X,Y\right)\\&+\mathcal{L}_{\text{GAN}}\left(\mathbb{F},\mathbb{D}_2,Y,X\right),\\&+\mathcal{L}_{\text{cyc}}(\mathbb{K},\mathbb{F})+\mathcal{L}_{\text{id}}(\mathbb{K},\mathbb{F}),\end{aligned}\tag{3.14}$$

and we aim to solve

$$\mathbb{K}^*,\mathbb{F}^*=\arg\min_{\mathbb{F},\mathbb{K}}\max_{\mathbb{D}_1,\mathbb{D}_2}\mathcal{L}\left(\mathbb{K},\mathbb{F},\mathbb{D}_1,\mathbb{D}_2\right).\tag{3.15}$$

For the training of the CycleGAN model in this paper, we have $X\doteq\{\mathbf{y}_{o,\text{tg},i},\forall i=1,\cdots,N_{ot}\}$ and $Y\doteq\{\mathbf{y}_{o,\text{sr},i},\forall i=1,\cdots,N_{os}\}$. After the training of the CycleGAN model, we can translate observations of the target miter gate into their counterparts of the source miter gates as

$$\hat{\mathbf{y}}_{o,\text{tg2sr},i}\approx\mathbb{F}(\mathbf{y}_{o,\text{tg},i}),\forall i=1,\cdots,N_{ot}.\tag{3.16}$$

Additionally, we can obtain the reconstrcuted observations of the source miter gates through the CycleGAN model as

$$\hat{\mathbf{y}}_{o,\text{sr2sr},i}\approx\mathbb{F}(\mathbb{K}(\mathbf{y}_{o,\text{sr},i})),\forall i=1,\cdots,N_{os}.\tag{3.17}$$

3.3.3.2 Domain-Adversarial Neural Network (DANN) -based domain translation

The Domain-Adversarial Neural Network (DANN) introduces a representation learning method for domain adaptation, where training and testing data originate from similar yet different distributions [77]. This model is designed to work with labeled data from the source domain and unlabeled data from the target domain, eliminating the need for labeled target-domain data. Throughout training, DANN encourages the development of features that are both discriminative for the task at hand within the source domain and non-discriminative regarding the domain differences. This dual capability is facilitated by introducing a few standard layers and a gradient reversal layer, enabling training via standard backpropagation and stochastic gradient descent [77].

Fig. 3.6 illustrates the architecture of DANN for the application to domain translation of miter gates. DANN was originally developed primarily for classification tasks. However, when adapting DANN for a regression task instead, the structural modifications primarily involve the output layer and the loss functions used, while retaining the core adversarial architecture. Here's how the model is structured

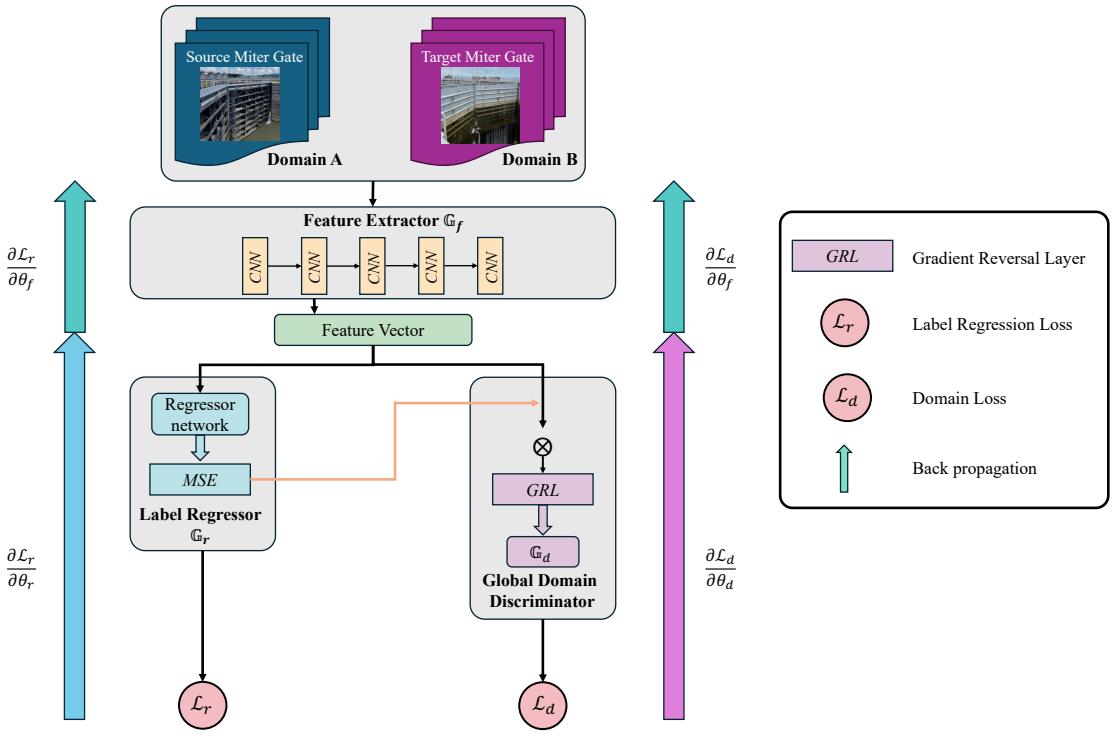


Figure 3.6: The structure of DANN for translation of miter gate observations.

- *Feature Extractor*: This is a neural network that processes input data to extract useful

features. The architecture of this network includes multiple dense layers designed to capture the nuances necessary for predicting continuous values.

- *Domain Classifier*: This component tries to determine the domain of the input features (source vs. target) and is pivotal for ensuring that the features are domain-invariant. It typically consists of a few dense layers followed by a binary classification output that indicates the domain.

- *Gradient Reversal Layer (GRL)*: Positioned between the feature extractor and the domain classifier, the GRL is crucial for the adversarial training aspect of DANN. It reverses the gradient during backpropagation, effectively making the domain classifier's job harder, which in turn forces the feature extractor to produce more domain-invariant features.

- *Regressor*: A regressor is used at the output of the feature extractor for the primary task of regression. The activation function is linear.

DANN incorporates concepts from GANs to facilitate the development of features that are effective across different domains. This adversarial learning approach involves a two-player game: the first player is a domain classifier $\mathbb{G}_d(\cdot)$ tasked with differentiating between source and target domain data. In contrast, the second player is a feature extractor $\mathbb{G}_f(\cdot)$ that aims to generate features indistinguishable by $\mathbb{G}_d(\cdot)$, effectively confusing the discriminator by producing domain-invariant features.

The training of these components is adversarial. The feature extractor $\mathbb{G}_f(\cdot)$'s parameters $\boldsymbol{\theta}_f$ are optimized to maximize the discriminator $\mathbb{G}_d(\cdot)$'s loss, thereby fooling $\mathbb{G}_d(\cdot)$ into misclassifying the domain of the features. Conversely, the parameters $\boldsymbol{\theta}_d$ of $\mathbb{G}_d(\cdot)$ are trained to minimize its loss, improving its ability to classify the domain of the data correctly. Simultaneously, the loss of a label regressor $\mathbb{G}_r(\cdot)$, responsible for the source domain regression, is minimized. The regression loss is given as

$$\mathcal{L}_{\mathrm{r}} = \frac{1}{n_s} \sum_{i=1}^{n_s} \left| r_i - \mathbb{G}_r\left(\mathcal{X}_i\right) \right| \tag{3.18}$$

where n_s represents the sample size of the source domain data and r_i is the label/response of the source domain data.

The domain loss, $\mathcal{L}_{\mathrm{d}}$ is expressed as introduced in [77] by evaluating the difference between the domain label (0 for source and 1 for target domain) and the domain classifier output, $\mathbb{G}_d\left(\mathbb{G}_f\left(\mathcal{X}\right)\right)$. Because there are no labels for target domain data in the training set, the total loss is given as

$$\mathcal{L}\left(\boldsymbol{\theta}_f, \boldsymbol{\theta}_r, \boldsymbol{\theta}_d\right) = \mathcal{L}_{\mathrm{r}} + \lambda \mathcal{L}_{\mathrm{d}}, \tag{3.19}$$

where λ is a weighting parameter that balances the contribution of the adversarial loss. And we intend to solve [77]

$$\begin{aligned} \left(\hat{\boldsymbol{\theta}}_f, \hat{\boldsymbol{\theta}}_r\right) &= \arg\min_{\theta_f, \theta_r} \mathcal{L}\left(\boldsymbol{\theta}_f, \boldsymbol{\theta}_r, \hat{\boldsymbol{\theta}}_d\right), \\ \hat{\boldsymbol{\theta}}_d &= \arg\max_{\theta_d} \mathcal{L}\left(\hat{\boldsymbol{\theta}}_f, \hat{\boldsymbol{\theta}}_y, \boldsymbol{\theta}_d\right). \end{aligned} \tag{3.20}$$

For the training of the DANN model in this paper, the data used include $\{\mathbf{y}_{o,\mathrm{tg},i}, \forall i = 1, \cdots, N_{ot}\}$, $\{\boldsymbol{\theta}_{\mathrm{sr},i}, i = 1, \cdots, N_{os}\}$, and $\{\mathbf{y}_{o,\mathrm{sr},i}, \forall i = 1, \cdots, N_{os}\}$. After the training of the DANN model, we have the common features of the observations of the target miter gate and the source miter gates as

$$\begin{aligned} \boldsymbol{\eta}_{\mathrm{DANN,tg},i} &\approx \varphi_{\mathrm{DANN}}(\mathbf{y}_{o,\mathrm{tg},i}),\ i = 1, \cdots, N_{ot}, \\ \boldsymbol{\eta}_{\mathrm{DANN,sr},i} &\approx \varphi_{\mathrm{DANN}}(\mathbf{y}_{o,\mathrm{sr},i}),\ i = 1, \cdots, N_{os}, \end{aligned} \tag{3.21}$$

where $\boldsymbol{\eta}_{\mathrm{DANN,tg},i}$ and $\boldsymbol{\eta}_{\mathrm{DANN,sr},i}$ are respectively the translated observations of the target and source miter gates, and $\varphi_{\mathrm{DANN}}(\cdot)$ represents the overall DANN model.

3.3.4 Bayesian inference-based damage diagnostics in the translated domain

Using the approaches described in Sec. 3.3.3, we can translate the observations of the source and target miter gates into a common domain. Based on the translated data from the source miter gates, in this section, we construct an inverse model to estimate the damage state of the target gate for given observations in the translated domain (i.e., Eqs. (3.6) and (3.7)). Given the complex nonlinear relationship between the damage state and the transformed observations, along with various sources of uncertainty in the domain translation, we employ a normalizing flow-based approach [78, 79, 80]. This method uses conditional invertiable neural networks (cINN) to construct the inverse model and perform Bayesian inference-based damage diagnostics of the target miter gate in the translated domain. In what follows, before we provide a brief overview of Bayesian inference using cINN, we explain the basics of normalizing flow.

3.3.4.1 Conditional invertible neural networks as a regression model for probabilistic estimation

Normalizing flows are generative techniques that can flexibly approximate any arbitrary probability distribution $f(\boldsymbol{\theta}|\boldsymbol{\beta})$ over a continuous domain. This is achieved by starting with a regular/base distribution $f_z(\mathbf{z})$ and applying a series of bijective transformations [78, 79]. Let $\boldsymbol{\theta}$ be a vector of damage state variables and $\boldsymbol{\beta}$ be a vector denoting observations, as illustrated in Fig. 3.7, there are two types of flow transformation, namely generative direction and normalizing direction. In the *generative direction*, we can map a regular or base distribution $f_z(\mathbf{z})$ (e.g., Gaussian distribution) to an irregular distribution. This allows for the generation of samples of $\boldsymbol{\theta}$ from distribution $f(\boldsymbol{\theta}|\boldsymbol{\beta})$ by sampling $\mathbf{z}$ from distribution $f_z(\mathbf{z})$. In the *normalizing direction*, we can operate the opposite by transforming an irregular distribution to a normal form of the base distribution $f_z(\mathbf{z})$.

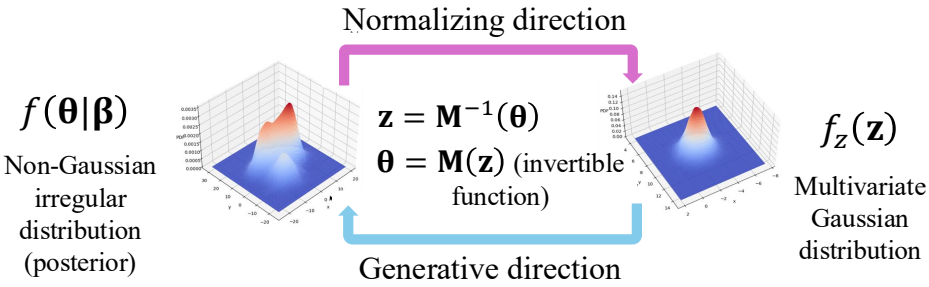


Figure 3.7: Transformation of distributions in generative models [3]

A key to enable the above bijective transformations is an invertible function $\boldsymbol{\theta} = M(\mathbf{z})$, where $M(\cdot)$ is an invertible function and both $M(\cdot)$ and $M^{-1}(\cdot)$ are differentiable. If such a invertible and differentiable function exists, we can compute the probability density function $f(\boldsymbol{\theta}|\boldsymbol{\beta})$ of $\boldsymbol{\theta}$ as

$$\begin{aligned} f(\boldsymbol{\theta}|\boldsymbol{\beta}) &= f_z(\mathbf{z})|\det\ J_M(\mathbf{z})|^{-1} \\ &= f_z(M^{-1}(\boldsymbol{\theta}))\,|\det\ J_{M^{-1}}(\boldsymbol{\theta})|\,, \end{aligned} \tag{3.22}$$

where $\mathbf{z} = M^{-1}(\boldsymbol{\theta})$, $J_M(\mathbf{z})$ is the Jacobian matrix of $\boldsymbol{\theta} = M(\mathbf{z})$, and $J_{M^{-1}}(\boldsymbol{\theta})$ is the Jacobian of $M^{-1}(\cdot)$.

In practice, the invertible function $\boldsymbol{\theta} = M(\mathbf{z})$ is usually not directly available or explicitly defined. The added complexity of translating the domain of the observations further complicates the construction of such functions. One approach to overcome the challenge of constructing the invertible function is to learn the function based on data using invertible neural networks (INN) [81, 82, 83]. INNs are neural network architectures proposed in Ref.

[84]. The basic building block of an INN is the affine coupling block as illustrated in Fig. 3.8. Each block consists of two complementary affine coupling layers that split the input vector $\boldsymbol{\omega}$ into two halves, namely $\boldsymbol{\omega}_1$ and $\boldsymbol{\omega}_2$. The split inputs are then transformed by an affine function using element-wise multiplication and addition to become inputs of the next block as follows [81]

$$\begin{aligned} \boldsymbol{\alpha}_1 &= \boldsymbol{\omega}_1 \odot \exp(s_2(\boldsymbol{\omega}_2)) + t_2(\boldsymbol{\omega}_2), \\ \boldsymbol{\alpha}_2 &= \boldsymbol{\omega}_2 \odot \exp(s_1(\boldsymbol{\alpha}_1)) + t_1(\boldsymbol{\alpha}_1), \end{aligned} \tag{3.23}$$

where $\odot$ is element-wise multiplication.

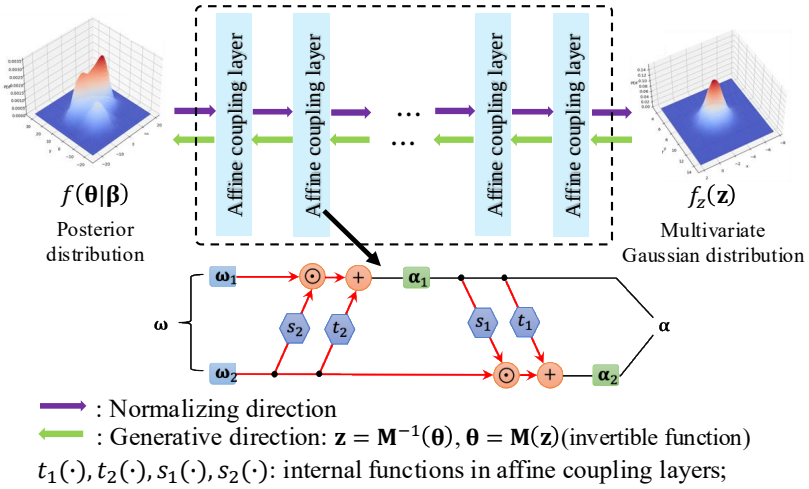


Figure 3.8: Illustration of a cINN model as a regression model for Bayesian inference

Similar to Eq. (3.23), the inverse direction operation can also be easily performed using the affine coupling blocks. By stacking multiple blocks together, the invertible neural networks can approximate the invertible function $\boldsymbol{\theta} = M(\mathbf{z})$. By further integrating the INN model architecture with an additional neural network, we can approximate the posterior distribution $f(\boldsymbol{\theta}|\boldsymbol{\beta})$, which represents the conditional probability of $\boldsymbol{\theta}$ given the observations $\boldsymbol{\beta}$. This integrated architecture is known as a conditional invertible neural network (cINN) [85]. Consequently, the cINN model serves as a probabilistic regression model for Bayesian inference. For more detailed information on INN and cINN, we refer interested readers to Refs. [81] and [82].

3.3.4.2 Training a cINN model based on domain translation

Let $\hat{\boldsymbol{\lambda}}$ represent the parameters of the cINN model $\mathbf{z} = \gamma_{\boldsymbol{\lambda}}(\boldsymbol{\theta}; \boldsymbol{\beta})$, which include both the INN for bijective mapping and an additional neural network for conditioning. These

parameters are estimated by solving the following optimization problem

$$\begin{aligned}\hat{\boldsymbol{\lambda}} &= \arg\min_{\boldsymbol{\lambda}} \mathbb{E}_{p(\boldsymbol{\beta})}\left[\mathbb{KL}(f(\boldsymbol{\theta}|\boldsymbol{\beta}) \,\|\, f_{\boldsymbol{\lambda}}(\boldsymbol{\theta}|\boldsymbol{\beta}))\right], \\ &= \arg\min_{\boldsymbol{\lambda}} \iint f(\boldsymbol{\theta},\boldsymbol{\beta}) \log\left\{f_{\boldsymbol{\lambda}}(\boldsymbol{\theta}|\boldsymbol{\beta})\right\} \mathbf{d}\boldsymbol{\beta}\mathbf{d}\boldsymbol{\theta},\end{aligned} \tag{3.24}$$

where $\mathbb{E}(\cdot)$ represents expectation and $\mathbb{KL}(\cdot)$ is the Kullback–Leibler (KL) divergence between the true and the approximate posterior distributions.

The objective function of the above equation can be approximated by a Monte Carlo estimate as follows [82]

$$\hat{\boldsymbol{\lambda}} = \arg\min_{\boldsymbol{\lambda}} \frac{1}{N_o} \sum_{j=1}^{N_o} \left(\frac{\left\| \gamma_{\boldsymbol{\lambda}}(\boldsymbol{\theta}^{(j)}; \boldsymbol{\beta}^{(j)}) \right\|_2^2}{2} - \log\left| \det\ J_{\gamma_{\boldsymbol{\lambda}}}^{(j)} \right| \right), \tag{3.25}$$

where N_o is the number of training data and $J_{\gamma_{\boldsymbol{\lambda}}}^{(j)}$ is the Jacobian of $\mathbf{z} = \gamma_{\boldsymbol{\lambda}}(\boldsymbol{\theta}; \boldsymbol{\beta})$ evaluated at the j-th sample of $\boldsymbol{\theta}$ and $\boldsymbol{\beta}$.

If the CycleGAN-based approach (i.e., Sec. 3.3.3.1) is employed to translate the observations of the target miter gate to that of the source miter gates, we have the training data as $\{\boldsymbol{\theta}_{\mathrm{sr},j}, j = 1, \cdots, N_{os}\}$ and $\{\boldsymbol{\beta}^{(j)} \doteq \hat{\mathbf{y}}_{o,\mathrm{sr2sr},j}, j = 1, \cdots, N_{os}\}$. If DANN-based approach (i.e., Sec. 3.3.3.2) is employed to perform observation translation, we have the training data as $\{\boldsymbol{\theta}_{\mathrm{sr},j}, j = 1, \cdots, N_{os}\}$ and $\{\boldsymbol{\beta}^{(j)} \doteq \boldsymbol{\eta}_{\mathrm{DANN,sr},j}, j = 1, \cdots, N_{os}\}$ in Eq. (3.25). After the estimation of $\hat{\boldsymbol{\lambda}}$, the trained cINN model can be used to approximate the posterior distribution of $\hat{f}(\boldsymbol{\theta}|\boldsymbol{\beta})$ for given observation $\boldsymbol{\beta}$.

3.3.4.3 Damage diagnostics using Bayesian inference in the translated domain

Once a cINN model has been successfully trained in the translated domain, we can generate samples from the posterior distribution of the damage state for any given observation $\boldsymbol{\beta}$ of the target miter gate. This is achieved by first generating N_{MCS} samples of $\mathbf{z}$ from $f_z(\mathbf{z})$. Let the samples be $\{\mathbf{z}^{(i)}, i = 1, \cdots, N_{MCS}\}$, then we have the posterior samples of $\hat{f}(\boldsymbol{\theta}|\boldsymbol{\beta})$ using the cINN model as

$$\boldsymbol{\theta}^{(i)} = \gamma_{\boldsymbol{\lambda}}^{-1}(\mathbf{z}^{(i)}; \boldsymbol{\beta}), i = 1, \cdots, N_{MCS}, \tag{3.26}$$

where $\gamma_{\boldsymbol{\lambda}}^{-1}(\cdot)$ stands for the generative direction of operation of the model, $\boldsymbol{\beta}$ is $\hat{\mathbf{y}}_{o,\mathrm{tg2sr},j}, j = 1, \cdots, N_{ot}$, if CycleGAN-based approach is employed for domain translation and is $\boldsymbol{\eta}_{\mathrm{DANN,tg},i}$ if DANN-based approach is used.

Based on the posterior samples, the posterior distribution $f_{DT}(\boldsymbol{\theta}_{\mathrm{tg},i})$ (i.e., given in Eq.

(3.9)) of the damage state can be obtained using domain translation by leveraging information from similar, but different miter gates.

3.3.5 Bayesian model averaging for the integration of different estimations

As discussed in Sec. 4.2 and depicted in Fig. 3.4, we can also conduct damage diagnostics using Bayesian inference through computational simulation of the target miter gate. This method complements the previously described diagnostics based on domain translation. In this section, we first introduce damage diagnostics utilizing physics-based Bayesian inference. Subsequently, we explore the integration of estimations from different approaches through Bayesian model averaging.

3.3.5.1 Damage diagnostics using physics-based Bayesian inference

In damage diagnostics using physics-based computational simulation, we calibrate the damage variable of the simulation model to minimize the discrepancy between the model prediction and actual observations. A commonly used approach to solve such an inverse problem using computational simulation is Bayesian inference or Bayesian model updating [86], which estimates the damage state as follows

$$\begin{aligned} &f_{\boldsymbol{\theta}|\mathbf{y}}(\boldsymbol{\theta}_{\mathrm{tg},i}|\mathbf{y}_{o,\mathrm{tg},i}, \mathbf{u}_{\mathrm{tg},i}; \mathbb{G}) \\ &\propto L_{\mathbf{y}|\boldsymbol{\theta}}(\mathbf{y}_{o,\mathrm{tg},i}|\boldsymbol{\theta}_{\mathrm{tg},i}, \mathbf{u}_{\mathrm{tg},i}; \mathbb{G}) f_{\boldsymbol{\theta}}(\boldsymbol{\theta}_{\mathrm{tg},i}), \end{aligned} \tag{3.27}$$

where $f_{\boldsymbol{\theta}}(\boldsymbol{\theta}_{\mathrm{tg},i})$ is the prior distribution of the damage state and $L_{\mathbf{y}|\boldsymbol{\theta}}(\mathbf{y}_{o,\mathrm{tg},i}|\boldsymbol{\theta}_{\mathrm{tg},i}, \mathbf{u}_{\mathrm{tg},i}; \mathbb{G})$ is a likelihood function computed using the computational model $\mathbb{G}$ (i.e., $\mathbf{y}_{\mathrm{tg}} = G(\boldsymbol{\theta}, \mathbf{u})$) by considering uncertainty sources. Fig. 3.9 shows an example of a computational simulation model of structural analysis for a miter gate.

If the major uncertainty sources considered are the observation noises which could include both the sensor measurement noises and the uncertainty in the model prediction due to unmodeled physics, the likelihood function given in Eq. (3.27) can be derived as [32]

$$\begin{aligned} &L_{\mathbf{y}|\boldsymbol{\theta}}(\mathbf{y}_{o,\mathrm{tg},i}|\boldsymbol{\theta}_{\mathrm{tg},i}, \mathbf{u}_{\mathrm{tg},i}; \mathbb{G}) \\ &= \frac{\exp\left(-0.5(\mathbf{y}_{o,\mathrm{tg},i} - \mathbf{y}_{\mathrm{tg},i})^T \boldsymbol{\Sigma}^{-1} (\mathbf{y}_{o,\mathrm{tg},i} - \mathbf{y}_{\mathrm{tg},i})\right)}{\sqrt{(2\pi)^{N_t \times N_s} |\boldsymbol{\Sigma}|}}, \end{aligned} \tag{3.28}$$

in which $\boldsymbol{\Sigma}$ is the covariance matrix which is usually modeled as a diagonal matrix with

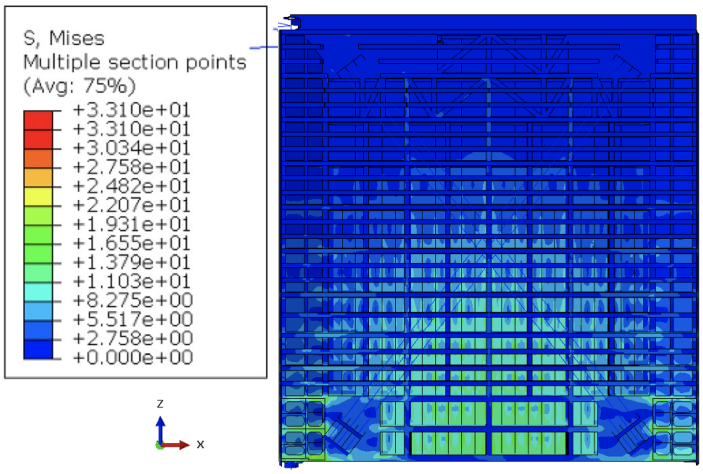


Figure 3.9: An example of FEA model of miter gates

the diagonal elements being the unknown variance σ_ε^2 of the observation noises, and $\mathbf{y}_{\text{tg},i} \in \mathbb{R}^{N_t \times N_s}$ is the predicted strain response using the physics-based simulation model $\mathbb{G}$. In this paper, the unknown variance σ_ε^2 is also treated as part of the estimation parameters $\boldsymbol{\theta}$ which includes both the gap length and the observation noise variance.

Once the likelihood function is derived, the posterior distribution of the target miter gate's damage state, as given in Eq. (3.27), can be estimated using various Bayesian methods, such as Markov Chain Monte Carlo simulation and particle filtering [87, 34]. While the physics-based Bayesian inference method can estimate the posterior distribution of the damage state, this estimate may be biased due to imperfections in the physics-based simulation model, such as the unmodeled physics due to misunderstanding or simplification, numerical discretization error, etc. Although numerous approaches have been developed over the past decades to address model uncertainty in Bayesian model calibration [88, 89, 75, 90], the bias in posterior distribution estimation can only be mitigated, not eliminated. Furthermore, a limited number of observations or monitoring data for the target miter gate can introduce additional bias or significantly increase uncertainty in the damage state estimation.

The method proposed in Sec. 3.3.4 offers a data-driven alternative to estimating the posterior distribution of the damage state for the target miter gate, based on domain translation and normalizing flows. By leveraging inspection and monitoring data from similar, but different, miter gates, this approach bypasses the need for a physics-based simulation model of the target miter gate. Consequently, it has the potential to overcome the limitations and disadvantages associated with physics-based methods (i.e., Sec. 3.3.5.1), as well as the constraint posed by the limited monitoring data for the target gate. Given that both approaches have their own strengths and weaknesses, we will next discuss how to combine

their estimations to achieve a final posterior distribution estimation of the damage state for the target miter gate.

3.3.5.2 Bayesian model averaging for damage diagnostics

In this paper, we combine the posterior distribution from the two different approaches above through a Bayesian model averaging framework. Note that there are actually three approaches/models if we consider the two different domain translation methods as two different variants. The three approaches are (1) the CycleGAN-based method, (2) the DANN-based method, and (3) the physics-based Bayesian inference method. The Bayesian model averaging of the results from different methods avoids the risk of double-counting the monitoring data of the target miter gate in Bayesian damage diagnostics. In the Bayesian model averaging setting, we have the final estimation of the posterior distribution of the damage state as

$$\begin{aligned} & f(\boldsymbol{\theta}_{\mathrm{tg},i}|(\tilde{\mathbf{y}}_{o,\mathrm{sr}}, \tilde{\boldsymbol{\theta}}_{\mathrm{sr}}, \mathbf{y}_{o,\mathrm{tg},i}; \mathbb{G})) \\ & \approx f_{DT}(\boldsymbol{\theta}_{\mathrm{tg},i}; \mathbb{H}_{\mathrm{CG}}) p(\mathbb{H}_{\mathrm{CG}}|\mathbf{D}) \\ & + f_{DT}(\boldsymbol{\theta}_{\mathrm{tg},i}; \mathbb{H}_{\mathrm{DN}}) p(\mathbb{H}_{\mathrm{DN}}|\mathbf{D}) \\ & + f(\boldsymbol{\theta}_{\mathrm{tg},i}|(\mathbf{y}_{o,\mathrm{tg},i}, \mathbf{u}_{\mathrm{tg},i}; \mathbb{G})) p(\mathbb{G}|\mathbf{D}), \end{aligned} \tag{3.29}$$

where $f_{DT}(\boldsymbol{\theta}_{\mathrm{tg},i}; \mathbb{H}_{\mathrm{CG}})$ represents the posterior distribution of the damage state estimated based on domain translation using CycleGAN and normalizing flow using historical data from source miter gates (i.e., Secs. 3.3.3 and 3.3.4), $f_{DT}(\boldsymbol{\theta}_{\mathrm{tg},i}; \mathbb{H}_{\mathrm{DN}})$ denotes the posterior distribution obtained using DANN-based domain translation, $f(\boldsymbol{\theta}_{\mathrm{tg},i}|(\mathbf{y}_{o,\mathrm{tg},i}, \mathbf{u}_{\mathrm{tg},i}; \mathbb{G}))$ is the posterior distribution obtained using physics-based method in Sec. 3.3.5.1, and $p(\mathbb{H}_{\mathrm{CG}}|\mathbf{D})$, $p(\mathbb{H}_{\mathrm{DN}}|\mathbf{D})$, and $p(\mathbb{G}|\mathbf{D})$ are weights of the three different approaches, which are respectively given by

$$p(\mathbb{G}|\mathbf{D}) = \frac{p(\mathbf{D}|\mathbb{G})P(\mathbb{G})}{p(\mathbf{D})}, \tag{3.30}$$

$$p(\mathbb{H}_{\mathrm{CG}}|\mathbf{D}) = \frac{p(\mathbf{D}|\mathbb{H}_{\mathrm{CG}})P(\mathbb{H}_{\mathrm{CG}})}{p(\mathbf{D})}, \tag{3.31}$$

and

$$p(\mathbb{H}_{\mathrm{DN}}|\mathbf{D}) = \frac{p(\mathbf{D}|\mathbb{H}_{\mathrm{DN}})P(\mathbb{H}_{\mathrm{DN}})}{p(\mathbf{D})}, \tag{3.32}$$

where $P(\mathbb{G})$, $P(\mathbb{H}_{\text{CG}})$, and $P(\mathbb{H}_{\text{DN}})$ are the prior weights of the three methods. Without any additional information available, we can assign equal weights to all approaches, and $p(\mathbf{D})$ is given by

$$\begin{aligned} p(\mathbf{D}) &= p(\mathbf{D}|\mathbb{G})P(\mathbb{G}) + p(\mathbf{D}|\mathbb{H}_{\text{CG}})P(\mathbb{H}_{\text{CG}}) \\ &+ p(\mathbf{D}|\mathbb{H}_{\text{DN}})P(\mathbb{H}_{\text{DN}}). \end{aligned} \tag{3.33}$$

In the above equations, $p(\mathbf{D}|\mathbb{G})$ can be easily computed based on Eq. (3.28) as

$$p(\mathbf{D}|\mathbb{G}) = \int L_{\mathbf{y}|\boldsymbol{\theta}}(\mathbf{y}_{o,\text{tg},i}|\boldsymbol{\theta}_{\text{tg},i}, \mathbf{u}_{\text{tg},i}, \mathbb{G}) f_{\boldsymbol{\theta}}(\boldsymbol{\theta}_{\text{tg},i}|\mathbb{G}) d\boldsymbol{\theta}_{\text{tg},i}, \tag{3.34}$$

where $f_{\boldsymbol{\theta}}(\boldsymbol{\theta}_{\text{tg},i}|\mathbb{G})$ is the prior distribution of $\boldsymbol{\theta}_{\text{tg},i}$ when the computational simulation model $\mathbb{G}$ is employed for Bayesian model updating. In this paper, a non-informative uniform prior distribution is used.

The estimation of $p(\mathbf{D}|\mathbb{H}_{\text{CG}})$ and $p(\mathbf{D}|\mathbb{H}_{\text{DN}})$ can be accomplished similarly. Taking $p(\mathbf{D}|\mathbb{H}_{\text{CG}})$ as an example, we can compute $p(\mathbf{D}|\mathbb{H}_{\text{CG}})$ as follows

$$p(\mathbf{D}|\mathbb{H}_{\text{CG}}) = \int f_H(\boldsymbol{\eta}_{o,\text{tg},i}|\boldsymbol{\theta}_{\text{tg},i}, \mathbb{H}_{\text{CG}}) f_{\boldsymbol{\theta}}(\boldsymbol{\theta}_{\text{tg},i}|\mathbb{H}_{\text{CG}}) d\boldsymbol{\theta}_{\text{tg},i}, \tag{3.35}$$

in which $f_H(\boldsymbol{\eta}_{o,\text{tg},i}|\boldsymbol{\theta}_{\text{tg},i}, \mathbb{H}_{\text{CG}})$ is the likelihood of observing $\boldsymbol{\eta}_{o,\text{tg},i}$ (i.e., $\hat{\mathbf{y}}_{o,\text{tg2sr},i}$ if CycleGAN is employed for domain translation) in the translated domain for given $\boldsymbol{\theta}_{\text{tg},i}$, and $f_{\boldsymbol{\theta}}(\boldsymbol{\theta}_{\text{tg},i}|\mathbb{H}_{\text{CG}})$ is the prior distribution of $\boldsymbol{\theta}_{\text{tg},i}$ when the domain translation model $\mathbb{H}_{\text{CG}}$ is employed for Bayesian damage diagnostics.

However, $f_H(\boldsymbol{\eta}_{o,\text{tg},i}|\boldsymbol{\theta}_{\text{tg},i}, \mathbb{H}_{\text{CG}})$ in Eq. (3.35) is not explicitly known due to the complicated domain translation and the analytically intractable relationship between $\boldsymbol{\theta}_{\text{tg},i}$ and $\boldsymbol{\eta}_{o,\text{tg},i}$. It poses a significant challenge to the Bayesian model averaging given in Eq. (3.29). To tackle this challenge, this paper employs a normalizing flow-based approach to estimate the intractable likelihood [91, 92]. It is based on the change of variables in generative modeling. Similar to the estimation of the posterior distribution in Sec. 3.3.4.1 (i.e., Eq.(3.22)), a normalizing flow model is constructed to estimate the likelihood function as follows

$$\begin{aligned} f(\boldsymbol{\eta}|\boldsymbol{\theta}) &= f_z(\mathbf{z}_\eta)\left|\det\ J_{M_\eta}(\mathbf{z})\right|^{-1} \\ &= f_z({M_\eta}^{-1}(\boldsymbol{\eta}))\left|\det\ J_{{M_\eta}^{-1}}(\boldsymbol{\eta})\right|, \end{aligned} \tag{3.36}$$

where $M_\eta(\cdot)$ represents the invertible function constructed using cINN for the estimation of $f(\boldsymbol{\eta}|\boldsymbol{\theta})$. In this paper, the Bayesflow Python library is employed to learn the likelihood function estimator [92].

Once we have estimated the likelihood function $f(\boldsymbol{\eta}|\boldsymbol{\theta})$ using Eqs. (3.28) and (3.36), we can determine the final estimation of the damage state through Bayesian model averaging as given in Eq. (3.29). Additionally, we can also perform model selection using Bayes factor (i.e., Bayesian Hypothesis Testing) based on Eqs. (3.34) and (3.35) as

$$\begin{aligned} B_{GC} &= \frac{p(\mathbf{D}|\mathbb{G})}{p(\mathbf{D}|\mathbb{H}_{\mathrm{CG}})},\ B_{GD} = \frac{p(\mathbf{D}|\mathbb{G})}{p(\mathbf{D}|\mathbb{H}_{\mathrm{DN}})}, \\ B_{\mathrm{CD}} &= \frac{p(\mathbf{D}|\mathbb{H}_{\mathrm{CG}})}{p(\mathbf{D}|\mathbb{H}_{\mathrm{DN}})}, \end{aligned} \tag{3.37}$$

where B_{GC}, B_{GD}, and B_{CD} are respectively Bayes factor for pairwise model comparison between physics-based method and domain translation using CycleGAN, and physics-based method and domain translation using DANN, and CycleGAN-based method and DANN-based method.

Up to this point, we have discussed all the details of the proposed framework, which is summarized in Fig. 3.4. In Sec. 6.4, we will demonstrate the application of this framework with a case study.

3.4 Case Study

3.4.1 Problem description

In this section, the proposed damage diagnostics framework is applied to two similar but different miter gates to demonstrate its efficacy. As depicted in Fig. 3.10, the two miter gates located upstream and downstream of the same basin have been chosen to serve as the source and target miter gates, respectively.

To verify the efficacy of the proposed BiEDT framework and due to the lack of real-world data, synthetic observations are generated for the miter gates. In this paper, the observations of both the source and target miter gates are generated through a physics-based finite element analysis (FEA) model by adding noises. Noise is introduced to the observation data to reflect the variability and imperfections typically found in actual measurements.

Let $\mathbf{s}_i$ represent the i-th simulated strain measurements of the source miter gates from the model given in Fig. 3.9. The strain observations of the source miter gates, denoted as $\mathbf{y}_{o,\mathrm{sr},i}$, can be described by the following equation $\mathbf{y}_{o,\mathrm{sr},i} = \mathbf{s}_i + \boldsymbol{\epsilon}_i$, where $\boldsymbol{\epsilon}_i$ is the noise term added to each strain measurement and $\boldsymbol{\epsilon}_i$ follows a normal distribution with a mean of 0 and a standard deviation that is a function of the magnitude of the corresponding strain measurement. Specifically: $\boldsymbol{\epsilon}_i \sim \mathcal{N}(0, 0.01 \times |\mathbf{s}_i|)$. The strain response of the target miter

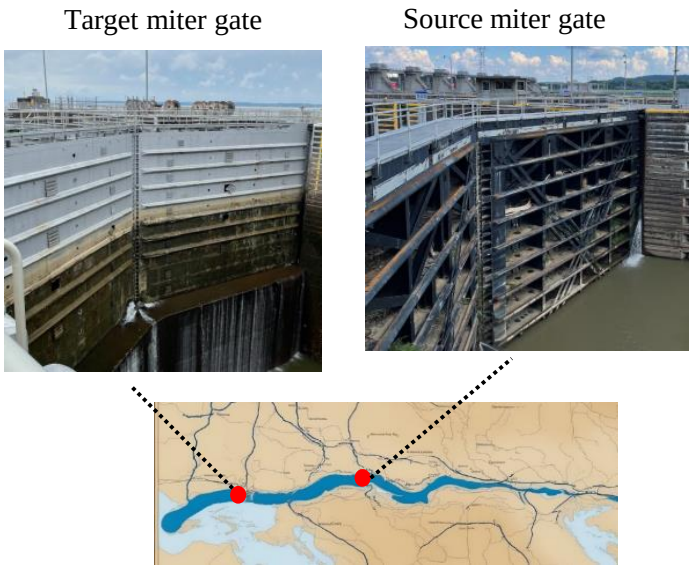


Figure 3.10: Illustration of source and target miter gates on a river.

gate is hypothetically derived by altering the response from the source miter gates. This modification ensures that while the two miter gates are relevant or share similarities, they remain different. The simulation model of the target miter gate is defined as follows

$$\mathbf{y}_{\text{tg},i} = \mathbf{s}_i(\cos(350\mathbf{s}_i) + \sin(650\mathbf{s}_i)). \tag{3.38}$$

The observation model of the target miter gate is given by

$$\mathbf{y}_{o,\text{tg},i} = \mathbf{s}_i(\cos(500\mathbf{s}_i) + \sin(800\mathbf{s}_i)) + \boldsymbol{\epsilon}_{\text{tg},i}, \tag{3.39}$$

where $\boldsymbol{\epsilon}_{\text{tg},i} \sim \mathcal{N}(0, 0.01 \times |\mathbf{s}_i|)$ is the observation noise of the target miter gate. As we can see from the above two equations that the simulation model is different from the observation model to mimic model form uncertainty that is inherent in computational simulation models.

The strain responses of the two miter gates (specifically, the source and target miter gates) are affected by their damage states (such as gap length) and water levels. The water levels at the two miter gates are different and are modeled using an Autoregressive Moving Average (ARMA) model. It is assumed that each miter gate has eight strain gauges (i.e., $N_s = 8$). Using the above-defined observation models, we generate 200 datasets of strain observations for the source miter gates (i.e., $N_{os} = 200$), and 180 datasets of strain observations for the target miter gate with unknown damage states (i.e., $N_{ot} = 180$). Each dataset consists of strain observation data of eight strain gauges over twenty time steps (i.e., $N_t = 20$). The damage states (i.e., gap length) corresponding to the 200 datasets of the source miter gates

are different and are available from historical inspections. Their counterparts of the target miter gate are unknown and are to be estimated.

As mentioned earlier, the observations of different miter gates are inherently related but different. This similarity and difference can be observed when comparing both the water level data and the strain observations. Fig. 3.11 shows the differences in upstream and downstream water levels between the source and target miter gates. It is important to note that the difference in water levels, particularly the downstream water levels, are not very significant. This is because we assume that the operational conditions of the two miter gates are similar and related, since they operate on the same river. Following that, Fig. 3.12 presents a comparison of the monitoring data from the source and target miter gates. The data clearly shows that even though the operational conditions are very similar, the strain responses are quite different due to structural differences between the two miter gates. Fig. 3.13 presents a comparison of the strain data from sensors placed on both the source and target miter gates, illustrating very different distributions for the same sensors across different structures. The top graph illustrates the strain readings between Sensor 1 and Sensor 2, while the bottom graph compares the data from Sensor 1 and Sensor 3. From the figure, it can be seen that the strain responses for corresponding sensors vary significantly between the two gates. The data points from the source miter gate, represented in red, form a cluster different from those of the target miter gate, shown in blue.

In addition to the differences highlighted in Figs. 3.11, 3.12, and 3.13, we calculate the Structural Similarity Index (SSIM) and Root Mean Square Error (RMSE) between the monitoring data of the source miter gate and the target miter gate. The average SSIM between the data of the source and targe miter gates is 0.4429, and the RMSE is 0.4231. These values are noticeably lower than those for the comparison between synthetic and real miter gate measurement data, as shown in Table 3.2. This quantification highlights the differences and similarities between the two types of miter gates in the case study. It is important to note that we have not yet established a standard threshold for similarity between miter gates, which is essential for applying the methods discussed in this paper for practical applications. Developing an effective similarity metric and determining an appropriate threshold to ensure the effectiveness of the proposed framework will be the focus of our future research.

The objective of this study is to accurately estimate the damage state of the target miter gate by leveraging strain observations and the inspected damage states of the source miter gates. It is worth mentioning that while the true damage state is used to generate these observations of the target miter gate, it is assumed to be unknown during the actual process of damage diagnostics.

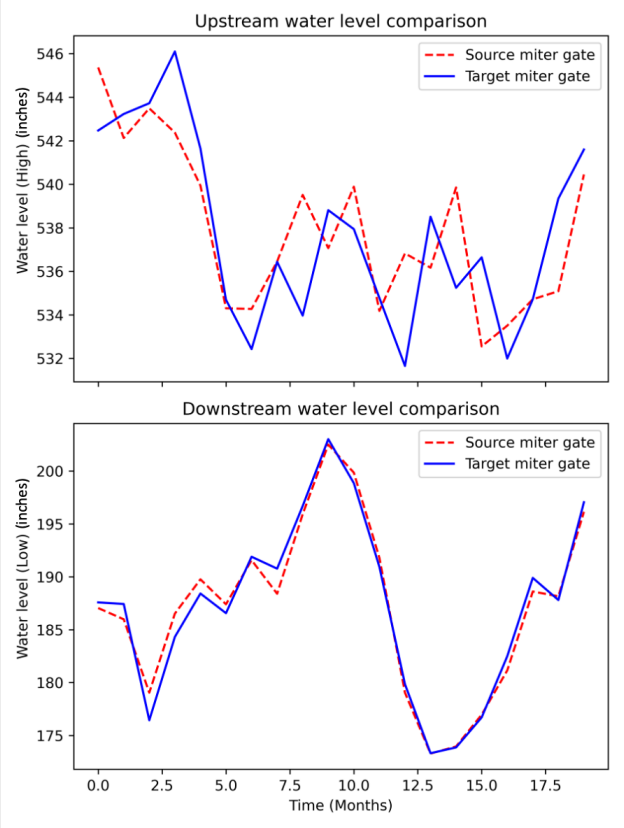


Figure 3.11: Water level comparison of source and target miter gates

3.4.2 Domain translation of the observations

3.4.2.1 CycleGAN-based domain translation

CycleGAN is first employed to perform domain adaptation/translation on the data from the target miter gate using the data from the source miter gates. During the training phase, the model establishes a relationship between data from two different miter gates by utilizing data from the source miter gate as input and data from the target miter gate as output. The training process for the CycleGAN model involves several key steps. Preprocessing is first performed to transform the raw data into a format suitable for image-based analysis. This preprocessing phase includes sorting the data, normalizing it, and ultimately converting it to image format. Specifically, in this study, the data is sorted based on the highest strain response values across the samples. This approach is adopted because the gap length significantly influences the strain response values, ensuring that the CycleGAN can accurately learn the relationships between the two different domains. After the data is sorted, it undergoes normalization to limit the range of data values, and then it is converted into an image format. Each data sample is represented as an image, with the two dimensions corresponding

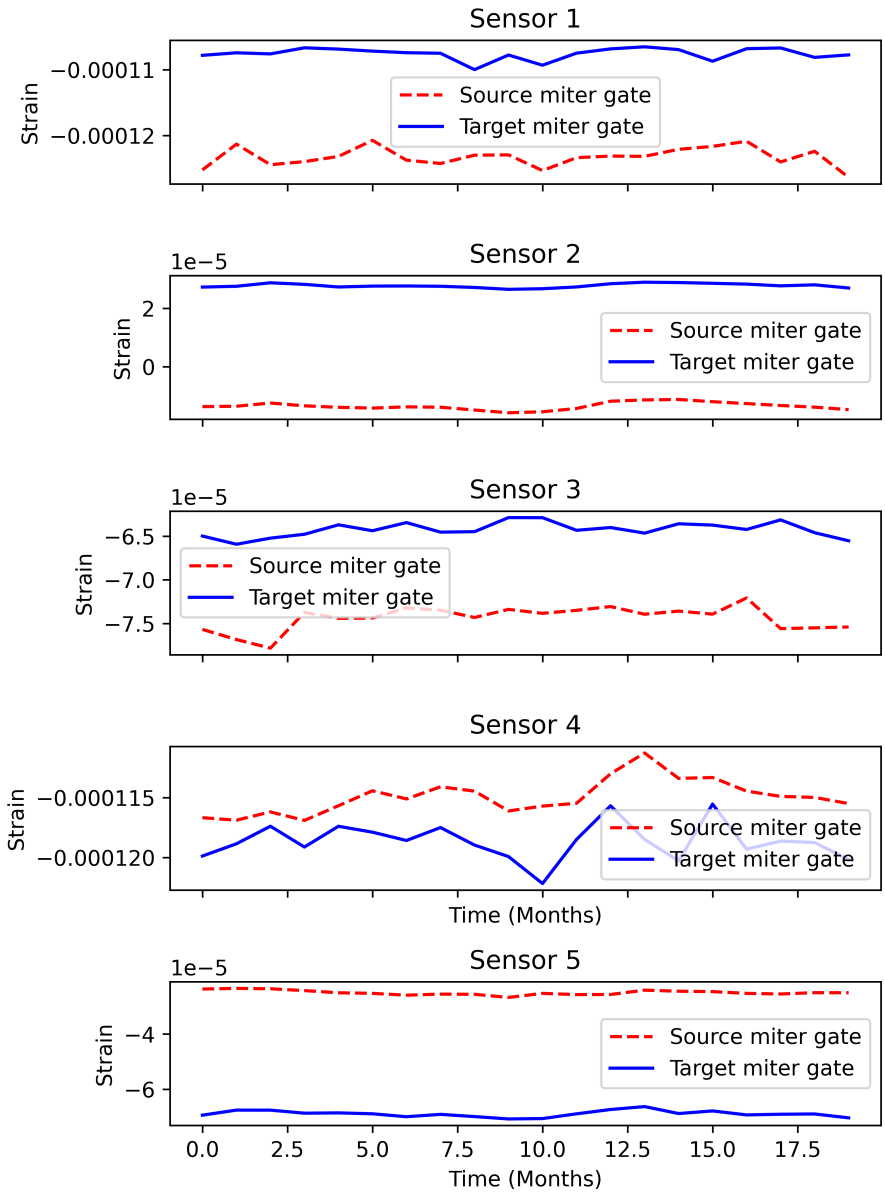


Figure 3.12: A comparison of monitoring data of source and target miter gates (Data of only five sensors are plotted for illustration purpose)

to time steps and sensor indices. Following preprocessing, this image-formatted data is used as the input for training the CycleGAN.

Table 3.1 presents the hyper-parameters for the training of CycleGAN. In the evaluation of CycleGAN's performance for domain translation, specifically in transforming images related to miter gate observations, we adopt three key metrics: SSIM, RMSE, and Fréchet Inception Distance (FID). These metrics are crucial for validating the effectiveness of CycleGAN in performing realistic and accurate image translations. Additionally, we focus on evaluating

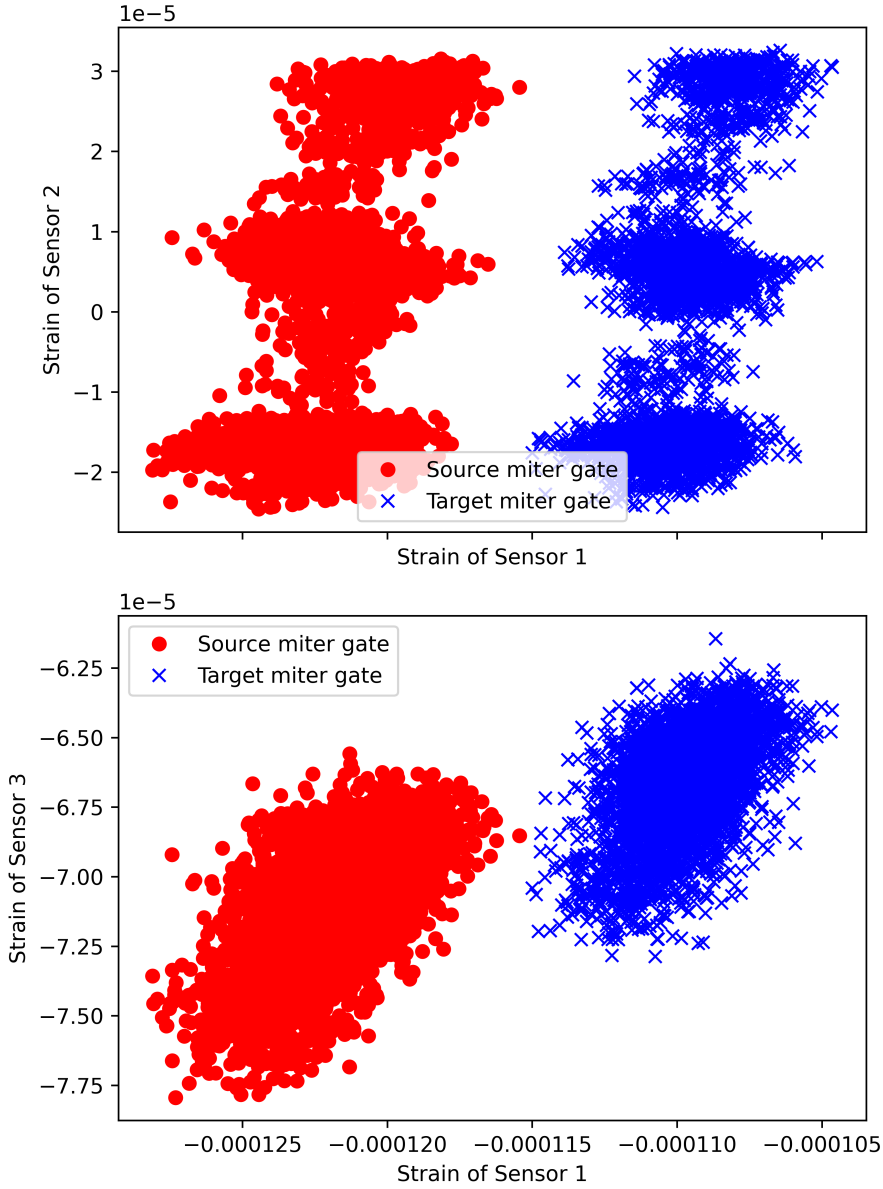


Figure 3.13: Scatter plot comparison of monitoring data of the source and target miter gates

the capability of CycleGAN under three different scenarios: (1) Synthetic Source vs. Real Source: This comparison assesses the ability of CycleGAN to generate synthetic image data of source miter gates from the target miter gate. (2) Original Source vs. Reconstructed Source. (3) Target vs. Reconstructed Target. Table 3.2 shows those comparison results for CycleGAN performance evaluation. Overall, these scores affirm that CycleGAN trains well and is effective in generating realistic and structurally consistent translated data.

Fig. 3.14 presents the results of domain adaptation using CycleGAN. The figure contains

Table 3.1: Model parameters of CycleGAN

Variable Name	**Conv U-net**
Number of epochs	3000
Batch size	5
Learning rate for generators	0.0001
Learning rate for discriminators	0.0001
Dropout	0.1
$\mathcal{L}_{\text{cyc}}$ (cycle consistency loss)	20
$\mathcal{L}_{\text{id}}$ (identity loss)	10
Optimizer	Adam

Table 3.2: Comparison of metrics for different objects after using CycleGAN

Object of Comparison	**SSIM**	**RMSE**	**FID**
Syn_{source} vs. $Real_{source}$	0.8000	0.0808	0.9247
$Real_{source}$ vs. $Reconst_{source}$	0.9649	0.0572	0.4020
$Real_{target}$ vs. $Reconst_{target}$	0.9294	0.0709	0.7108

two graphs; the upper graph plots the strain readings from Sensor 1 against Sensor 2, and the lower graph compares the strain readings from Sensor 1 against Sensor 3. In these figures, the data from the source miter gate is marked with red circles, while the data translated from the target miter gate is represented by blue crosses. Notably, the figure shows that the synthetic source data, generated from the target miter gate using CycleGAN, appears to cluster closely with the actual observations of the source miter gates. This suggests that the CycleGAN has effectively adapted the domain of the target gate data to align with the data characteristics of the source miter gates. The overlapping clusters indicate a successful translation, where the adapted synthetic observations mimic the real observations of the source miter gates. This highlights the potential of CycleGAN in bridging the gap between different domains by generating synthetic yet realistic data representations.

3.4.2.2 DANN-based domain adaptation

The second domain adaptation model we explore is DANN. DANN performs domain adaptation by extracting domain-independent features from observations between source miter gates and the target miter gate. The label predictor and feature extractor are trained together iteratively, as different features extracted require a different label predictor. Note that only labels from the source miter gates are used in training label predictor while both observations from the source and target miter gates are used to train the feature extractor with a multi-layer discriminator. The discriminator forces the feature extractor to extract

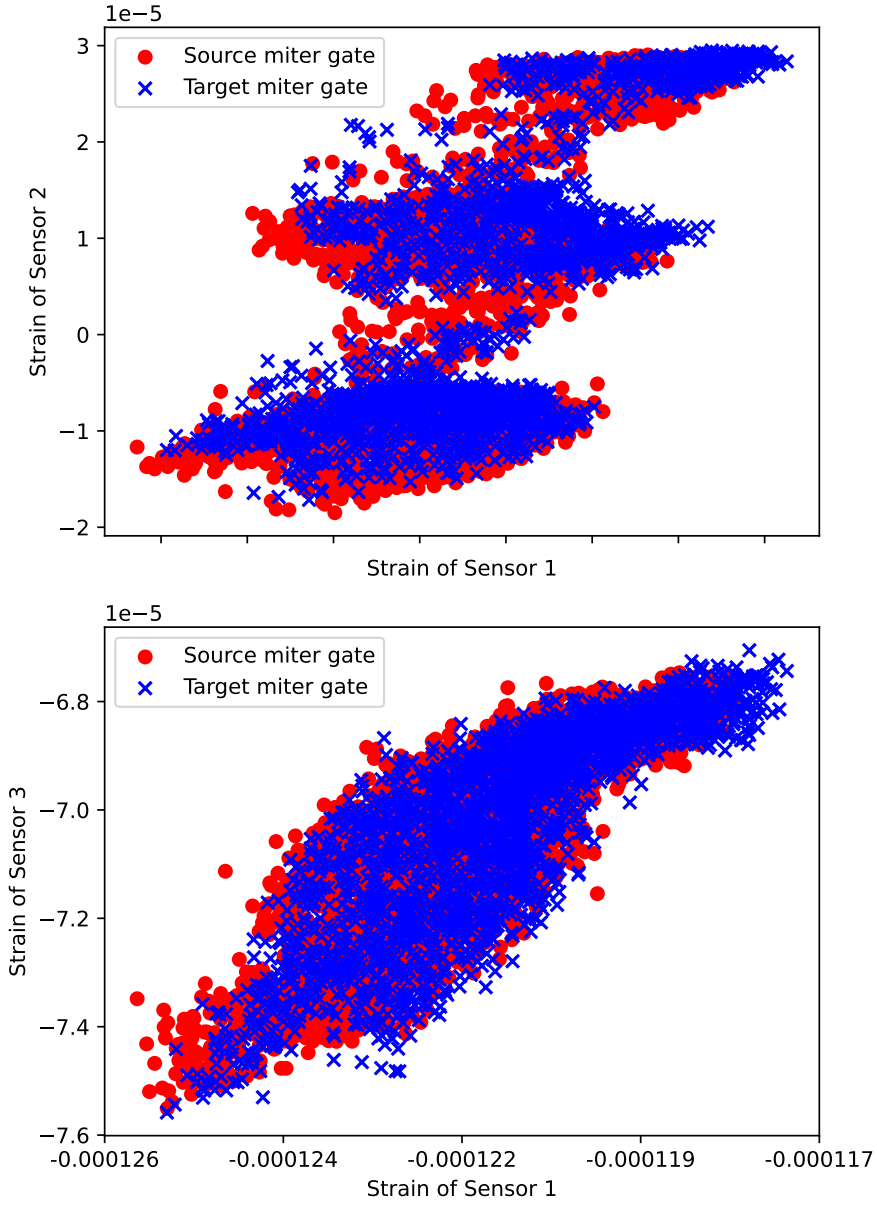


Figure 3.14: Scatter plot comparison of monitoring data of source and target miter gates after observation translation using CycleGAN

domain-independent features from both domains such that the discriminator can not distinguish them from the features. The observation data and labels (i.e., gap length of the source miter gates) are normalized with min-max normalization, respectively.

Table 3.3 shows the hyper-parameters for the training of the DANN model.

Fig. 3.15 shows the results of features extracted from the source and target miter gates. Among the 80 features, we depict the comparisons of feature pairs 2 vs. 4, 3 vs. 5, 2 vs.

Table 3.3: Model parameters of DANN

Variable Name	**DANN**
Number of epochs	1500
Batch size	16
Feature size	80
Learning rate for generators	0.001
Learning rate for discriminators	0.001
Optimizer	SGD

6, and 3 vs. 7 for illustration purposes. Notably, for feature pairs 3 vs. 5 and 3 vs. 7, the features from the source domain are very close to those from the target domain, which shows the efficacy of the DANN approach. Consequently, the relation between these feature pairs remains consistent across both domains. Conversely, other feature pairs demonstrate significant differences between the source and target domains, resulting in a change in their relation from source to target. However, for feature pairs 2 vs. 4 and 2 vs. 6, while the points are generally close between both domains, there are noticeable differences. For example, feature 2 from the source miter gate shows a bidirectional relationship with changes in features 4 and 6. In contrast, feature 2 from the target miter gate appears to follow only one direction of change with these features.

3.4.3 Bayesian damage diagnostics in translated domain

Following the domain translations, cINN is employed to perform Bayesian inference in the two translated domains, respectively. Herein, cINN consists of three conditional affine coupling layers. Each conditional affine coupling layer contains four separate fully connected layer neural networks with exponential linear units. The entire training process goes through 500 epochs, each containing 100 iterations.

For illustrative purposes, Fig. 3.16 lists the comparison between the true value, the prior distribution, and the posterior distribution estimated by cINN and cycleGAN-based domain translation for the four gap lengths. It can be found that except for a slight deviation in gap 2, the estimated posterior distribution of the remaining three gaps well captures the information of the true value. Meanwhile, Fig. 3.17 gives the errorbar plot of the posterior distribution estimated by cINN and cycleGAN-based domain translation for all gap lengths of the 180 obseravation datasets. Overall, the posterior mean is consistent with the true value, and the posterior variance does not show much fluctuation.

Following that, Fig. 3.18 presents the comparison between the true value, the prior distribution, and the posterior distribution estimated by cINN and DANN-based domain

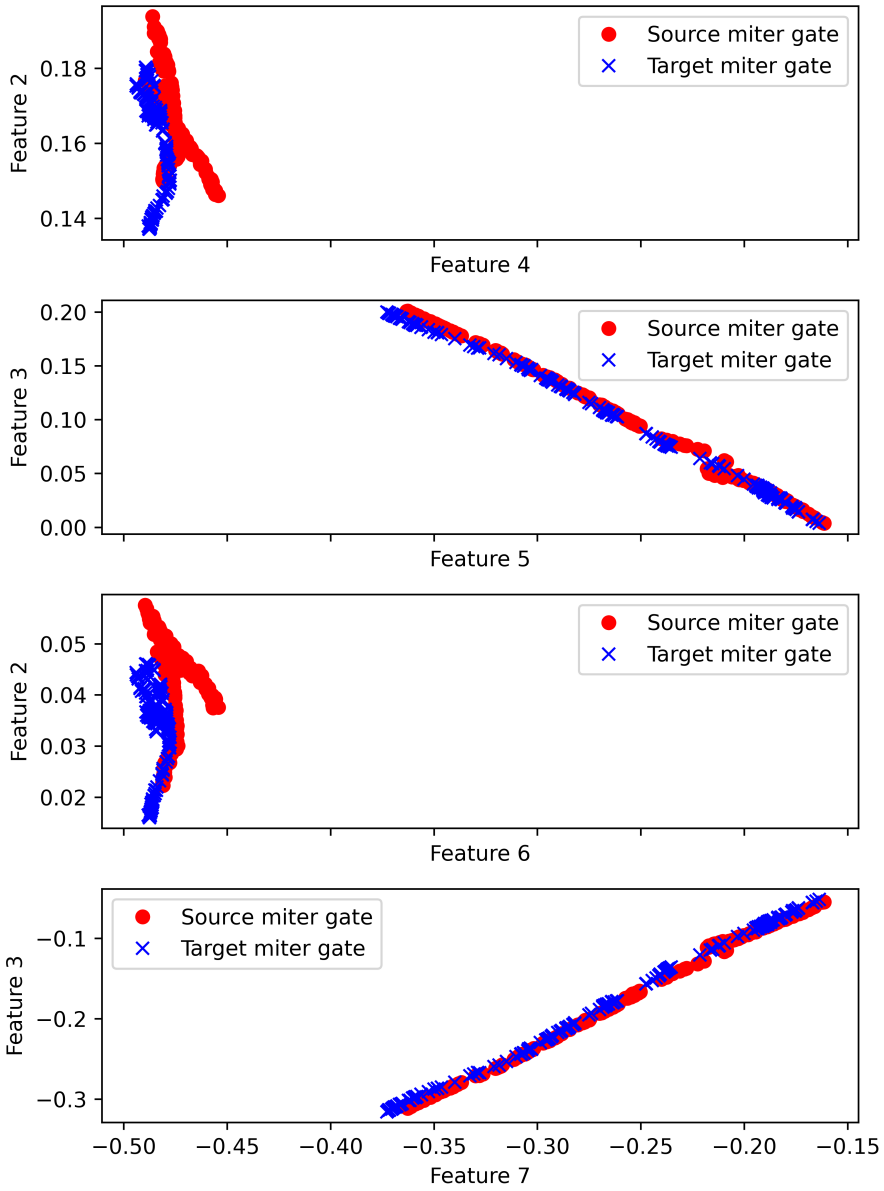


Figure 3.15: Comparison of DANN features of the source and target miter gates

translation for the four gap lengths. Similar to the cycleGAN-based translation, except for gap 2, the posterior distributions match the true value very well for the other three gap lengths. Fig. 3.19 shows the errorbar plot of the posterior distribution estimated by cINN and DANN-based domain translation for all gap lengths. Different from the results of the cycleGAN-based method, the posterior estimations from cINN and DANN-based domain translation exhibit large fluctuations between 40 and 60, deviating from the true value.

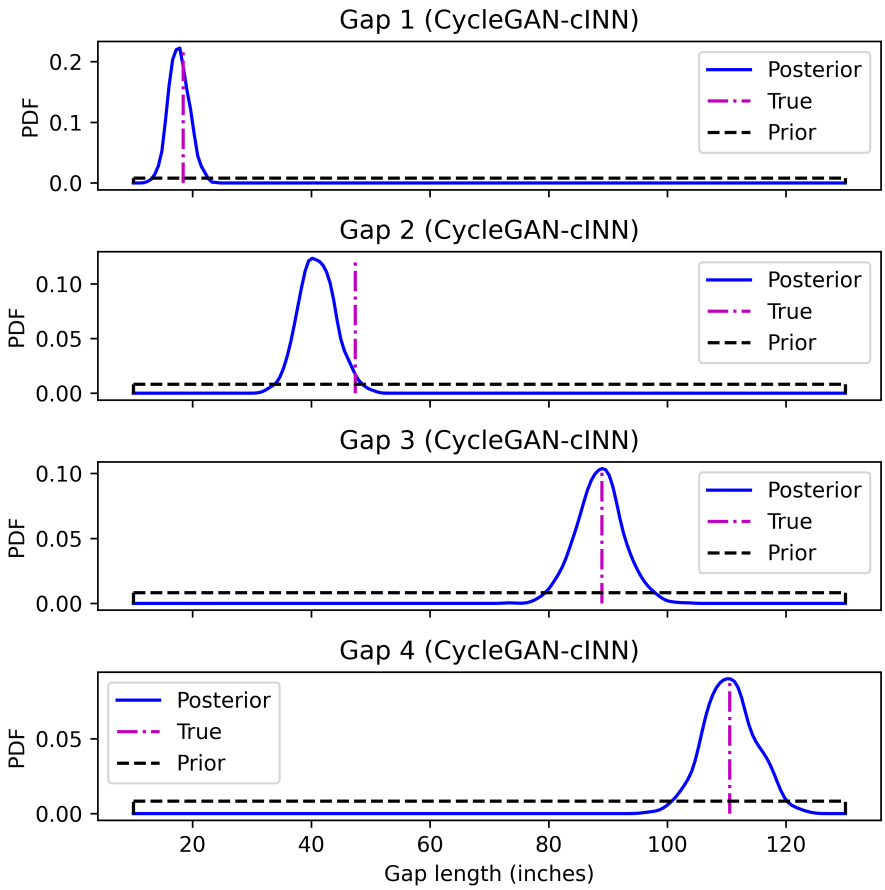


Figure 3.16: Posterior distribution of gap length obtained using cINN and CycleGAN-based domain translation

3.4.4 Bayesian model averaging for damage diagnostics of miter gates

3.4.4.1 Physics-based damage diagnostics using Bayesian method

As described in Sec. 3.3.5.1, we can also perform physics-based Bayesian inference for damage diagnostics, in addition to the above methods using domain translation and cINN. In the physics-based damage diagnostics using Bayesian inference, the prior distributions of the gap length and the standard deviation of the observation noises are non-informative uniform distributions, which are respectively $[10, 130]$ inches and $[3 \times 10^{-4}, 3 \times 10^{-4}]$. Fig. 3.20 presents the posterior distribution of gap lengths obtained from physics-based Bayesian inference for four illustrative datasets. From the results, we can see that the posterior distributions for gap lengths over 60 (as seen in Gap 3,4) show a high degree of accuracy, with the peak of the posterior probability density function closely aligning with the true

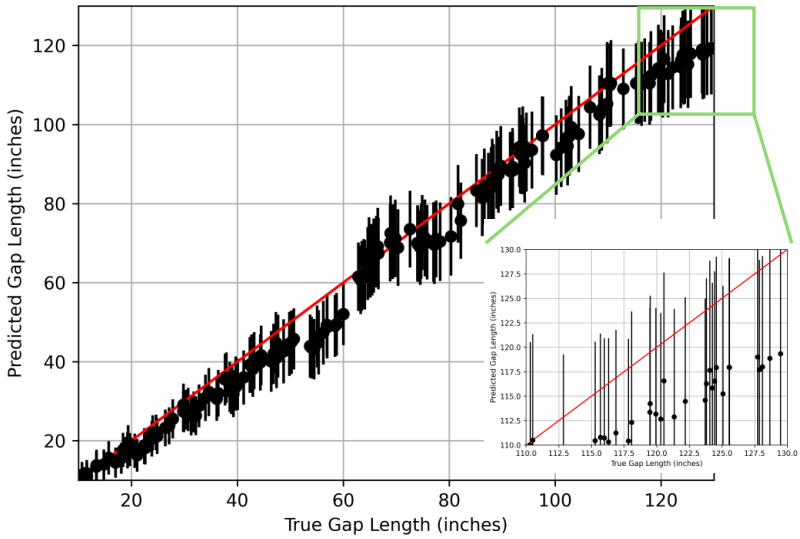


Figure 3.17: Errorbar plot of the posterior distribution of different gap length estimates (CycleGAN-based domain translation)

value. The sharp, narrow peaks of the posterior distribution indicate high confidence in the estimates for large gap lengths. However, for smaller gaps (especially for Gap 1 and Gap 2), the posterior distributions do not align as closely with the true values. This misalignment indicates potential model error or limitations in the model's ability to estimate small gap lengths. The significant error in estimating smaller gap lengths could be attributed to the uncertain variance σ_ε^2 of the observation noises.

Fig. 3.21 presents an error-bar plot of the posterior distribution for various gap length estimates using physics-based Bayesian inference, showing a comparison between the predicted and true gap lengths. It shows that the model has a good predictive performance for larger gap lengths, as indicated by the clustering of points along the red line in the higher range. Fig. 3.21 corroborates the findings from the previous analysis (i.e., Fig. 3.20), where the model shows better accuracy for larger gap lengths and faces difficulties with smaller gaps.

3.4.4.2 Bayesian model averaging of posterior distributions

After implementing Bayesian damage diagnostics in the translated domain and using a physics-based model, we integrate these posterior distributions using a Bayesian model averaging framework. This application of BMA using posterior distributions is obtained from three different approaches: CycleGAN-cINN, DANN-cINN, and physics-based approaches.

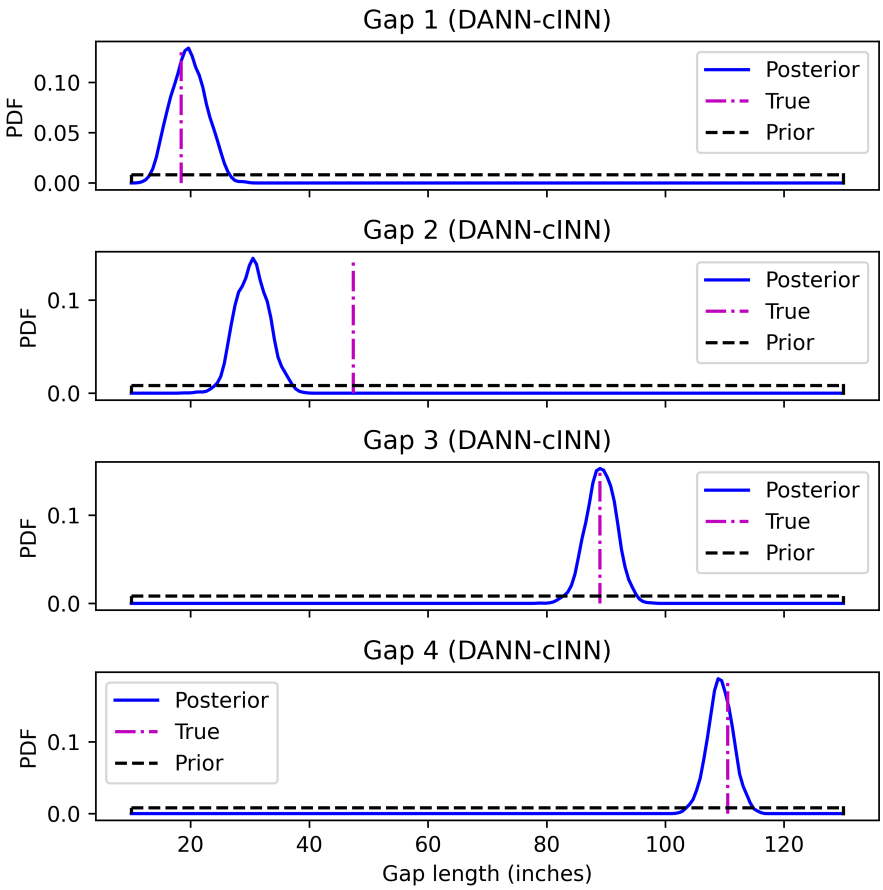


Figure 3.18: Posterior distribution of gap length obtained using cINN and DANN-based domain translation

Fig. 3.22 shows the combined posterior distributions for a gap length around 70. The posterior curves for each method (DANN-cINN, CycleGAN-cINN, Physics-based) indicate their individual estimations before averaging. The posterior distribution of the physics-based method exhibits a multi-modal pattern. This phenomenon may be attributed to the large uncertainty in the variance of the observation noises. As mentioned above, the prior distribution of σ_ε is $[3\times10^{-4}, 3\times10^{-4}]$. It leads to a certain identifability issue for the physics-based method. The BMA curve, which is the combined result of these three approaches, seeks to average the estimates of each method. The bar chart on the right illustrates the weights assigned to each method in the BMA process (i.e., Eqs. (3.30)-(3.32)). It shows that in this case, the physics-based method has the highest weight, suggesting its predictions are considered the most reliable for this particular gap length scenario based on the likelihood estimates.

Fig. 3.23 represents a different gap length scenario around 125. Similar to Fig. 3.22, it

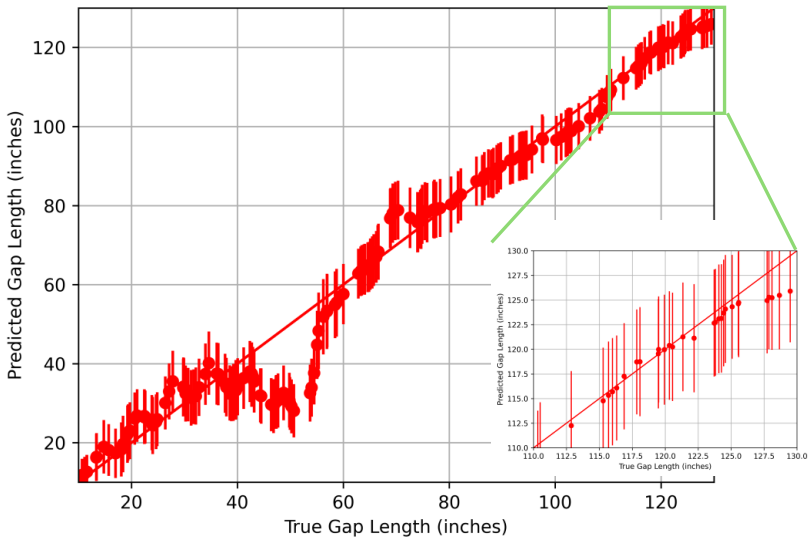


Figure 3.19: Errorbar plot of the posterior distribution of different gap length estimates (DANN-based domain translation)

compares the posterior estimates from the three methods alongside their BMA result. The weights are notably different; the DANN-cINN method is given the most weight, indicating its superior performance or reliability for this larger gap scenario.

Fig. 3.24 shows the predicted versus true gap lengths for each method after using the proposed model selection method (see Sec. 3.3.5.2). It shows the predictive accuracy of each method across the entire range of gap lengths. By integrating the results of different methods through model selection using the Bayes factor, we aim to mitigate the individual limitations of each method, leveraging their combined strengths to enhance predictive accuracy. As indicated in this figure, the CycleGAN-based method is selected when the gap length is small, and the DANN-based is selected when the gap length is large. For the other regions, the physics-based Bayesian inference method is selected. Results in this figure demonstrate the efficacy of the proposed approach in combining different Bayesian diagnostic models to provide a robust estimation of gap lengths, which is crucial for accurate damage assessment in structural health monitoring.

3.5 Conclusions and Discussion

This paper presents a novel BiEDT framework, for damage diagnostics that integrates domain translation with Bayesian inference. By transforming observations into a common

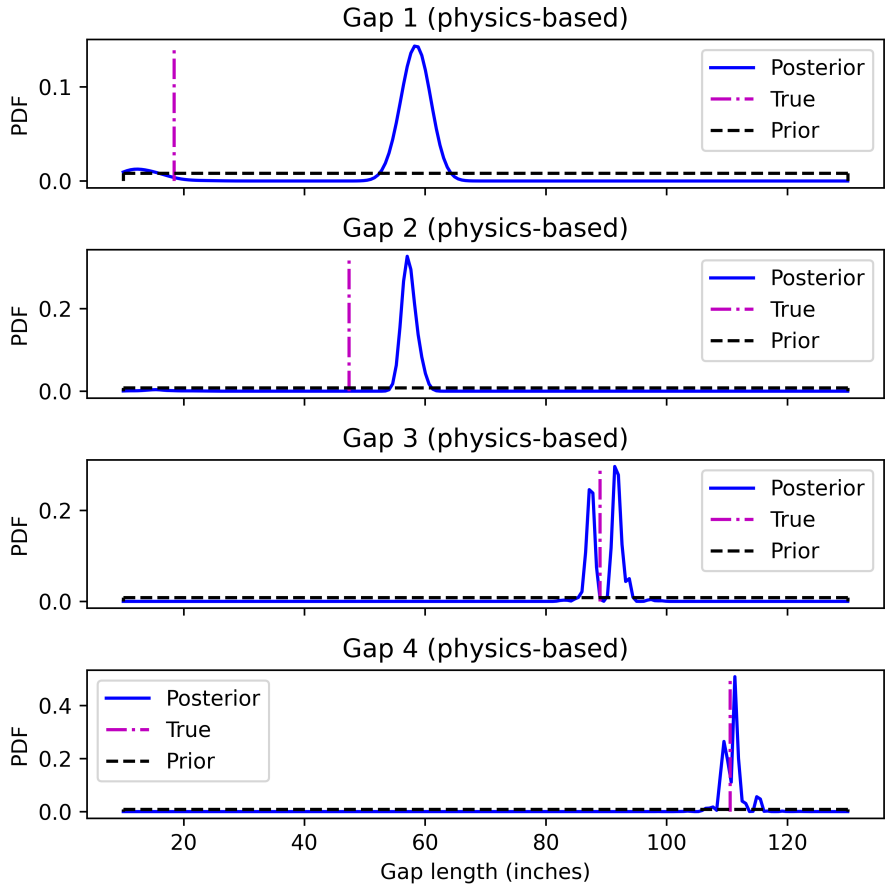


Figure 3.20: Posterior distribution of gap length obtained using physics-based Bayesian inference

domain and employing Bayesian model averaging, BiEDT leverages both physics-based models and advanced machine learning techniques to enhance the accuracy and reliability of damage state estimations for miter gates. The method not only combines the strengths of different damage diagnostic approaches but also quantifies uncertainty in the damage estimates, making it a robust tool in structural health monitoring. The method presented in this paper tackles the challenge of damage diagnostics of miter gates by synthesizing multiple data sources: real-time structural monitoring of gates of interest with unknown damage status, historical data from similar but different miter gates, and predictive insights from physics-based computational simulation models.

Despite its numerous advantages, the BiEDT framework does have certain limitations that must be acknowledged. One key limitation is that the accuracy of the translations and Bayesian inference is heavily dependent on the assumptions and quality of the underlying machine learning models. Any mis-specifications or inaccuracies in these models could

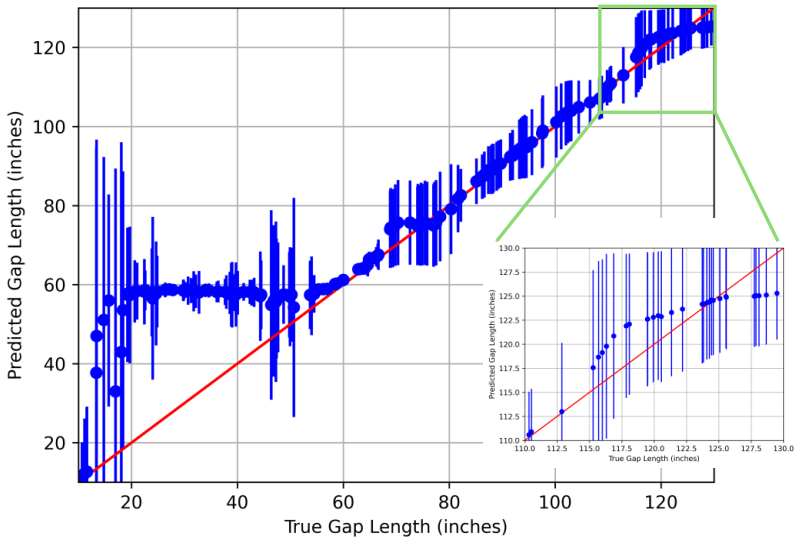


Figure 3.21: Errorbar plot of the posterior distribution of different gap length estimates (Physics-based Bayesian inference)

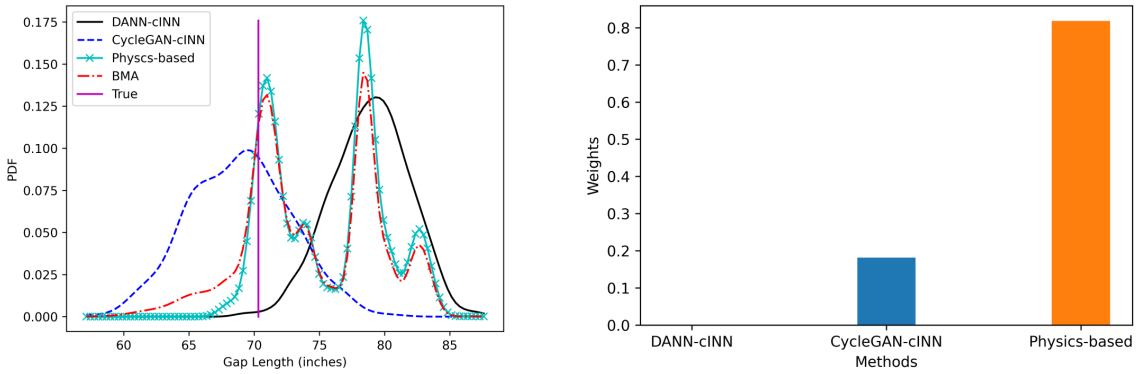


Figure 3.22: Bayesian model averaging of posterior distributions (Case 1)

lead to biased or erroneous damage state estimations. Additionally, the domain translation models used in this framework may not always adequately handle the full range of diverse operational conditions encountered in real-world applications. Moreover, the current framework primarily focuses on miter gates, and its application to other types of structures or systems has not been extensively validated. Future research could extend the BiEDT framework to a broader range of SHM scenarios, ensuring its adaptability and effectiveness across different domains.

In addition, an important issue to consider when applying the proposed framework is the

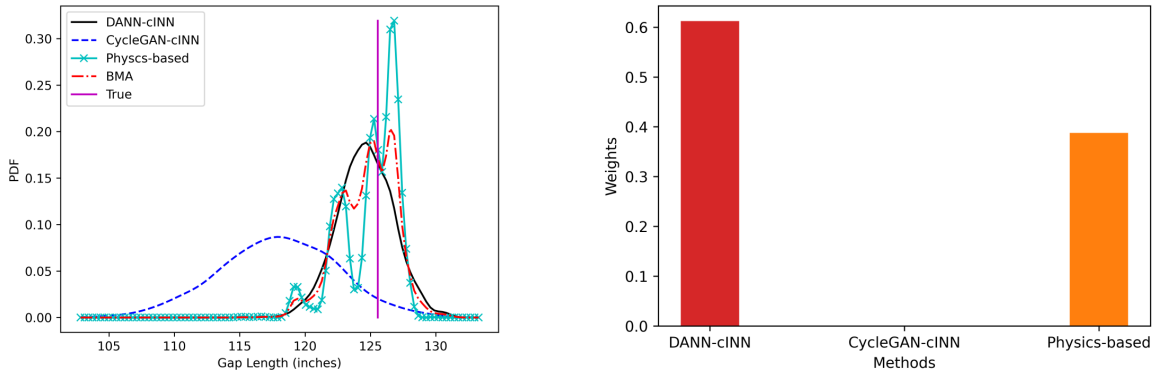


Figure 3.23: Bayesian model averaging of posterior distributions (Case 2)

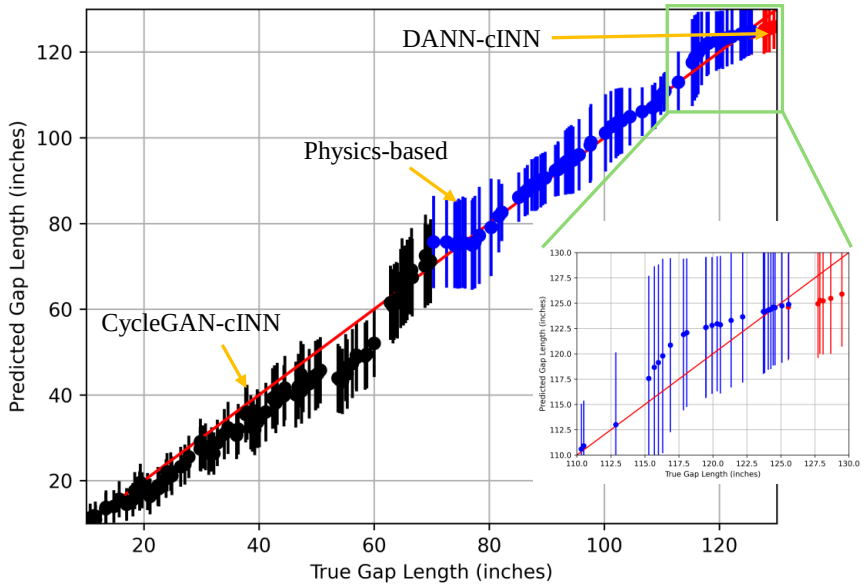


Figure 3.24: Errorbar plot of the posterior distribution of different gap lengths after model selection using Bayes factor

assessment of the similarity between two miter gates. As been pointed out in Refs. [67, 93], *negative transfer* occurs when utilizing data or knowledge from the source domain undesirably diminishes the learning performance in the target domain. It poses a long-standing challenge in transfer learning. Developing appropriate similarity metrics and determining a threshold to decide when the proposed framework can be applied and to avoid negative transfer is a research direction that worth studying in our future work. Another important area of future research involves the quantification of uncertainty in these machine learning models. Understanding and effectively managing this uncertainty is crucial for improving

the reliability and confidence in the damage state estimations provided by the BiEDT framework. This could involve incorporating advanced methods of uncertainty quantification and exploring ways to integrate this uncertainty into the overall decision-making process [94].

CHAPTER 4

Hybrid Modeling for Streamflow Prediction in Ungauged Rivers with Uncertainty Quantification: A Comparative Analysis

4.1 Introduction

Streamflow prediction is vital for managing flood risks, facilitating river navigation, and ensuring sustainable water management [95, 96]. In long-term water management, reliable streamflow forecasts made months in advance can greatly enhance efficiency [97].

A key aspect of decision-making in this field is predicting streamflow discharge within river networks. Recent studies emphasize the need for improved discharge predictions in ungauged locations, where the lack of sensors creates forecasting challenges. River routing models, widely used in management systems, simulate hydrograph changes as water moves through networks, offering critical insights into flow dynamics. For example, the US Army Corps of Engineers (USACE) employs the Routing Application for Parallel computatIon of Discharge (RAPID) model [98] for streamflow forecasting and inland waterway management [99]. Based on McCarthy's Muskingum Routing method introduced in 1938 [100], RAPID has been extensively studied over the past decades [98]. However, the labor-intensive calibration process of the Muskingum method limits its efficiency and scalability [101, 102], underscoring the need to transition to more adaptable and efficient alternatives, such as machine learning (ML) models.

Interest in ML has surged in the hydrologic community, driven by expanding hydrologic data repositories and successful applications across academic and commercial domains. Xu *et al.* [103] provide an overview for newcomers on the application and growing importance of ML in hydrology. Shamshirband *et al.* [104] model hydrological droughts by analyzing their duration, severity, and magnitude using Support Vector Regression (SVR), Gene Expression Programming (GEP), and M5 model trees (MT), incorporating key factors such

as precipitation, evapotranspiration, and runoff. Zhao *et al.* [101] relate hydrological model parameters to river and topographic characteristics using models such as Gaussian process regression, Gaussian mixture copula, Random Forest, and XGBoost. Difi *et al.* [105] combine ML with signal decomposition for daily streamflow forecasting, while Magni *et al.* [106] develop a hybrid framework that improves streamflow predictions by integrating outputs from the PCR-GLOBWB global hydrological model.

Long Short-Term Memory (LSTM), a type of recurrent neural network, is well-suited for streamflow prediction due to its ability to handle time-series data effectively [107, 108, 109, 110, 102]. Rahimzad *et al.* [111] evaluate various data-driven techniques for daily streamflow forecasting in the Kentucky River basin, finding LSTM to be the most accurate. Kratzert *et al.* [112] propose a data-driven LSTM approach for rainfall-runoff modeling. Feng *et al.* [113] enhance streamflow forecasts by integrating recent observations with varying schedules and types (e.g., moving average, snapshot, regularly spaced) using an LSTM model. More recently, Nearing *et al.* [114] show that AI-based forecasting with LSTM can predict extreme riverine events in ungauged watersheds up to five days in advance, matching or surpassing zero-day lead-time predictions from the Copernicus Emergency Management Service Global Flood Awareness System. Data quality also plays a key role—Schutte *et al.* [115] find that GRU and LSTM models perform reliably with commercial weather data but show moderate accuracy with freely available datasets.

While deterministic ML models have been widely adopted in hydrology for their ability to capture complex nonlinear relationships, they inherently fail to represent the uncertainties present in hydrological systems. This limitation often leads to overconfident predictions with narrow uncertainty intervals, potentially resulting in misleading conclusions and inadequate preparation for extreme events. Recognizing this, the focus has shifted toward probabilistic streamflow forecasting, which provides predictions as probability distributions or prediction intervals rather than single deterministic values. Such approaches enable stakeholders to quantify risks and make better-informed decisions under uncertainty. Consequently, uncertainty quantification (UQ) has become a critical component in advancing hydrological modeling. Several methods have been explored to address UQ in hydrology. Traditional approaches include Bayesian frameworks that integrate prior knowledge and systematically update uncertainty based on observations [116]. Information-theoretic and informal Bayesian methods, such as the generalized likelihood uncertainty estimation (GLUE) and limits-of-acceptability approaches, have also been reviewed extensively [117]. Techniques like first-order Taylor series expansion have been used to quantify predictive uncertainty in neural network outputs [118], while Zhao *et al.* [119] introduced a Bayesian joint probability model to address uncertainty in deterministic forecasts. Furthermore, Clark *et al.* [120] emphasized

the importance of explicitly accounting for hydrologic uncertainties given their implications for climate adaptation and societal resilience. In parallel, recent advances in deep learning have provided new tools for probabilistic hydrologic forecasting. Kratzert *et al.* [121] demonstrated the effectiveness of LSTM networks for rainfall–runoff modeling, inspiring further exploration into integrating deep learning with UQ methods. Liu *et al.* [122] propose PI3NN to enhance the trustworthiness of ML models, particularly LSTM networks, for streamflow prediction. Lu *et al.* [95] employ Bayesian LSTM for UQ in hydrologic predictions for data-scarce rural watersheds. Techniques such as Monte Carlo dropout, deep ensembles, and Bayesian neural networks have been developed to estimate both aleatoric (data-related) and epistemic (model-related) uncertainties, thereby enhancing prediction reliability [123]. Moreover, hybrid modeling approaches that integrate physics-based hydrologic models with ML frameworks have shown promise for improving predictive performance and generalization, particularly in data-scarce or ungauged basins [124]. Li *et al.* [125] used stochastic variational inference in an LSTM residual error model for process-based hydrology, improving CRPS by over 10% and reliability while scaling to high-dimensional applications. Klotz *et al.* [126] proposed an uncertainty benchmarking procedure for hydrological deep learning, showing that mixture density networks and Monte Carlo dropout provide strong baselines for accurate and nuanced predictions. What's more, Sattari *et al.* [127] developed a probabilistic LSTM-based framework with Monte Carlo sampling and wavelet decomposition to capture both aleatory and epistemic uncertainties in streamflow forecasting, outperforming a basic LSTM by up to 45% during extreme events such as Hurricane Harvey.

Although existing studies enhance model confidence through UQ methods, accurately predicting streamflow in ungauged rivers remains a challenge. This paper addresses probabilistic river discharge prediction in ungauged locations. The absence of discharge gauge data complicates forecasting, but by leveraging insights from calibrated physics based models such as RAPID, selecting key features, and capturing causal relationships including runoff data and neighboring gauged river discharge, it is possible to develop a hybrid model using advanced machine learning techniques. This model not only generates accurate discharge predictions for ungauged rivers but also provides uncertainty bounds. By integrating various UQ methods into ML models, it produces probabilistic forecasts essential for risk-informed decision-making. In this study, both aleatoric (data-related) and epistemic (model-related) uncertainties are explicitly quantified using probabilistic deep learning approaches, enabling a comprehensive characterization of prediction uncertainty in streamflow forecasts.

The primary contribution of this paper is a module-based hybrid modeling framework that leverages state-of-the-art ML algorithms to predict the probabilistic discharge of ungauged rivers. This is achieved through the following key components:

1. *Integrating insights from calibrated physics-based models like RAPID into hybrid model training.* This module incorporates predictions from RAPID, a first-order physics-based model grounded in conservation laws. While RAPID provides a first-order estimate, it may contain errors but serves as a causality check and effectively captures the fundamental physics of river dynamics.

2. *Incorporating carefully selected sensitive features into hybrid model training.* These features, which significantly influence river discharge, help bridge gaps in the understanding of river behavior that are not captured by first-order models like RAPID.

3. *Providing probabilistic forecasts of river discharge.* These forecasts are particularly valuable for risk-informed decision-making. UQ methods are integrated into the hybrid model to achieve this.

4. *Training the hybrid model using discharge data from gauged rivers in the target basin or neighboring rivers.* "Neighboring rivers" refers to rivers that are geographically near the target river location and share a hydrological similarity, such as being part of the same upstream catchment or drainage basin. This enables the model to capture causal relationships between input variables and river discharge.

The framework assumes that causal relationships between input variables (e.g., runoff data, selected features), RAPID predictions, and measured discharge in gauged rivers also apply to ungauged rivers due to the neighborhood effect (i.e., similar geography, hydrology, and topology). Even in regions with few gauged rivers, discharge characteristics may vary. Training the hybrid model on multiple gauged rivers enables it to capture diverse river dynamics, including variations in peak discharge levels.

In the next section (i.e., Section 4.2), we provide an overview of the proposed framework, offering a bird's-eye view of its structure. Subsequently, the paper's organization is discussed, followed by the details.

4.2 Overview of the proposed framework

This section provides an overview of the proposed framework, illustrated in Figure 4.1. It comprises four main modules: (1) *Calibrated Physics-Based RAPID Model*, (2) *Feature Selection*, (3) *Machine Learning Data-Driven Model*, and (4) *Uncertainty Quantification* for ML models. These modules integrate systematically to form a hybrid, comprehensive streamflow prediction system.

The process begins with calibrating the physics-based RAPID model, where river network data and runoff information are assimilated to update its parameters. Next, features from multiple sources are analyzed for their sensitivity to neighboring gauged river discharge. Only the most sensitive features are selected to reduce computational costs and enhance accuracy.

The selected features, runoff data in a reduced-dimensional space, and RAPID-based discharge predictions serve as inputs to the ML model, creating a hybrid system that captures causal relationships between inputs (e.g., runoff and selected features) and river discharge while incorporating first-order river flow physics from RAPID. Training output consists of discharge data from gauged rivers near the target ungauged river. The framework obtains probabilistic discharge predictions by integrating a UQ module into the training process.

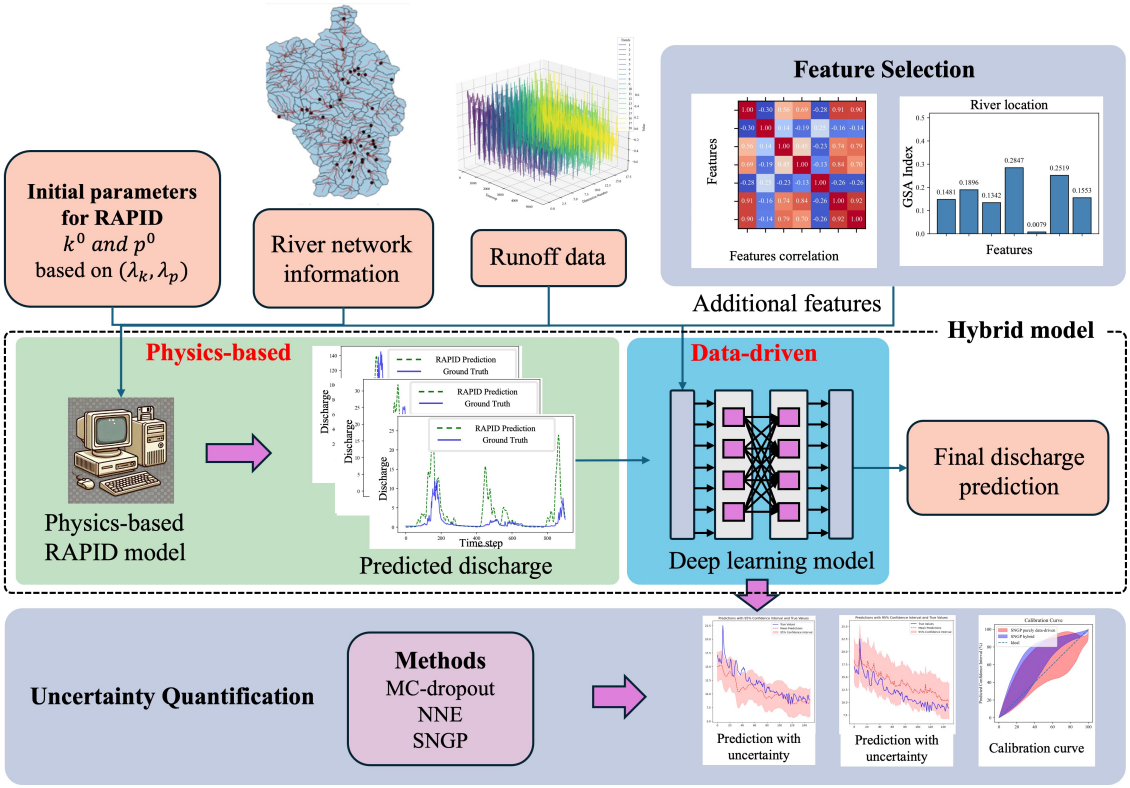


Figure 4.1: Overview of the proposed framework.

The remainder of the paper is organized as follows: Section 4.3 discusses the hybrid modeling approach for streamflow prediction in ungauged rivers. Section 5.3.6 covers uncertainty quantification methods for the proposed ML models. Section 6.4 presents a case study on ungauged streamflow prediction in Colorado. Section 6.5 outlines the model results, and Section 4.7 concludes with key findings and insights.

4.3 Hybrid modeling for streamflow prediction of ungauged rivers

Hybrid modeling integrates multiple modeling approaches to leverage their strengths and mitigate weaknesses. While numerous studies apply hybrid techniques to streamflow prediction, most do not combine physics-based and machine learning (ML) models. Lin *et al.* [128] propose DIFF-FFNN-LSTM, a hybrid model integrating first-order difference (DIFF) equations, a feedforward neural network (FFNN), and LSTM for hourly streamflow prediction. Kilinc *et al.* [129] combine LSTM with a genetic algorithm (GA) for streamflow forecasting. Zhong *et al.* [130] develop a model integrating physics-based approaches, specifically the variable infiltration capacity (VIC) and catchment-based macroscale floodplain (CaMa-Flood) models, with LSTM. Physics-based models can complement ML-based time series prediction by enhancing performance and incorporating physical constraints to prevent spurious predictions. These hybrid models leverage advanced ML algorithms to capture nonlinear trends and dependencies in river discharge.

This paper proposes a hybrid model integrating RAPID with either LSTM-based or SNGP ResNet-based models. This section details the hybrid modeling framework, its components, and their collaboration to improve prediction performance.

4.3.1 Physics-based RAPID

RAPID is a river routing model designed to efficiently compute discharge across large, complex river networks. It is based on the Muskingum method [98], later enhanced by Cunge [131] into the Muskingum–Cunge method, where parameters are derived from the river channel's mean physical characteristics and flow wave.

In networks with hundreds or thousands of reaches, matrices are essential for defining connectivity and calculating flow. RAPID's core is a vector-matrix implementation of the Muskingum method, as shown in Eq. (4.1) and detailed in this section [98].

$$(\mathbf{I} - \mathbf{C}_a \cdot \mathbf{N}) \cdot \mathbf{Q}(t + \Delta t) = \mathbf{C}_a \cdot \mathbf{Q}^{\mathrm{l}}(t) + \mathbf{C}_b \cdot \left[\mathbf{N} \cdot \mathbf{Q}(t) + \mathbf{Q}^{\mathrm{l}}(t)\right] + \mathbf{C}_c \cdot \mathbf{Q}(t), \tag{4.1}$$

where $\mathbf{C}_a$, $\mathbf{C}_b$, and $\mathbf{C}_c$ are diagonal matrices with diagonal elements representing Muskingum method coefficients; $\mathbf{I}$ is the identity matrix, and $\mathbf{N}$ the river network matrix; $\mathbf{Q}$ is the vector of reach outflows, while $\mathbf{Q}^{\mathrm{l}}$ represents lateral inflows; t denotes time, and Δt the river routing time step.

The elements of $\mathbf{C}_a$, $\mathbf{C}_b$, and $\mathbf{C}_c$ depend on parameters k_m and p_m, where k_m is a storage constant with the dimension of time and p_m a dimensionless weighting factor representing

the relative influence of inflow and outflow on volume of reach m. For any reach m, the relationship $C_{am} + C_{bm} + C_{cm} = 1$ holds. Muskingum method parameters k_m and p_m vary across river reaches.

For robust predictions, the RAPID model must be calibrated using observed data. This involves estimating the parameters k_m and p_m for all reaches ($\forall\ m$) using assumed initial values and river network information. An inverse approach is employed to optimize these parameters, minimizing a cost function that quantifies the difference between model outputs and observations.

After RAPID is calibrated, rainfall-runoff data and river properties (such as length and slope) are used to generate discharge predictions. Let $\mathbf{F}_{\text{RAPID},k} \in \mathbb{R}^{N_T \times 1}$ be the RAPID predictions vector for river k, containing N_T time steps. For more details on RAPID calibration and predictions, readers are referred to [98] or Section 2 of Zhao *et al.* [101].

In the next section, we present the methods used for feature selection in the ML models.

4.3.2 Feature selection for ML-based river discharge prediction

Feature selection is vital for data preprocessing and dimensionality reduction, playing a key role in ML and time-series prediction. It aims to create simpler and more interpretable models, enhance data-mining performance, and prepare clean and understandable data [132]. For the hybrid model, we gathered multiple data sources, which require preprocessing before feature selection.

- **Runoff data**: The runoff dataset comprises time series data for all rivers in the basin, each associated with a watershed. In this study, runoff data refer to surface runoff estimates simulated using the Noah with multiparameterization (Noah-MP) land surface model within the National Water Model framework [133]. Let N_{WS} be the number of watersheds and N_T the number of time steps (observations). Define $\mathcal{S}_{\text{runoff}}$ as the set of features derived from runoff, with its i-th element representing the runoff time series for the i-th watershed.

  Let $\mathbf{F}_{\text{runoff}} \in \mathbb{R}^{N_T \times N_{\text{WS}}}$ denote the runoff data matrix, where the i-th column represents the runoff time series for the i-th watershed. The basin of interest includes multiple watersheds and their associated rivers. Since these rivers share the same basin and interrelated dynamics, their runoff data are expected to be correlated. Therefore, rather than using the entire runoff dataset, reducing its dimensionality is advantageous. It simplifies the model, improves computational efficiency, mitigates overfitting, and enhances interpretability by identifying relevant features. To achieve this, Singular Value Decomposition (SVD) [134] is applied, expressing $\mathbf{F}_{\text{runoff}}$ as:

$$\mathbf{F}_{\text{runoff}} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T, \tag{4.2}$$

where $\mathbf{U} \in \mathbb{R}^{N_T \times N_T}$ is the left singular matrix, $\mathbf{\Sigma} \in \mathbb{R}^{N_T \times N_{\text{WS}}}$ is the diagonal matrix of singular values with σ_i as its i-th diagonal element, and $\mathbf{V}^T \in \mathbb{R}^{N_{\text{WS}} \times N_{\text{WS}}}$ is the right singular matrix, whose columns represent the right singular vectors of $\mathbf{F}_{\text{runoff}}$.

The reduced dimensionality N_{WS_d} is chosen such that $N_{\text{WS}_d} < N_{\text{WS}}$, using the cumulative energy criterion to retain most of the data variance. It is computed as:

$$\mathcal{E}(N_{\text{WS}_d}) = \frac{\sum_{i=1}^{N_{\text{WS}_d}} \sigma_i}{\sum_{i=1}^{\min(N_T, N_W)} \sigma_i^2}. \tag{4.3}$$

Here, $\mathcal{E}(N_{\text{WS}_d})$ denotes the variance captured by N_{WS_d} features. To retain the desired variance, N_{WS_d} is the smallest integer satisfying $\mathcal{E}(N_{\text{WS}_d}) > \tau$, where τ (typically 0.90–0.95) is the variance retention threshold. Once determined, we select the first N_{WS_d} columns of $\mathbf{U}$, the top N_{WS_d} singular values in $\mathbf{\Sigma}$, and the first N_{WS_d} rows of $\mathbf{V}^T$. Let $\mathbf{U}_d \in \mathbb{R}^{N_T \times N_{\text{WS}_d}}$ be the truncated left singular matrix, $\mathbf{\Sigma}_d \in \mathbb{R}^{N_{\text{WS}_d} \times N_{\text{WS}_d}}$ the diagonal matrix of the largest N_{WS_d} singular values, and $\mathbf{V}_d^T \in \mathbb{R}^{N_{\text{WS}_d} \times N_{\text{WS}_d}}$ the truncated right singular matrix. The reduced runoff matrix $\mathbf{F}_{\text{runoff-SVD}}$ is given by:

$$\mathbf{F}_{\text{runoff-SVD}} = \mathbf{U}_d\mathbf{\Sigma}_d = \mathbf{F}_{\text{runoff}}\mathbf{V}_d. \tag{4.4}$$

The matrix $\mathbf{F}_{\text{runoff-SVD}} \in \mathbb{R}^{N_T \times N_{\text{WS}_d}}$ is a low-dimensional representation of $\mathbf{F}_{\text{runoff}}$. Let $\mathcal{S}_{\text{runoff-SVD}}$ denote the reduced feature space corresponding to the runoff data, which now contains N_{WS_d} features.

- **Dynamic Environmental Variables (DEV)**: A significant portion of the input consists of time-variant features that exhibit substantial short-term and seasonal fluctuations over time. They are highly responsive to daily environmental changes, such as evapotranspiration and temperature. A detailed description of these features in the proposed hybrid model is provided in Section 6.4. Dynamic Environmental Variables (DEV) can be grouped into separate feature sets. For instance, $\mathcal{S}_{\text{evap}}$ represents a set of features evapotranspiration-derived features (containing N_{evap} variables), while $\mathcal{S}_{\text{temp}}$ represents a set of temperature-derived features (containing N_{temp} variables). Since these features originate from different sources, they must share the same number of time stamps.

  Let $\mathcal{S}_{\text{DEV}}$ be the combined set of DEV features, representing the union of all sources:

$\mathcal{S}_{\text{DEV}} = \mathcal{S}_{\text{evap}} \cup \cdots \cup \mathcal{S}_{\text{temp}}$. Define $\mathbf{F}_{\text{DEV},k} \in \mathbb{R}^{N_T \times N_{\text{DEV}}}$ as the data matrix of dynamic environmental variables for watershed k and its associated river reach. Here, $N_{\text{DEV}} = N_{\text{evap}} + \cdots + N_{\text{temp}}$ is the total number of DEV features, and N_T the number of time stamps. Section 6.4 provides details on the variables considered as DEVs

- **Quasi-Static Environmental Variables (QSEV)**: Unlike DEV, some features remain stable over short-term records but evolve over decades due to environmental or geological processes. In our case study, these features are treated as constant over the period of interest and will be detailed in Section 6.4.

  QSEV features form a set where each element represents a feature that changes minimally over time. Examples include soil properties and river slope. Let $\mathcal{S}_{\text{soil}}$ (containing N_{soil} soil-related features) and $\mathcal{S}_{\text{slope}}$ (containing N_{slope} slope-related features) denote features extracted from watershed and river data. The combined QSEV set is $\mathcal{S}_{\text{QSEV}}$, containing $N_{\text{QSEV}} = N_{\text{soil}} + \cdots + N_{\text{slope}}$ features, and is the union of features obtained from all the sources: $\mathcal{S}_{\text{QSEV}} = \mathcal{S}_{\text{soil}} \cup \cdots \cup \mathcal{S}_{\text{slope}}$.

  Since each feature has a unique scalar value over time, we represent the data in matrix form as $\mathbf{F}_{\text{QSEV},k} \in \mathbb{R}^{N_T \times N_{\text{QSEV}}}$ for watershed k and its associated river reach. The i-th column denotes a vector of N_T rows, where each row holds the same scalar value for the i-th feature.

4.3.2.1 Feature correlation analysis

Feature correlation analysis is essential for feature selection, assessing statistical relationships within a dataset. It helps identify linear dependencies, detect multicollinearity, and understand feature interactions. A widely used method is Pearson's correlation, which quantifies the linear relationship between two continuous variables. After performing correlation analysis using the Pearson correlation coefficient, we can cluster potential features. Features within the same cluster can be retained or discarded based on their importance in the feature selection process. The following section introduces two methods for selecting these features.

4.3.2.2 Feature selection methods

In this section, we introduce two feature selection methods used to identify the most sensitive combination of features for streamflow prediction.

(a) Gradient-based feature selection Gradient-based feature importance is a model-based selection technique that evaluates input feature significance by analyzing the gradients

of the model's output with respect to its inputs. A foundational approach, *vanilla gradients*, computes straightforward gradients to assess how changes in each feature impact the output [135].

This method measures how sensitive the model's output is to each input feature. Features with higher gradient magnitudes are deemed more important, as small changes in these features cause larger output variations. Tailored to the specific model, it captures how the trained model utilizes each feature and provides detailed, sample-level importance scores, which can be aggregated for overall feature importance.

Suppose we establish a neural network model g that takes the j-th sample of input vector $\mathbf{f}^{(j)} = \left[f_1^{(j)}, f_2^{(j)}, \ldots, f_{N_{\text{total}}}^{(j)}\right]$, where N_{total} indicates the total number of features in the input vector. The gradient of the output $g(\mathbf{f}^{(j)})$ with respect to an input feature $f_i^{(j)}$ is given by the partial derivative:

$$w_i^{(j)} = \frac{\partial g(\mathbf{f}^{(j)})}{\partial {f_i}^{(j)}}. \tag{4.5}$$

To assess the overall importance of each feature across all samples, we compute the average absolute gradient magnitude. For a dataset with N_{samples} samples, this is computed as [135]:

$$\bar{G}_i = \frac{1}{N_{\text{samples}}} \sum_{j=1}^{N_{\text{samples}}} \left| \frac{\partial g(\mathbf{f}^{(j)})}{\partial {f_i}^{(j)}} \right| = \frac{1}{N_{\text{samples}}} \sum_{j=1}^{N_{\text{samples}}} \left| w_i^{(j)} \right|, \tag{4.6}$$

where the quantity $\bar{G}_i$ represents the importance of the i^{th} input feature, $\left|w_i^{(j)}\right|$ represents the absolute gradient magnitude of the function g with respect to the i^{th} input feature for the j^{th} sample.

This method depends on the trained and validated ML model, making it time-consuming as models must be trained before feature importance can be assessed. Therefore, an alternative, model-independent approach that efficiently selects relevant features without extensive processing is valuable. The next section introduces variance decomposition-based feature selection as such an alternative.

(b) Variance decomposition-based feature selection Another feature selection approach is the Sobol' index, a variance-based sensitivity analysis method widely used in global sensitivity analysis (GSA) to quantify the contribution of input variables to output variance [136]. The Sobol' index comprises a set of variance-based sensitivity indices that partition and quantify the variance of an output as a function of contributions from various input variables and their interactions. It helps identify which input variables are the most influential in driving the variability of the output.

The Sobol' indices are typically divided into two types: the First-Order Sobol' Index and the Total Sobol' Index. The First-Order Sobol' Index measures the main effect of an individual input variable $\mathbf{f}_i = [f_1^{(i)}, f_2^{(i)}, \ldots, f_{N_{\text{samples}}}^{(i)}]$ (a vector containing all the samples of i-th feature) on the corresponding output vector $\mathbf{y}$, representing the fraction of the output variance attributable to $\mathbf{f}_i$ alone. It can be expressed as:

$$S_i = \frac{\text{Var}_{\mathbf{f}_i}\left[\mathbb{E}\left(\mathbf{y} \mid \mathbf{f}_i\right)\right]}{\text{Var}(\mathbf{y})}, \tag{4.7}$$

where $\text{Var}_{\mathbf{f}_i}\left[\mathbb{E}\left(\mathbf{y} \mid \mathbf{f}_i\right)\right]$ is the variance of the expected value of $\mathbf{y}$ given $\mathbf{f}_i$, and $\text{Var}(\mathbf{y})$ is the total variance of $\mathbf{y}$.

The Total Sobol' Index quantifies the overall effect of the i-th feature vector $\mathbf{f}_i$, including its main effect and interactions with other variables. It is given by:

$$S_{Ti} = 1 - \frac{\text{Var}_{\sim\mathbf{f}_i}\left[\mathbb{E}\left(\mathbf{y} \mid \mathbf{f}_{\sim i}\right)\right]}{\text{Var}(\mathbf{y})}. \tag{4.8}$$

Here, $\text{Var}_{\sim\mathbf{f}_i}\left[\mathbb{E}\left(\mathbf{y} \mid \mathbf{f}_{\sim i}\right)\right]$ is the variance of $\mathbf{y}$ when the input $\mathbf{f}_i$ is fixed, focusing on all other variables $\mathbf{f}_{\sim i}$. A higher Sobol' index S_i or S_{Ti} signifies greater feature contribution to output variability, emphasizing its importance. This article utilizes the method proposed by Li *et al.* [136]. It is a model-free approach that proposes a direct estimation of sensitivity indices using just model inputs and outputs and does not require a model that connects the input to the output.

Remark 1 On a practical note, when performing gradient-based feature selection, all three feature sources—the reduced runoff data matrix $\mathbf{F}_{\text{runoff-SVD}}$, dynamic environmental variables $\mathbf{F}_{\text{DEV},k}$, and quasi-static environmental variables $\mathbf{F}_{\text{QSEV},k}$—are used as model inputs. However, variance decomposition-based feature selection considers only $\mathbf{F}_{\text{runoff-SVD}}$ and $\mathbf{F}_{\text{DEV},k}$ to identify globally sensitive features, while quasi-static variables in $\mathbf{F}_{\text{QSEV},k}$ remain unchanged, as they represent constant time series. For hybrid model training, data from multiple rivers are used, with their respective matrices vertically concatenated.

4.3.3 ML models for time-series river discharge prediction of ungauged rivers

The primary idea is to obtain all the reduced features and input data that need to be trained against the gauged river discharge time series. The premise of developing this hybrid ML model, trained using gauged rivers and then applying it to predict discharge for ungauged rivers, is based on the expectation that the model learns the causal relationship between the

input features and the discharge data of neighboring rivers (with gauges). The model is expected to learn reasonably generic river dynamics, and this learned knowledge can then be used to predict the discharge of ungauged rivers, provided the necessary input data is available for the ungauged river of interest. In this paper, we primarily employ two types of deep learning models: an LSTM-based model and an SNGP ResNet-based model.

4.3.3.1 LSTM-based model

Long Short-Term Memory (LSTM) is a specialized recurrent neural network (RNN) architecture designed to effectively manage and utilize long-range dependencies in sequential data. Introduced by Hochreiter and Schmidhuber in 1997 [137], LSTM addresses the limitations of traditional RNNs, particularly the issues of vanishing and exploding gradients, which hindered RNNs from learning long-term dependencies effectively.

The LSTM model used in this paper begins with an input layer that processes time-series data. This is followed by three stacked LSTM layers to capture temporal dependencies across various time scales. To mitigate overfitting, dropout layers are applied after each LSTM layer. A reshape layer then formats the outputs, which are flattened before reaching the final densely connected output layer, tailored for regression tasks. This architecture allows flexibility in forecasting methods by adapting the final layer based on specific prediction needs. Section 4.5.2.1 provides more detailed information about the LSTM model used in this study.

4.3.3.2 SNGP Resnet model

While LSTM models excel at handling time-series data, they are limited in their ability to effectively quantify uncertainty, a limitation explored in the next section. To address this, integrating the Spectral Normalization Gaussian Process (SNGP) with ResNet has proven beneficial [138], offering enhanced robustness, improved uncertainty estimation, and better generalization. To mitigate the high memory and inference costs of Bayesian neural networks and deep ensembles in real-time, industrial-scale applications, Liu *et al.* [138] proposed SNGP, which enables high-quality uncertainty estimation using a single deep neural network (DNN). Before discussing the model, we first introduce the residual-based DNN used in our study: ResNet. Residual Networks (ResNets) mark a key advancement in deep learning, allowing the training of much deeper networks. Introduced by He *et al.* [139], ResNets address the degradation problem, where deeper networks suffer performance deterioration due to training challenges rather than overfitting.

ResNet is based on the idea that each layer in a deep network learns a residual mapping

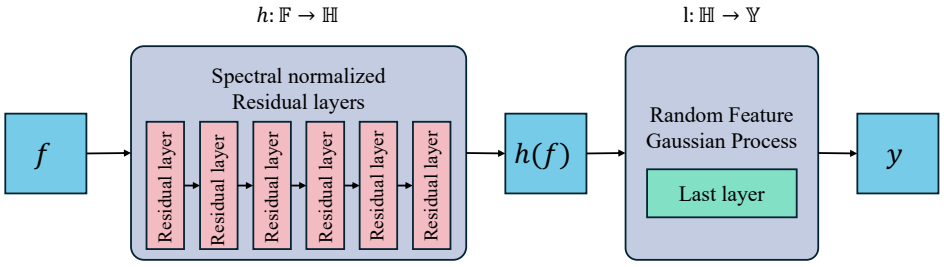


Figure 4.2: The diagram of SNGP Resnet-based model structure.

instead of a direct input-output mapping. Mathematically, if $H(\mathbf{f})$ is the desired mapping for a set of layers and $\mathbf{f}$ is the input, ResNet models the residual function as $F(\mathbf{f}) = H(\mathbf{f}) - \mathbf{f}$. The final output is $H(\mathbf{f}) = F(\mathbf{f}) + \mathbf{f}$, implemented via skip connections that perform identity mapping, adding their outputs to the stacked layers.

To incorporate both aleatoric (data-related) and epistemic (model-related) uncertainty in ResNet and improve its robustness, the SNGP technique can be integrated. This involves two key components: Spectral Normalization (SN) and a Gaussian Process (GP) layer. The structure of the SNGP model is shown in Figure 4.2, with details of both components provided below.

Spectral Normalization (SN) Spectral Normalization (SN) is used in neural network training to control the Lipschitz constant of network layers. By constraining the spectral norm (the largest singular value) of weight matrices, SN limits how much a layer's transformation expands distances between input points. It normalizes each layer's weight matrix $\mathbf{W}$ by its spectral norm λ, estimated using the power iteration method—an efficient technique for finding a matrix's dominant singular value [140]. The normalization adjusts weights as [138]:

$$\mathbf{W}_l = \begin{cases} \frac{c \cdot \mathbf{W}_l}{\sqrt{\lambda}} & \text{if } c < \lambda \\ \mathbf{W}_l & \text{otherwise} \end{cases} \tag{4.9}$$

Here, $\mathbf{W}_l$ is the weight matrix of layer l, λ is the estimated spectral norm, and c is a hyperparameter that adjusts the effective upper bound of the spectral norm, ensuring $|\mathbf{W}_l|_2 \leq c$. This constraint helps maintain the desired Lipschitz continuity, ensuring that the network does not distort input distances excessively.

In ResNet, SN is applied to each residual block's weight matrix W to maintain a controlled output range. This ensures that the addition of the residual block's output to its input (i.e., $\mathbf{f} + F(\mathbf{f})$) does not lead to disproportionate scaling of the input features, thereby preserving the geometric structure in the hidden space. During training, SN is enforced at

each step before forward and backward passes, stabilizing training and preventing exploding gradients in deep networks. By controlling the Lipschitz constant, it enhances generalization, mitigates overfitting, and improves robustness to input perturbations and adversarial attacks by reducing sensitivity to small input changes.

Gaussian Process (GP) Layer A GP layer is incorporated into the SNGP model to enhance its distance-aware learning capabilities by replacing the standard dense output layer of ResNet with a GP layer that leverages a Laplace approximation and Random Fourier Features (RFF). This substitution helps manage computational complexity while preserving end-to-end trainability.

A GP is defined over the hidden representations $\mathbf{h} = h(\mathbf{f})$ of the input $\mathbf{f}$, where h is a mapping learned by preceding residual neural network layers. The GP output layer models $\boldsymbol{y}$ as drawn from a multivariate normal distribution ($\mathcal{MVN}$):

$$g(\mathbf{h}) \sim \mathcal{MVN}(\mathbf{0}, \mathbf{K}), \tag{4.10}$$

where $\mathbf{K}$ is the covariance matrix, built using a kernel function, which encodes the similarity between pairs of data points in terms of their hidden representations. Each element K_{sp} of the matrix represents the covariance between the outputs for data points s and p. In this case, $K_{sp} = \exp\left(-|\mathbf{h}_s - \ \mathbf{h}_p|^2/2\right)$.

To approximate the GP kernel efficiently, RFF is employed [138]:

$$\mathbf{K} \approx \Phi\Phi^T, \tag{4.11}$$

where Φ is constructed from the hidden layer outputs as:

$$\Phi_{D_l} = \sqrt{\frac{2}{D_l}} \cos\left(\mathbf{W}\mathbf{h} + b_l\right). \tag{4.12}$$

Here, RFF samples random "frequencies" $\mathbf{W}$ and random phases b_l. $\mathbf{W}$ consists of weights sampled from $N(0, 1)$, and b_l is sampled from a uniform distribution $U(0, 2\pi)$. D_l represents the dimensionality of $\mathbf{h}$.

Posterior inference in Gaussian Processes is computationally expensive due to the inversion of the covariance matrix $\mathbf{K}$. The Laplace approximation simplifies this by approximating the posterior around the mode (MAP estimate) of the likelihood:

$$\beta \sim \mathcal{MVN}\left(\hat{\beta}, \Sigma\right), \tag{4.13}$$

where, $\hat{\beta}$ is the MAP estimate, and Σ is the inverse of the Hessian matrix $\mathbf{H}$ of the log posterior likelihood evaluated at $\hat{\beta}$. The posterior mean β is updated using stochastic gradient descent [138]:

$$\beta \leftarrow \beta - \eta \nabla_{\beta} \left(\log p(\mathbf{h} \mid \beta) + \frac{1}{2} \|\beta\|^2 \right), \tag{4.14}$$

here, η is the learning rate of the GP layer.

In summary, SNGP enhances deep learning models like ResNets by integrating spectral normalization and a GP layer. Spectral normalization regulates the spectral norm of layer weights, preserving input space geometry and stabilizing training. The GP layer, leveraging a Laplace approximation and RFF, efficiently approximates the kernel matrix and models final predictions, enabling end-to-end training and improved uncertainty estimation. This setup enhances robustness, reliability, and interpretability, making SNGP ideal for critical applications requiring predictive uncertainty assessment.

4.3.4 Hybrid model architecture

The proposed hybrid model integrates a data-driven machine learning approach with the physics-based RAPID model, combining the former's ability to capture complex patterns with the latter's detailed physical insights into river dynamics. The next two sections introduce the purely data-driven and hybrid models.

Consider N_{gauged} gauged rivers near the ungauged river of interest. Various input data (discussed in Section 4.3.2) are used, depending on the feature extraction approach (see Remark 1), yielding the reduced data matrix $\mathbf{F}_{\text{reduced}} \in \mathbb{R}^{(N_T \times N_{\text{gauged}}) \times N_{\text{reduced}}}$. Let $\mathbf{F}_{\text{reduced},k} \in \mathbb{R}^{N_T \times N_{\text{reduced}}}$ represent the reduced input data for river reach k, and $\mathbf{y}_{\text{obs},k} \in \mathbb{R}^{N_T \times 1}$ the observed discharge data for gauged river k.

4.3.4.1 Purely data-driven model

The LSTM-based model that is purely data-driven, is trained using the *reduced-feature matrix* as input and the *observed discharge of all gauged rivers* as output. A single model is trained for all gauged rivers to effectively capture both shared and river-specific flow dynamics.

The organization of training data is crucial and is discussed here. At time step t, the model is trained on the past N_{W} time steps (including t), represented as $\mathbf{F}_{\text{reduced},k}([N_{\text{W}} - 1 : t], :) \in \mathbb{R}^{N_{\text{W}} \times N_{\text{reduced}}}$. The combined training data for all gauged rivers is given by $\mathbf{X}_{\text{train}} \in \mathbb{R}^{((N_T - N_W) \times N_{\text{gauged}}) \times N_{\text{W}} \times N_{\text{reduced}}}$, with a sample size of $((N_T - N_W) \times N_{\text{gauged}})$

and an input size per sample of $(N_{\mathrm{W}} \times N_{\mathrm{reduced}})$. The corresponding output is $\mathbf{Y}_{\mathrm{train}} \in \mathbb{R}^{((N_T - N_W) \times N_{\mathrm{gauged}}) \times 1}$. For instance, the input at time step t for river k is $\mathbf{X}_{\mathrm{train}}(t, k, :, :) = \mathbf{X}_{\mathrm{train},t,k} = \mathbf{F}_{\mathrm{reduced},k}([N_{\mathrm{W}} - 1 : t], :) \in \mathbb{R}^{N_{\mathrm{W}} \times N_{\mathrm{reduced}}}$, with the corresponding output $\mathbf{Y}_{\mathrm{train}}(t, k, 1) = \mathbf{Y}_{\mathrm{train},t,k} = \mathbf{y}_{\mathrm{obs},k}(t + 1) \in \mathbb{R}^1$.

The number of training samples is reduced from N_T due to the sliding window approach with size N_W, yielding $(N_T - N_W + 1)$ samples. Since the output is the discharge at the next time step, the sample size is further reduced by one to $(N_T - N_W)$.

The LSTM model $\mathcal{G}$ learns the mapping from the input sequence $\mathbf{X}_{\mathrm{train}}$ to the output sequence $\mathbf{Y}_{\mathrm{train}}$:

$$\hat{\mathbf{Y}}_{t,k} = \mathcal{G}(\mathbf{X}_{\mathrm{train},t,k}), \tag{4.15}$$

where $\hat{\mathbf{Y}}_{t,k} \in \mathbb{R}^1$ is the predicted output at time t for river k. LSTM model in this case is used as an operator that maps time-sequence inputs of river discharge drivers to the discharge prediction at t.

4.3.4.2 Data-driven model augmented with RAPID prediction

Unlike the purely data-driven model, the hybrid model incorporates RAPID predictions $\mathbf{F}_{\mathrm{RAPID},k}$ alongside features $\mathbf{F}_{\mathrm{reduced},k}$ for each gauged river k. The appended feature matrix is defined as: $\mathbf{F}_{\mathrm{total},k} = [\mathbf{F}_{\mathrm{reduced},k}, \mathbf{F}_{\mathrm{RAPID},k}] \in \mathbb{R}^{N_T \times (N_{\mathrm{reduced}}+1)}$. The training process remains unchanged except that $\mathbf{F}_{\mathrm{reduced},k}$ is replaced with $\mathbf{F}_{\mathrm{total},k}$.

4.4 Uncertainty quantification of river discharge prediction of un-gauged rivers

4.4.1 Methods for UQ of ML models

Streamflow prediction using ML techniques relies on historical data and can suffer from significant extrapolation errors under changing environmental conditions. Thus, UQ is crucial for enhancing model robustness and reliability. This section presents three UQ methods for LSTM-based and SNGP ResNet-based models: MC-dropout for LSTM, Neural Network Ensemble (NNE) for LSTM, and SNGP for ResNet.

4.4.1.1 MC-dropout based on LSTM

MC-dropout leverages dropout layers during both training and inference to approximate Bayesian inference. By enabling dropout at inference, the model performs multiple stochastic forward passes, capturing the distribution of possible predictions rather than a single

deterministic output. Gal and Ghahramani [123] established a theoretical framework interpreting dropout training in deep neural networks as approximate Bayesian inference in deep Gaussian processes, providing a scalable and computationally efficient means to quantify model (epistemic) uncertainty. This approach requires only a modest increase in computation compared to standard inference and is highly versatile, as it can be applied to any neural network with dropout layers. While MC-dropout addresses epistemic uncertainty, aleatory uncertainty can also be quantified when the model samples from a predictive distribution, for example, by employing a pinball loss function. Together, these capabilities make MC-dropout a widely used and practical method for uncertainty quantification in hydrological deep learning models.

In MC-dropout, each weight matrix $\mathbf{W}_l$ in a neural network layer l is multiplied elementwise with a random binary mask. This mask is sampled from a Bernoulli distribution with a dropout probability p_l:

$$z \sim \text{Bernoulli}\left(p_l\right). \tag{4.16}$$

If $z = 0$, the corresponding weight connection is "dropped" (set to zero). This creates a sparse network for each forward pass, with randomly selected neurons deactivated.

MC-dropout approximates Bayesian inference by introducing stochasticity through dropout during training and testing. This allows the model to generate diverse outputs for the same input via multiple forward passes, effectively creating an ensemble of predictions. Mathematically, each pass is a sample from a distribution $q(\boldsymbol{\theta}; \lambda)$ over the model weights, where $\boldsymbol{\theta}$ are the weights and λ is the dropout probability. It approximates the posterior of the model weights conditioned on the input $\mathbf{X}$ and output $\mathbf{Y}$, denoted as $p(\boldsymbol{\theta} \mid \mathbf{X}, \mathbf{Y})$, using a variational distribution $q(\boldsymbol{\theta}; \lambda)$ that factorizes over the layers:

$$q(\boldsymbol{\theta}; \lambda) = \prod_{l=1}^{L} q_{\mathcal{M}}\left(\mathbf{W}_l\right). \tag{4.17}$$

Here, L is the number of layers in the network, $q_{\mathcal{M}}\left(\mathbf{W}_l\right)$ represents a Gaussian mixture model for each layer's weights, which approximates the posterior distribution as [94]:

$$q_{\mathcal{M}}\left(\mathbf{W}_l\right) = p_l \mathcal{N}\left(M_l, \sigma_l^2 \mathbf{I}\right) + \left(1 - p_l\right) \mathcal{N}\left(0, \sigma_l^2 \mathbf{I}\right). \tag{4.18}$$

This represents a Bernoulli mixture of two Gaussians with the same standard deviation, σ_l, but different means: one Gaussian has a mean of M_l, and the other has a mean of zero.

During the testing phase, the model performs multiple forward passes (Monte Carlo sampling) with active dropout. This generates a set of predictions $\left\{\hat{\mathbf{y}}^{(1)}, \hat{\mathbf{y}}^{(2)}, \ldots, \hat{\mathbf{y}}^{(N_{\text{passes}})}\right\}$ for

a given input $\mathbf{X}_{\text{test}}$. The mean prediction and uncertainty can then be estimated as [94]

$$\begin{aligned} \hat{\boldsymbol{\mu}} &\approx \frac{1}{N_{\text{passes}}} \sum_{n=1}^{N_{\text{passes}}} \hat{\mathbf{y}}^{(n)}, \\ \hat{\boldsymbol{\sigma}}^2 &\approx \frac{1}{N_{\text{passes}}} \sum_{n=1}^{N_{\text{passes}}} \left(\hat{\mathbf{y}}^{(n)} - \hat{\boldsymbol{\mu}}\right) \circ \left(\hat{\mathbf{y}}^{(n)} - \hat{\boldsymbol{\mu}}\right). \end{aligned} \tag{4.19}$$

Here, $\circ$ is the Hadamard product, or element-wise multiplication operator.

4.4.1.2 Neural Network Ensemble based on LSTM

Neural Network Ensemble (NNE) estimates uncertainty by combining predictions from multiple independently trained neural networks. Leveraging model diversity, it enhances robustness and reliability in streamflow predictions. A key advantage of NNE is its ability to reduce overfitting by averaging multiple models, which improves generalization to unseen data. Additionally, the spread of predictions naturally quantifies uncertainty. NNE is highly flexible and applicable to any neural network architecture, making it a versatile solution for various machine learning problems [94].

NNE captures both aleatoric and epistemic uncertainty. To estimate epistemic uncertainty (related to the model), multiple neural networks $\{\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_M\}$ are trained independently with different random initializations. Aleatoric uncertainty, representing inherent data noise, is modeled as a variance term associated with each prediction. For input $\mathbf{X}$, each model $\mathcal{M}_m$ outputs a mean prediction $\hat{\boldsymbol{\mu}}_m(\mathbf{X})$ and an aleatoric uncertainty $\hat{\boldsymbol{\sigma}}_m^2(\mathbf{X})$. This is achieved by incorporating a final Gaussian layer into $\mathcal{M}_m$, such that the predicted discharge $\hat{\mathbf{y}}_m$ is given as:

$$\hat{\mathbf{y}}_m \sim \mathcal{N}\left(\hat{\boldsymbol{\mu}}_m(\mathbf{X}), \hat{\boldsymbol{\sigma}}_m^2(\mathbf{X})\right). \tag{4.20}$$

To estimate epistemic uncertainty, we use an ensemble of neural networks, $\{\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_M\}$, capturing uncertainty in model parameters due to limited or incomplete training data. For input $\mathbf{X}$, the ensemble mean prediction, $\boldsymbol{\mu}(\mathbf{X})$, is obtained by averaging the predictions from all networks as

$$\hat{\boldsymbol{\mu}}\left(\mathbf{X}\right) = \frac{1}{M} \sum_{m=1}^{M} \hat{\boldsymbol{\mu}}_m(\mathbf{X}). \tag{4.21}$$

The total predictive uncertainty (which includes both aleatoric and epistemic components)

is given by [94]

$$\hat{\boldsymbol{\sigma}}^2(\mathbf{X}) = \overbrace{\frac{1}{M}\sum_{m=1}^{M}\hat{\boldsymbol{\sigma}}_m^2(\mathbf{X})}^{\text{Aleatory uncertainty}} + \overbrace{\frac{1}{M}\sum_{m=1}^{M}\left(\hat{\boldsymbol{\mu}}_m(\mathbf{X}) - \boldsymbol{\mu}(\mathbf{X})\right) \circ \left(\hat{\boldsymbol{\mu}}_m(\mathbf{X}) - \boldsymbol{\mu}(\mathbf{X})\right)}^{\text{Epistemic uncertainty}}. \tag{4.22}$$

The structure of the NNE LSTM-based model is illustrated in Figure 4.3. Each ensemble member is trained independently with different random initializations to capture epistemic uncertainty. No explicit selection or weighting of ensemble members based on probabilistic criteria is performed after training. Instead, diversity in predictions is achieved through randomization in training, and each member's reliability is verified using validation performance to ensure consistency and avoid unstable predictions. This architecture adheres to the principles of the NNE and incorporates LSTM layers throughout the entire model.

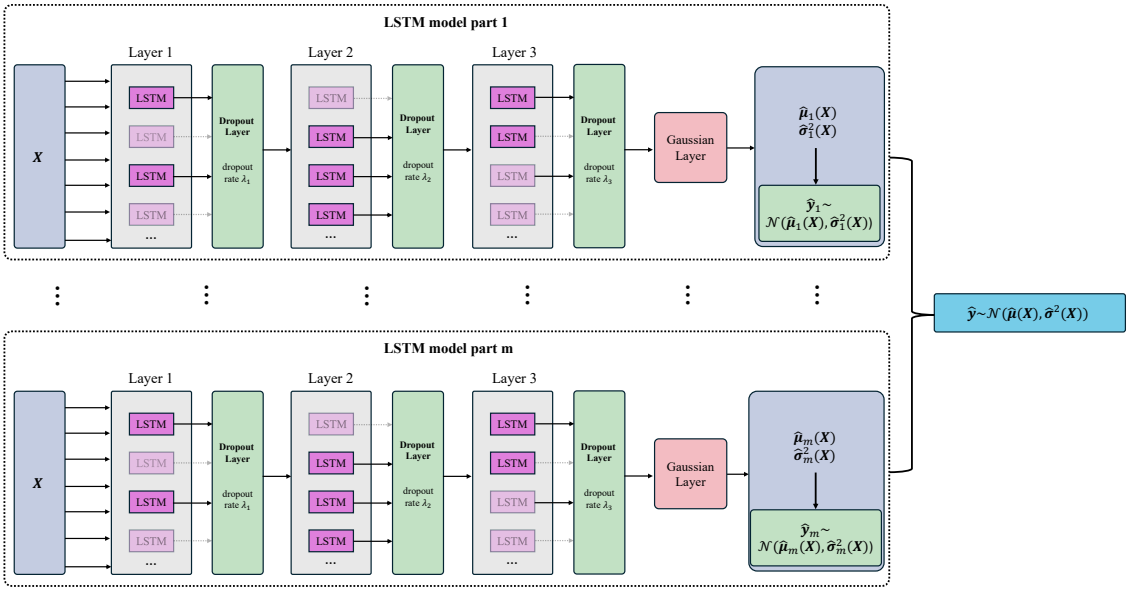


Figure 4.3: The structure of the NNE LSTM-based model.

4.4.1.3 SNGP based on Resnet

SNGP combines Gaussian processes with neural networks by normalizing spectral norms of weight matrices. This retains neural network flexibility while incorporating Gaussian process-based uncertainty quantification. A key advantage of SNGP is its improved calibration, ensuring probability estimates better reflect true prediction confidence. Spectral normalization is computationally efficient and scales well, making SNGP suitable for large-scale applications. A detailed description is given in Section 4.3.3.2, and the model structure

is shown in Figure 4.2.

4.4.2 Evaluation method for river discharge prediction uncertainty

Validating the qaulity of the UQ methods requires appropriate performance evaluation. A common method for evaluating the quality of predictive uncertainty is to create a calibration curve, a technique that has been widely used in the past. For example, Zadrozny *et al.* [141] demonstrated how to achieve accurate probability estimates for multiclass problems by combining calibrated binary estimates. The calibration curve applies to both classification and regression tasks and is constructed through three key steps [94]:

- **Determine the Confidence Level**: We first determine the confidence levels to assess. To model prediction uncertainty, we assume the probabilistic output at the i^{th} test step, $\hat{y}_i$, follows a Gaussian distribution with the probability density function:

$$\hat{y}_i \sim \mathcal{N}\left(\hat{\mu}_i, \hat{\sigma}_i\right), \tag{4.23}$$

  where, for a given input $X_{\text{test},i}$ (denoting the i-th row of the test input data matrix $\mathbf{X}_{\text{test}}$), $\hat{\mu}_i$ is the predicted mean (the i-th element of $\hat{\boldsymbol{\mu}}(\mathbf{X}\text{test})$), and $\hat{\sigma}_i$ is the standard deviation in the prediction of $\hat{y}_i$ (the i-th component of $\hat{\boldsymbol{\sigma}}(\mathbf{X}\text{test})$).

  For a specified confidence level $c \in [0, 1]$, we can determine a two-sided $100c\%$ confidence interval for the Gaussian random variable $\hat{y}_i$ as follows [94]

$$CI_{i,c} = \left[\hat{\mu}_i - z_{\frac{1+c}{2}}\hat{\sigma}_i, \hat{\mu}_i + z_{\frac{1+c}{2}}\hat{\sigma}_i\right], \tag{4.24}$$

  Here, $z_{\left(\frac{1+c}{2}\right)}$ represents the $\left(\frac{1+c}{2}\right)$-th quantile of the standard normal distribution, given by $z_{\left(\frac{1+c}{2}\right)} = \Phi^{-1}\left(\frac{1+c}{2}\right)$, where $\Phi(\cdot)$ denotes the cumulative distribution function of the standard normal distribution.

- **Calculate the Observed Confidence**: To assess the reliability of the model's uncertainty estimates, the observed confidence level is computed by comparing predictions to actual outcomes. This involves checking how often the predicted confidence intervals contain the true values. For example, at a 95% confidence level, the proportion of intervals including the true values should ideally approach 95%. This involves checking how often the predicted confidence intervals contain the true values. For example, at a nominal 95% confidence level, the proportion of intervals including the true values

should ideally approach 95%, indicating good calibration. As discussed in Khosravi *et al.* [142], this concept is formalized through the Prediction Interval Coverage Probability (PICP) metric, which measures the fraction of true observations that fall within the predicted intervals. A well-calibrated model should exhibit a PICP close to the nominal confidence level, providing a direct indication of the model's uncertainty reliability.

- **Plot Observed vs. Expected Confidence**: Finally, the results are visualized by plotting observed confidence against expected confidence levels. In a well-calibrated model, this plot aligns with the diagonal from bottom-left to top-right, indicating observed confidence matches expected confidence. Deviations from this line reveal potential calibration issues. This evaluation using calibration curves follows the same principle outlined by Renard *et al.* [143], which recommends assessing predictive reliability through consistency between nominal and empirical coverage.

Beyond the calibration curve, several metrics assess uncertainty quantification performance. Root Mean Square Error (RMSE) measures the difference between predicted and actual values and is used here to evaluate point prediction accuracy only. To assess the quality of predictive intervals and probabilistic forecasts, we use metrics such as the calibration curve and Negative Log-Likelihood (NLL). The calibration curve evaluates how well the predicted probabilities or intervals align with the actual observations, ensuring proper coverage. NLL quantifies how well the predicted distribution matches the observed data [94]. A detailed description of NLL is provided in Section 4.5.3. Together, these metrics comprehensively evaluate a model's ability to produce accurate point predictions and reliable uncertainty estimates.

4.5 Case Study

The United States is divided into seven main catchment zones. Building on Zhao *et al.* [101], this study focuses on part of the upper Colorado River basin, which includes 431 rivers. This basin features a range of discharge rates, including rivers regulated by dams. However, discharge data is available at only six locations, identified by USGS IDs: 700739316, 700749972, 700777836, 700782048, 700811946, and 700825284, highlighted in Figure 4.4. Our objective is to use discharge data from select gauged rivers to train predictive models. The remaining gauged rivers are treated as ungauged for testing and validation, with predicted discharge rates compared against observed data to assess model accuracy. The six river locations are split into training and test sets, with five randomly selected for training and

one for test. This simulates using gauged river data to predict discharge rates for ungauged rivers.

This case study requires two main datasets. The first is the input data, which includes all river-related features: time-series data from Climate Engine ($\mathbf{F}_{\text{DEV},k}$), runoff data after dimension reduction ($\mathbf{F}_{\text{runoff-SVD}}$), constant features such as soil properties, elevation, and slope ($\mathbf{F}_{\text{QSEV},k}$), and RAPID discharge predictions ($\mathbf{F}_{\text{RAPID},k}$). The second dataset is the model's output, the gauged discharge data for selected river locations, $\mathbf{y}_{\text{obs},k}$. The next section details these datasets.

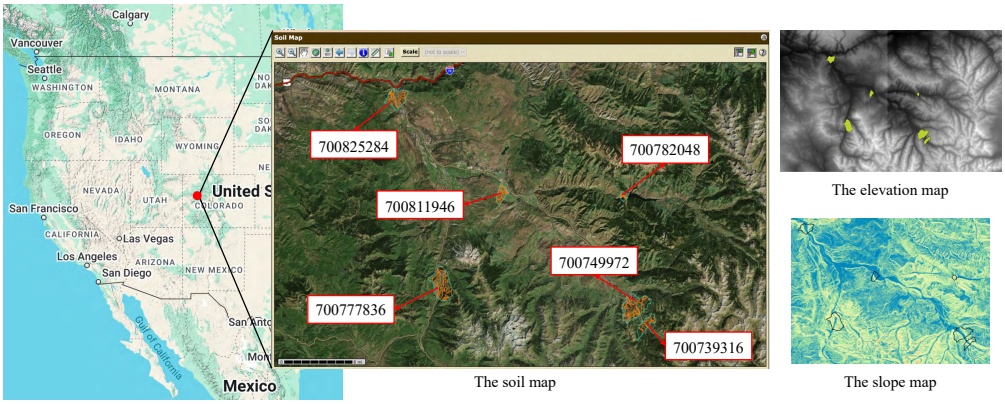


Figure 4.4: The map of the six gauged river locations of interest.

4.5.1 Raw data extraction and preprocessing

The dataset includes gauged river discharge data and river network details of the Colorado Basin for training and calibrating the RAPID model. Selective time-series features (DEV) were obtained from ClimateEngine.org (https://app.climateengine.org/climateEngine), which offers an application and API that integrates climate and remote sensing data for environmental analysis. Eight features were selected from the GridMET dataset (https://www.climatologylab.org/gridmet.html), which provides daily high-resolution ($\sim$ 4 km, 1/24th degree) surface meteorological data for the contiguous United States since 1979. These features include Mean Temperature, Precipitation, Hargreaves Potential Evapotranspiration, Specific Humidity, Wind Speed, Downward Shortwave Radiation, Vapor Pressure Deficit, and Burning Index, recorded as daily time-series data.

The slowly changing environmental features (QSEV) are gathered from various data sources. The first component to consider is soil data. Soil plays a vital role in streamflow prediction as it regulates water storage and release within a watershed. It influences the proportion of precipitation that infiltrates the ground versus becoming runoff,

thereby affecting streamflow. The soil's moisture content determines whether water will infiltrate into the soil or flow across the surface during rainfall, with dry soils promoting infiltration and wet soils increasing surface runoff. The soil data is sourced from the Web Soil Survey (WSS) (https://websoilsurvey.sc.egov.usda.gov/app/), which offers comprehensive soil data and information produced by the National Cooperative Soil Survey. This resource includes various features, such as hydrologic soil groups, among others. Figure 4.4 displays the soil map for the gauged river locations, highlighting the soil features pertinent to this case study. The raw data for the rest environmental features such as elevation and slope are sourced from the USGS online data portals, including The USGS Earth Explorer (https://earthexplorer.usgs.gov/), NHDPlus HR (https://www.usgs.gov/national-hydrography/nhdplus-high-resolution), and The National Map Apps (https://apps.nationalmap.gov/downloader/). By uploading the relevant shapefile, users can access Landsat satellite imagery, radar data, and more. We utilize the seamless map provided by the 3D Elevation Program (3DEP) with a resolution of 1/3 arc second. The seamless 3DEP data are processed from diverse sources to maintain a consistent coordinate system and unit of vertical measurement. This data is distributed using geographic coordinates in decimal degrees, adhering to the North American Datum of 1983 (NAD 83). All elevation values are measured in meters and, for the contiguous United States, are referenced to the North American Vertical Datum of 1988 (NAVD 88). Two map sections covering the six river locations can be accessed through links: https://www.sciencebase.gov/catalog/item/61b19ba7d34eb8f531255a1b and https://www.sciencebase.gov/catalog/item/620de4aed34e6c7e83ba9fdc. Figure 4.4 also illustrates the elevation map and the corresponding slope map derived from it. The image data can be processed using geographic information system (GIS) software to extract key statistics such as the mean, median, minimum, maximum, and standard deviation values for the selected river locations. Given that much of the Colorado region is mountainous with highly variable terrain and slopes, the mean and standard deviation of elevation and slope for each river location are primarily chosen as the features for this case study.

One of the input features not yet mentioned is the runoff dataset. This dataset is considered large as it contains information on 431 rivers within the area of interest, often leading to redundancy. To address this, the SVD technique is applied to the runoff data, $\mathbf{F}_{\text{runoff}}$, reducing its dimensionality from 431 to 18. The resulting reduced data matrix, $\mathbf{F}_{\text{runoff-SVD}}$, effectively captures the non-redundant information contained in $\mathbf{F}_{\text{runoff}}$.

Table 4.1 provides an overview of all features used in this case study. The abbreviation of the feature names will be used in the following sections. After data preprocessing, all the data is compiled and utilized for the subsequent feature selection process.

Full Term	Dimension	Feature Type	Abbreviation
Reduced runoff data	18	Time-series	M^3
RAPID predictions	1	Time-series	RP
Soil data	10	Constant (QSEV)	SD
Mean temperature	1	Time-series (DEV)	MT
Hargreaves potential evapotranspiration	1	Time-series (DEV)	HE
Wind Speed	1	Time-series (DEV)	WS
Precipitation	1	Time-series (DEV)	PR
Vapor pressure deficit	1	Time-series (DEV)	VP
Specific humidity	1	Time-series (DEV)	SH
Downward shortwave radiation	1	Time-series (DEV)	SR
Elevation difference	1	Constant (QSEV)	ED
Elevation max and min	1	Constant (QSEV)	EM
Slope mean	1	Constant (QSEV)	SM
Slope standard deviation	1	Constant (QSEV)	SS

Table 4.1: An overview of features in this case study

4.5.2 Hybrid model setup

The hybrid model integrates the physics-based RAPID model with ML using selected features. The following sections detail two ML models: an LSTM-based model (Section 4.5.2.1) and a ResNet-SNGP model (Section 4.5.2.2).

4.5.2.1 LSTM-based model setup

LSTM-based models are utilized for both streamflow prediction and uncertainty quantification. Establishing a solid foundational structure and selecting appropriate hyperparameters for the LSTM model can significantly enhance its performance. The structure of the LSTM model is discussed in Section 4.3.3.1, and the following text introduces the hyperparameters used in the case study.

Hyperparameters were fine-tuned using a grid search strategy across a predefined search space, including 1 to 4 layers, 50 to 200 units per layer, dropout rates from 0.0 to 0.3, and learning rates ranging from 0.0001 to 0.001. The optimal input window size was selected from 10 to 100 days. Each configuration was assessed based on validation performance and calibration criteria to ensure both accuracy and robust uncertainty quantification. The optimal LSTM model consists of three layers, each with 100 units, providing a moderately complex architecture capable of capturing intricate data patterns while ensuring computational efficiency. The choice of 100 units per layer balances the model's capacity to learn detailed features with its operational efficiency, supporting scalability and responsiveness

during training and inference. A batch size of 32 is employed based on experience. The model is trained for 200 epochs, a duration deemed sufficient for convergence on effective parameters without overfitting, especially with the application of regularization techniques like dropout. A dropout rate of 0.1 is applied in most cases (except in MC-dropout), meaning 10% of neurons are randomly excluded during each training iteration, which helps prevent reliance on individual neurons and enhances network resilience. The learning rate is set to 0.0001, a relatively low value that ensures stable and gradual convergence by making small adjustments to the weights, minimizing the risk of overshooting the optimal loss values during training. Finally, the Adam optimizer is employed for its efficiency in handling sparse gradients and its adaptive learning rate capabilities.

The optimal input window size is determined to be 35 days before the prediction date, with the output being the next day's streamflow discharge. This was determined through a grid search over 10 to 100 days, considering practical factors such as seasonal variations. The 35-day window was selected for its balance between predictive accuracy, computational efficiency, and reduced overfitting risk.

4.5.2.2 SNGP setup

As previously discussed, the SNGP model enhances robustness and uncertainty estimation in deep learning by integrating spectral normalization and Gaussian processes within a residual network. For the SNGP model used in this paper, it consists of 6 spectrum norm layers with each layer has 32 units. The spectrum norm bound used is 0.9. The learning rate is set to 0.0001 and a dropout rate of 0.1 is used to avoid overfitting. The model is trained using 1000 epochs and the best model is saved using callback.

4.5.3 Uncertainty quantification methods

Building on the benchmark models from Nemani *et al.* [94], the UQ models adopt similar architectures with modifications such as LSTM cells and varying hyperparameters. This section details three approaches: MC-dropout (LSTM-based), Neural Network Ensemble (LSTM-based), and SNGP (ResNet-based), outlining their structures and hyperparameters. To ensure the reliability of predictive intervals and guide decisions on sampling size, we evaluate probabilistic performance using calibration curves, NLL, and PICP. These metrics comprehensively assess how well the predicted uncertainty captures the true observations, supporting robust decision-making and ensuring the generated intervals are informative and reliable.

MC-dropout based on LSTM We develop an MC-dropout model following the methodology in Section 4.4.1.1. This model comprises three LSTM blocks, each equipped with a dropout layer that remains active during the prediction phase. The dropout rate for these layers varies randomly within the range of [0.05, 0.15]. This approach ensures that while an individual LSTM model might produce predictions with varying levels of confidence, the ensemble of models collectively generates reliable forecasts with a well-defined uncertainty range. The loss function utilized is Mean Squared Error (MSE) loss. The Elbow Method is employed to determine the optimal ensemble size for our case study. The Elbow Method determines the optimal ensemble size by testing values from 2 to 16, identifying 6 as optimal. The model is iterated 15 times to capture run-to-run variation.

Neural Network Ensemble based on LSTM As discussed in Section 4.4.1.2, each individual model in the ensemble incorporates a Gaussian layer as its final layer. For instance, in the m^{th} model at the i^{th} time step during the training process, this Gaussian layer outputs a predicted mean, $\hat{\mu}_{m,i}$ (i.e., the i-th element of $\hat{\boldsymbol{\mu}}_m(\mathbf{X}_{\text{train}};\boldsymbol{\theta})$), and variance $\hat{\sigma}^2_{m,i}$ (i.e., the i-th element of $\hat{\boldsymbol{\sigma}}^2_m(\mathbf{X}_{\text{train}};\boldsymbol{\theta})$). Here, $\boldsymbol{\theta}$ represents the parameter vector for the Neural Network model. These parameters are optimized by minimizing the Negative Log-Likelihood (NLL) loss, which is expressed as:

$$\text{NLL Loss for model } m: \quad \mathcal{L}_m(\boldsymbol{\theta}) = \sum_{i=1}^{(N_T-N_W)} \left[\frac{\log \hat{\sigma}^2_{m,i}}{2} + \frac{(Y_{\text{train},i} - \hat{\mu}_{m,i})}{2\hat{\sigma}^2_{m,i}} \right]. \tag{4.25}$$

The Elbow Method is utilized to determine the optimal ensemble size for this case study. By varying the ensemble size from 2 to 16, it was observed that an ensemble size of 4 is optimal for the NNE model. Subsequently, this model is iterated 15 times to account for run-to-run variation.

SNGP based on Resnet As discussed in Section 4.5.2.2, the SNGP model inherently quantifies uncertainty. It outputs the predicted next-day discharge and its associated standard deviation. To capture run-to-run variation, the model is iterated 15 times.

4.6 Results and Discussion

4.6.1 Feature selection analysis results

As discussed in Section 4.3.2.1, the Pearson correlation coefficient is used to identify linear dependencies between time-series streamflow features. The Pearson correlation coefficients

#	**Feature Name**
1	Specific humidity
2	Hargreaves potential evapotranspiration
3	Downward shortwave radiation
4	Reduced runoff M_{10}^3
5	RAPID

Table 4.2: Top 5 features using gradient-based feature selection

indicate a clear linear relationship between different features at six river locations of interest. The analysis reveals that certain features exhibit strong linear correlations over time. For instance, some features share similar seasonality patterns over extended periods, resulting in high linear relationships. Consequently, these features are grouped, and only a subset from each group is selected as potential features. One distinct group of features that exhibits a high degree of correlation includes *mean temperature*, *vapor pressure deficit*, and *Hargreaves potential evapotranspiration*. After that, variance decomposition-based feature selection was applied to all six river locations, analyzing all time-series features. Figure 4.5 presents the results, showing variations in feature importance across sites. Overall, *downward shortwave radiation* (SR) and *Hargreaves potential evapotranspiration* (HE) emerged as the most significant features. Similar results were found using gradient-based feature selection, as shown in Table 4.2, where M_{10}^3 represents the 10^{th} element in the *reduced runoff data*. Table 4.3 summarizes model validation performance with different feature combinations. Notably, including more features does not always improve performance. The best-performing model uses only *runoff data* (M^3), *RAPID predictions* (RP), *soil data* (SD), *evapotranspiration* (HE), and *radiation* (SR). For efficiency, subsequent research focuses on this feature set.

One river in the training set has a dam, leading to controlled flow patterns distinct from natural rivers. This discrepancy affects model predictions. To evaluate its impact, we removed this human-controlled river (river 4) from the training set. The results showed worsened performance, indicating its inclusion benefits prediction accuracy. Therefore, the human-controlled river remains in the training set.

4.6.2 Model performance and the results of uncertainty quantification analysis

MC-dropout Figure 4.6 presents a detailed prediction comparison of a selected region over different time periods of prediction with and without RAPID using MC-dropout. In this figure, the discharge values for both actual and predicted data are shown over time. The predictions include the mean values bounded by confidence intervals, which represent both

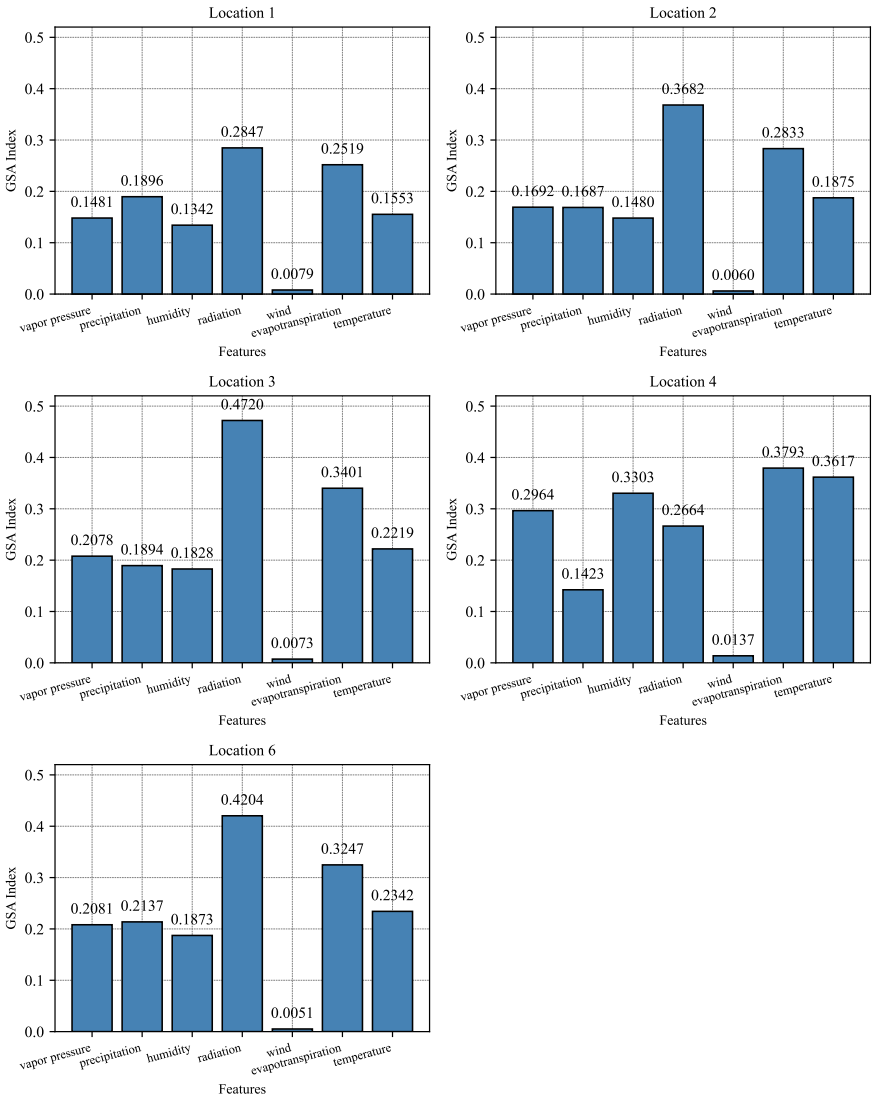


Figure 4.5: Variance decomposition-based feature selection.

the predicted values and their associated uncertainties. To generate Figure 4.6, we focus specifically on the peaks of the time-series data, as accurately predicting peak discharge is critical in practice. During the intervals from days 0 to 300, 400 to 600, 1150 to 1350, 2950 to 3200, and 4000 to 4300, the hybrid model clearly outperforms the purely data-driven model. However, for the peak occurring between days 1500 and 1700, both models overpredict discharge values before and after the peak. This discrepancy is primarily attributed to the relatively smaller size of this peak compared to others, which is closely linked to unexpected environmental conditions during that period. Overall, the hybrid model, incorporating MC-dropout with a 95% confidence interval, successfully captures most true values and demonstrates superior prediction performance compared to the purely data-driven model.

Features Combination	**Features**	*MSE*	*RMSE*
M^3, RP, SD, MT, HE, WS, PR, VP, SH, SR, ED, EM, SM, SS	40	62.43	7.90
M^3, RP, SD, MT, HE, WS, PR, VP, SH, SR, SM, SS	38	**53.94**	**7.34**
M^3, RP, SD, MT, HE, WS, PR, VP, SH, SR	36	90.97	9.54
M^3, RP, SD, HE, SR, SM, SS	33	55.59	7.46
M^3, RP, SD, HE, SR, ED, EM	33	65.64	8.10
M^3, RP, SD, HE, SR	31	**49.13**	**7.01**
M^3, RP, SD	29	162.57	12.75

Table 4.3: LSTM-based model validation performance using different input feature combinations

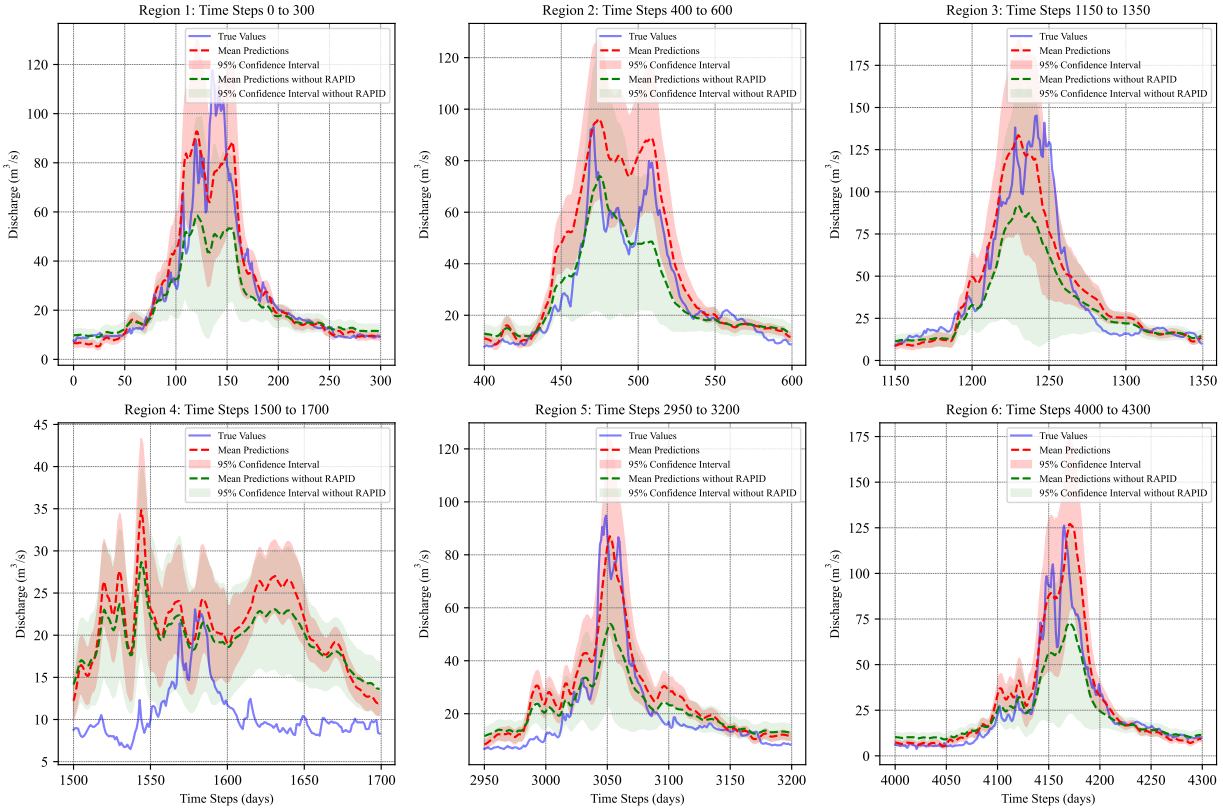


Figure 4.6: Divided time-series prediction plot with 95% confidence interval and true values (MC-dropout) between purely data-driven and hybrid model.

Figure 4.7 (a) presents the violin plot that compares the absolute errors between the models' predictions and observed discharge values using MC-dropout. This violin plot shows the distribution of absolute errors for both models, which is crucial for assessing model performance in terms of error minimization. By comparing the height and spread of the violins for each model, we can discern which model tends to have lower prediction errors, indicating better performance. The plot highlights areas where errors are most concentrated, showing

the commonality of error sizes. A taller, narrower violin suggests more consistent error sizes, whereas a wider base indicates greater variability. This plot reveals that the purely data-driven model has a wider error distribution, suggesting less reliability and potentially higher average errors, including more high-value errors. Overall, these findings demonstrate that the hybrid model using MC-dropout outperforms the purely data-driven model, underscoring the benefits of using a hybrid approach for prediction.

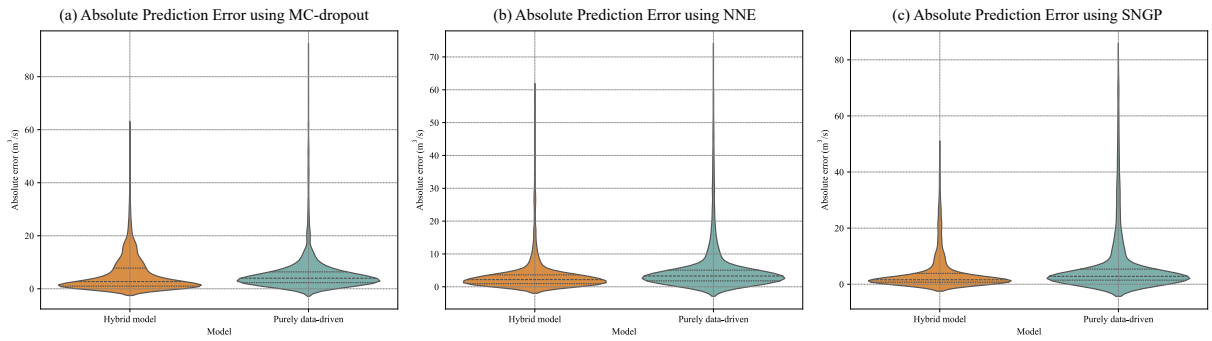


Figure 4.7: Violin plots of absolute error between models and observation discharge using MC-dropout (a), NNE (b), and SNGP (c).

Neural network ensemble Figure 4.8 illustrates the comparisons between model predictions with and without RAPID using NNE of different time periods. Similar to MC-dropout methods, the discharge values for both actual and predicted data are depicted over time, showcasing how well each model captures the dataset's dynamics over a 5000-day period. The prediction values include the mean predictions and the previously discussed confidence intervals, which represent both the predicted values and their associated uncertainties. Six peaks have been selected from the entire time period shown in Figure 4.8. It is evident that from days 0 to 300, 400 to 600, 1150 to 1350, 2950 to 3200, and 4000 to 4300, the hybrid model surpasses the purely data-driven model in performance. It is important to highlight that both models effectively capture the peak between days 1500 to 1700, a detail that the MC-dropout model overlooks. Overall, the hybrid model using NNE with a 95% confidence interval successfully captures most true values and delivers superior prediction performance compared to the purely data-driven model. Additionally, the performance of both NNE models surpasses that of the MC-dropout models.

Figure 4.7 (b) presents the violin plot that illustrates the absolute errors between the models' predictions and observed discharge values using NNE. This plot reveals that the purely data-driven model has a wider error distribution, suggesting less reliability and potentially higher average errors, including more high-value errors. Overall, these findings demonstrate

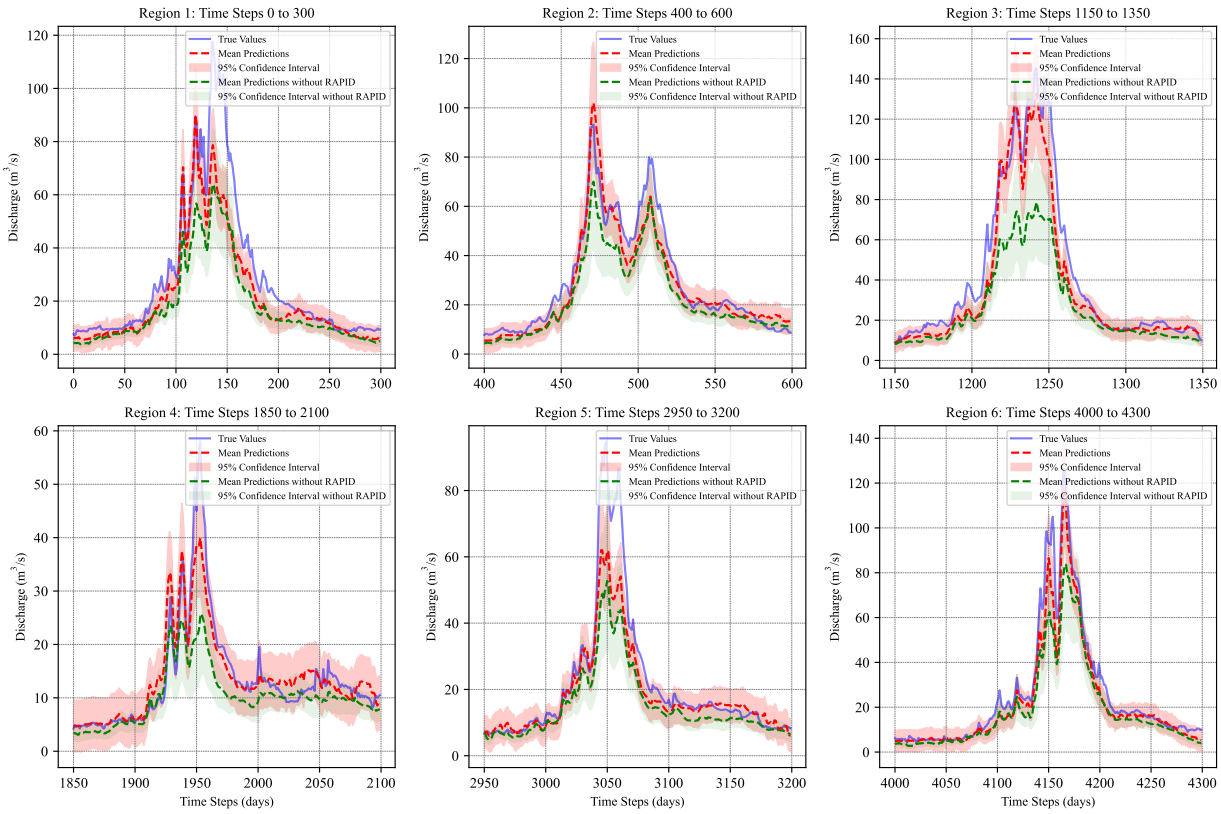


Figure 4.8: Divided time-series prediction plot with 95% confidence interval and true values (NNE) between purely data-driven and hybrid model.

that the hybrid model using NNE outperforms the purely data-driven model, underscoring the benefits of using a hybrid approach for prediction. This conclusion is consistent with the results observed from the MC-dropout models.

Spectral Normalization Gaussian Process Figure 4.9 illustrates detailed comparisons between model predictions with and without RAPID using SNGP for different time periods. It is evident that from days 0 to 300, 400 to 600, 1150 to 1350, 2950 to 3200, and 4000 to 4300, the hybrid model surpasses the purely data-driven model in performance. For the peak occurring between days 1500 and 1700, both models underpredict discharge values before and after the peak. However, the hybrid model still outperforms the purely data-driven model by better capturing the peak rather than failing to detect it. One notable feature of the SNGP model is that it has a considerably wider confidence interval compared to the other two UQ methods. Overall, the hybrid model using SNGP with a 95% confidence interval successfully covers most true values and delivers superior prediction performance compared to the purely data-driven model.

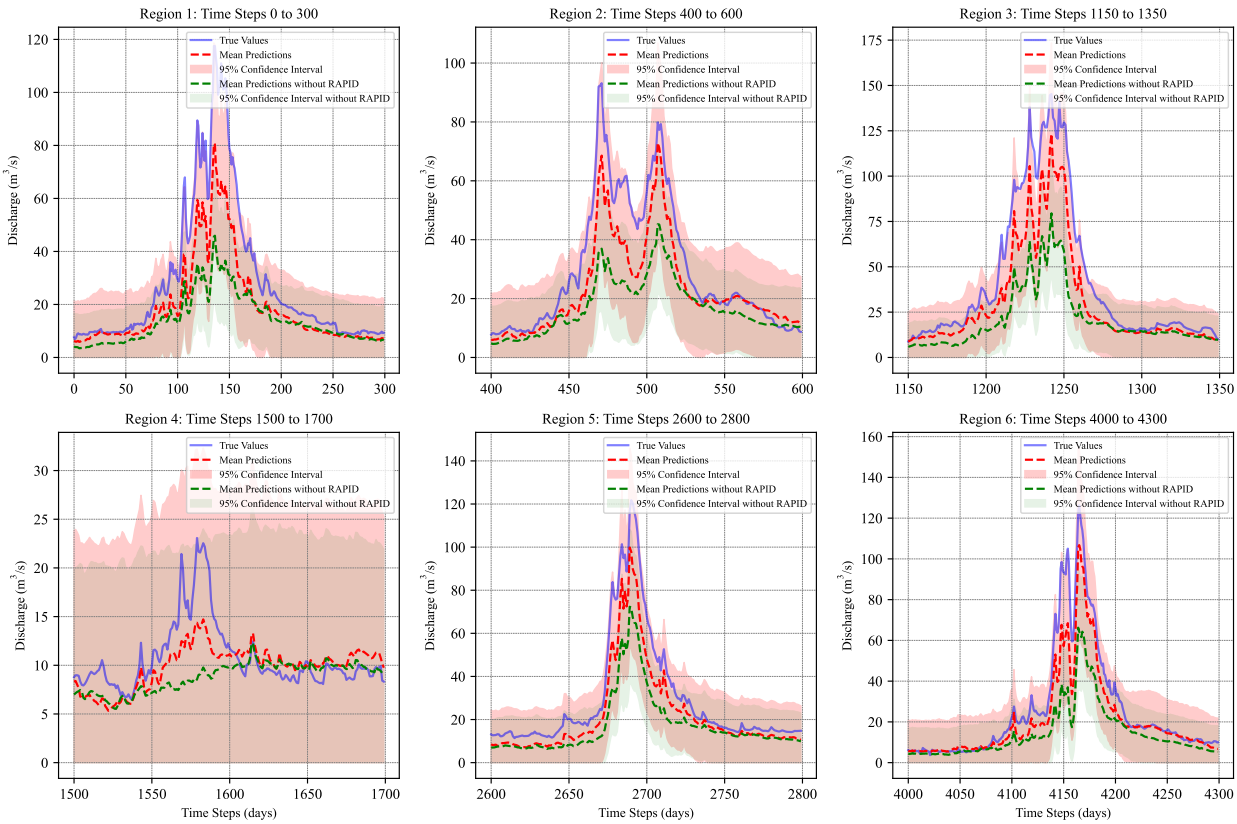


Figure 4.9: Divided time-series prediction plot with 95% confidence interval and true values (SNGP) between purely data-driven and hybrid model.

Figure 4.7 (c) presents the violin plot that illustrates the absolute errors between the models' predictions and observed discharge values using SNGP. Notably, the right plot reveals that the purely data-driven model has a wider error distribution, suggesting less reliability and higher average errors, including more high-value errors. Overall, these findings demonstrate that the hybrid model using SNGP outperforms the purely data-driven model, highlighting the benefits of a hybrid approach for prediction.

4.6.3 Comparison of uncertainty quantification quality using calibration curves

Figure 4.10 (a) shows the calibration curves of MC-dropout for both the hybrid model and the purely data-driven model. The range of each calibration curve reflects the run-to-run variations of MC-dropout, achieved by repeating the algorithm multiple times. The results indicate that the model is conservative in its predictions, and both models exhibit underconfidence in predicting streamflow, meaning the certainty expressed by the models

about their predictions is less than what is justified by their actual accuracy or performance. It is important to note that while this finding highlights that the uncertainty is higher for the purely data-driven model compared to the hybrid model, it does not directly represent the actual prediction accuracy. Figure 4.10 (b) presents the calibration curves of NNE for both the hybrid model and the purely data-driven model. The results indicate that both models exhibit underconfidence in predicting streamflow, meaning the certainty expressed by the models about their predictions is less than what is justified by their actual accuracy or performance. Figure 4.10 (c) depicts the calibration curves of SNGP for both the hybrid model and the purely data-driven model. Different from the previous two UQ methods, the calibration curves are closely aligned with the ideal line, indicating that both models generally make well-calibrated predictions. Specifically, the hybrid model exhibits slight overconfidence, although it remains near the ideal line. In contrast, the purely data-driven model closely follows the ideal line throughout, suggesting accurate calibration between predicted probabilities and observed outcomes.

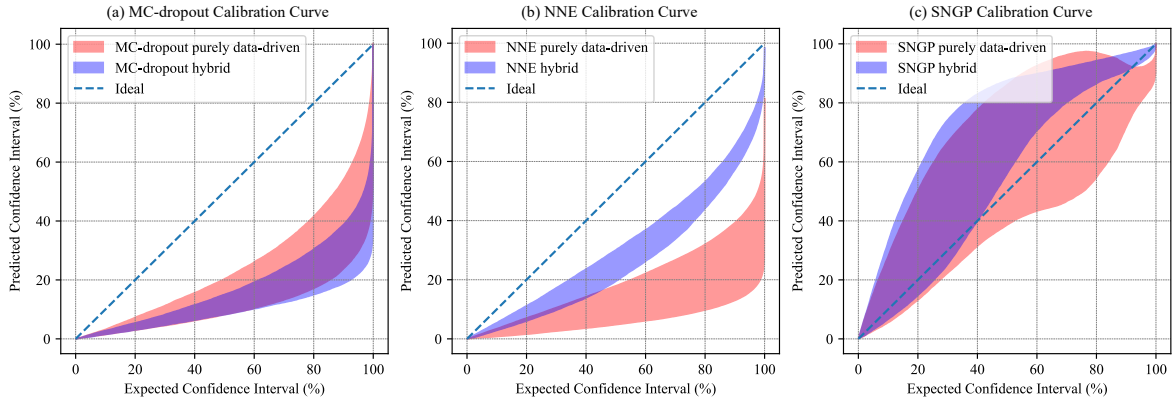


Figure 4.10: Calibration curve comparisons between hybrid model and purely data-driven model using (a) MC-dropout, (b) NNE, and (c) SNGP.

4.6.4 Comparison of computational time

In this subsection, we compare the computational time required by different methods. All the computational costs are based on an Apple M3 Pro GPU, leveraging PyTorch's Metal Performance Shaders (MPS) backend for optimized computation on macOS. The computational costs required by different methods are summarized as follows.

- MC Dropout: Since the model is trained repeatedly for several times in MC dropout, the training process is notably time-intensive. During the training, the model operates

at an average speed of 95.20 iterations per second. With 6597 batches per epoch, each epoch is completed in approximately 69.30 seconds. Each MC-dropout model takes an average of 23.10 hours to train. For prediction, each model requires an average of 27.5 minutes to complete.

- Neural Network Ensemble: Considering all parameters, the model achieves an average training speed of 104.10 iterations per second. With 6597 batches per epoch, each epoch is completed in approximately 63.37 seconds. Each NNE model requires approximately 14.08 hours to train. During the prediction phase, each model takes an average of 33 minutes to predict the streamflow for a single river location.

- SNGP: The SNGP training process is significantly faster compared to the LSTM-based models (i.e., MC Dropout and NNE). Each epoch requires only 6.70 seconds on average, and training a single model takes approximately 1.86 hours. The SNGP model is also notably faster during the prediction phase compared to LSTM-based models. On average, it takes just 33 seconds for each SNGP model to make predictions for a single river location.

4.7 Conclusions

This paper addresses the critical challenge of predicting river discharge in ungauged locations, with the goal of enhancing the accuracy and reliability of streamflow predictions. The absence of gauged data for these rivers presents a significant obstacle. To address the issue of high dimensionality, this study employs SVD to reduce the number of dimensions. However, using a purely SVD-based approach for all variables carries the risk of excluding important hydrological variables from the model. To mitigate this risk, we adopt a two-stage approach that integrates SVD with a feature selection method. In the first stage, SVD is applied exclusively to the runoff data to reduce its dimensionality. Subsequently, a feature selection method is used to identify the most important features for constructing a machine learning model for streamflow prediction. By leveraging physics-based models like RAPID and incorporating carefully selected features along with discharge data from neighboring gauged rivers, this study demonstrates that a hybrid model, built on ML algorithms, can provide accurate predictions with uncertainty bounds in such scenarios.

The primary objectives of this paper are twofold: (1) to develop a hybrid model trained on selected features, runoff data, and physics-based model outputs as inputs, with discharge data from neighboring gauged rivers as outputs. This approach enables the model to capture both the river dynamics described by physics-based models and the nonlinear

relationships between river dynamics and environmental factors. Additionally, training on data from multiple gauged rivers improves the model's ability to generalize across diverse conditions; and (2) to integrate multiple UQ methods that offer probabilistic predictions of discharge in ungauged rivers. Given the complexity of river dynamics and the noise-ridden nature of hydrological data, quantifying uncertainty is crucial for enabling risk-informed decision-making. Our findings highlight the potential of hybrid modeling approaches and UQ methods in tackling the challenges of hydrological predictions, particularly for ungauged rivers. While effective in the presented case studies, the approach depends on historical data and pre-trained models like RAPID, which assume homogeneity and transferability of hydrological behaviors—assumptions that may not hold universally across all regions and climates. Additionally, hybrid modeling and UQ methods can be computationally demanding, with their performance reliant on the quality and availability of hydrological data, which may be sparse or unreliable in certain areas.

Future research directions could focus on expanding the geographic scope of these methods, incorporating additional hydrological and meteorological variables, improving computational efficiency, exploring real-time prediction capabilities, and enhancing data quality through initiatives such as remote sensing techniques. In addition, the proposed method is demonstrated using a catchment in Colorado. Testing the generalization of the proposed method using catchments beyond this study area is worth studying in the future. In conclusion, this study represents a significant advancement in predicting river discharge in ungauged locations. However, addressing current limitations and continuing to refine these advanced modeling techniques will be essential for unlocking their full potential.

CHAPTER 5

A Data-Driven Channel Infilling Rate Forecasting Framework with Quantified Uncertainty for Southwest Pass and Houston Ship Channel

5.1 Introduction

Shoaling in navigational waterways reduces channel depth, and threatens the safe passage of deep-draft vessels. This disruption imposes significant logistical and economic burdens on maritime trade, including increased shipping costs and delays. To maintain navigability, the U.S. Army Corps of Engineers (USACE) maintains an estimated 40,200 km of channels across both coastal and riverine areas [144], investing heavily in maintenance dredging, with nearly a billion dollars contracted annually across the U.S. [145]. Ports and channels require periodic maintenance dredging to ensure minimum draft for safe vessel access [146]. However, shoaling forecasts and dredging decisions often depend on rule-of-thumb practices, short-term stage trend extrapolations, or institutional knowledge gained from years of experience to estimate dredging patterns offering limited lead time (typically 10-14 days) and restricting proactive planning [145, 144].

The accurate forecasting of shoaling remains challenging, due to highly non-linear, noisy, and non-stationary channel infilling processes, influenced by upstream flow, weather, geomorphology, and anthropogenic activities [147, 148, 149, 150]. Traditional models often fail to capture this complexity, leading to inefficiencies in dredging operations [147]. In this context, developing robust, data-informed forecasting models for shoaling is not only a scientific priority but also a practical necessity. Over the last two decades, Artificial Intelligence (AI) and Machine Learning (ML) techniques have become increasingly popular alternatives to traditional empirical or physics-based models. A comprehensive review by

Afan *et al.* outlines the evolution of ML techniques in this domain, emphasizing the superior performance of hybrid and data-driven models [147]. Early contributions include Jain [151], who developed a sediment rating curve using feedforward neural networks, and Nagy *et al.* [152], who demonstrated that multilayer perceptrons (MLP) based on backpropagation could outperform classical formulas as described by Engelund *et al.* [153]. Subsequent studies in Refs. [154], [155], and [156] confirmed the efficacy of artificial neural network (ANN) models in predicting suspended sediment loads across diverse environments. Melesse *et al.* [157] and Kisi [158] further highlighted ANN's superiority over multiple linear regression (MLR), autoregressive models, and conventional sediment rating curves, particularly in complex river systems. Beyond ANN, researchers have explored alternative techniques like Adaptive Neuro-Fuzzy Inference Systems (ANFIS), Support Vector Machines (SVM), and ensemble methods. ANFIS, introduced by Ref. [159], has been shown to more effectively handle uncertainty and non-linearity in hydrological processes [160, 161, 162, 163]. SVM and Least Squares SVM (LSSVM) have demonstrated competitive accuracy in sediment estimation and offered advantages in regularization and generalization [164, 146]. Evolutionary techniques such as Linear Genetic Programming (LGP) and Gene Expression Programming (GEP) have been used by Refs. [165] and [166], outperforming traditional methods in several sediment forecasting scenarios. Preprocessing techniques like wavelet transforms have been widely integrated with ML models to address non-stationarity in time-series data, yielding Wavelet-ANN, Wavelet-ANFIS, and Wavelet-LSSVM hybrids that capture high-frequency variability and improve peak prediction accuracy [167, 168, 169].

In parallel to the aforementioned efforts, a growing body of work has focused specifically on dredging-related applications and shoaling prediction. Bhattacharya and Solomatine [148] used MLPs and M5 model trees to predict sedimentation in the Port of Rotterdam by incorporating discharge, wave energy, and sediment potential. Chou and Lin [149] developed a novel risk-informed prediction model for dredging project duration in Taiwan using support vector regression (SVR) and Monte Carlo simulations, making it the first such effort in that region. In ports and coastal systems, Mateo-Perez *et al.* [146] used SVM and Random Forests to derive satellite-based bathymetry in highly anthropized environments, providing rapid and cost-effective alternatives to sonar surveys. Graw *et al.* [170] applied Random Forest Regressors for the first high-resolution global prediction of seafloor sediment density, incorporating uncertainty estimates to support real-world use. Dunkin *et al.* [171] showed how combining AIS-based vessel tracking with hydrographic surveys can reveal shoaling "hot spots" and limiting depths, improving navigational safety and planning. Yeates *et al.* [145] linked inland streamflow to dredging requirements using the USACE's Streamflow Prediction Tool (SPT), and found that high flows could sometimes reduce shoaling by scouring

sediment, suggesting that upstream hydrology is a critical—but often underused—predictor of dredging needs. Building on these advances, Asborno *et al.* [144] recently proposed a multivariate, multistep ML framework for shoaling prediction in the Southwest Pass (SWP) of the Mississippi River. Their use of Long Short-Term Memory (LSTM) networks, combined with physics-informed variables and Extreme Gradient Boosting (XGB) for feature selection, achieved significantly better 45-day forecasts than autoregressive integrated moving average (ARIMA) and univariate baselines. Although ML has shown promise for dredging-related applications, substantial challenges remain. For example, Asborno *et al.* [144] found that ML models can capture the direction of future shoaling (i.e., increasing or decreasing) but struggle to accurately predict the actual shoaling volume. Moreover, model performance can depend strongly on difficult-to-forecast variables such as turbidity, which limits the practical application of many proposed approaches. To enable a reliable real-world application of ML for dredging decision support, models should be built using input that can be predicted operationally, such as river discharge, rather than variables that are inherently unpredictable in practice.

Motivated by the limitations of existing data-driven approaches to shoaling forecasting, we propose a new framework that departs from the conventional paradigm by predicting the channel infill rate (CIR) rather than forecasting sedimentation volume directly. The CIR is a critical quantity in the U.S. Army Corps of Engineers' Corps Shoaling Analysis Tool (CSAT), which projects sediment volume linearly in time using the most recently computed CIR. However, CSAT does not explicitly forecast CIR; instead, it treats CIR as an input estimated from historical shoaling records (CSAT is described in more detail later). By modeling CIR as a process that varies over time, the proposed approach captures the temporal dynamics of sedimentation, improves sensitivity to short-term trends, and reduces the influence of abrupt changes introduced by external interventions. This perspective is analogous to the challenges in structural dynamics, such as modeling crack-growth [172] and shape reconstruction [173], where forecasting physically meaningful rates (e.g., crack propagation rate or strain) and then integrating them produces accumulated quantities such as crack length [172] or structural deformation [173]. In addition, we investigate modeling strategies tailored to different levels of shoaling-survey availability. We develop and compare multiple data-driven CIR models to evaluate performance across scenarios with limited versus abundant survey data. Because CIR predictions are subject to uncertainty arising from measurement noise and limited training data, we also incorporate uncertainty quantification to provide confidence intervals for shoaling forecasts. The framework is demonstrated in two key U.S. navigation channels: Southwest Pass (SWP), with dense bathymetric survey coverage, and the Houston Ship Channel (HSC), where bathymetric surveys are comparatively

sparse. The main contributions of this paper can be summarized as follows:

- Introduction of a CIR-based shoaling predictive formulation that is more robust to external disruptions.
- Implementation of a data-cleaning protocol to exclude outliers from human or extreme natural events.
- Tailoring the modeling strategies to sparse and dense data regions.
- Comparison of different data-driven CIR modeling methods including linear regression, Long Short-Term Memory (LSTM), Convolutional Neural Network (CNN)–LSTM, and Temporal Convolutional Network (TCN)–LSTM.
- Uncertainty quantification of the shoaling forecasting.
- Practical applications of the developed framework to two important navigation channels in the U.S.

The remainder of the paper is organized as follows: Section 5.2 presents the background on shoaling, the two study areas of interest (i.e., SWP and HSC), and available data sources for shoaling forecasting. Section 5.3 defines the problem statement, processes the CSAT data to obtain CIR, explains the input and output variables and the data processing techniques used, and outlines the solution methodology. Section 5.4 and Section 5.4.2 discuss the results and key findings in respectively the HSC region and the SWP region, while Section 5.5 concludes the paper with a summary of limitations and potential directions for future work.

5.2 Background

5.2.1 The process of shoaling

In the context of river channels and hydrology, shoaling is the process by which sediment, such as sand, silt and gravel, accumulates on a riverbed, resulting in a decrease in water depth and the formation of submerged or exposed landforms called shoals. Navigable channels often necessitate dredging to remove shoals and maintain navigable depths. The term shoaling encompasses both vertical and lateral sedimentation processes [174]. *Bed infilling*, or deposition, occurs when sediments settle at the bottom of a channel, directly decreasing the available navigable depth. *Lateral sediment encroachment*, by contrast, involves features such as spits, shoals, or sand waves migrating from the banks into the channel, thereby

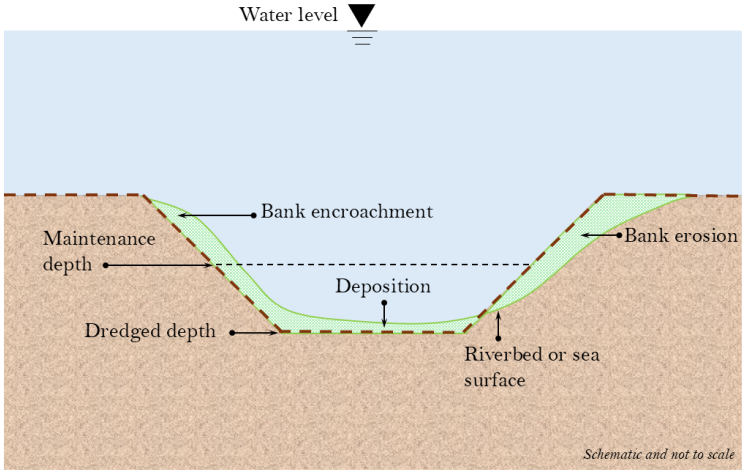


Figure 5.1: Schematic diagram showing progressive Morphological Changes in Channel Geometry

reducing its effective width. Figure 5.1 schematically illustrates the effect on channel's cross-section due to shoaling.

Shoaling is driven by a complex interplay of sediment transport dynamics, hydro-meteorological forces, sediment properties, geological features, and human activities. At its core, it involves the nonlinear interaction between water flow and sediment particles, from detachment to deposition. Density currents from freshwater discharges and salt-wedge intrusions in estuaries can shift sediments upstream, while seasonal stratification can trap them in deeper channels [148, 150]. Wind, waves, and currents further influence sediment movement, with reduced river discharge and colder temperatures favoring deposition. Sediment traits, like size, density, and concentration, affect settling behavior, while biological activity can enhance or stabilize deposits. Geographical features and anthropogenic factors such as land use changes and dredging practices also shape sediment delivery and accumulation. Since the primary source of sediment transport is the upstream reaches that feed into the downstream channel, the upstream river discharge data contains crucial information about shoaling in the downstream reaches [145].

5.2.2 Study sites: HSC and SWP

5.2.2.1 The Houston Ship Channel (HSC)

The HSC, as shown in Fig. 5.2, is a navigable waterway located in the state of Texas and the USACE Galveston District (SWG). It extends 53 miles upstream from Galveston Bay to

the Turning Basin in Houston, facilitating the movement of vessels from the Gulf of Mexico to the city. In 2005, the SWG undertook an expansion project for the HSC, resulting in varying authorized channel dimensions along its length: (1) a depth of 45 ft and a width of 530 ft for the segment extending from Bolivar Roads to Boggy Bayou, (2) a depth of 40 ft and a width of 300 ft for the segment extending between Boggy Bayou and Sims Bayou, and (3) a depth of 36 ft and a width of 300 ft for the third segment extending between Sims Bayou and the Main Turning Basin [175]. The upstream gauge stations and their corresponding IDs are also shown in this figure. These stations serve as the input variables for the GSA described in Section 5.3.3.

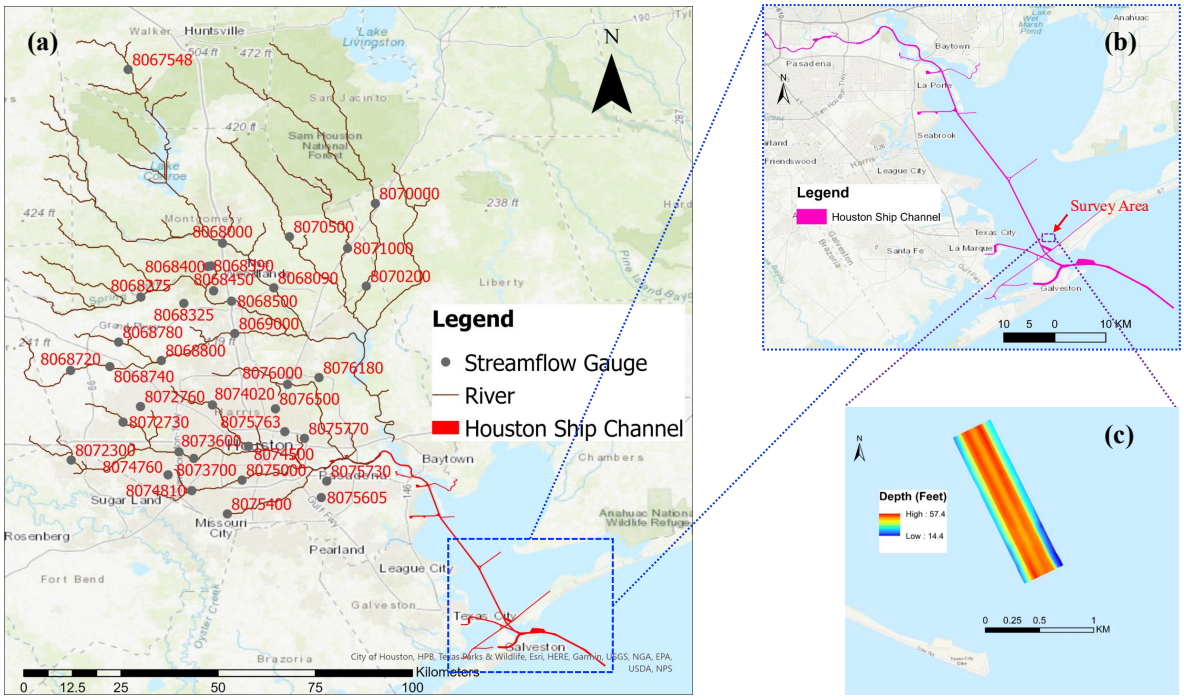


Figure 5.2: Houston Shipping Channel (HSC). (a) Buffalo Bayou and San Jacinto River Basins showing data collection stations considered for sensitivity analysis to select inputs for building the forecasting model for the HSC region. (b) Zoomed-in view of the HSC. (c) Bathymetric survey for a portion of channel in HSC.

Based on USACE hydrographic surveys (*eHydro* [176]), the HSC is divided into 12 main sections, also known as reaches. In addition to being a deep-draft navigation channel, the HSC is home to the Port of Houston—one of the busiest ports in the United States in terms of total tonnage. It also connects the nation's largest petrochemical complex to global markets. These attributes make it the country's leading waterway by waterborne tonnage, supporting 3.37 million jobs nationwide and generating $906 billion in economic impact across the United States [177].

Maintaining the HSC's safety, efficiency, and navigability, especially given the high vessel traffic, is therefore critical. This is achieved through ongoing improvement and maintenance strategies, including new work dredging for channel expansions and maintenance dredging to address shoaling hotspots. The HSC remains an active expansion project, which aims to widen certain segments from 530 to 700 ft and deepen upstream segments to 46.5 ft [178, 179].

5.2.2.2 The Southwest Pass (SWP)

Located at the USACE New Orleans District (MVN), the SWP, as shown in Fig. 5.3, is a critical navigable channel in the United States, located in the state of Louisiana. It extends from the mouth of the Mississippi River to the Gulf, serving as the primary gateway for maritime traffic traveling between the Gulf and the interior states [180], particularly where the Port of New Orleans and other key ports are located. Similar to the HSC region, the upstream gauge stations and their corresponding IDs are also shown in this figure. These stations serve as the input variables for the GSA described in Section 5.3.3.

The SWP currently maintains a navigable depth of 50 ft, with a recommended draft limit of 45 ft [181]. Its width ranges from 600 to 750 ft along a stretch of approximately 35 miles (Hartman *et al.* 2022), divided into 13 reaches. These characteristics make the SWP the nation's premier waterway for deep-draft vessels, with over 250 million tons of cargo passing through annually—transported by an average of 13 such vessels per day [182]. Given its critical role, any disruption to the route of these oceangoing vessels can result in significant economic losses. As such, the SWP is a top priority for the USACE and MVN, particularly in terms of maintaining full navigability through timely dredging operations.

5.2.3 Data sources and existing tools

5.2.3.1 Bathymetric surveys and eHydro tool

Bathymetric surveys, which measure channel depth to support navigation, engineering, and sediment management, and is hosted on a centralized data platform known as *eHydro* [183, 176]. *eHydro* is an enterprise-level tool developed by USACE to process, store, and standardize hydrographic survey data nationwide. It organizes bathymetric data by individual channel reaches as defined by the National Channel Framework (NCF)—an enterprise Geographic Information System (eGIS) database that documents congressionally authorized navigation channels [184]. *eHydro* compiles the survey data into structured files with uniform spatial grids, maintaining consistent X and Y coordinates while capturing temporal changes in bathymetry.

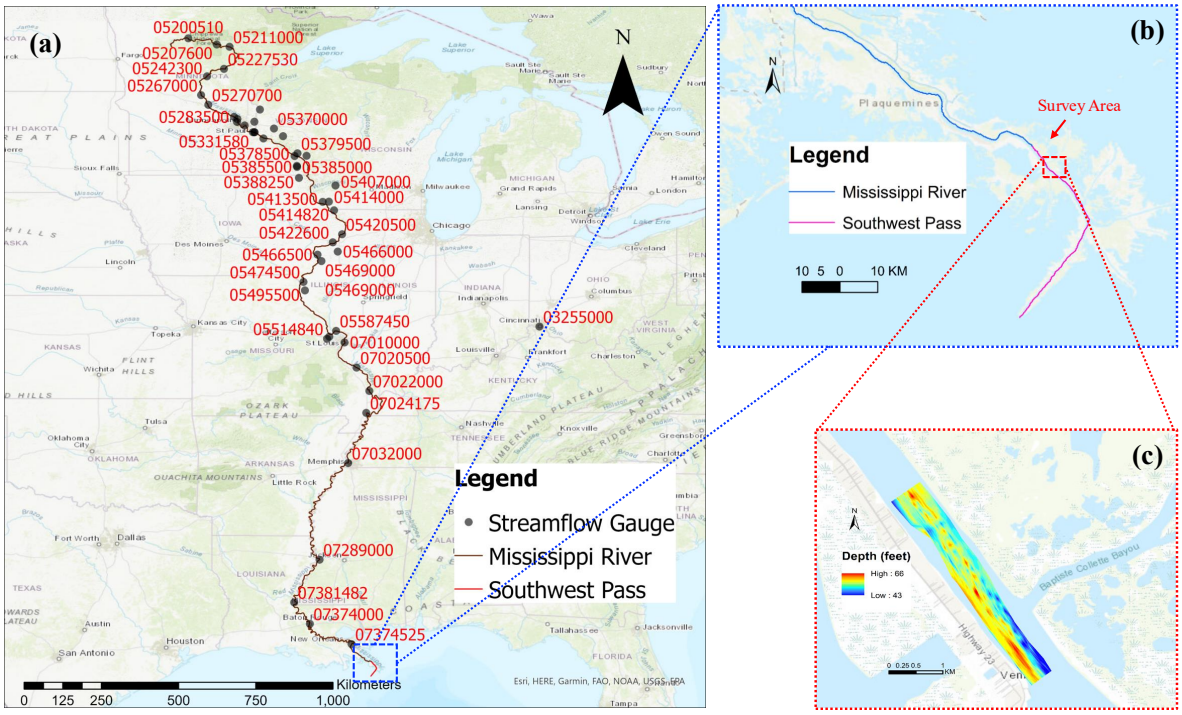


Figure 5.3: The Southwest Pass (SWP). (a) Mississippi River and its tributaries showing discharge and the locations of data collection stations considered for sensitivity analysis to select inputs for building the forecasting model for the SWP region. (b) Zoomed-in view of SWP. (c) Bathymetric survey for a portion of channel in SWP.

Bathymetric surveys [185, 186] capture underwater terrain by measuring the horizontal position (X-Y) and the bottom elevation using mechanical or acoustic methods, with modern systems offering high precision. Depths are often measured using single-beam echo sounders, such as the Navisound 210, which provide centimeter-level vertical precision across a range of acoustic frequencies [146]. These measurements are georeferenced using Differential GPS and corrected for tidal variations to ensure accuracy [146, 186]. The resulting point data are interpolated into digital terrain models of the channel bed, typically referenced to standard projection systems like UTM/WGS84 [146]. Figures 5.2(c) and 5.3(c) illustrate respectively an instance of a Bathymetric survey for a portion of the channel in the HSC and SWP region. Survey accuracy is influenced by sea state, instrument calibration, and vessel motion, with errors potentially arising from random, systematic, or human sources. These are managed through rigorous calibration and quality control to limit such errors. The data are typically processed into Triangulated Irregular Networks (TINs) or gridded models, with TINs preferred when data are sparse. Interpolation and terrain irregularities introduce uncertainty, which is minimized through repeated measurements and surface comparisons.

5.2.3.2 Corps Shoaling Analysis Tool (CSAT)

The CSAT calculates channel shoaling volumes using historical channel surveys (available through *eHydro* [176]) and provides a first-order prediction of future dredging volumes [187]. The tool consists of two parts. The first is the *CSAT input*, which contains pre-processed and cleaned bathymetric data from eHydro, stored in NetCDF format and includes details from historical surveys. Since these surveys span wide geographic areas, the region is divided into grids referred to as bathymetric survey points (typically 10-by-10 ft), with each point representing the center of a grid cell and identified by its *latitude* and *longitude*. Each survey, identified by a unique *survey ID*, provides elevation data measured as positive downward. In essence, the elevation value is treated as a function of the *survey ID*, *survey time*, and the *latitude* and *longitude* of each bathymetric survey point (see Fig. 10 in [188]).

The second part of the tool uses the provided Python code [187] to process the *CSAT input* and generate various *CSAT output* files, including: (a) a table of average, minimum, and maximum CIR for each bathymetric survey point; (b) a table of volume differences between surveys by reach; and (c) a table of forecast dredging volumes by reach, which includes projected volumes for various cut depths at 6-month intervals, among others as listed in Section 5.4 of Ref. [188]. For a fixed cut depth, the forecast is provided at 6-month intervals, covering periods from 6 to 60 months. However, these forecasts are first-order, piecewise linear projections informed solely by past sedimentation volume change data. They are intended to indicate long-term sedimentation trends but lack more informed short-term

projections (e.g., 30 to 45 days ahead, which this paper aims to provide), which are more useful for immediate decision-making.

The *CSAT output* in its current form [187] does not provide historical time series of sedimentation volume. Therefore, it is not the *CSAT output* that interests us, but rather the *CSAT input*, which contains data from historical bathymetric surveys derived from *eHydro*. The *CSAT input* (in the ascending time order) is utilized to derive the historical sedimentation volume time series at the reach level.

5.2.3.3 Survey frequencies in HSC and SWP

When comparing the number of conducted hydrographic surveys between the HSC and the SWP—based on a portion of the *CSAT input* database, which compiles all hydrographic surveys for each reach into a single file [189]—the SWP shows a significantly higher number of surveys per reach across different years. From 2012 to 2024, the SWP recorded 5.65 times more surveys than the HSC, with a total of 21,208 survey entries compared to just 3,751 for HSC. While district-specific survey processing practices (MVN for SWP and SWG for HSC) could introduce minor procedural differences, a comparison of representative surveys from both regions showed that the spatial coverage of individual surveys evaluated using reach length, shape area, and shape length, does not systematically vary with survey frequency. In the SWP, reach lengths are relatively uniform (approximately 2.7–3 miles), whereas in the HSC they range more widely (0.3–11.3 miles). Although HSC reaches vary widely in size, as reflected in their corresponding survey shape area and length, this variability does not correlate with the number of surveys conducted. Moreover, even for smaller HSC reaches, whose survey shape area and length are less than those of SWP reaches, the number of conducted surveys remains significantly lower than in the SWP. Thus, the disparity primarily reflects the SWP's inherently dynamic morphology and rapid shoaling behavior, which compel the MVN to perform near-daily hydrographic surveys to monitor and maintain navigable conditions [182]. Figure 5.4 illustrates the number of annual hydrographic survey entries for the SWP and the HSC from 2012 to 2024. The bar chart (left y-axis) compares the total number of surveys conducted in each region per year, while the overlaid line plot (right y-axis) depicts the ratio of SWP to HSC survey counts. This visualization highlights the consistently higher frequency of surveys in the SWP, underscoring the channel's more aggressive shoaling behavior and the increased monitoring efforts required to maintain navigability.

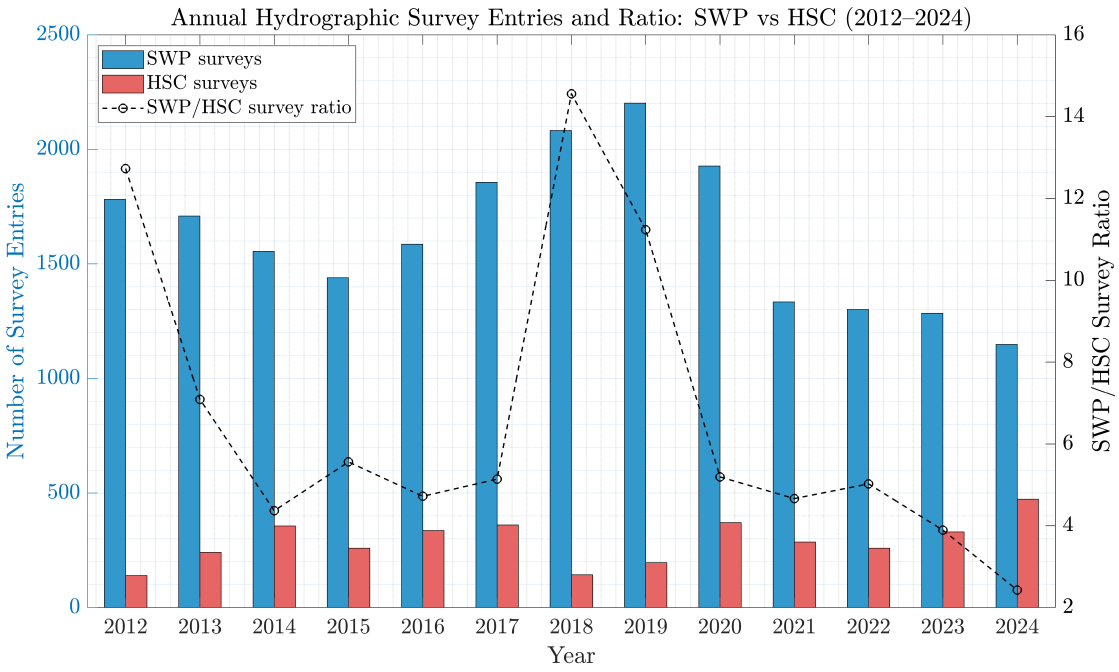


Figure 5.4: Annual hydrographic survey entries for the Southwest Pass (SWP) and Houston Ship Channel (HSC) from 2012 to 2024.

5.2.3.4 Streamflow gauges near HSC and SWP

Previous studies by [145] and [144] highlight the critical role of upstream streamflow measurements in influencing shoaling. In the United States, streamflow measurements are primarily sourced from the National Water Information System (NWIS) maintained by the U.S. Geological Survey (USGS) (https://waterdata.usgs.gov/nwis). Supplementary, shorter-duration data are sometimes provided by local state or municipal agencies, though these are generally considered less reliable than the NWIS records [190, 96]. We considered 36 gauge stations in the HSC (Buffalo Bayou and San Jacinto River) basin and 43 gauge stations in the SWP (Mississippi River) basin for sensitivity analysis. Figures 5.2(a) and 5.3(a) illustrate respectively the gauge stations along with the corresponding station IDs for the HSC and SWP regions.

It is worth noting that, although other predictors, such as turbidity, have been used to forecast shoaling in prior work [144] (see Sec. 6.1), we focus primarily on streamflow as the main driver of shoaling. Streamflow can be forecast using physics-based simulations, and river-discharge prediction remains an active area of research in hydrology. In contrast, variables such as turbidity are generally not directly predictable with physics-based models, which limits their practical utility for operational shoaling forecasting.

5.2.4 Challenges and needs of data-driven models for shoaling forecasting

Shoaling forecasting remains difficult to implement in practice for several reasons, summarized below.

- First, physics-based sediment transport models, although theoretically capable of simulating sediment dynamics, are impractical for large-scale and long-term operational forecasting in complex systems, such as the HSC and SWP shown in Figs. 5.2 and 5.3. Process-based numerical models have also been widely used to investigate sediment transport mechanisms and system-scale sediment dynamics in complex coastal and estuarine environments (e.g., [191, 192, 193]). These models require extensive bathymetric detail, boundary conditions, and computational resources, making them unsuitable for routine forecasting at the spatial and temporal scales required by navigation authorities..

- Second, while data-driven models offer a promising alternative for shoaling forecasting, training them has proven challenging because the data are affected by various sources of noise. For example, dredging interventions introduce abrupt, extreme jumps in the CSAT data, obscuring the underlying natural shoaling process. Consequently, using the raw CSAT data directly for model training can lead to substantial prediction errors.

- Third, bathymetric survey frequency varies by region. In areas such as the SWP, surveys are conducted frequently, producing a densely sampled sedimentation-volume time series that captures finer temporal variability. In contrast, the HSC has a much sparser survey record, yielding a sedimentation-volume curve that is effectively a coarse, piecewise-linear approximation. These differences require distinct data-driven modeling strategies; no universal approach is suitable for all scenarios.

- Fourth, although upstream river discharge is a critical predictor for shoaling forecasting (Figs. 5.2(a) and 5.3(a)), discharge data are available from a large number of gauging stations. This high-dimensional input space complicates training of data-driven shoaling models. Moreover, more features are not necessarily better: if some predictors are weakly related to the quantity of interest, including them can degrade model performance rather than improve it. Therefore, it is important to identify the most relevant stations among the many upstream discharge gauges.

- Fifth, while CSAT provides valuable insights into past shoaling patterns, it does not

offer predictive capabilities and therefore cannot support forward-looking decision-making for navigation management [144, 145].

In the following section, we propose a systematic shoaling forecasting framework that complements existing survey-based tools and enables data-driven prediction of shoaling volume. Unlike approaches that focus only on trend, the proposed framework forecasts the actual shoaling-volume values.

5.3 Methodology

5.3.1 Overview

Fig. 5.5 provides an overview of the proposed framework. As shown in this figure, it consists of three main stages.

- The first stage focuses on data collection for both target variable (i.e., raw NetCDF files from the eHyrdo database [183, 176]) and the explanatory inputs (i.e., streamflow data from the upstream of the region of interest).
- The second stage centers on data preprocessing, which converts the raw datasets obtained from the first stage into physically meaningful and model-ready inputs. For the target variable, sedimentation volume is first computed using the CSAT tool, followed by data cleaning and transformation into the CIR. On the input side, gauge data go through cleaning and quality control before being analyzed to select the most important features. This step identifies the most influential predictors, allowing the modeling effort to focus on a reduced and physically interpretable set of input variables.
- The final stage develops data-driven shoaling forecast models using the processed CIR data and selected input features. Modeling strategies are tailored to the survey frequency in the region of interest. Prediction uncertainty is also quantified to produce confidence intervals for the shoaling forecasts.

5.3.2 Data Preprocessing

Before analysis, the *CSAT input* data must be preprocessed. As noted above, this paper forecasts the CIR rather than sedimentation volume directly, because CIR captures the temporal dynamics of sedimentation and helps distinguish gradual, natural shoaling from abrupt shifts induced by external interventions. Next, we describe the preprocessing steps for the CSAT input and the computation of the CIR after that.

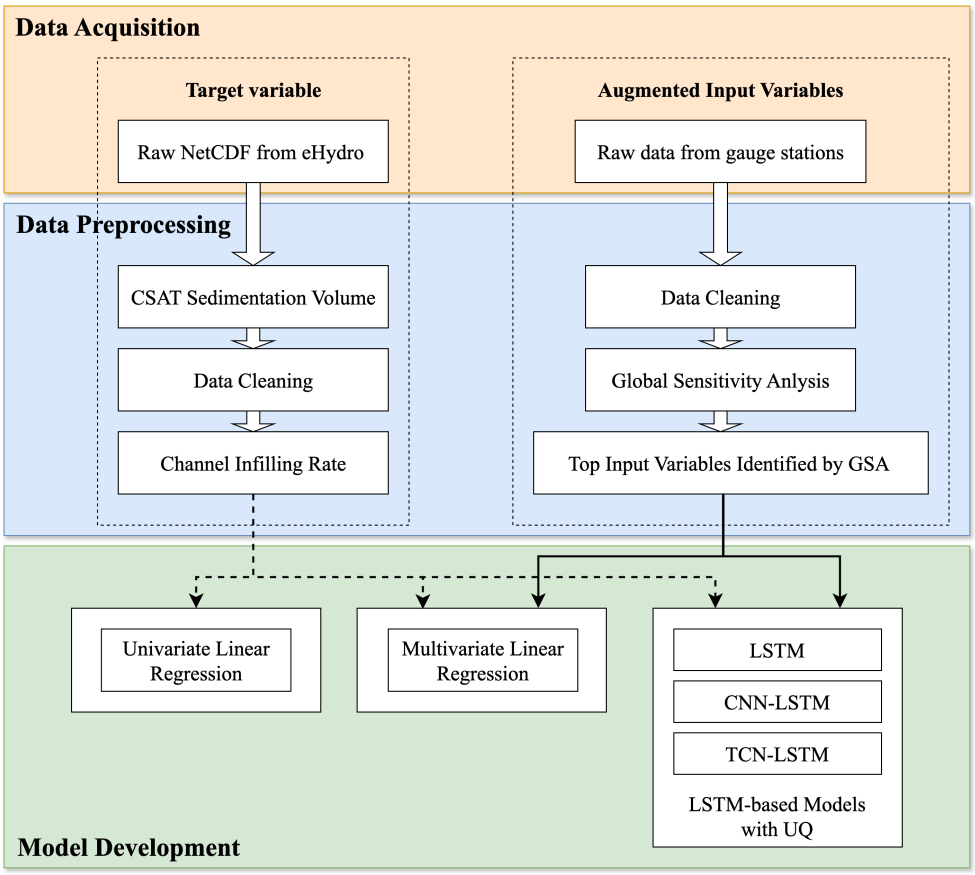


Figure 5.5: Proposed framework of the solution approaches.

5.3.2.1 Computation of Channel Infilling Rate

There are several data issues that need to be addressed before the computation of the CIR. (a) For a given reach, bathymetric surveys typically focus on the channel; grid cells outside the channel are often unsurveyed and are therefore recorded as either a very large value (e.g., $\sim 10^{38}$) or *NaN*. (b) Multiple surveys (distinguished by survey ID) may be conducted on the same day. (c) Surveys are irregularly spaced in time and can be sparse. Notably, CSAT applies region-specific, preprogrammed datum corrections.

These issues present both a *spatial* (item *(a)*) and a *temporal* (items *(b)* and *(c)*) challenge. To address the spatial issue, we first replace all extremely large values (e.g., $\sim 10^{38}$) with *nan*. For the temporal issue, we apply linear interpolation over the *nan* values to estimate reasonable elevation values for the grids that are part of the channel but were not surveyed in certain instances. Grids that have *nan* values across all survey IDs are assumed to be outside the channel and are excluded from volume calculations. When there are multiple surveys for the same day (item *(c)*), we combine them by computing the mean elevation

value (excluding *nan*) for each grid. Finally, the elevation values are *linearly* interpolated over time, yielding continuous elevation data at a daily frequency (which can be adjusted to weekly or monthly if desired) for each grid. This process is schematically illustrated in the flowchart below (Figure 5.6).

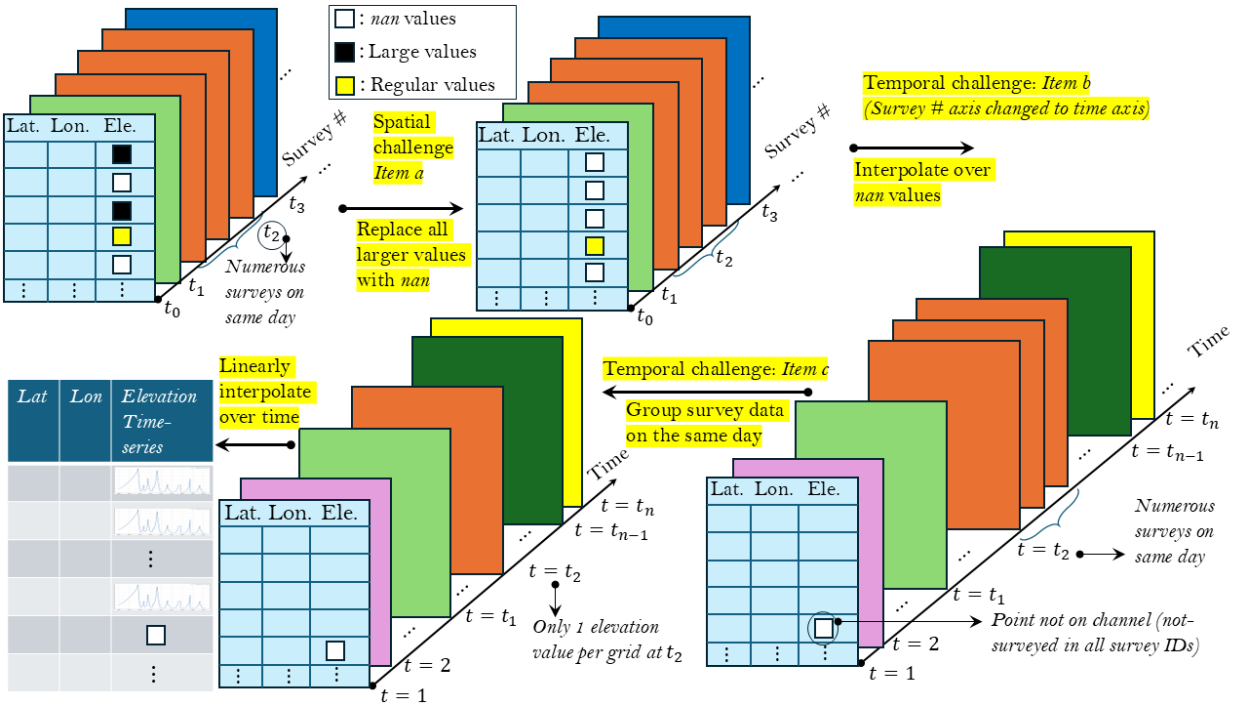


Figure 5.6: Pre-processing of historic bathymetry survey data extracted from *CSAT input*

The sedimentation volume of the reach at time t is obtained by first computing the volume associated with each grid and then summing the volumes across all grids within the reach to obtain the total volume. Let $v_i(t)$ denote the volume of the grid i at time t, and let $d_{\mathrm{e}}(t)$ represent the elevation of the sea surface (obtained from *CSAT input*) and d_{m} the depth of maintenance. Both quantities are assumed to be positive in the downward direction, measured in feet from a fixed datum. For a grid size of 10-by-10 ft, the volume $v_i(t)$ is given by

$$v_i(t) = \begin{cases} 0, & \text{if } d_{\mathrm{e}}(t) - d_{\mathrm{m}} > 0 \\ \frac{(d_{\mathrm{e}}(t) - d_{\mathrm{m}}) \times 10 \times 10}{27} \mathrm{yard}^3, & \text{if } d_{\mathrm{e}}(t) - d_{\mathrm{m}} \leq 0. \end{cases} \tag{5.1}$$

The two scenarios presented in Eq. (5.1), one corresponding to the channel bed being below the maintenance depth (i.e., $d_{\mathrm{e}}(t) - d_{\mathrm{m}} > 0$), and the other where the channel bed is above the maintenance depth (i.e., $d_{\mathrm{e}}(t) - d_{\mathrm{m}} \leq 0$), are schematically illustrated in Fig. 5.7.

The volume at the reach level, denoted by reach r, at time t, represented as $V_{\mathrm{raw}_r}(t)$, is computed by summing the contributions from all N_{grids} individual grids, as defined in

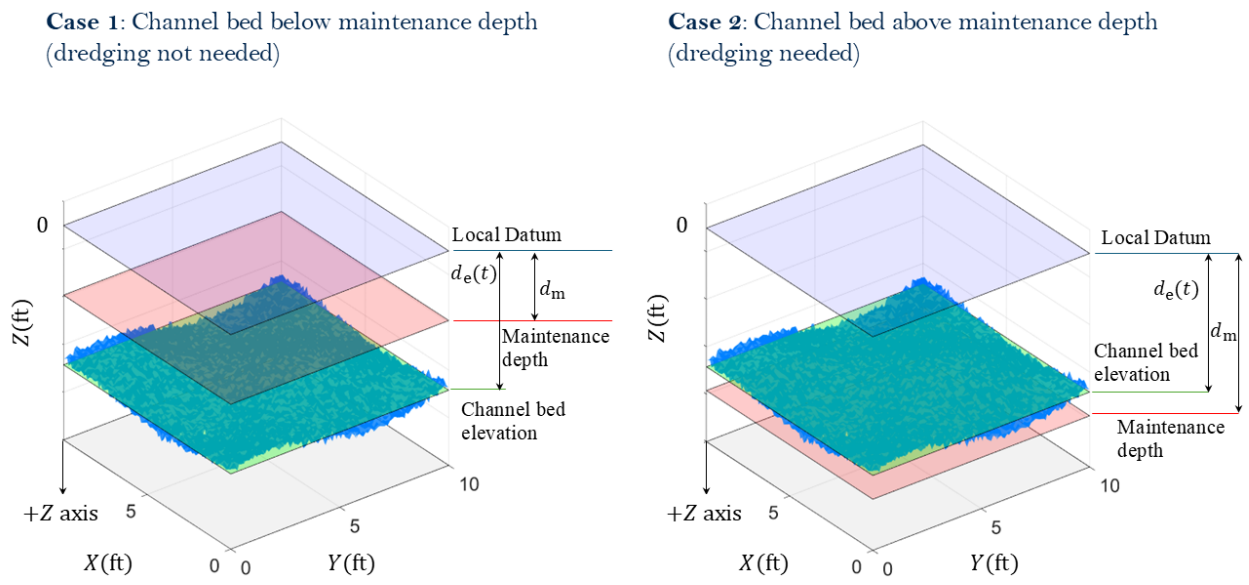


Figure 5.7: Calculating the sediment volume for a fixed survey grid and time instance

Eq. (5.2):

$$V_{\text{raw}_r}(t) = \sum_{i=1}^{N_{\text{grids}}} v_i(t). \tag{5.2}$$

The raw volume $V_{\text{raw}_r}(t)$ is often contaminated by short-term, high-frequency fluctuations resulting from measurement noise and interpolation artifacts in survey data. To mitigate this, we applied the *Savitzky-Golay filter* (SGF) with a 30-day window size to obtain a smoothed volume time series at the reach level. Let $V_r(t)$ denote the smoothed sedimentation volume for reach r. Mathematically, this can be expressed as:

$$V_r(t) = \text{SGF}(V_{\text{raw}_r}(t), 30). \tag{5.3}$$

The *SGF* is a digital smoothing technique that fits a low-degree polynomial (typically of degree 2 or 3) to a moving window of consecutive data points using least squares. Unlike simple moving averages, it preserves the shape, local trends, and features—such as peaks and inflection points—of the data while reducing noise. This makes it particularly suitable for our application, where preserving the underlying trend in sedimentation volume is crucial for reliable CIR computation and downstream modeling. The 30-day window was chosen to strike a balance between removing short-term noise and retaining meaningful temporal

variations in sedimentation dynamics.

Occasionally, it becomes important to consider combinations of reaches. Let $\mathcal{C}$ denote the set of reaches to be combined or aggregated, consisting of N_{reaches} reaches. The total sedimentation volume for the combined set can be obtained by summing the volumes of the individual reaches.

$$V_{\mathcal{C}}(t) = \sum_{r=1}^{N_{\text{reaches}}} V_r(t). \tag{5.4}$$

The CIR, computed using a lag of Δ days at the reach level (denoted by $R_{\Delta,r}(t)$) and at the aggregate level (denoted by $R_{\Delta,\mathcal{C}}(t)$), is given by:

$$R_{\Delta,r}(t) = \frac{V_r(t) - V_r(t-\Delta)}{\Delta}, \tag{5.5a}$$

$$R_{\Delta,\mathcal{C}}(t) = \frac{V_{\mathcal{C}}(t) - V_{\mathcal{C}}(t-\Delta)}{\Delta}. \tag{5.5b}$$

5.3.2.2 Identification of extreme jumps caused by non-natural human activities

We used an *Interquartile Range* (IQR)–based approach to filter abrupt shifts induced by external non-natural interventions, since the goal of this paper is to model the natural shoaling process. The procedure consists of the following four steps:

1. *Compute the quartiles:* Obtain the 25-th and 75-th percentile values of the CIR time series (either $R_{\Delta,r}(t)$ or $R_{\Delta,\mathcal{C}}(t)$), denoted as Q_{25} and Q_{75}, respectively.

2. *Compute the interquartile range (IQR):* The IQR is defined as the difference between the 75th and 25th percentiles:

$$\text{IQR} = Q_{75} - Q_{25} \tag{5.6}$$

3. *Determine the outlier bounds:* Compute the lower and upper cutoff thresholds, denoted as Q_{lower} and Q_{upper}, using the following expressions:

$$Q_{\text{lower}} = Q_{25} - q\,\text{IQR} \tag{5.7a}$$

$$Q_{\text{upper}} = Q_{75} + q\,\text{IQR} \tag{5.7b}$$

   Here, q is the outlier threshold chosen by the modeler. A smaller value of q results in a more sensitive outlier filter.

4. *Identify the extreme events:* Values of the CIR time series that are either above Q_{upper} (e.g., due to sudden volume addition from extreme rainfall) or below Q_{lower} (e.g., due to dredging) are identified as extreme values, and the corresponding time instances are omitted for the purpose of training. Let N_{time} denote the total number of time instances in the CIR data after removing (i.e., not counting) the outliers.

Figures 5.8 and 5.9 illustrate the IQR-based filter applied to the aggregated CIR, denoted by $R_{\Delta,\mathcal{C}}$, using $\Delta = 1$ days. We set $q = 2$ in this case study. This outlier threshold value was chosen because it provides a good balance between sensitivity and robustness in filtering abnormal data points. Several threshold values were tested (e.g., $q = 1, 1.5, 2, 2.5, \ldots, 10$), and it was found that $q = 2$ effectively eliminated most abnormal trends, such as sudden artificial drops due to dredging or spikes from extreme events, while preserving the meaningful variability in sediment dynamics. For reproducibility purpose, the corresponding quartile statistics are reported as follows:

- **HSC**: $Q_{\text{lower}} = -28,854.70$ yd^3/day, $Q_{\text{upper}} = 30,946.55$ yd^3/day for $q = 2$.
- **SWP**: $Q_{\text{lower}} = -152,820.23$ yd^3/day, $Q_{\text{upper}} = 159,095.30$ yd^3/day for $q = 2$.

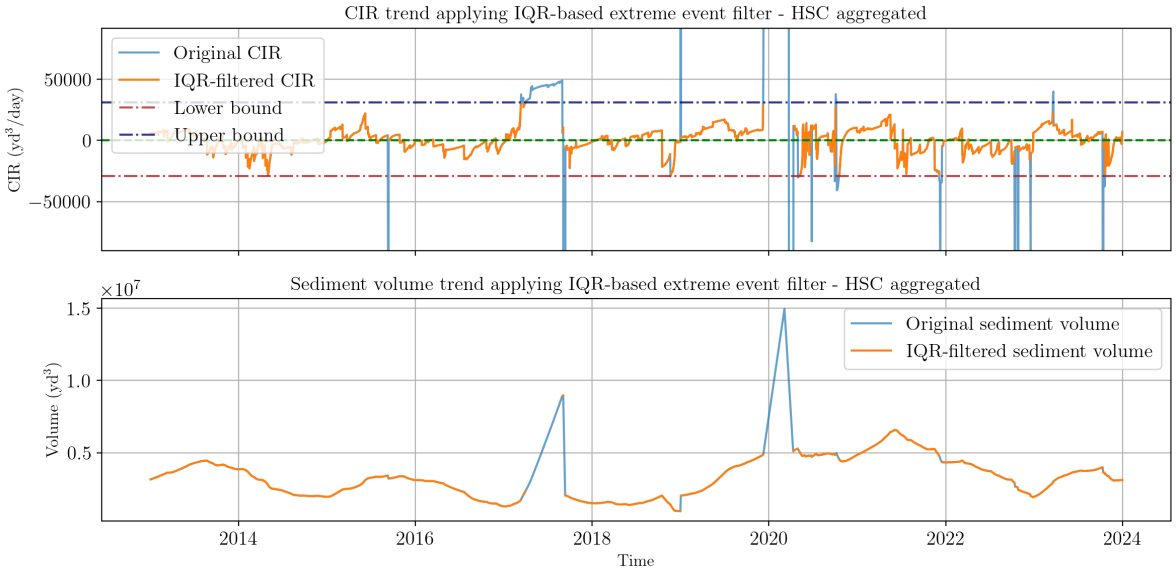


Figure 5.8: IQR-based extreme event filter for HSC region considering aggregated sedimentation data

Since the survey data are linearly interpolated, the ability of the sedimentation volume time series to capture the true volume evolution for a given reach depends heavily on the density of available surveys. When surveys are sparse, the resulting sedimentation volume

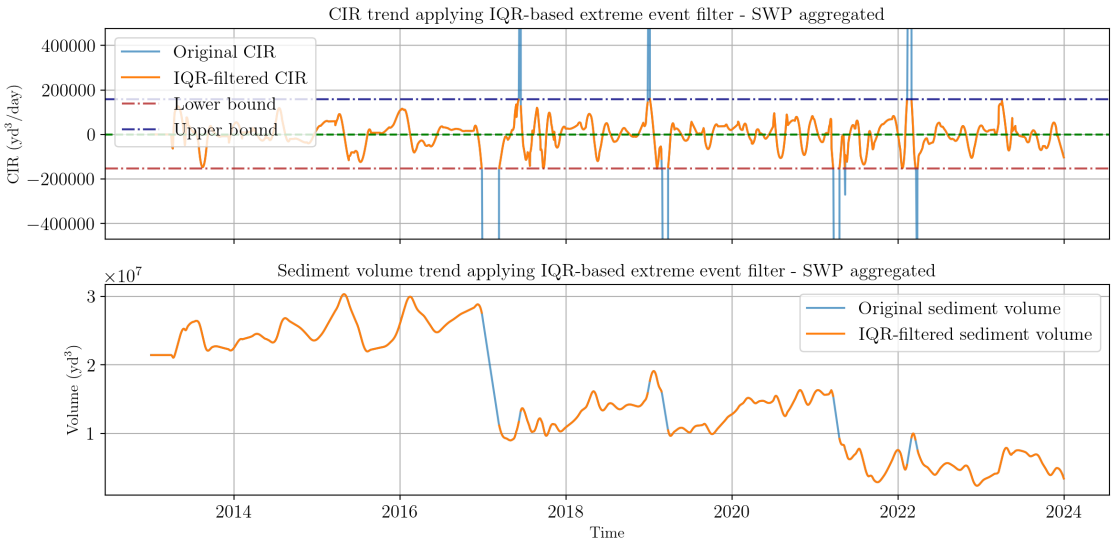


Figure 5.9: IQR-based extreme event filter for SWP region considering aggregated sedimentation data

curve reduces to an approximate piecewise linear fit. While this may capture the overall trend at the system level, it serves only as a first-order approximation of short-term variations.

5.3.3 Feature selection

Before we present the proposed solution strategy for forecasting CIR in Section 5.3.4, we first outline the approach employed to select input features used in the modeling framework. As mentioned in Sec. 5.2.4, we use the upstream streamflow gauge data as inputs (i.e., predictors) for CIR forecasting. However, the large number of available streamflow gauges (see Figs. 5.2(a) and 5.3(a)) makes the problem high-dimensional, which poses challenges to the data-driven CIR modeling. In order to identify the most informative streamflow gauges for CIR forecasting, the variance decomposition-based Global Sensitivity Analysis (GSA) is employed to select the N_{sg} gauges with highest sensitivity to the CIR time series.

Let $G_i(t)$ (or, equivalently, its abbreviated form G_i) denote the streamflow gauge time series for the i-th upstream gauge, where $1 \leq i \leq N_{\text{g}}$ and N_{g} denote the total number of upstream gauges considered for the sensitivity analysis. To ensure compatibility with the CIR data, the time instances corresponding to extreme event outliers were removed, resulting in each $G_i(t)$ vector containing N_{time} elements. These time series were then arranged as columns to form a matrix of streamflow gauge data, denoted by $\mathbf{G} \in \mathbb{R}^{N_{\text{time}} \times N_{\text{g}}}$. The objective of GSA is to identify which columns of $\mathbf{G}$ (i.e., which upstream gauges) are most sensitive to the CIR data, which serves as the output variable in the forecasting models (more details

on modeling in Section 5.3.4). For description, we use reach-level CIR data, denoted by $R_{\Delta,r} \in \mathbb{R}^{N_{\text{time}}}$.

In variance-based GSA (most commonly known as Sobol' sensitivity analysis), the variance of the reach-level CIR data $R_{\Delta,r}$ is decomposed into contributions from individual inputs $G_i, \forall i = 1, \cdots, N_g$ and their interactions as follows [136].

$$\text{Var}(R_{\Delta,r}) = \sum_{i=1}^{N_g} V_i + \sum_{1 \le i < j \le N_g} V_{ij} + \sum_{1 \le i < j < k \le N_g} V_{ijk} + \cdots + V_{1 \ldots N_g}, \tag{5.8}$$

where V_u in which $u \in \{1, \cdots, N_g\}$ is the variance contribution form the interaction involving exactly those input factors in u. For instance, $V_i, i = 1, \cdots, N_g$ are the variance contributions involve only the i-th streamflow gauge G_i.

Based on the above variance decomposition, two types of Sobol' indices are mostly widely used, namely the *First-Order Sobol' Index* and the *Total Sobol' Index*. The First-Order Sobol' Index quantifies the main effect of an individual input vector $G_i \in \mathbb{R}^{N_{\text{time}}}$ on the output vector $R_{\Delta,r}$, representing the fraction of the output variance attributable to G_i alone. It is defined as:

$$S_i = \frac{\text{Var}_{G_i}\left[\mathbb{E}\left(R_{\Delta,r} \mid G_i\right)\right]}{\text{Var}(R_{\Delta,r})}, \tag{5.9}$$

where $\text{Var}_{G_i}\left[\mathbb{E}\left(R_{\Delta,r} \mid G_i\right)\right]$ is the variance of the conditional expectation of $R_{\Delta,r}$ given G_i, and $\text{Var}(R_{\Delta,r})$ is the total variance of the output.

A high value of S_i indicates that the corresponding streamflow gauge G_i has a strong influence on the output variability and is therefore a strong candidate for inclusion in the forecasting model. Although the *Total Sobol' Index* captures both the main effects and interaction effects among variables, it is computationally more expensive to estimate. In our case, the primary goal was to identify the most influential individual gauges with minimal computational burden. We adopted the approach proposed by Ref. [136] to estimate the *First-Order Sobol' Index* S_i. This is a model-free method that directly computes sensitivity indices from the input and output data without requiring an explicit surrogate or simulation model linking them.

For the proposed model, additional computation was necessary because it operates on sequences of input data segments and produces corresponding output sequences. Since the model's predictions and sensitivity may vary across different positions within these sequences, it is important to aggregate the GSA indices over all time steps to obtain a comprehensive measure of input importance. Firstly, the local Sobol' index for each input variable $i \in \{1, \ldots, N_{\text{g}}\}$ at each time step $m \in \{1, \ldots, W_{\text{past}}\}$ within the input sequence can be defined

as follows:

$$S_{i,m,n} = \text{Sobol}\left(G_{i,t-w+m} \rightarrow R_{\Delta,r,t+n}\right), \tag{5.10}$$

where $G_{i,t-w+m}$ denotes the value of the i-th input variable at the m-th time step in the input sequence and $R_{\Delta,r,t+n}$ represents the n-th time step in the output sequence.

Building on the calculation of the local Sobol' index, the aggregated GSA indices are calculated using the following equation:

$$S_i^{(\text{agg})} = \sum_{m=1}^{W_{\text{past}}} \sum_{n=1}^{W_{\text{pred}}} S_{i,m,n}, \tag{5.11}$$

We then selected the top N_{sg} most sensitive streamflow gauges, based on each variable's aggregated Sobol' index $S_i^{(\text{agg})}$, and denote the k-th most sensitive gauge as $\tilde{G}_k$, where $1 \leq k \leq N_{\text{sg}}$. In this paper, we tried different N_{sg} values and finally considered two input configurations, top 3 and top 8 gauges, to balance parsimony and coverage: the top 3 represent the most dominant controls, while the top 8 incorporate additional gauges to capture broader spatial variability.

5.3.4 CIR Forecasting Problem Description and Solution Strategies

5.3.4.1 Modeling Strategies

In this paper, our objective is to predict sedimentation volume using two different modeling strategies.

1. **Strategy 1 (Disaggregated Strategy)**: This strategy treats each reach independently and forecasts the sedimentation volume at the individual reach level. The forecasted values are then aggregated to obtain the cumulative volume across multiple reaches.

2. **Strategy 2 (Aggregated Strategy)**: This strategy begins with the cumulative volume and the respective CIR obtained for the desired set of reaches $\mathcal{C}$ and directly forecasts the aggregate volume.

For *Strategy 1*, which involves modeling at the reach level, $R_{\Delta,r}(t)$ is used, while for *Strategy 2*, which involves modeling at the aggregate level, $R_{\Delta,\mathcal{C}}(t)$ is used. These two quantities are defined in Eq. (6.4) and are preprocessed to remove extreme outliers, as discussed in Section 5.3.2.2. For both of these strategies, we consider different data-driven forecasting

approaches. The only differences among these data-driven approaches are the CIR data used to train the forecasting models and the inclusion of upstream streamflow gauge data.

5.3.4.2 Simple Linear Regression

(a) Univariate Linear Regression (ULR) The first approach used the CIR data only for training. For the sake of describing this approach, we use the CIR of the reach level, $R_{\Delta,r}(t)$. A past window of W_{past} days of CIR data was used to fit a best-fit linear curve (via univariate linear regression, denoted by $\mathcal{G}_1(\cdot)$ in Eq. (5.12)), which was then used to predict the CIR over the next W_{pred} days. Specifically, for the current time t, the model was trained on the past W_{past} days of CIR data (including time t), denoted by $R_{\Delta,r}([t - W_{\text{past}} + 1 : t])$. The predicted CIR at future time $t < t + h \leq t + W_{\text{pred}}$ is functionally defined as:

$$\hat{R}_{\Delta,r}(t + h) = \mathcal{G}_1 \left(t + h; a_{\text{uni}}, b_{\text{uni}}\right), \forall h = 1, \cdots, W_{\text{pred}}. \tag{5.12}$$

Here, a_{uni} and b_{uni} are model parameters for univariate linear regression, as discussed later in Section 5.3.5.1.

(b) Multivariate Linear Regression (MLR) The second approach performs a $(N_{\text{sg}}+1)$-dimensional multivariate linear regression, denoted by $\mathcal{G}_2(\cdot)$, to regress the output variable CIR on a multi-dimensional input vector consisting of time and the most sensitive streamflow gauge data (i.e., identified in Sec. 5.3.3). Since the most sensitive gauges were selected based on the CIR data, these had to be obtained at the individual reach level for *Strategy 1*, or at the aggregate level for *Strategy 2*.

For the sake of describing MLR, we use the reach-level CIR, denoted by $R_{\Delta,r}(t)$. At the current time t, the model was trained using the past W_{past} days of CIR data as output, denoted by $R_{\Delta,r}([t - W_{\text{past}} + 1 : t])$, which was regressed against streamflow data from N_{sg} selected gauges, denoted by $\tilde{G}_i([t - W_{\text{past}} + 1 : t])$, and the corresponding time vector $[t - W_{\text{past}} + 1 : t]$ as inputs. The predicted CIR at a future time $[t < t + h \leq t + W_{\text{pred}}]$ is functionally defined as:

$$\hat{R}_{\Delta,r}(t+h) = \mathcal{G}_2 \left(t + h, \tilde{G}_1(t + h), \cdots, \tilde{G}_{N_{\text{sg}}}(t + h); \mathbf{a}_{\text{multi}}, b_{\text{multi}}\right), \forall h = 1, \cdots, W_{\text{pred}}. \tag{5.13}$$

Here, $\mathbf{a}_{\text{multi}}$ and b_{multi} are model parameters for multivariate linear regression, as discussed later in Section 5.3.5.1.

5.3.4.3 Advanced machine learning (ML) forecasting models

(a) Long Short-Term Memory (LSTM) Model This scenario utilizes historical CIR and streamflow gauge data as input, with the future CIR time series serving as the output. The LSTM algorithm, denoted by $\mathcal{G}_3(\cdot)$, was employed to construct the predictive model. Unlike the linear regression models used in *Approaches 1* and *2*, which are trained independently at each time instance t using only the most recent W_{past} days of data (i.e., $[t - W_{\text{past}} + 1 : t]$), the LSTM model is trained *once* on the entire available training history. Once trained, the same LSTM model is used to predict the future output for any given time step.

In this regard, the linear regression approach can be viewed as a *localized fitting technique*, capturing short-term or local effects that may vary from one time step to another. In contrast, the LSTM approach captures *global temporal patterns* by learning from the complete time history, allowing it to generalize and detect long-term dependencies across the data sequence.

The proposed LSTM neural network was specifically designed for the prediction of the sequence-to-sequence of future CIR. The core of the proposed architecture is the LSTM cell, a type of recurrent neural network (RNN) unit specifically designed to capture long-term dependencies in sequential data [137]. Unlike traditional RNN cells, which can suffer from vanishing or exploding gradients, the LSTM cell incorporates a memory mechanism through the use of gating functions that regulate the flow of information. By selectively retaining or discarding information via its gating mechanisms, the LSTM can model complex dependencies across widely varying timescales, improving robustness to noise and irregularities in the data. This makes the LSTM particularly well-suited for forecasting the CIR, where the underlying physical processes may involve delayed responses and long-range correlations.

The architecture of the LSTM model is illustrated in Fig. 5.10. There are three stacked LSTM layers in the model, each followed by a dropout layer to reduce overfitting by randomly zeroing a fraction of the activations during training. The final hidden state of the last LSTM layer is passed to a fully connected output layer, which maps the learned temporal representation (i.e., streamflow discharge and past CIR values) to the desired output sequence length W_{pred} (i.e., future CIR).

Defining the functional form of $\mathcal{G}_3(\cdot)$ requires further clarification. The input data was structured as a two-dimensional matrix, denoted by $\mathbf{X}_{\text{train}}$, where each row corresponds to a time instance. Each row contains a total of $W_{\text{past}}.(N_{\text{sg}} + 1)$ columns, representing the input features: specifically, W_{past} days of past CIR data and streamflow data from N_{sg} gauges.

The number of available days of data, denoted by N_{time}, was divided into N_{train} training days and N_{test} testing days, such that $N_{\text{time}} = N_{\text{train}} + N_{\text{test}}$. Because each training input requires W_{past} days of historical data, the total number of training samples (i.e., the number

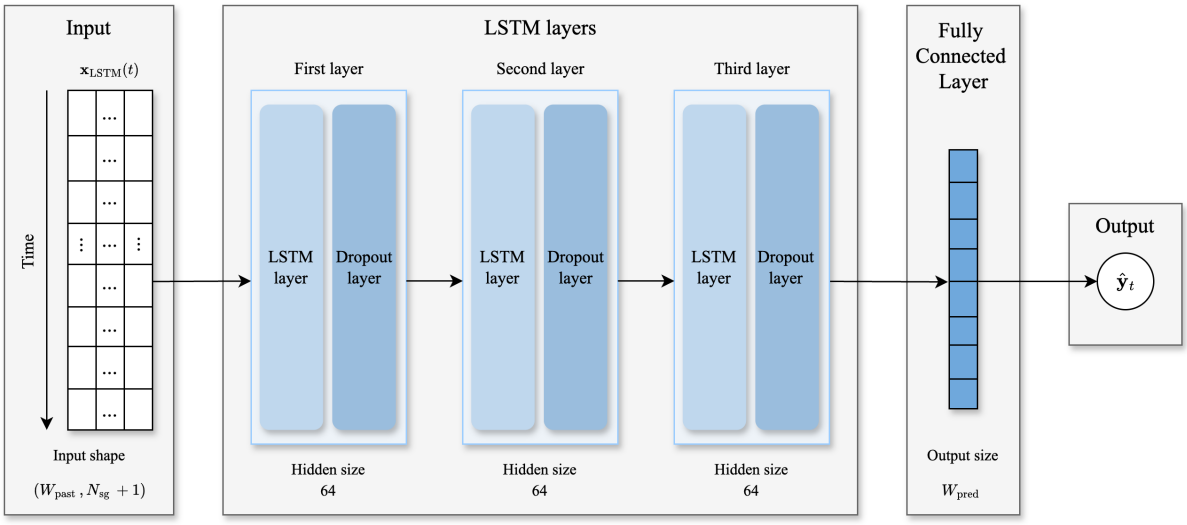


Figure 5.10: The overall structure of the LSTM model.

of rows in $\mathbf{X}_{\text{train}}$) is given by $(N_{\text{train}} - W_{\text{past}} + 1)$.

The output used for training the LSTM model is denoted by $\mathbf{Y}_{\text{train}}$, which contains the same number of rows as $\mathbf{X}_{\text{train}}$, corresponding to each training time instance. $\mathbf{Y}_{\text{train}}$ consists of W_{pred} columns, representing the next W_{pred} days of future CIR data.

$$\mathbf{X}_{\text{train}} \in \mathbb{R}^{(N_{\text{train}} - W_{\text{past}} + 1) \times (W_{\text{past}} \cdot (N_{\text{sg}} + 1))}; \tag{5.14a}$$

$$\mathbf{Y}_{\text{train}} \in \mathbb{R}^{(N_{\text{train}} - W_{\text{past}} + 1) \times W_{\text{pred}}}. \tag{5.14b}$$

The LSTM model $\mathcal{G}_3(\cdot)$ learns the mapping from the input $\mathbf{X}_{\text{train}}$ to the output $\mathbf{Y}_{\text{train}}$. For the sake of describing the functional form of $\mathcal{G}_3(\cdot)$, we use the reach-level CIR, denoted by $R_{\Delta,r}(t)$. At time t, let $\mathbf{x}_{\text{LSTM}}(t) \in \mathbb{R}^{W_{\text{past}}(N_{\text{sg}}+1)}$ denote the feature vector at time t within a given input window. The predicted CIR, denoted by $\hat{R}_{\Delta,r}([t+1 : t+W_{\text{pred}}])$, is given by:

$$\hat{R}_{\Delta,r}([t+1 : t+W_{\text{pred}}]) = \mathcal{G}_3\left(\mathbf{x}_{\text{LSTM}}(t); \mathbf{W}_{\text{LSTM}}, \mathbf{b}_{\text{LSTM}}\right). \tag{5.15}$$

Here, $\mathbf{W}_{\text{LSTM}}$ and $\mathbf{b}_{\text{LSTM}}$ are model parameters obtained by training the LSTM model, as discussed later in Section 5.3.5.2.

(b) Convolutional neural network–long short-term memory (CNN–LSTM) model
To improve the representation of temporal dynamics in CIR forecasting, we also employ a hybrid convolutional neural network–long short-term memory (CNN–LSTM) architecture. This structure combines the strengths of convolutional layers for local temporal feature extrac-

tion with recurrent layers for modeling long-term temporal dependencies. The CNN–LSTM structure was proposed in the context of deep learning for speech recognition by Sainath *et al.* [194] and has since been successfully applied to a range of temporal prediction tasks.

Unlike the standalone LSTM model, which directly processes the raw input sequence, the CNN–LSTM architecture first applies one-dimensional convolutional filters along the temporal dimension to learn localized temporal features and interactions among the input variables. These learned features are then passed to stacked LSTM layers to model longer-range temporal dependencies. That is to say, the overall structure of the CNN–LSTM model consists of three main components: (1) a convolutional feature extractor, (2) stacked LSTM layers with dropout regularization, and (3) a fully connected output layer. This architecture is illustrated in Fig. 5.11.

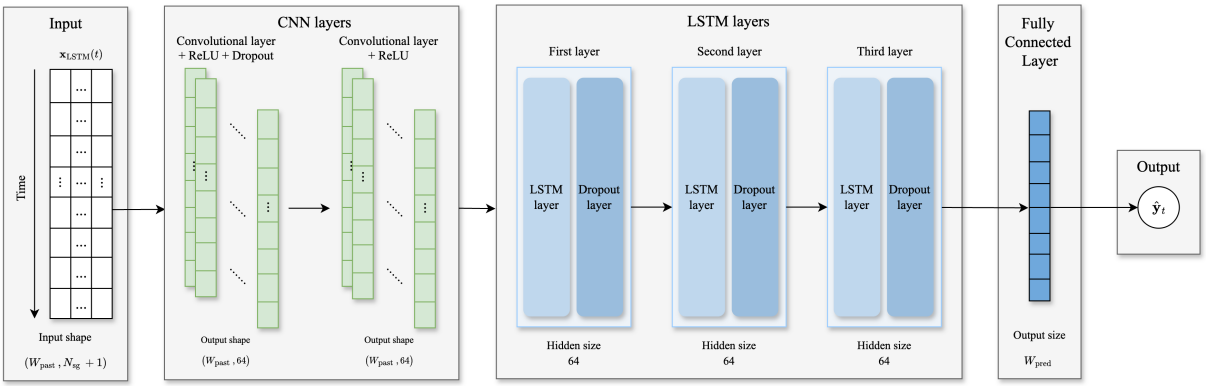


Figure 5.11: The overall structure of the CNN-LSTM model for CIR forecasting.

The input and output data structures are similar to those of the LSTM model. The difference is that there are CNN layers between the input and the LSTM layers. The convolutional stage applies one-dimensional convolutional filters along the temporal axis. Two convolutional layers with kernel size k are used, each followed by a ReLU activation function. Dropout is applied after the first convolutional layer to reduce overfitting (dropout rate = 0.2). These convolutional layers learn local temporal patterns and short-term variations in the CIR and streamflow signals within the input window. After the convolutional feature extraction stage, the feature representation maintains the same temporal length while expanding the feature dimension to 64 channels. Let $\mathcal{C}(\cdot)$ denote the nonlinear transformation implemented by the convolutional feature extractor. The output of the CNN block is given by

$$\mathbf{Z}(t) = \mathcal{C}\left(\mathbf{x}_{\mathrm{CNN-LSTM}}(t)\right), \tag{5.16}$$

where $\mathbf{Z}(t)$ is a sequence of learned temporal features with reduced noise and enhanced local

structure, $\mathbf{x}_{\mathrm{CNN-LSTM}}(t) \in \mathbb{R}^{W_{\mathrm{past}} \times (N_{\mathrm{sg}}+1)}$ denotes the input feature vector at each prediction time t.

The extracted feature sequence $\mathbf{Z}(t)$ is then passed to stacked LSTM layers. Similar to the standalone LSTM model, the LSTM component is trained once using the entire available training history and is subsequently applied to all prediction time steps. The LSTM layers capture global temporal dependencies and long-range correlations across the filtered feature sequence. The final hidden state of the last LSTM layer is passed through a ReLU activation function and a dropout layer, and then mapped to the output space via a fully connected layer.

The CNN–LSTM model, denoted by $\mathcal{G}_4(\cdot)$, learns the mapping from the input sequence to the future CIR sequence as

$$\hat{R}_{\Delta,r}\left([t+1:t+W_{\mathrm{pred}}]\right) = \mathcal{G}_4\left(\mathbf{x}_{\text{CNN-LSTM}}(t); \mathbf{W}_{\text{CNN-LSTM}}, \mathbf{b}_{\text{CNN-LSTM}}\right), \tag{5.17}$$

where $\mathbf{W}_{\text{CNN-LSTM}}$ and $\mathbf{b}_{\text{CNN-LSTM}}$ represent the trained parameters of both the convolutional and LSTM layers.

(c) Temporal Convolutional Network–Long Short-Term Memory (TCN–LSTM) model This scenario extends the LSTM-based forecasting framework by incorporating a Temporal Convolutional Network (TCN) as a front-end feature extractor, followed by stacked LSTM layers for sequence-to-sequence prediction. Unlike standard convolutional neural networks with fixed receptive fields, the TCN employs causal and dilated one-dimensional convolutions, enabling the model to capture temporal dependencies over multiple timescales while preserving the temporal ordering of the input sequence [195]. By stacking convolutional layers with increasing dilation factors, the TCN effectively expands its temporal receptive field without increasing model depth. This design enables the TCN to encode long-range temporal patterns directly within the convolutional feature maps prior to recurrent processing.

The overall structure of the TCN–LSTM model is illustrated in Fig. 5.12. The input data are first processed by a sequence of dilated causal convolutional layers, forming the TCN component. It applies a series of dilated one-dimensional convolutional layers to capture temporal patterns at multiple scales. In this study, three dilated convolution layers are used with dilation factors $d = 1, d = 2,$ and $d = 4$, respectively. Each convolutional layer contains 32 filters and is followed by a ReLU activation function and a dropout layer (dropout rate = 0.2) to enhance nonlinearity and reduce overfitting. After the TCN module, the extracted features maintain the same temporal length while expanding the feature dimension to 32 channels, resulting in an output tensor of shape $(W_{\mathrm{past}}, 32)$. This output feature sequence

is then passed to a stack of LSTM layers, followed by a fully connected output layer that maps the learned temporal representation to the desired prediction horizon W_{pred}. At a given time t, let $\mathbf{x}_{\text{TCN-LSTM}}(t) \in \mathbb{R}^{W_{\text{past}}(N_{\text{sg}}+1)}$ denote the input feature vector within the sliding window. The predicted CIR sequence is expressed as

$$\hat{R}_{\Delta,r}\left([t+1:t+W_{\text{pred}}\,]\right) = \mathcal{G}_5\left(\mathbf{x}_{\text{TCN-LSTM}}(t); \mathbf{W}_{\text{TCN-LSTM}}, \mathbf{b}_{\text{TCN-LSTM}}\right), \tag{5.18}$$

where $\mathbf{W}_{\text{TCN-LSTM}}$ and $\mathbf{b}_{\text{TCN-LSTM}}$ denote the learned parameters of the TCN, LSTM, and fully connected layers.

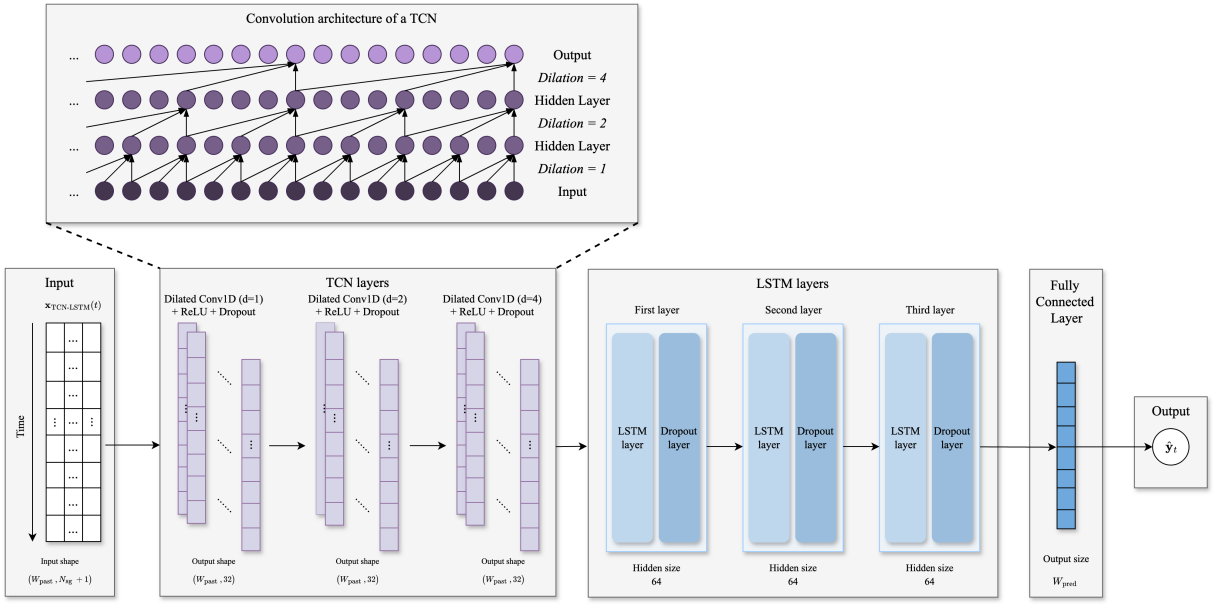


Figure 5.12: The overall structure of the TCN-LSTM model for CIR forecasting.

Remark 1 Recall from Fig. 5.4 that the HSC region has sparse survey data, whereas the SWP region has relatively abundant survey data. Therefore, the advanced ML model-based approaches, such as variants of LSTM-based models adopted in this paper, which provide a flexible global fitting capability, are more suitable for capturing the complex dynamics of the SWP region. This modeling approach can effectively learn temporal depth and non-linear patterns from rich data. For the SWP region modeled using LSTM, CNN-LSTM, and TCN-LSTM models, we set both W_{past} and W_{pred} to 30 days. The 30-day input window was selected as it provided the optimal balance between model accuracy and computational efficiency, while also representing the typical hydrologic memory of discharge and sediment processes at the monthly scale. On the other hand, the sparse survey data in the HSC region

produces a sedimentation volume curve that approximates a coarse, piecewise linear form. In such cases, the data structure aligns more naturally with simpler linear models (as suggested by ULR and MLR), which are better suited to learning local behavior. The sparsity of data limits the ability to capture long-term patterns. Additionally, such localized linear fitting techniques help avoid overfitting and align well with the first-order characteristics introduced by interpolation. For the HSC region modeled using ULR and MLR, we set relatively lower values of W_{past} to 5 days.

5.3.4.4 Obtaining the volume from CIR data

For all three approaches and both scenarios, the CIR data used for training were obtained using consecutive days of volume data with $\Delta = 1$.

Consider *Strategy 1* and a current time instance t, such that the past CIR and streamflow gauge data are available. Once $\hat{R}_{\Delta=1,r}([t+1 : t+W_{\text{pred}}])$ is obtained (using any of the three approaches), and since $\Delta = 1$, the forecasted volume of the reach, denoted by $\hat{V}_r$, and the combined volume, denoted by $\hat{V}_{\mathcal{C}}$, at time $(t+1)$ for *Strategy 1* is computed as:

$$\hat{V}_r(t+i) = \begin{cases} V_r(t) + \hat{R}_{\Delta=1,r}(t+i), & \text{if } i = 1 \\ \hat{V}_r(t+i-1) + \hat{R}_{\Delta=1,r}(t+i), & \text{if } i > 1 \end{cases}; \tag{5.19a}$$

$$\hat{V}_{\mathcal{C}}(t+1) = \sum_{i=1}^{N_{\text{reaches}}} \hat{V}_r(t). \tag{5.19b}$$

Similarly, for *Strategy 2*, the predicted sedimentation volume, denoted by $\hat{V}_{\mathcal{C}}$, is computed as:

$$\hat{V}_{\mathcal{C}}(t+i) = \begin{cases} V_{\mathcal{C}}(t) + \hat{R}_{\Delta=1,\mathcal{C}}(t+i), & \text{if } i = 1 \\ \hat{V}_{\mathcal{C}}(t+i-1) + \hat{R}_{\Delta=1,\mathcal{C}}(t+i), & \text{if } i > 1 \end{cases} \tag{5.20}$$

As discussed in *Remark 1*, since HSC has sparse data that are linearly interpolated to obtain the volume time series, it serves as a natural choice to demonstrate ULR and MLR. On the other hand, SWP, being rich in survey data, is more suitable for illustrating the workings of LSTM-based models (i.e., LSTM, CNN-LSTM, and TCN-LSTM). All methods will be discussed in more detail in Section 5.3.5.

5.3.5 Forecasting framework

The remainder of this section presents a comprehensive overview of the linear regression and LSTM-based ML methods utilized in this study. To describe the regression techniques

used across the three approaches, we focus on the reach-level CIR (i.e., *Scenario 1*). The regression methods remain the same for *Scenario 2*, except that CIR data is considered at the aggregate level, with all other aspects held constant.

5.3.5.1 Linear regression based model

Solution ULR and MLR are two different linear regression strategies: a univariate approach using only the recent history of the target variable, and a multivariate approach that incorporates additional gauge data.

1. In the ULR case, predictions rely exclusively on past observations of the target variable, $R_{\Delta,r}([t - W_{\text{past}} + 1 : t])$. For each prediction step, a simple linear regression model is fitted using the CIR data over the past W_{past} days as the output and time as the input. The model estimates a linear trend—characterized by a single slope and intercept—across these time points. This linear model is then extrapolated to predict sediment volumes for the subsequent W_{pred} days.

   The fitted linear function is expressed as

   $$R_{\Delta,r}(t) = a_{\text{uni}} \cdot t + b_{\text{uni}}. \tag{5.21}$$

   Here, t is the time index, a_{uni} is the estimated slope, and b_{uni} is the intercept—both obtained by least-squares fitting of the training input $[t - W_{\text{past}} + 1 : t]$ to the output $R_{\Delta,r}([t - W_{\text{past}} + 1 : t])$.

   Prediction for the next W_{pred} days ($h = 1, \ldots, W_{\text{pred}}$) are given by:

   $$\hat{R}_{\Delta,r}(t + h) = a_{\text{uni}} \cdot (t + h) + b_{\text{uni}}. \tag{5.22}$$

   where the same slope and intercept are applied for all forecast steps $1 \leq h \leq W_{\text{pred}}$.

2. Similar to ULR, in MLR, a local model is trained at each time step using a multivariate linear regression. The regression models the CIR over the past W_{past} days as the output variable, using the corresponding time steps and the most sensitive gauge data over the same period as input features. The fitted linear model is expressed as:

   $$R_{\Delta,r}(t) = \mathbf{x}_{\text{multi}}(t) \cdot \mathbf{a}_{\text{multi}} + b_{\text{multi}}. \tag{5.23}$$

   Here, $\mathbf{x}_{\text{multi}}(t)$ denotes the input vector defined as:

   $$\mathbf{x}_{\text{multi}}(t) = [t, \tilde{G}_1(t), \ldots, \tilde{G}_{N_{\text{sg}}}(t)]. \tag{5.24}$$

The parameters $\mathbf{a}_{\text{multi}}$ and b_{multi} are obtained by least-squares fitting of the training input $\mathbf{x}_{\text{multi}}(t_{\text{train}})$ to the output $R_{\Delta,r}(t_{\text{train}})$ for all training time steps $t_{\text{train}} \in [t - W_{\text{past}} + 1 : t]$. The term $\mathbf{x}_{\text{multi}}(t) \cdot \mathbf{a}_{\text{multi}}$ denotes a dot product between input vector $\mathbf{x}_{\text{multi}}(t)$ and the parameter vector $\mathbf{a}_{\text{multi}}$.

Prediction for the next W_{pred} days ($h = 1, \ldots, W_{\text{pred}}$) are given by:

$$\hat{R}_{\Delta,r}(t+h) = \mathbf{x}_{\text{multi}}(t+h) \cdot \mathbf{a}_{\text{multi}} + b_{\text{multi}}. \tag{5.25}$$

where the same slope and intercept are applied for all forecast steps $1 \leq h \leq W_{\text{pred}}$.

5.3.5.2 ML model-based based CIR forecasting

Let $\mathbf{x}_{\text{ML-based}}(t) \in \mathbb{R}^{W_{\text{past}}(N_{\text{sg}}+1)}$ denotes the feature vector at time t within a given input window, where $\mathbf{x}_{\text{ML-based}}(t)$ is $\mathbf{x}_{\text{LSTM}}(t)$ for the LSTM model, is $\mathbf{x}_{\text{CNN-LSTM}}(t)$ after passing through the CNN layers for the CNN-LSTM model, and is $\mathbf{x}_{\text{TCN-LSTM}}(t)$ after passing through the TCN layers for the TCN-LSTM model. For each LSTM layer $l = 1, 2, 3$, the hidden state and cell state are recursively updated according to:

$$\left[\mathbf{h}_t^{(l)}, \mathbf{c}_t^{(l)}\right] = \text{LSTM}^{(l)}\left(\mathbf{h}_{t-1}^{(l)}, \mathbf{c}_{t-1}^{(l)}, \mathbf{z}_t^{(l)}\right) \tag{5.26}$$

where $\mathbf{z}_t^{(l)} = \mathbf{x}_{\text{ML-based}}(t)$ for the first layer and $\mathbf{z}_t^{(l)} = \mathbf{h}_t^{(l-1)}$ for subsequent layers.

After passing through the stacked LSTM layers (i.e., CNN and LSTM layers for the CNN-LSTM model, TCN and LSTM layers for the TCN-LSTM model) and dropout regularization, the final hidden state sequence is processed by a dense output layer:

$$\hat{\mathbf{y}}_t = \mathbf{W}_{\text{LSTM}}\mathbf{h}_t^{(3)} + \mathbf{b}_{\text{LSTM}} \tag{5.27}$$

Here, $\mathbf{W}_{\text{LSTM}}$ and $\mathbf{b}_{\text{LSTM}}$ are the weights and bias of the output layer in the LSTM module, and $\hat{\mathbf{y}} \in \mathbb{R}^{W_{\text{pred}}}$ is the vector of predicted values for the forecast window. For *Scenario 1*, $\hat{\mathbf{y}}_t = \hat{R}_{\Delta,r}([t+1 : t+W_{\text{pred}}])$; and for *Scenario 2*, $\hat{\mathbf{y}}_t = \hat{R}_{\Delta,\mathcal{C}}([t+1 : t+W_{\text{pred}}])$.

5.3.6 Uncertainty quantification of CIR forecasting

In addition to point predictions, uncertainty quantification (UQ) is incorporated into all LSTM-based models considered in this study, including the standard LSTM, CNN–LSTM, and TCN–LSTM architectures. Given the cumulative nature of sediment volume prediction, where small errors in predicted CIR can propagate over time, the predictive uncertainty is critical for reliable forecasting and risk-informed decision making. To achieve this, Monte

Carlo (MC) dropout is implemented as a computationally efficient, distribution-free approach for estimating predictive uncertainty in deep neural networks. MC-dropout provides an approximation to Bayesian inference by interpreting dropout as a variational approximation over the model parameters [196].

Let $\mathcal{G}_{\mathrm{ML}}(\cdot)$ denote a trained LSTM-based model or a hybrid model that combines LSTM model with a CNN or TCN model (i.e., LSTM, CNN–LSTM, or TCN–LSTM), parameterized by weights W. During standard training, dropout layers are used to randomly deactivate a subset of neurons, reducing overfitting. In MC-dropout, this stochastic behavior is retained during inference, enabling multiple forward passes through the network with different realizations of the dropout mask. For a given input sequence $\mathbf{x}(t)$, N stochastic forward passes are performed:

$$\hat{\mathbf{y}}^{(i)}(t) = \mathcal{G}_{\mathrm{ML}}\left(\mathbf{x}(t); \mathbf{W}^{(i)}\right), \quad i = 1, \ldots, N, \tag{5.28}$$

where $\mathbf{W}^{(i)}$ represents an implicit sample from the approximate posterior induced by dropout. Each forward pass produces a realization of the predicted CIR sequence over the forecast horizon W_{pred}.

The predictive mean is estimated as the ensemble average of the MC samples:

$$\overline{\mathbf{y}}(t) \approx \frac{1}{N} \sum_{i=1}^{N} \hat{\mathbf{y}}^{(i)}(t), \tag{5.29}$$

while the predictive uncertainty is quantified by the sample variance:

$$\sigma^2(t) \approx \frac{1}{N-1} \sum_{i=1}^{N} \left(\hat{\mathbf{y}}^{(i)}(t) - \overline{\mathbf{y}}(t)\right)^2. \tag{5.30}$$

Assuming approximate normality of the predictive distribution, a two-sided confidence interval can be constructed at each forecast horizon step. In this study, we select 95% confidence interval which can be find in the case study section. These uncertainty bounds are propagated through the sediment volume integration process, enabling uncertainty estimates not only for CIR predictions but also for the corresponding cumulative sediment volume forecasts.

5.3.7 Error Metrics

The efficacy of predicting the sedimentation volume, obtained by integrating the predicted CIR, is quantified using three different error metrics. For each time instance t in the test segment, the output consists of a predicted time series of W_{pred} future time steps. Let the

vector $\hat{\mathbf{V}}_{t+1:t+W_{\text{pred}}}$ denote the predicted volume time series and $\mathbf{V}_{t+1:t+W_{\text{pred}}}$ denote the corresponding observed true values for test time instance t, with the total number of test samples denoted by N_{test}. Let $\hat{V}_{t+k}$ and V_{t+k} denote the predicted and true values at time $t+k$ (i.e., the k-th elements of their respective vectors).

The first error metric used is the Root Mean Squared Error (RMSE) defined as:

$$\text{RMSE} = \sqrt{\frac{1}{N_{\text{test}} \cdot W_{\text{pred}}} \sum_{t=1}^{N_{\text{test}}} \sum_{k=1}^{W_{\text{pred}}} \left(V_{t+k} - \hat{V}_{t+k}\right)^2}. \tag{5.31}$$

Since observed sediment volumes vary substantially across test windows, RMSE alone may not provide a fair performance comparison. This issue is addressed by normalizing RMSE by the mean observed value within each window, which facilitates consistent comparison across periods and scales. The average normalized RMSE, denoted as nRMSE, is defined as:

$$\text{nRMSE} = \frac{1}{N_{\text{test}}} \left(\frac{\sqrt{\frac{1}{W_{\text{pred}}} \sum_{k=1}^{W_{\text{pred}}} \left(V_{t+k} - \hat{V}_{t+k}\right)^2}}{\frac{1}{W_{\text{pred}}} \sum_{k=1}^{W_{\text{pred}}} V_{t+k}} \right). \tag{5.32}$$

The third error metric, used to assess the relative accuracy of predictions across periods with varying sediment volume scales, is the Mean Absolute Percentage Error (MAPE). The average MAPE is computed as:

$$\text{MAPE} = \frac{1}{N_{\text{test}}} \left(\frac{100}{W_{\text{pred}}} \sum_{k=1}^{W_{\text{pred}}} \left| \frac{V_{t+k} - \hat{V}_{t+k}}{V_{t+k}} \right| \right). \tag{5.33}$$

5.4 Case Study

In this section, the two study sites of interest (i.e., HSC and SWP) as described in Sec. 5.2.2 are employed to demonstrate and validate the proposed framework. As mentioned previously, these two sites have different survey frequencies, which represent two different representative scenarios. Through this case study, we aim to identify the best modeling approach for each scenario.

5.4.1 Houston Ship Channel (HSC)

Figure 5.13 illustrates historic CIR and sedimentation volume of four representative reaches in the HSC region. These figures show that the HSC has relatively fewer measure-

ments and bathymetric surveys compared to the SWP region (also reflected in Figure 5.4 and Figure 5.20). In this case study, data spanning January 1, 2013, to the end of 2023 were used. The dataset was divided into three subsets: the training set (2013–early 2021), the validation set (2021–2022), and the test set (2022–end of 2023). The model was trained to produce rolling 30-day forecasts for each timestep within the test period.

Two strategies are evaluated and compared in the case study. The first is a disaggregated strategy, in which GSA is performed, and models are trained separately for each reach. After selecting an identical testing period for all reaches, the predicted sediment volumes from each reach are aggregated to obtain the total sediment volume trend for the entire target region. In contrast, the aggregated strategy directly applies GSA and trains the model using the total sediment volume aggregated across all reaches. Although the aggregated approach can retain a larger number of timesteps within the same testing period, the testing dataset is constrained to match that of the disaggregated strategy to ensure a fair comparison. The remainder of this section presents the results obtained for this region.

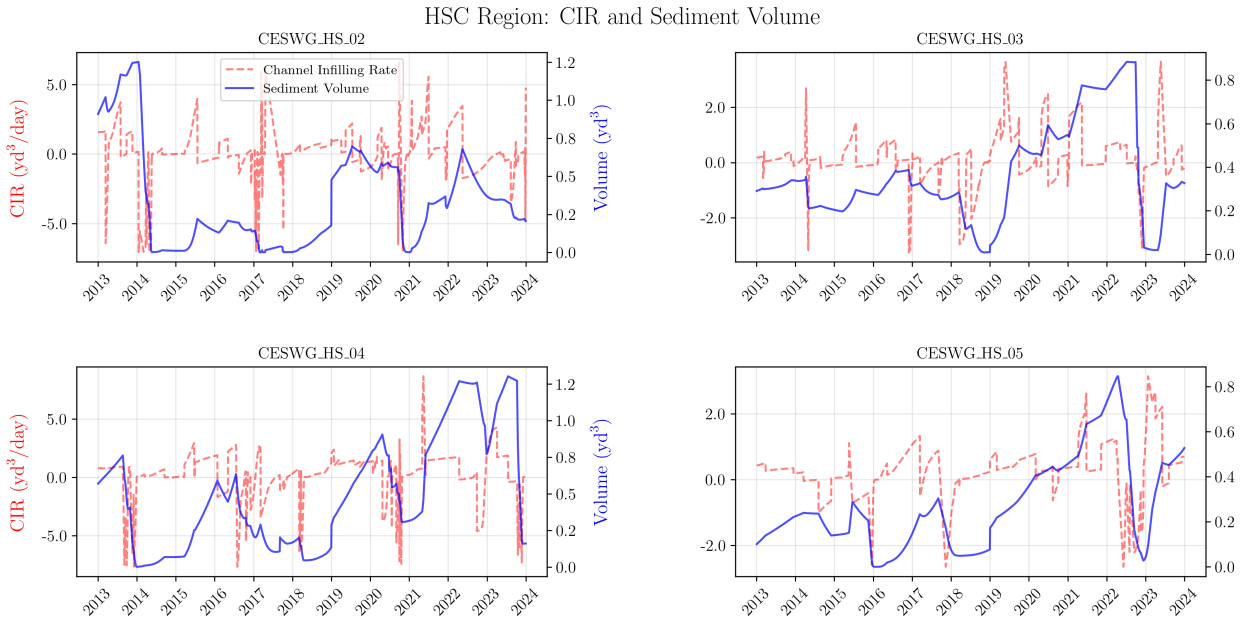


Figure 5.13: Sedimentation volume and channel infilling rate (CIR) for representative reaches in the HSC region (The legend for all subplots is identical to that of the top-left figure).

5.4.1.1 Disaggregated Strategy

The prediction results obtained using the disaggregated strategy are presented in Figure 5.14 and Table 5.1. Figure 5.14 illustrates the predicted sedimentation volume and CIR at the reach level for a representative reach (CESWG_HS_02_RFB_2) in the HSC region at

four different time instances. Table 5.1 reports the error metrics corresponding to these predictions. The full test set consists of 686 filtered time steps, covering the period from January 1, 2022, to December 31, 2023. From the results, it can be observed that:

- In the HSC region, linear regression (ULR and MLR) proves better suited for the HSC region than LSTM, which yields poor predictive performance with significantly larger errors. This weakness arises from the HSC's sparse survey record, which produces a sedimentation volume curve that is essentially a coarse, piecewise linear approximation, as shown in Figure 5.13. In such cases, the data structure aligns more naturally with simpler linear models, which not only avoid overfitting but also capture the first-order characteristics introduced by the interpolation process.

- The results of ULR and MLR are not very different. Based on the nRMSE and MAPE error metrics, MLR performs marginally better than ULR. Therefore, although MLR is objectively the best, the minimal difference between ULR and MLR arises from the sparse nature of the HSC data, where adding information from streamflow gauges provides little additional value, and the correlation between the CIR data and the gauge data is weak.

Table 5.1: HSC region: Evaluation metrics for reach-level predictions at a representative reach under the disaggregated modeling strategy, compared across different modeling approaches. Columns correspond to four selected forecast start dates and the full test set. For the full test set, the best value for each metric is highlighted in bold.

Method	**Metric**	**2022-03-20**	**2022-10-16**	**2023-02-01**	**2023-11-01**	**Full test set**
ULR	RMSE	68.56	26.18	197.12	282.07	**2226.55**
	nRMSE	0.0001	0.00006	0.0005	0.0013	0.0038
	MAPE (%)	0.007	0.003	0.041	0.09	0.27
MLR	RMSE	382.43	231.80	70.73	292.15	2301.74
	nRMSE	0.0007	0.0005	0.0002	0.0013	**0.0034**
	MAPE (%)	0.05	0.0425	0.01	0.09	**0.26**
LSTM	RMSE	17671.27	1033.39	6315.08	1487.23	15617.80
	nRMSE	0.0326	0.0024	0.0180	0.0069	0.0352
	MAPE (%)	2.49	0.21	1.61	0.49	2.79

The predicted sedimentation volume using MLR at the reach level is evaluated under two predictor configurations based on the most sensitive upstream gauges measuring streamflow discharge (ranking obtained from the GSA analysis). The first configuration uses the three most sensitive gauges, representing a parsimonious set that captures the dominant upstream

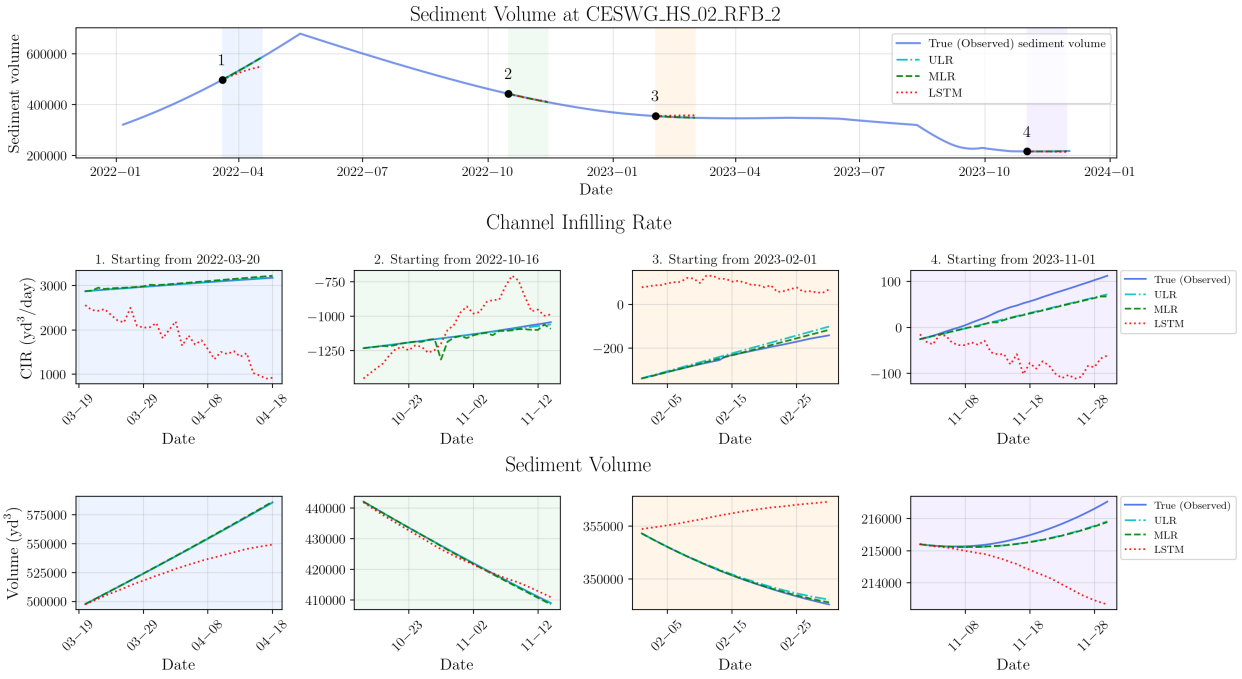


Figure 5.14: HSC region: Forecasted channel infilling rate (CIR) and sediment volume at a representative individual reach under the disaggregated strategy, compared across three modeling approaches. The top row shows the overall sediment volume time series together with the forecast windows. The second row presents the CIR forecasts for four selected forecast start dates, and the third row shows the corresponding sediment volume forecasts. Each column corresponds to one forecast window.

hydrological signals influencing CIR dynamics. The second configuration expands the predictor set to the top eight gauges to examine whether incorporating additional upstream observations provides additional predictive information. This comparison allows assessment of the trade-off between model simplicity and potential information gain from a larger set of hydrological predictors. Table 5.2 reports the error metrics corresponding to these predictions. The full test set consists of 686 time steps, covering the period from January 1, 2022, to December 31, 2023. Several prediction instances show stronger agreement between the observed and predicted CIR when using only the three most sensitive gauges. Consistent with these observations, the nRMSE and MAPE metrics computed for the full test set indicate that the three-gauge case consistently outperforms the eight-gauge case. This improvement can be attributed to the characteristics of the HSC region, where the inclusion of additional gauges may introduce noise rather than useful predictive information due to weak correlations between some gauges and the CIR data.

Table 5.2: HSC region: Evaluation metrics for reach-level predictions at a representative reach under the disaggregated modeling strategy, using different sets of sensitive gauges (MLR). Columns correspond to four selected forecast start dates and the full test set. Best values for the full test set are highlighted in bold.

Input	**Metric**	**2022-02-06**	**2022-07-15**	**2023-02-03**	**2023-08-28**	**Full test set**
Top 3 most sensitive gauges	RMSE	218.44	15.94	155.73	4,539.63	**2,301.74**
	nRMSE	0.0005	0.00002	0.0004	0.0190	**0.0034**
	MAPE (%)	0.04	0.001	0.03	1.54	**0.26**
Top 8 most sensitive gauges	RMSE	153.46	45.11	1,288.92	11,558.70	3,062.29
	nRMSE	0.0004	0.00008	0.0037	0.0484	0.0041
	MAPE (%)	0.03	0.006	0.28	3.55	0.29

5.4.1.2 Comparison Between Disaggregated and Aggregated Strategies

Figure 5.15 presents the predicted cumulative sedimentation volume using MLR for both Modeling *Scenario 1* (disaggregated strategy) and Modeling *Scenario 2* (aggregated strategy). The corresponding error metrics are reported in Table 5.3. Unlike the reach-level predictions, the full test set for cumulative volume predictions contains only 86 filtered time steps, spanning the period from January 1, 2022, to December 31, 2023. The limited size of the test dataset arises from the requirement to retain only the common test timesteps shared by both *Scenario 1* and *Scenario 2* in order to ensure a fair comparison. However, because the disaggregated strategy (*Scenario 1*) applies the IQR-filtering process separately to each reach, many participating reaches end up with a larger number of discarded time steps.

Therefore, only the time steps retained across all participating reaches were used for testing, resulting in a reduced number of test time steps. As shown in Figure 5.15 and Table 5.3, the cumulative volumes predicted for the two scenarios differ only marginally. Therefore, we conclude that cumulative volume prediction depends less on selecting the scenario and more on choosing the appropriate regression technique (ULR or MLR over LSTM-based models).

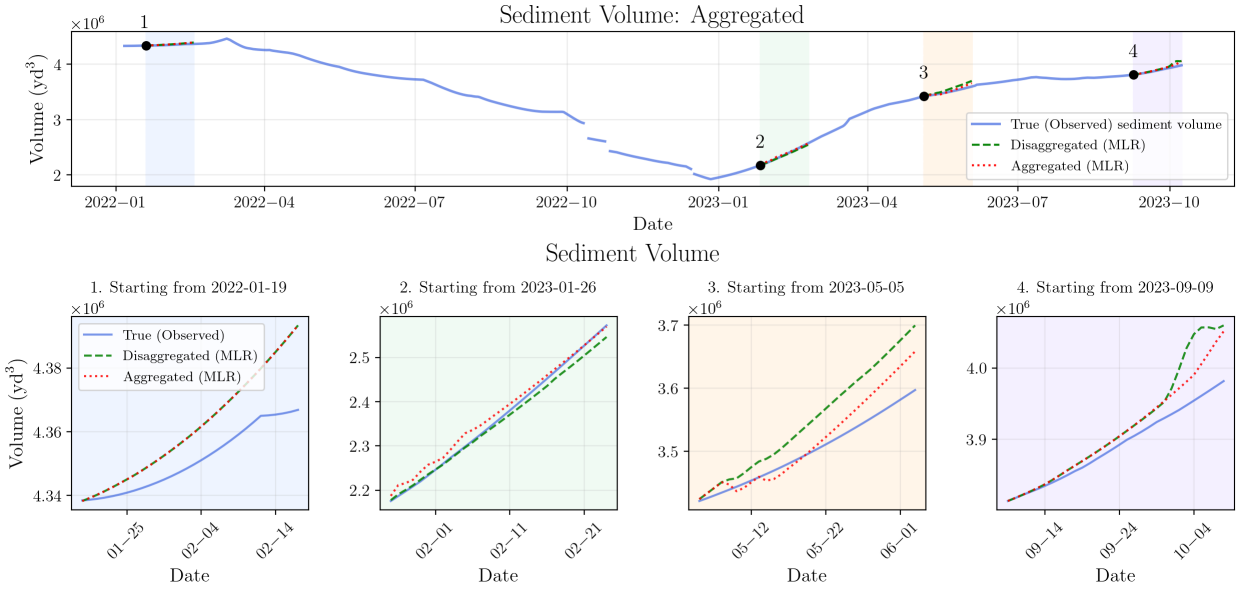


Figure 5.15: HSC region: Forecasting of cumulative sedimentation volume aggregated over all reaches using two modeling strategies, illustrated for four selected forecast start dates using the MLR model with the three most sensitive gauges. The top row shows the overall sediment volume time series together with the forecast windows. The second row presents the CIR forecasts for the four selected start dates, while the third row shows the corresponding sediment volume forecasts. Each column corresponds to one forecast window.

5.4.1.3 Model Comparison Under the Aggregated Strategy

From the results above, linear regression models demonstrate better overall predictive performance, which can be attributed to the limited number of bathymetric surveys and the resulting scarcity of informative data in the HSC region. Furthermore, the disaggregated and aggregated strategies have nearly identical prediction accuracy; however, the disaggregated strategy greatly reduces the number of available timesteps in the test set. Therefore, subsequent analyses in this section focus exclusively on the aggregated strategy, which provides comparable predictive performance while allowing a more flexible and expanded test set. Within this framework, more advanced LSTM-based models are introduced for further comparison. The following five approaches are compared. **(1) LSTM-Volume**: This model

Table 5.3: HSC region: Evaluation metrics for cumulative sediment volume prediction under different modeling strategies using the three most sensitive gauges (MLR). Columns correspond to four selected forecast start dates and the full test set. Best values for the full test set are highlighted in bold.

Scenario	Metric	2022-01-19	2023-01-26	2023-05-05	2023-09-09	Full test set
Scenario 1 *(Disaggregated)*	RMSE	12,294	13,003	57,246	41,596	41,002
	nRMSE	0.0028	0.0054	0.0163	0.0106	**0.0074**
	MAPE (%)	0.23	0.41	1.37	0.67	**0.58**
Scenario 2 *(Aggregated)*	RMSE	12,294	15,907	26,843	26,116	**39,806**
	nRMSE	0.0028	0.0067	0.0076	0.0067	0.0087
	MAPE (%)	0.23	0.59	0.56	0.47	0.65

uses an LSTM architecture trained directly on cumulative sediment volume data. The input features are the same as those used in the other models. **(2) MLR**: The MLR model is trained to predict the CIR data. The cumulative sediment volume is then calculated from the predicted CIR values. **(3) LSTM**: The LSTM model is designed to forecast the CIR data, which is subsequently used to compute the cumulative sediment volume. **(4) CNN-LSTM**: This model combines CNN and LSTM architectures to predict the CIR data. The predicted CIR values are then employed to estimate the cumulative sediment volume. **(5) TCN-LSTM**: Integrating TCN and LSTM models, this approach predicts the CIR data from which the cumulative sediment volume is derived.

Figure 5.16 presents four examples of the predicted CIR and the corresponding back-calculated cumulative sediment volume for the entire HSC region obtained using different models. Table 5.4 summarizes the associated performance metrics. The observed CIR and sediment volume trends are presented, along with a baseline model in which sediment volume is directly predicted using an LSTM model (i.e., LSTM-Volume).

The middle row of Figure 5.16 highlights CIR forecasts, which serve as the primary drivers of cumulative volume predictions under the aggregated strategy. MLR produces comparatively smooth and linear CIR trends, whereas deep learning–based models (LSTM, CNN–LSTM, and TCN–LSTM) show more flexible temporal dynamics. These differences are important because even small systematic biases in CIR can accumulate over time and lead to large errors in the predicted sediment volume. The bottom row of Figure 5.16 illustrates the resulting sediment volume forecasts. Consistent with Table 5.4, MLR achieves the best overall performance on the full test set, producing the lowest RMSE (39,806), nRMSE (0.0088), and MAPE (0.66). This result indicates that even with its simple structure, the linear model is able to produce more stable and robust volume estimates over long prediction horizons

when using the aggregated strategy. Among deep learning models, CNN–LSTM outperforms both standard LSTM and TCN–LSTM, achieving lower full test set errors (RMSE = 93,143; nRMSE = 0.0174; MAPE = 1.43).

It should be noted that the LSTM Volume model shows the weakest generalization performance, with much higher errors on the full test set, suggesting that directly predicting cumulative volume without explicitly modeling CIR leads to greater error accumulation over time. In addition to the study above, the error growth is also of concern. Figure 5.17 shows how sediment volume forecast error grows as the prediction horizon increases (day 0 to day 30) for the HSC region. Several findings are deduced from this figure:

- **Short-horizon vs long-horizon trade-off (MLR):** MLR is much better than other models for near-term forecasts (0 to 20 days), but it becomes unreliable for longer horizons. This is consistent with a linear model producing small systematic biases in CIR that don't look great at day 0 to 20, but accumulate rapidly once you integrate over many days.

- **Why the three rate-based LSTM-based models look similar:** When the input information is limited and the evaluation is on aggregated volume, different architectures can end up learning very similar rate dynamics, especially if the true signal is smooth and the main challenge is avoiding bias rather than extracting complex patterns.

- **A practical strategy:** If your use case cares about relatively short-term forecasting (less than 20 days), MLR is the better choice. If you are more into longer planning horizons (e.g., larger than 30 days), use CNN-LSTM / TCN-LSTM / LSTM (rate-based) over MLR.

5.4.1.4 Uncertainty Quantification of LSTM-Based Models

In this section, we focus on applying uncertainty quantification to the LSTM-based models (i.e., LSTM, CNN-LSTM, TCN-LSTM) for the HSC region using the aggregated strategy. Figure 5.18 illustrates representative prediction samples for both CIR and cumulative sediment volume at four different timesteps. For CIR predictions (top row), all three deep learning models generate wide uncertainty bands. They are capable of reflecting short-term variability and noise in daily dynamics. The uncertainty bands naturally widen when propagated to cumulative sediment volume forecasts (bottom row of Figure 5.18).

Figure 5.19 further evaluates the reliability of the probabilistic forecasts using calibration curves. The CNN-LSTM and TCN-LSTM models demonstrate calibration curves that lie

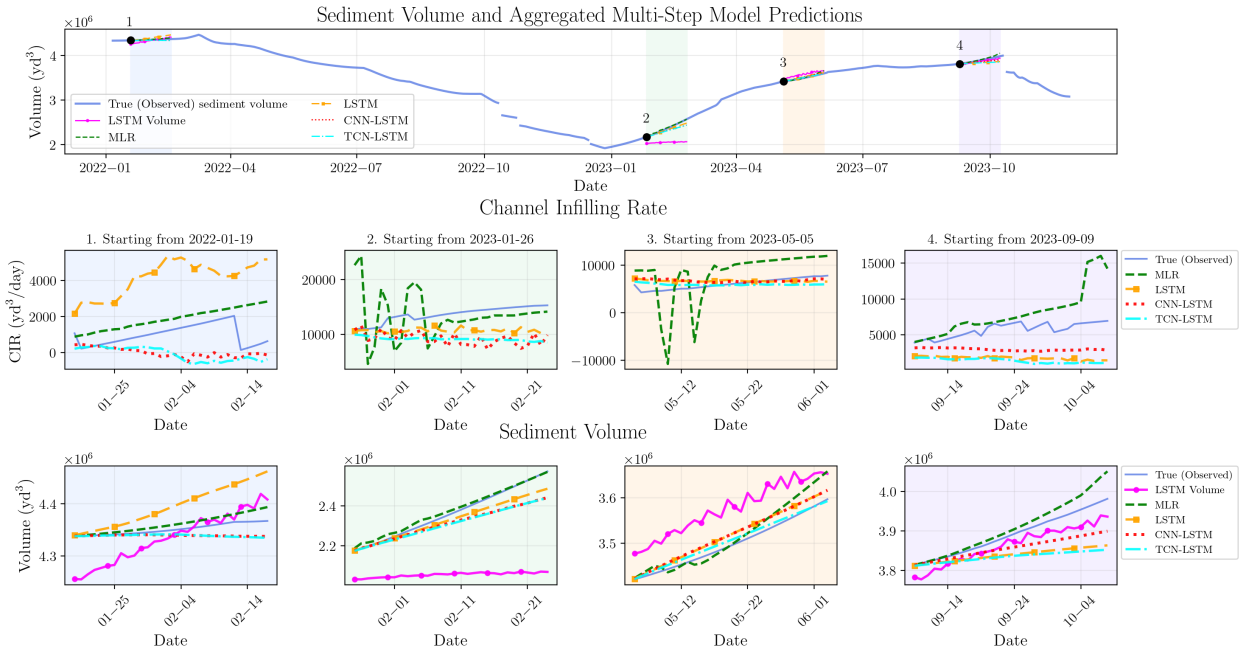


Figure 5.16: HSC region: Forecasting of cumulative sedimentation volume under the aggregated modeling strategy, illustrated for four selected forecast start dates and compared across different modeling approaches. The top row shows the overall sediment volume time series together with the forecast windows. The second row presents the CIR forecasts for the four selected start dates, while the third row shows the corresponding sediment volume forecasts. Each column corresponds to one forecast window.

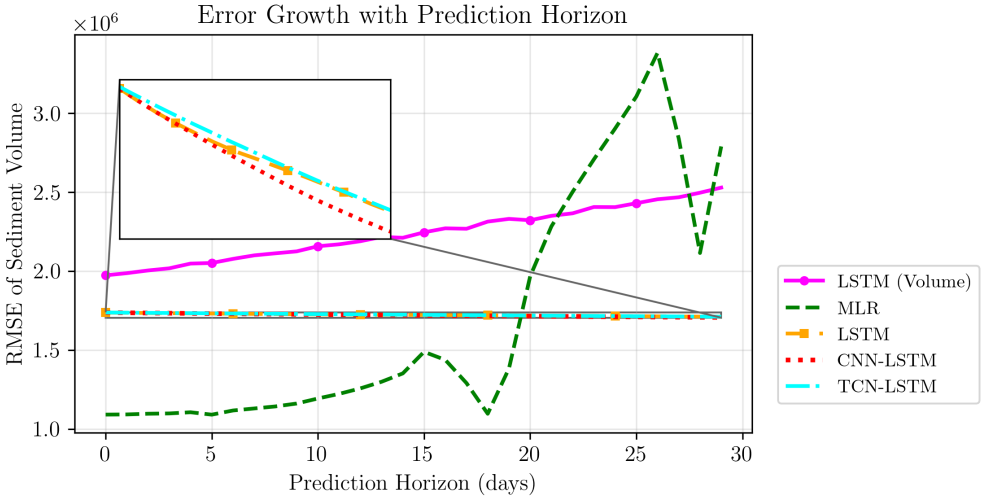


Figure 5.17: HSC region: Cumulative sedimentation volume forecasting error growth with prediction horizon using different approaches.

Table 5.4: HSC region: Evaluation metrics for cumulative sediment volume prediction under the aggregated modeling strategy, compared across different modeling approaches. Columns correspond to four selected forecast start dates and the full test set. For the full test set, the best value for each metric is highlighted in bold.

Method	Metric	2022-01-19	2023-01-26	2023-05-05	2023-09-09	Full test set
LSTM Volume	RMSE	40,876	328,469	78,217	29,865	884,755
	nRMSE	0.0094	0.1389	0.0224	0.0077	0.1167
	MAPE (%)	0.75	12.90	2.20	0.71	11.02
MLR	RMSE	12,294	15,907	26,843	26,116	**39,806**
	nRMSE	0.0028	0.0067	0.0077	0.0067	**0.0088**
	MAPE (%)	0.23	0.59	0.56	0.47	**0.66**
LSTM	RMSE	52,259	42,951	20,536	63,507	97,041
	nRMSE	0.0120	0.0182	0.0059	0.0163	0.0191
	MAPE (%)	1.02	1.37	0.55	1.34	1.57
CNN–LSTM	RMSE	16,128	66,579	20,589	43,083	93,143
	nRMSE	0.0037	0.0282	0.0059	0.0111	0.0174
	MAPE (%)	0.28	2.12	0.56	0.88	1.43
TCN–LSTM	RMSE	16,806	68,710	8,832	69,497	97,350
	nRMSE	0.0039	0.0291	0.0025	0.0179	0.0191
	MAPE (%)	0.29	2.27	0.23	1.46	1.57

closer to the perfect calibration line across a wide range of confidence levels, particularly in the mid-confidence range (approximately 0.3–0.8). In contrast, the LSTM model consistently performs worse than the other two, especially at higher confidence levels, indicating over-confident uncertainty estimates.

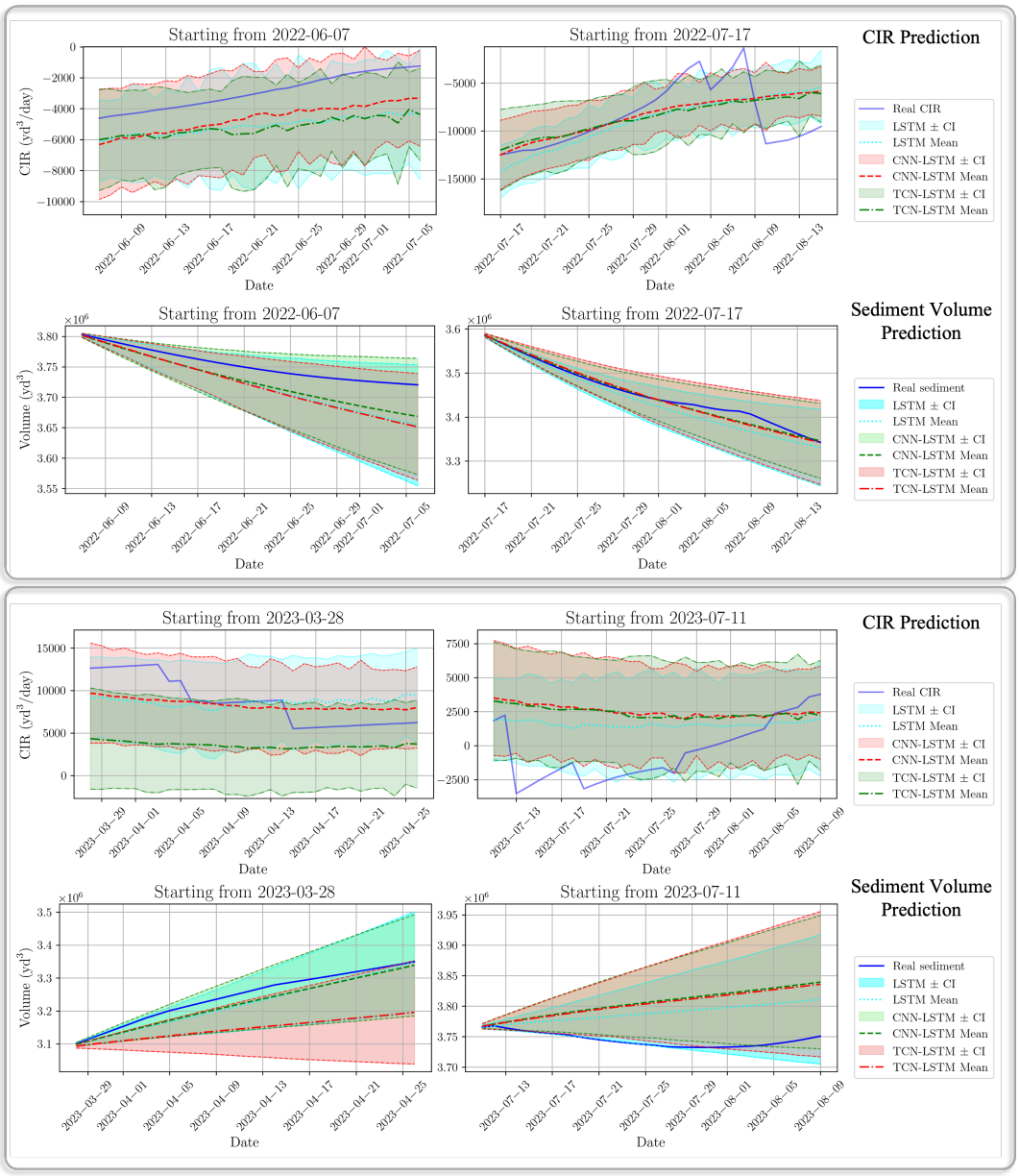


Figure 5.18: HSC region: Cumulative sedimentation volume forecasting uncertainty quantification. HSC region: Comparison of CIR forecasts and corresponding region-scale cumulative sedimentation volume predictions with uncertainty bounds, shown for four different time instances and compared across different modeling approaches.

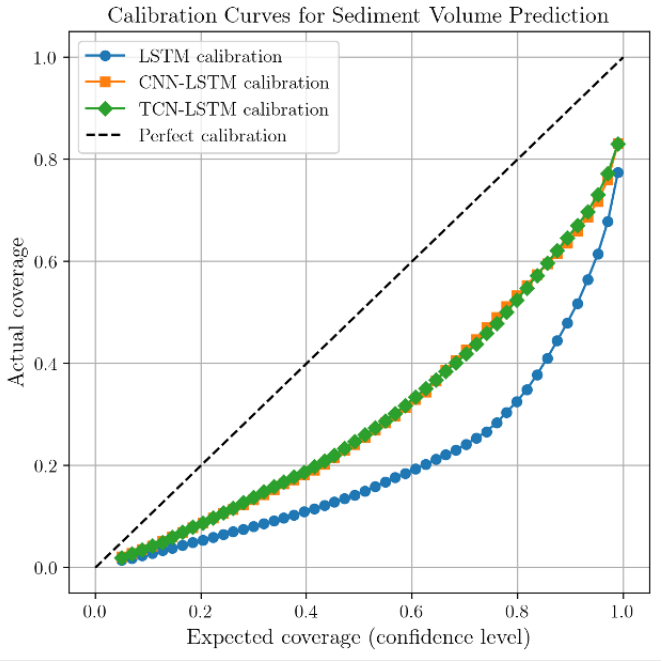


Figure 5.19: HSC region: Cumulative sedimentation volume forecasting calibration curve

5.4.2 The Southwest Pass (SWP)

This section discusses the results and observations inferred from the prediction outcomes in the SWP region. Figure 5.20 shows four reach samples of the sediment volume and CIR in the SWP region. The average number of surveys is noticeably higher than that observed in the HSC region. The following observations are drawn from the results obtained for the SWP region.

5.4.2.1 Disaggregated Strategy

Figure 5.21 illustrates the predicted sedimentation volume and CIR at the reach level for a representative reach (CEMVN_SW_05_SWP_01) in the SWP region at four different time instances. This reach is one of the most frequently dredged and most intensively surveyed locations within the SWP, making it a particularly informative test case for evaluating model performance under data-rich conditions. Table 5.5 reports the error metrics corresponding to these predictions. The full test set consists of 696 filtered time steps, covering the period from January 1, 2022, to December 31, 2023.Based on Figure 5.21 and Table 5.5, we observe that LSTM proves better suited for the SWP region than ULR or MLR, which yield poor predictive performance with significantly larger errors. Since the SWP region is richly surveyed and captures nonlinearities in the sedimentation process, using an advanced

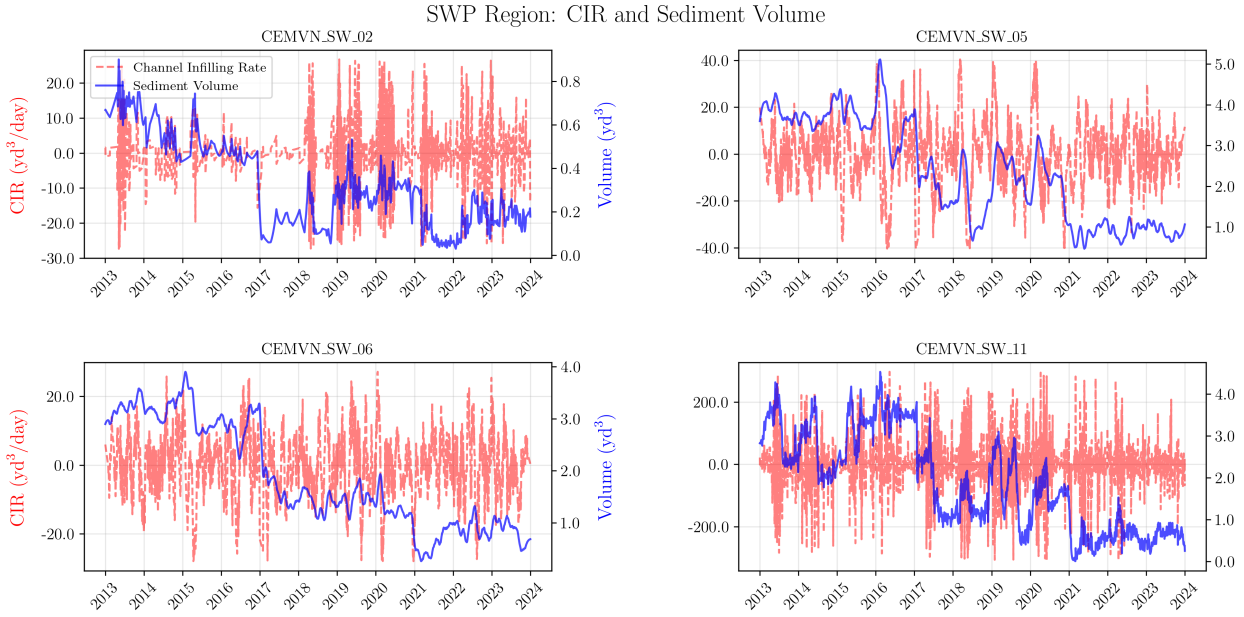


Figure 5.20: Sedimentation volume and CIR for representative reaches in the SWP region (The legend for all subplots is identical to that of the top-left figure).

regression technique with the capacity to learn such nonlinearities and patterns in the data (such as LSTM) is advantageous, particularly for capturing curvatures and inflection points in the sedimentation data. In contrast, a simple regression technique like linear regression is incapable of learning these patterns. Unlike the HSC region, where predictive superiority depended primarily on aligning with the nature of the interpolated training data, in the SWP region it arises from better learning of the underlying sedimentation dynamics. The following two conclusions are made considering LSTM as the chosen regression technique.

The predicted sedimentation volume using the LSTM model at the reach level is evaluated under two configurations based on the most sensitive upstream gauges. In the first configuration, three gauges are used, while in the second configuration, eight gauges are incorporated. Table 5.6 summarizes the corresponding error metrics. The full test set consists of 696 filtered time steps, covering the period from January 1, 2022, to December 31, 2023. Based on the results summarized in Table 5.6, the eight-gauge configuration generally achieves better predictive performance over the full test period in terms of nRMSE and MAPE. However, when examining individual prediction instances, the performance differences between the two configurations are less consistent, with each configuration outperforming the other in some cases. Overall, incorporating a limited number of additional gauge observations appears beneficial. Since the LSTM model can effectively handle higher-dimensional inputs, and because the SWP dataset derived from frequent surveys captures the underlying sedi-

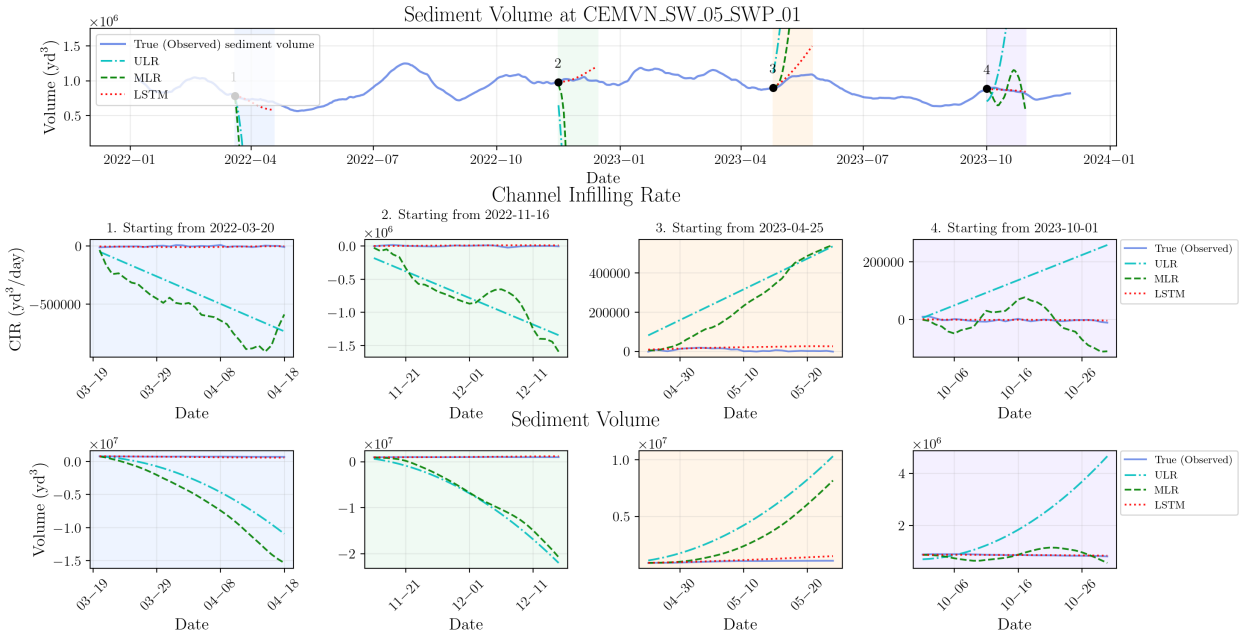


Figure 5.21: SWP region: Forecasts of CIR and sediment volume at a representative individual reach under the disaggregated modeling strategy, compared across three modeling approaches. The top row shows the overall sediment volume time series together with the forecast windows. The second row presents the CIR forecasts for four selected forecast start dates, while the third row shows the corresponding sediment volume forecasts. Each column corresponds to one forecast window.

Table 5.5: SWP region: Evaluation metrics for reach-level predictions at a representative reach under the disaggregated modeling strategy, compared across different modeling approaches. Columns correspond to four selected forecast start dates and the full test set. Best values for the full test set are highlighted in bold.

Method	Metric	2022-03-20	2022-11-16	2023-04-25	2023-10-01	Full test set
ULR	RMSE	5,520,827	11,339,814	4,539,048	1,741,658	10,463,000
	nRMSE	7.5616	11.1807	4.4317	2.0111	7.8219
	MAPE (%)	590.62	887.58	340.00	151.30	612.07
MLR	RMSE	8,128,286	10,394,928	3,013,574	179,755	34,662,213
	nRMSE	11.1329	10.2491	2.9423	0.2076	11.6732
	MAPE (%)	898.81	796.47	192.88	18.16	864.63
LSTM	RMSE	76,599	92,002	186,806	15,158	**140,650**
	nRMSE	0.1049	0.0907	0.1824	0.0175	**0.1373**
	MAPE (%)	8.64	6.47	13.07	1.49	**11.49**

mentation dynamics, the inclusion of sensitive upstream gauge information likely provides meaningful supplementary signals that enhance prediction accuracy.

Table 5.6: SWP region: Evaluation metrics for reach-level predictions at a representative reach under the disaggregated modeling strategy using different sets of sensitive gauges (LSTM). Columns correspond to four selected forecast start dates and the full test set. Best values for the full test set are highlighted in bold.

Input	Metric	2022-02-06	2022-07-15	2023-02-03	2023-08-28	Full test set
Top 3 most sensitive gauges	RMSE	235,650	65,035	248,821	55,733	152,711
	nRMSE	0.2366	0.0555	0.2326	0.0800	0.1403
	MAPE (%)	16.89	4.78	19.97	6.47	12.15
Top 8 most sensitive gauges	RMSE	44,253	109,620	65,651	110,977	**140,650**
	nRMSE	0.0444	0.0936	0.0614	0.1593	**0.1374**
	MAPE (%)	3.84	7.16	5.50	11.93	**11.49**

5.4.2.2 Comparison Between Disaggregated and Aggregated Strategies

Figure 5.22 illustrates the predicted cumulative sedimentation volume using LSTM for both Modeling *Scenario 1* (disaggregated strategy) and Modeling *Scenario 2* (aggregate strategy). Table 5.7 reports the corresponding error metrics. The full test set consists of 629 filtered time steps, covering the period from January 1, 2022, to December 31, 2023. Considering the full test set, *Scenario 1* performs marginally better than *Scenario 2*. Of the eight time instances reported, *Scenario 1* outperforms *Scenario 2* in four cases, while *Scenario 2* performs better in the remaining four. On balance, *Scenario 1* does slightly better overall. In this disaggregated scenario, each individual reach prediction generally maintains consistent accuracy, with errors showing little variation across neighboring reaches. For example, the prediction results for reach IDs SWP_04, SWP_05, and SWP_06 are relatively similar due to their spatial proximity and shared hydrological patterns. This level of accuracy is consistent across most reaches, though slightly worse for reaches such as SWP_01 and SWP_11, where data are more sparse. Therefore similarly to the HSC region, for the SWP region, the clear conclusion is that cumulative volume prediction depends less on selecting the scenario and more on choosing the appropriate regression technique (variants of LSTM models over ULR or MLR).

5.4.2.3 Model Comparison Under the Aggregated Strategy

From the results above, LSTM-based models demonstrate much better overall predictive performance, which can be attributed to the larger number of bathymetric surveys and the

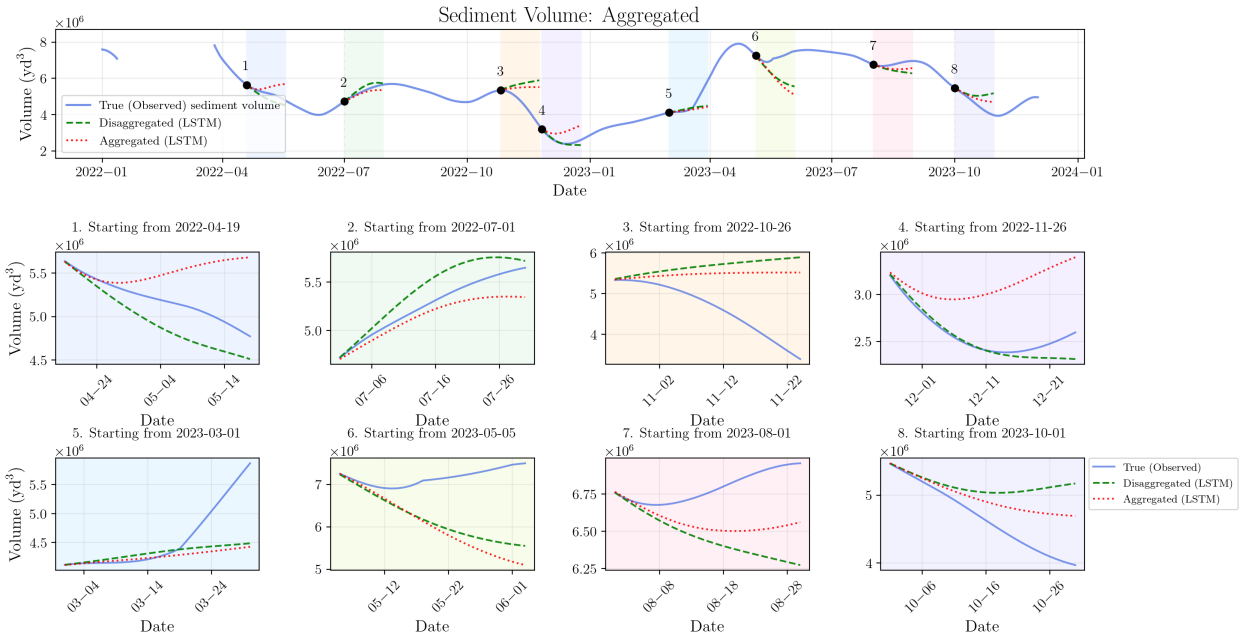


Figure 5.22: SWP region: Forecasting of cumulative sedimentation volume aggregated over all reaches using two modeling strategies, illustrated for four selected forecast start dates using the LSTM model with the eight most sensitive gauges. The top row shows the overall sediment volume time series together with the forecast windows. The second row presents the CIR forecasts for the four selected start dates, while the third row shows the corresponding sediment volume forecasts. Each column corresponds to one forecast window.

Table 5.7: SWP region: Evaluation metrics for cumulative sediment volume predictions under two modeling strategies using the eight most sensitive gauges (LSTM). The eight forecast windows are shown in two blocks for readability, and the overall performance over the full test set is reported in the final column.

Scenario	Metric	2022-04-19	2022-07-01	2022-10-26	2022-11-26
Scenario 1 *(Disaggregated)*	RMSE	271,367	178,091	1,288,059	99,201
	nRMSE	0.0521	0.0338	0.2781	0.0382
	MAPE (%)	4.70	2.94	25.22	2.57
Scenario 2 *(Aggregated)*	RMSE	438,511	147,380	1,070,186	576,597
	nRMSE	0.0842	0.0280	0.2310	0.2225
	MAPE (%)	6.51	2.25	20.58	20.69

Scenario	Metric	2023-03-01	2023-05-05	2023-08-01	2023-10-01	Full test set
Scenario 1 *(Disaggregated)*	RMSE	516,189	1,165,344	399,398	624,958	**752,475**
	nRMSE	0.1132	0.1626	0.0588	0.1337	**0.1285**
	MAPE (%)	5.91	13.29	4.77	11.25	**11.27**
Scenario 2 *(Aggregated)*	RMSE	554,622	1,366,200	269,922	389,097	771,546
	nRMSE	0.1216	0.1907	0.0397	0.0832	0.1317
	MAPE (%)	6.07	15.04	3.27	7.13	11.38

resulting rich data in the SWP region compared to the HSC region. Furthermore, the disaggregated and aggregated strategies have nearly identical prediction accuracy; Similar to the HSC region, the disaggregated strategy greatly reduces the number of available timesteps in the test set. Therefore, subsequent analyses in this section focus exclusively on the aggregated strategy. Within this framework, more advanced LSTM-based models are introduced for further comparison.

Figure 5.23 presents four representative examples of the predicted CIR and the corresponding back-calculated cumulative sediment volume for the entire SWP region using different modeling approaches. Table 5.8 summarizes the associated performance metrics. The figure shows the observed CIR and sediment volume evolution, together with model predictions obtained under the aggregated forecasting framework.

The middle row of Figure 5.23 highlights CIR predictions, which act as the primary drivers of cumulative sediment volume forecasts under the aggregated strategy. Similar to the HSC region, the MLR produces relatively smooth predictions, whereas the LSTM–based models (i.e., LSTM, CNN–LSTM, and TCN–LSTM) capture more complex temporal variability more accurately. The bottom row of Figure 5.23 shows the corresponding sediment volume forecasts. The baseline LSTM Volume model shows huge deviations from observations across

multiple starting times, which is consistent with its poor full test set performance reported in Table 5.8 (RMSE = 4,997,062; nRMSE = 1.0555; MAPE = 106.66). Among CIR-based aggregated models, linear regression models like MLR can't predict the sediment volume accurately compared to LSTM-based models. Among deep learning models, TCN–LSTM achieves the best overall performance (RMSE = 764,065; nRMSE = 0.1240; MAPE = 10.63).

In addition to the study above, the error growth is also investigated. Figure 5.24 shows how sediment volume forecast error grows as the prediction horizon increases (day 0 to day 30) for the SWP region. Several findings from this figure:

- **Linear regression is not suitable for this case:** From the results of MLR, the error increases drastically at longer horizons, reflecting strong error accumulation from small systematic biases in CIR. The low accuracy makes this method not suitable for the regions like SWP.

- **Direct volume prediction produces large error:** It can be observed that the direct sediment volume prediction model fails to produce reliable results. This limitation arises because, during normalization, the training set does not include the test set values; owing to a datum shift, the test set values are almost entirely outside the range of the training data. As a result, the model is unable to generate accurate predictions. These findings indicate that direct sediment volume prediction is not suitable for this case study and further highlight the advantages of the proposed methodology of using CIR as the predictor under such conditions.

- **A practical strategy:** For regions like SWP that have rich survey data, it is recommended to use deep learning CIR prediction models, such as the TCN-LSTM model used in this case study.

5.4.2.4 Uncertainty Quantification of LSTM-Based Models

In this section, we focus on applying uncertainty quantification to the LSTM-based models for the SWP region using the aggregated strategy. Figure 5.25 illustrates representative prediction samples for both CIR and cumulative sediment volume at four different timesteps. For CIR predictions (top row), similar to the results in the HSC region, all three deep learning models generate wide uncertainty bands. They are capable of reflecting short-term variability and noise in daily dynamics. The uncertainty bands naturally widen when propagated to cumulative sediment volume forecasts (bottom row of Figure 5.25). It is observed that, although the SWP region has more complex nonlinear behavior than the HSC region, the

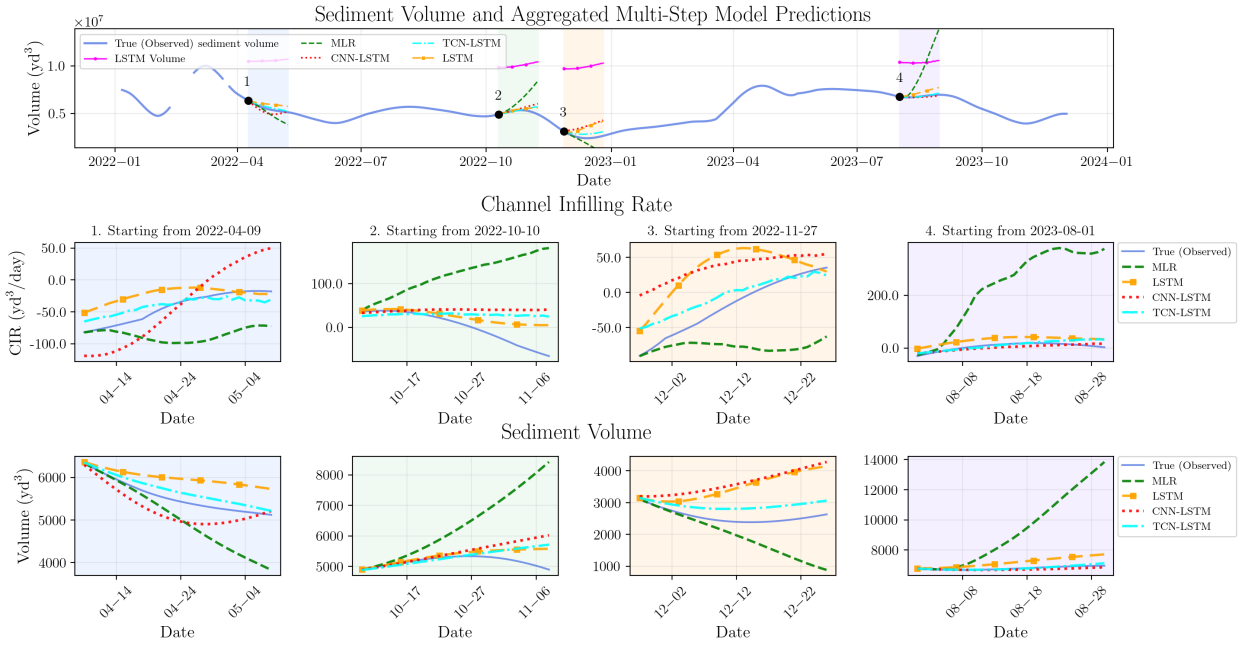


Figure 5.23: SWP region: Forecasting of cumulative sedimentation volume under the aggregated modeling strategy, illustrated for four selected forecast start dates and compared across different modeling approaches. The top row shows the overall sediment volume time series together with the forecast windows. The second row presents the CIR forecasts for the four selected start dates, while the third row shows the corresponding sediment volume forecasts. Each column corresponds to one forecast window.

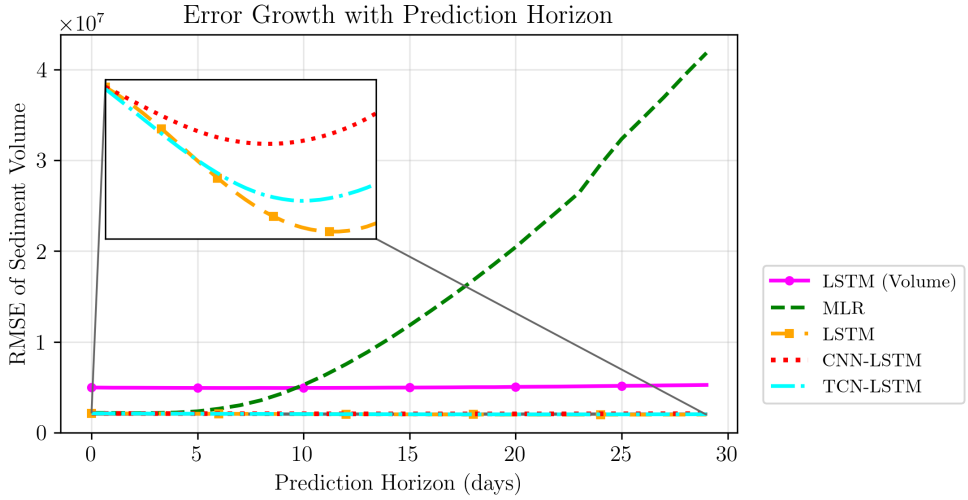


Figure 5.24: SWP region: Cumulative sedimentation volume forecasting error growth with prediction horizon using different approaches.

Table 5.8: SWP region: Evaluation metrics for cumulative sediment volume prediction under the aggregated modeling strategy, compared across different modeling approaches. Columns correspond to four selected forecast start dates and the full test set. Best values for the full test set are highlighted in bold.

Method	Metric	2022-04-09	2022-10-10	2022-11-27	2023-08-01	Full test set
LSTM Volume	RMSE	5,010,233	4,888,630	7,326,433	3,575,668	4,997,062
	nRMSE	0.9021	0.9465	2.8484	0.5264	1.0555
	MAPE (%)	90.73	94.69	286.90	52.65	106.66
MLR	RMSE	662,289	1,599,133	807,892	3,344,661	1,922,644
	nRMSE	0.1193	0.3096	0.3141	0.4924	0.4692
	MAPE (%)	9.38	22.89	23.40	36.10	34.60
LSTM	RMSE	483,983	266,726	1,054,880	448,818	771,546
	nRMSE	0.0871	0.0516	0.4101	0.0660	0.1317
	MAPE (%)	8.22	3.56	37.04	5.78	11.38
CNN–LSTM	RMSE	319,847	444,350	1,149,633	85,591	843,126
	nRMSE	0.0576	0.0860	0.4470	0.0126	0.1409
	MAPE (%)	5.15	5.56	41.84	1.01	12.22
TCN–LSTM	RMSE	162,343	296,768	364,625	50,991	**764,065**
	nRMSE	0.0292	0.0575	0.1418	0.0075	**0.1240**
	MAPE (%)	2.78	3.69	13.64	0.46	**10.63**

models are still able to accurately forecast sediment volume trends with uncertainty bounds that remain within an acceptable range.

Figure 5.26 further evaluates the reliability of the probabilistic forecasts using calibration curves. Three models show similar results, while the LSTM is slightly closer to the perfect calibration curve. Compared to Figure 5.19, the calibration curve indicates that the overall uncertainty quantification of the LSTM-based models in the SWP region is closer to the perfect calibration curve, suggesting reduced overconfidence relative to the results obtained for the HSC region.

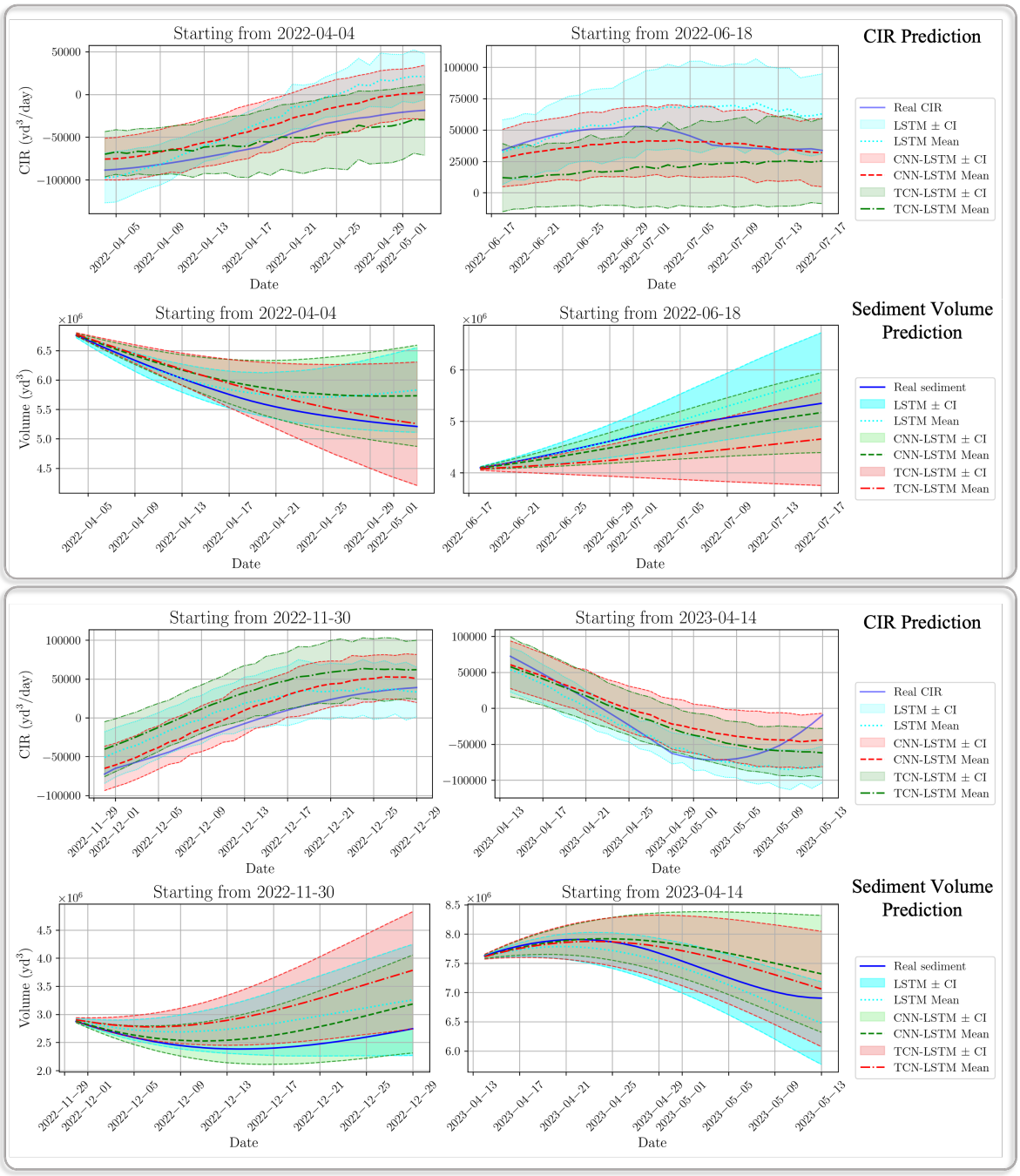


Figure 5.25: SWP region: Cumulative sedimentation volume forecasting uncertainty quantification. HSC region: Comparison of CIR forecasts and corresponding region-scale cumulative sedimentation volume predictions with uncertainty bounds, shown for four different time instances and compared across different modeling approaches.

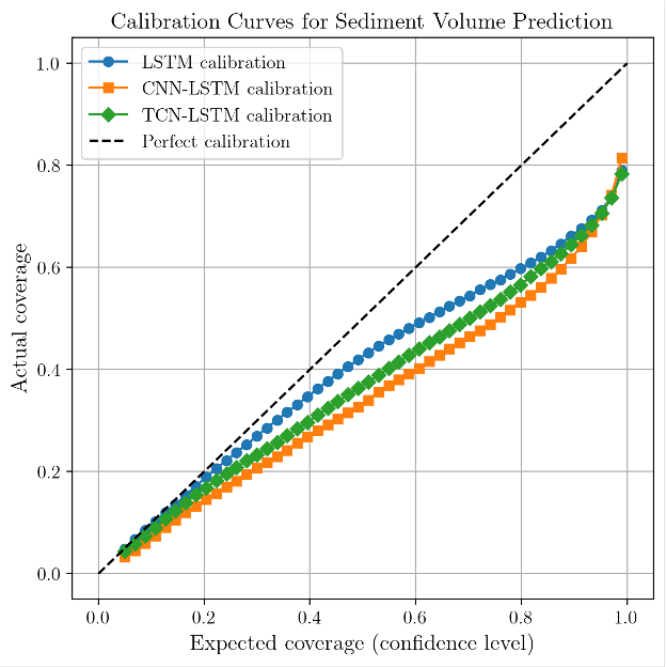


Figure 5.26: SWP region: Cumulative sedimentation volume forecasting calibration curve

5.5 Conclusion

This work proposed a CIR-based forecasting framework for sedimentation volume prediction in navigable river reaches, shifting the modeling focus from directly predicting cumulative volumes to forecasting the CIR. By reframing the problem in terms of CIR, the framework improves sensitivity to short-term sediment dynamics, reduces the influence of abrupt external interventions such as dredging, and improves interpretability relative to raw volume forecasts. The dual-modeling strategy, using linear regression in relatively data-scarce reaches such as the Houston Ship Channel (HSC) and LSTM-based models in data-rich reaches such as the Southwest Pass (SWP, which had on average 5 times more bathymetric surveys between 2012-2024 than HSC), demonstrated that model selection must align with survey density and data richness.

Results from HSC confirm that simpler linear regression approaches are more effective when bathymetric surveys are sparse and interpolation dominates the data structure, whereas LSTM-based models yield poor performance under such conditions. Conversely, in SWP, where dense survey records capture non-linear sedimentation dynamics, LSTM-based models significantly outperformed linear models, successfully learning complex patterns and improving prediction accuracy. Sensitivity analysis further highlighted that while additional upstream gauge data added little value in HSC, it improved predictions in SWP by providing complementary hydrologic signals. Among them, the CNN-LSTM and TCN-LSTM

architectures provide consistent, though small-scale improvements over the standard LSTM by incorporating convolutional feature extraction mechanisms that better capture local temporal patterns and multiscale dependencies in CIR. Moreover, differences between disaggregated (reach-level) and aggregated (system-level, consisting of numerous reaches) modeling strategies were marginal, indicating that forecasting accuracy depends more on the choice of regression technique than on scenario selection.

Beyond point predictions, this study integrates uncertainty quantification for all LSTM-based models using MC-dropout. The resulting predictive intervals provide valuable insights into forecast confidence and error growth with prediction horizon. The Uncertainty Quantification (UQ) results indicate that CNN-LSTM and TCN-LSTM models exhibit slightly improved calibration and more stable uncertainty bounds compared to the standard LSTM, particularly for longer lead times. These findings highlight that even when point-prediction accuracy is similar, differences in uncertainty behavior can be critical for risk-aware decision-making in sediment management.

Overall, the study underscores the importance of adapting predictive models to local data conditions and sediment dynamics. By integrating CIR-based formulation, gauge-informed feature selection, and adaptive modeling, the proposed framework offers a practical step toward robust, data-driven shoaling forecasts. These results have direct implications for proactive dredging management, enabling longer lead times, more efficient allocation of resources, and improved resilience of navigation infrastructure. As forecasting performance is significantly dependent on the availability of past bathymetric data, the findings also shed light on the need for comprehensive surveying of waterway segments that a critical for supply chains, and the use of modern methods to complement traditional surveying methods, like crowdsourced (voluntary) bathymetry.

Acknowledgment

This work was supported by the United States Army Corps of Engineers through the U.S. Army Engineer Research and Development Center Research Cooperative Agreement W912HZ24C0044. The support is gratefully acknowledged.

CHAPTER 6

A Recursive Shoaling Forecasting Framework with Shifted Delays and Its Application to the Southwest Pass in the United States

6.1 Introduction

Navigation channels form a vital transportation network that supports the economy and national security. They are widely used for safe and efficient exchange of goods and for the movement of vessels and passengers. In this regard, the US Army Corps of Engineers (USACE) maintains a total of 25,000 miles of coastal and inland navigation waterways, comprised of 13,000 miles of coastal channels and 12,000 miles of inland waterways [197]. Maintenance dredging activities ensure that these systems remain clear, accessible, and available for commercial and recreational vessels.

Maintenance dredging is a periodic underwater excavation that aims to restore congress-authorized navigable water depths by removing accumulated sediments and other materials restricting the depth or even the width of navigation channels. Each year, USACE dredging activities remove more than 210 million cubic yards of material across approximately 25,000 miles of USACE-maintained coastal and inland waterways [198]. These activities cost more than one billion U.S. dollars annually and involve an average of 760 dredging-related contract actions [199, 145]. To perform these dredging operations, starts planning navigation dredging projects years in advance. Dredging specifications, such as the estimation of the volume of material to be removed, become more accurate as the planned maintenance operation approaches. Early prediction of dredging volumes is therefore essential for estimating dredged material placement requirements, selecting appropriate dredging equipment, and ensuring the timely arrival of the dredging fleet. It also helps reduce disruptions to maritime operations and supports effective execution of dredging budgets [199, 200, 144]. This need formulates the foundation for the problem that this paper is addressing. In demonstrating

the importance of dredging activities, the Southwest Pass (SWP) channel reflects one of the crucial navigable channels in the U.S. Enabling the transit of over 250 million tons of cargo on a yearly basis, the SWP channel distinguishes itself as the nation's top waterway for deep-draft vessels. Between the years 2015 to 2019, USACE spent an average of U.S. $88 million on dredging activities per year in this area. Despite these dredging plans and efforts, the SWP experienced a total of 205 days of imposed draft restrictions in 2019. These restrictions were the outcome of unexpected shoaling patterns, resulting in insufficient clearance for vessels. In turn, these restrictions subsequently introduced disruptions to shipping activities for a daily average of 13 oceangoing vessels, adding significant cost burdens to shippers. Therefore, there is an indispensable need for accurate shoaling forecasts in the SWP channel [182].

In predicting dredging volumes, USACE has resorted to both qualitative and quantitative approaches. Qualitatively, USACE relied on the expertise of dredging managers and engineers to estimate the year-on-year variation in dredging [145, 144]. For example, the New Orleans District (MVN) project managers employ upstream river discharge rates combined with heuristics drawn from past experience to predict the shoaling condition in SWP. Nevertheless, this method yields only 14 days of predictive lead time at best, compromising the timely availability of dredging equipment [144, 182]. In an additional effort to obtain a comprehensive understanding of the dynamics of the shoaling process, a hybrid quantitative and qualitative method was formulated [182]. This method combines several USACE enterprise datasets. These include hydrographic surveys stored in the USACE eHydro system, which document channel bathymetric conditions [201], and the USACE Dredging Quality Management (DQM) geospatial data archive, which records dredge dates, dredge locations, and estimated dredged volumes [182]. It was also applied to the SWP channel, allowing trackage of sediment movement through sequential bathymetric surveys and attributing it to dredging operations as documented in DQM entries. More specifically, it enables the analysis of dredging schedules, dredged volumes, and their effects on bathymetric surveys. This provides a more robust understanding of rapid-onset shoaling, which can cause sudden and significant reductions in channel depth. Similarly, other enterprise datasets were superimposed to determine limiting depths and side-slope clearances along the channel. More precisely, time-stamped vessel position reports containing real-time vessel dimensions, including length, beam, and draft, were combined with bathymetric survey data. This integration enabled channel evaluation, dredging volume estimation, and prioritization of maintenance dredging requirements [171]. These two studies [182, 171] demonstrate the unique opportunity of leveraging USACE datasets for maintaining federal waterways, being a key component of the districts' navigation missions.

Furthermore, USACE established other key tools for systematic quantitative analyses. In particular, the U.S. Army Engineer Research and Development Center (USACE-ERDC) developed the Corps Shoaling Analysis Tool (CSAT) [202]. This tool embeds a hindcasting algorithm permitting the linear extrapolation of volumetric channel infilling rates, as derived from USACE bathymetric survey data. It then projects sediment volume into the future based on the most recent bathymetric survey. USACE has further optimized the CSAT output in distinct ways in an attempt to achieve more accurate shoaling forecasts. One approach incorporates records of GPS-indicated dredge position and status, together with a net volume change threshold. This helps prevent CSAT from misinterpreting a dredging event as natural scour [200]. This technique was tested in different channels and reaches within districts on the Pacific, Gulf, and Atlantic coasts. Another approach encompasses the merging of the Joint Airborne Lidar Bathymetry Technical Center of Expertise (JALBTCX) National Coastal Mapping Program (NCMP) topobathy lidar data with the hydrographic surveys serving as inputs to the CSAT [203]. This approach helps to account for the shoaling that occurs outside the channel, as these could impact the sediment movement patterns within the channel boundaries as defined by the National Channel Framework (NCF).

However, CSAT does not take into consideration the impact of other factors influencing the driving sedimentation processes. Such factors include the runoff in inland hydrologic systems and the inter-annual effects of coastal processes [145]. As a result, the USACE-ERDC adopted causal models by establishing spatially-informed regression relationships between precipitation-driven inland channel flow and subsequent dredged volumes. Through this approach, the effects of the upstream precipitation-driven streamflow are taken into account. To support a more rigorous data-driven approach for maintaining USACE ship channels, CSAT-derived daily sediment volumes were integrated into statistical and machine learning/artificial intelligence (ML/AI) regression models. These models included univariate Autoregressive Integrated Moving Average (ARIMA), Random Forest (RF), Artificial Neural Networks (ANN), and Long Short-Term Memory (LSTM) models [144]. These models used physics-informed hydrological input variables to forecast daily accumulated sediment volume in the SWP channel, providing significantly longer lead times of 45 days and acceptable accuracy. These favorable results, as yielded by this non-linear ML-based approach, offer a valuable basis for subsequent work and refinement. Thus, this paper builds on the findings of these shoaling ML forecasting models, while focusing on the improvement of the automated multivariate feature selection and LSTM time-series forecasting framework.

A review of ML-based time-series models for shoaling forecasting shows that most studies have focused on predicting the absolute value of the target variable. Examples include suspended sediment load (SSL), suspended sediment concentration [204], turbidity [205], and

sediment volume or deposition rate [148]. In contrast, this study is motivated to exploring non-linear ML methods to forecast the Channel Infilling Rate (CIR) rather than directly forecasting the sediment volume. The CIR refers to the rate at which sediments accumulate or erode within a navigation channel, river reach, or waterway. These processes lead to changes in channel depth and cross-sectional capacity over time. It quantifies how rapidly sedimentation occurs and is typically expressed as volume per unit time. Positive CIR values indicate sediment accumulation, whereas negative values indicate erosion. Focusing on CIR helps the model capture the natural evolution of sedimentation dynamics. It also supports the identification and filtering of external factors that cause non-stationary conditions and abrupt local fluctuations. Such abrupt fluctuations may be the result of dredging activities, extreme meteorological events, or datum changes. This choice is consistent with previous studies that forecast changes in quantities rather than absolute quantities [206]. Therefore, adopting the CIR may improve forecasting performance. For example, optimized deep learning (DL) models achieved improved chlorophyll forecasts when the rate of change (ROC) was used as the model output [207]. In another study, using the ROC led to higher accuracy in streamflow forecasts [208]. Therefore, we expect that utilizing the CIR in this study will minimize carry-on errors [209] in the adopted ML forecasting framework.

Streamflow has also been identified as an important input to sedimentation-based models [144, 204]. Upstream streamflow plays a crucial role, as water discharge not only drives sediment transport but also governs whether particles remain suspended or settled. While previous models incorporated streamflow in real-time, several researchers have emphasized the importance of considering the delayed response in establishing relationships between upstream and downstream. On this matter, lag-time is mostly determined by the travel distance from one point to another [210]. Streamflow travel time from one station to another is influenced by river characteristics and morphology. For example, narrower and deeper channels may require longer travel times than wider and shallower channels [210]. Similarly, sediment size controls the time needed to transport sediments, with coarse constituents requiring longer travel times compared to smaller constituents contributing to turbidity [211]. While ML approaches are characterized by identifying patterns (including lagged patterns) in the data, the time lag phenomenon associated with the hydrological response is often overlooked in data-driven approaches [212]. Therefore, this study explores the incorporation of streamflow, accounting for the effect of travel lag-time, as an additional input variable to the ML forecasting model.

As mentioned, dredging planning requires long lead times and relatively long forecasting horizons, which makes the forecasting problem more challenging to solve. This challenge is related to the trade-off between the forecasting horizon and the forecasting accuracy. Previ-

ous studies employing ML-based time series models have either used single-output or multi-output approaches in solving their multi-step forecasting problem. In this regard, multi-output approaches, such as the multi-input multi-output (MIMO) strategy, retain stochastic dependence among predicted values. They can outperform single-output approaches, such as the direct single-output strategy, which assumes independence among future predictions [213, 214]. Therefore, in this paper, we focus on MIMO-based framework while adopting the ROC in sediment volume. The MIMO strategy produces a time sequence of future values simultaneously, as opposed to predicting one step at a time. However, stochastic dependencies, as retained by the MIMO strategy, may limit forecasting flexibility. Therefore, we additionally propose to optimize the forecasting strategy by developing and applying a Recursive MIMO approach. This Recursive MIMO approach is capable of retaining complex dependencies between the predicted values, especially essential for estimating long-term trends [213, 214]. The proposed approach predicts future CIR values in sub-windows and iteratively uses the set of predictions in one sub-window as part of the input to predict the next window. This method also outperforms the recursive single-output strategy by considering a lower number of iterations (number of iterations corresponds to the number of sub-windows), thus reducing the error accumulation [213]. Furthermore, while the Mean Square Error (MSE) is commonly used with LSTMs to optimize the model prediction results, it treats all predictions equally. This equal weighting may fail to emphasize the importance of accurately predicting unique crucial events such as extreme sedimentation changes. Hence, it is indispensable to introduce alternative loss functions capable of accurately predicting such events. For instance, Ding *et al.* [215] integrated a novel extreme-value loss function. This function adjusts the weights assigned to extreme events and increases the model's attention to such values. Similarly, Man *et al.* [216] designed two loss functions to introduce a heavier penalty for errors in peak runoff predictions than in other areas. As a result, one of the utilized loss functions demonstrated superior performance over the standard loss function. Mirroring these initiatives, this study attempts to refine the LSTM framework by introducing a new loss function.

This paper presents an advancement in sediment volume prediction by developing a robust framework that leverages state-of-the-art ML techniques. Our approach addresses the complexities inherent in modeling geophysical systems with multiple interrelated variables and varying time lags. The contributions of this paper are summarized as follows:

- A comprehensive CIR-based framework is proposed to model the evolution of natural sedimentation. The framework identifies and filters extreme events and integrates a recursive MIMO LSTM model with Global Sensitivity Analysis (GSA)-identified shifted inputs and a weighted loss function. This approach leads to improved sediment

volume prediction.

- A time-resolved GSA is used to identify crucial input variables and determine their optimal time shifts, measured in days. This method of quantifying the lag-time is especially useful for variables like discharge data, where gauge stations vary in distance from the target area. By determining which variables are significant and how much they should be shifted, prediction accuracy of the LSTM model is improved.
- A custom loss function is introduced, specifically designed for the LSTM model to better capture the unique characteristics of the CIR, thereby improving the model's predictive performance.
- The use of a recursive MIMO LSTM model contributes to more accurate prediction results, increasing the overall effectiveness of automated sediment volume prediction.

The remainder of the paper is organized as follows: Section 6.2 provides background on the process of shoaling and reviews current ML research on this topic. Section 6.3 describes the methodology details of the proposed framework. Section 6.4 presents a case study conducted in the SWP region. Section 6.5 presents and discuss details of the results. Finally, Section 6.6 concludes this paper with future research directions.

6.2 Background

6.2.1 Factors Impacting Shoaling Forecasting

Sedimentation is a complex process resulting from a combination of several natural factors and human activities. Figure 6.1 shows the various classes of these factors. Natural factors that influence shoaling are multifaceted and encompass sediment characteristics such as its source and size, natural channel geometry such as its morphology, depth and width, hydrodynamics including physical oceanography such as water flow, tides and waves, and other water properties such as temperature and density. On the other hand, human activities entail several practices such as construction, deforestation, ship traffic, maintenance dredging, and new work dredging (changing channel dimensions).

6.2.1.1 Natural Factors

One of the primary sources of shoaling is sediment generated by the erosion of adjacent islands and material from sediment placement areas, which contributes to elevated sedimentation in navigation channels. Similarly, the scour of the surrounding shallow water bottom

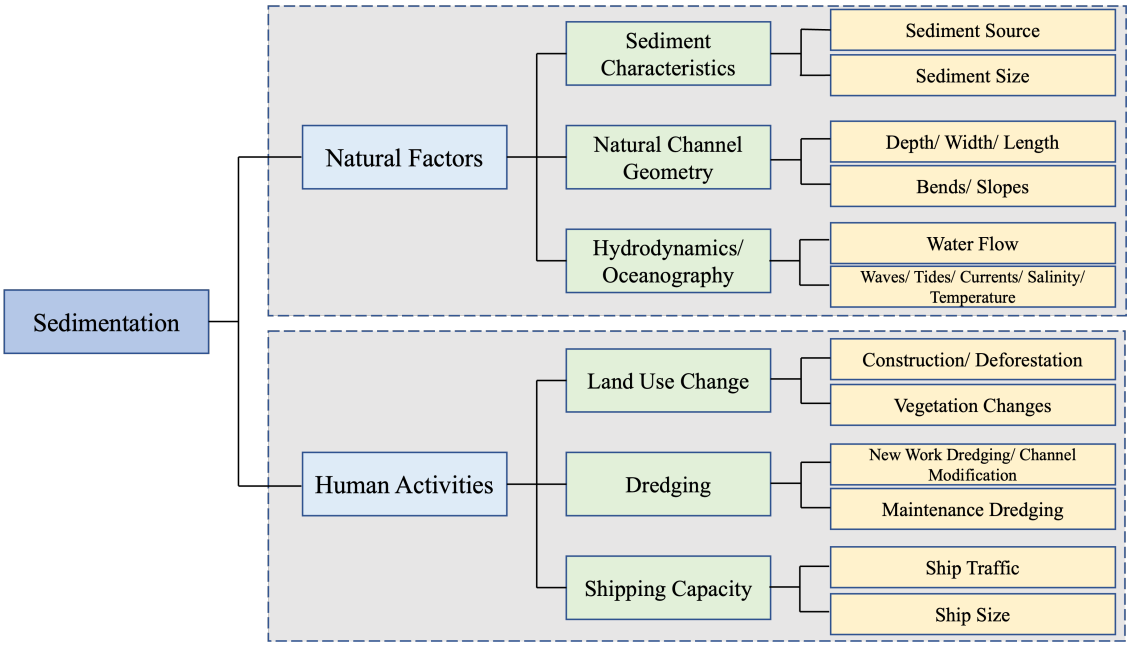


Figure 6.1: Factors affecting sedimentation.

further adds to the deposition of sediments. These two sources were identified as the main suppliers of sedimentation in the Corpus Christi Ship Channel [217]. In this respect, the extent of shoaling is influenced by the composition of the original soils (sand, silt, clay, etc.) that make up these sediment sources [218]. Parent soils are characterized by their grain size, density, and particle shape. Of these, sediment size largely controls settling velocity, which in turn governs key sediment dynamics such as deposition and accumulation. More specifically, larger sediments are relatively heavy, featuring a high settling velocity. Therefore, they are prone to quicker deposition as compared to smaller sediments, which constitute the major proportion of suspended sediment concentration (SSC) in water bodies. Nevertheless, additional physical and chemical processes, such as flocculation, may result in the aggregation of multiple cohesive fine particles, altering the behavior of the original fine sediments by increasing their settling velocity [219].

The interplay of different channel characteristics influences sediment dynamics particularly by affecting water velocities. Channel dimensions such as large widths or depths induce lower flow velocities, leading to heavier deposits [220]. Conversely, flow velocity drops on the inside radius of a channel curve, leading to higher shoaling [221]. For example, while investigating shoaling patterns in the Porto Novo Access Channel, the deposition on the convex margin of the channel was relatively higher than on its concave margin [220]. The influence of sediment characteristics on the CIR is not isolated, but it is strongly mediated by hydrodynamic conditions. Upstream hydrology—including streamflow, stage, and velocity— is a

key determinant to downstream shoaling, such that the interaction between water flow and sediment size governs whether a sediment is transported, deposited, or eroded/re-suspended [222]. According to Hjulstrom's curve, fine sediments are most likely to be deposited under extremely low flow velocities, transported under moderate velocities, and eroded under extremely high velocities [223].

Beyond upstream hydrology, sediment dynamics are further influenced by external forces such as currents, tides, and waves. Tides and currents are closely interrelated: tidal fluctuations induce vertical oscillations of the water column that generate currents, thereby enhancing horizontal sediment transport—an effect particularly pronounced in coastal and estuarine systems. Other factors are also fundamental in driving currents; among these are waves, wind, and water temperature [224]. Waves, in particular, generate longshore currents that move water parallel to the shoreline, while the combined action of waves and tides serves as a major driver of coastal erosion, sediment transport, and the formation of diverse coastal landforms [225, 226]. Tides influence both water levels and currents in rivers, pushing seawater inland and altering flow patterns that can extend far upstream. During the flood tidal phase, sediments are transported landward, either as bedload or in suspension, where they may become entrapped in navigation channels and subsequently deposited in coastal waterways. Conversely, during the ebb tidal phase, sediments are transported seaward. When tidal phase durations are asymmetric, the imbalance produces a net sediment transport in the direction of the dominant phase [227]. Moreover, sediments can remain in suspension for much of the tidal cycle, allowing them to become widely dispersed [228].

Waves are primarily generated by winds and, similar to tides, play a major role in re-suspending sediments, maintaining them in suspension, and redistributing them both vertically and horizontally [229]. Their effect depends on several factors, particularly wave amplitude and wave period [230]; the amplitude itself is often influenced by wind direction [231]. Wave processes also interact with tidal water levels. For instance, Brand *et al.* [229] observed that most wave-induced re-suspension events in the South San Francisco Bay occurred following low tides. Such non-linear interactions between waves and tidal currents enhance bottom shear stress, thereby increasing sediment re-suspension. Likewise, the interaction between winds and currents governs the direction of sediment transport [232].

6.2.1.2 Human Activities

Waves can also be generated by human activities such as ship traffic passing through navigation channels. As shipping traffic increases, additional pressure is exerted on the hydrodynamic conditions, leading to the generation of waves and accompanying currents. Such ship-induced waves are often associated with alterations in sediment transport within

rivers and estuaries. In particular, they cause shoreline erosion, leading to the (re-)suspension and dispersion of sediments [233].

Other human activities that play a vital role in sediment deposition include deforestation. The removal of natural vegetation in lands adjacent to ship channels for urbanization or agricultural purposes accelerates soil erosion. In turn, soil erosion increases the sediment supply to water bodies, including navigation channels, intensifies turbidity, and elevates the likelihood of deposition [218, 234]. Moreover, channel modifications such as deepening or widening—often undertaken to improve navigation and accommodate the increasing size of vessels—can unintentionally raise maintenance demands, as altered geometries may act as sediment traps. While some examples indicate the opposite, channel deepening and widening increased the sedimentation rate in St. Mary's Entrance, FL [174], Freeport, TX, Calcasieu Pass, LA, and Houston-Galveston Entrance and Bay, TX [235].

6.2.1.3 Sedimentation Modeling in Navigation Channels

As a result of their impact on sediment dynamics, the aforementioned variables have been employed in different numerical models developed to estimate sedimentation in navigation channels. Among these numerical models is the Coastal Modeling System (CMS), developed under the Coastal Inlets Research Program (CIRP) at the US Army ERDC. It is primarily designed to model coastal inlets and navigation applications and results from the coupling of two models: (1) a two-dimensional hydrodynamic model (CMS-Flow), which calculates currents, water surface elevation, salinity, and sediment transport; and (2) a wave transformation model (CMS-Wave), which improves the accuracy of wave transformation simulations [236]. CMS was used to simulate and estimate channel shoaling and sedimentation in the Corpus Christi Ship Channel improvement project, using several input variables such as river inflow rates, coastal wind and wave forcing, and water level forcing [217].

Another numerical sedimentation model is the Hydrologic Engineering Center - River Analysis System (HEC-RAS). HEC-RAS was developed by USACE to primarily model water through rivers and channels as well as to simulate sediment transport. This model was used to forecast the future sedimentation along the Lower Puyallup River. Similar to CMS, several sediment-driving factors were collected to develop and calibrate the HEC-RAS model including the flow rates to identify temporal variability of flow duration characteristics associated with the river, the tidal stage historical data to account for the water surface elevation, the channel cross-sections as surveyed by W&H Pacific and as measured by National Hurricane Center (NHC), the sediment concentration in the form of site measurements of SSL, and the grain size distribution in the form of measurements of existing bed material [237].

Similar to numerical sedimentation models, ML sedimentation models also rely on a range

of input variables spanning hydrodynamics, meteorology, water quality, and human activities. For example, Heng *et al.* [238] used discharge and rainfall as inputs to an ANN to predict SSL. Likewise, Bharti *et al.* [239] employed an extensive set of hydro-meteorological variables—including average temperature, relative humidity, solar radiation, and wind speed—in addition to rainfall, to predict sediment in the Pokhariya watershed, India, using five ML models. Other studies have incorporated water quality indicators such as turbidity to predict sediment accumulation and suspended sediment concentration [150, 240]. Reflecting on human activities such as urbanization and deforestation, Mateo-Pérez *et al.* [146] leveraged satellite images including bands characterizing vegetation to estimate coastal bathymetry. Table 6.1 summarizes studies that incorporated these different categories of input variables to predict sediment-related target variables using ML models.

Table 6.1: Categories of input variables for ML-based sediment prediction with corresponding references.

Category of Input Variables	**Cited References**
Hydrology/Hydrodynamics: discharge/streamflow, runoff, water level, waves, tides, surges, velocity	[241, 238, 242, 148, 243, 244, 150, 208, 204, 245, 246, 247]
Climate/Meteorology: rainfall, temperature, evaporation, radiation, wind, humidity	[238, 242, 149, 150, 208, 248, 239, 249, 250]
Sediment/Water Quality: SSC, turbidity, sediment	[241, 251, 240, 252, 253, 244, 204, 245, 249, 250]
Human Factors: vegetation change, dredging, cost	[252, 146]

6.2.2 Machine Learning Approaches for Modeling Sediment Transport and Dredging Operations

As noted in Section 6.1, ML models have been widely applied to sediment prediction in shoaling environments. These models are usefully classified into five major groups, as summarized in Table 6.2. Among them, ANNs are the most frequently employed, with studies primarily focused on forecasting target outputs, such as SSL, suspended sediment concentration, turbidity, and sediment volume or deposition rate. For example, Afan *et al.* [147] present a comprehensive review of ML approaches applied to sediment modeling, with a particular emphasis on suspended sediment prediction, including SSL and suspended sediment concentration. The review evaluates the performance of a range of ML-based techniques: most notably ANNs [241, 251, 238, 242, 247, 240] in modeling the complex and dynamic

behavior of sediment transport in rivers. Overall, these models demonstrated sufficient accuracy in predicting the target variables. Bhattacharya *et al.* [148, 243, 252] conducted a series of studies advancing the application of ML in sediment transport and prediction modeling using ANNs, beginning in 2006. In their initial investigation, they developed both Multi-Layer Perceptron model (MLP) and the M5 model tree, demonstrating that data-driven approaches can achieve sufficient predictive accuracy for practical operational decision-making [148, 243]. Building on this work, their third study [252] explored hybrid modeling by using ANN outputs to generate time-varying sediment boundary conditions for numerical simulations of suspended particulate matter (SPM) along the Dutch Coast, highlighting the potential of hybrid modeling strategies for complex coastal systems.

Table 6.2: Categories of ML Models applied to sediment transport and dredging operations with corresponding references.

Category of Models	**References**
Artificial Neural Networks	[241, 251, 238, 242, 247, 240, 148, 243, 252, 149, 244, 150, 208, 204, 245, 239, 250, 246]
Support Vector Methods	[253, 149, 146, 204, 239]
Tree-Based and Ensemble Models	[253, 149, 146, 204, 239]
Regression & Statistical Models	[253, 149, 249]
Hybrid Approaches	[146, 150, 248, 249]

Other ML models, including Linear Regression, Regression Trees, Support Vector Machines (SVM), Gaussian Process Regression, Kernel Approximation, Ensemble Tree methods, and wavelet-based models, have also been applied to sediment and dredging duration prediction [253, 149, 146]. Some models have demonstrated strong performance, further confirming the potential of ML in this field. For instance, Li *et al.* [244] developed a hybrid, wavelet-based NARX model (INARX) enhanced with genetic programming for improved suspended sediment forecasting, which enhances the performance in non-stationary and flood conditions. Mateo-Pérez *et al.* [146] investigated the use of satellite data and ML algorithms to estimate seabed bathymetry in port environments, with a focus on supporting efficient dredging operations. Their study applied SVM, RF, and Multi-Adaptive Regression Splines (MARS) to predict bathymetric depths in the highly anthropized ports of Candás and Luarca in Asturias. The results show that SVM and RF algorithms provided more accurate and robust estimates than traditional single-beam sonar surveys.

In addition to traditional sediment-related target variables, a recent trend involves utilizing CSAT data to calculate sediment volume [183], which is directly relevant to shoaling and dredging activities in commercial navigation channels. Asborno *et al.* [150] integrate automated, data-driven variable selection using XGB into a time-series forecasting MLP model for sediment accumulation, thereby enhancing both the accuracy and adaptability of shoaling predictions in response to evolving monitoring conditions.

However, a review of recent literature reveals that few studies have explored the use of the CIR as the target variable in ML model development. Modeling the CIR—essentially the first-order derivative of the sediment volume time series—remains uncommon, as most previous research has focused on refining or comparing different ML models without reconsidering the choice of the target variables themselves. Tan *et al.* [208] explored the incorporation of the rate of change in input data for developing advanced AI models for flood forecasting. They compared the performance of MLP and LSTM models using historical rainfall data. The results revealed that while the MLP model underperformed, the LSTM model showed substantial improvements, particularly when the CIR was included as an input feature. Predicting the CIR instead of sediment volumes may offer a more reliable and stable target for ML models. The CIR is inherently less sensitive to changes in datum, as it captures the ongoing pattern of sediment accumulation regardless of shifting baseline depths. By focusing on the CIR, the model may more effectively capture the underlying dynamics of sediment transport, resulting in predictions that are more consistent, generalizable, and interpretable.

6.3 Methodology

6.3.1 Overview of the Proposed Framework

Figure 6.2 presents an overview of the proposed AI-based CIR forecasting model. This framework begins with data acquisition and preprocessing of the target variable, as detailed in Section 6.3.2, including the collection of source data, the CSAT calculation method, the CIR calculation, and the Interquartile Range (IQR) based filtering method. This is followed by a discussion on data acquisition and preprocessing of the exogenous input variables, as described in Section 6.3.3. The subsequent step involves feature engineering with GSA, during which the most crucial input variables are selected, as covered in Section 6.3.4. Finally, we present a MIMO LSTM model, configured in a NARX-style framework, which incorporates a custom loss function and a recursive prediction scheme, as discussed in Sections 6.3.4, 6.3.5, and 6.3.6.

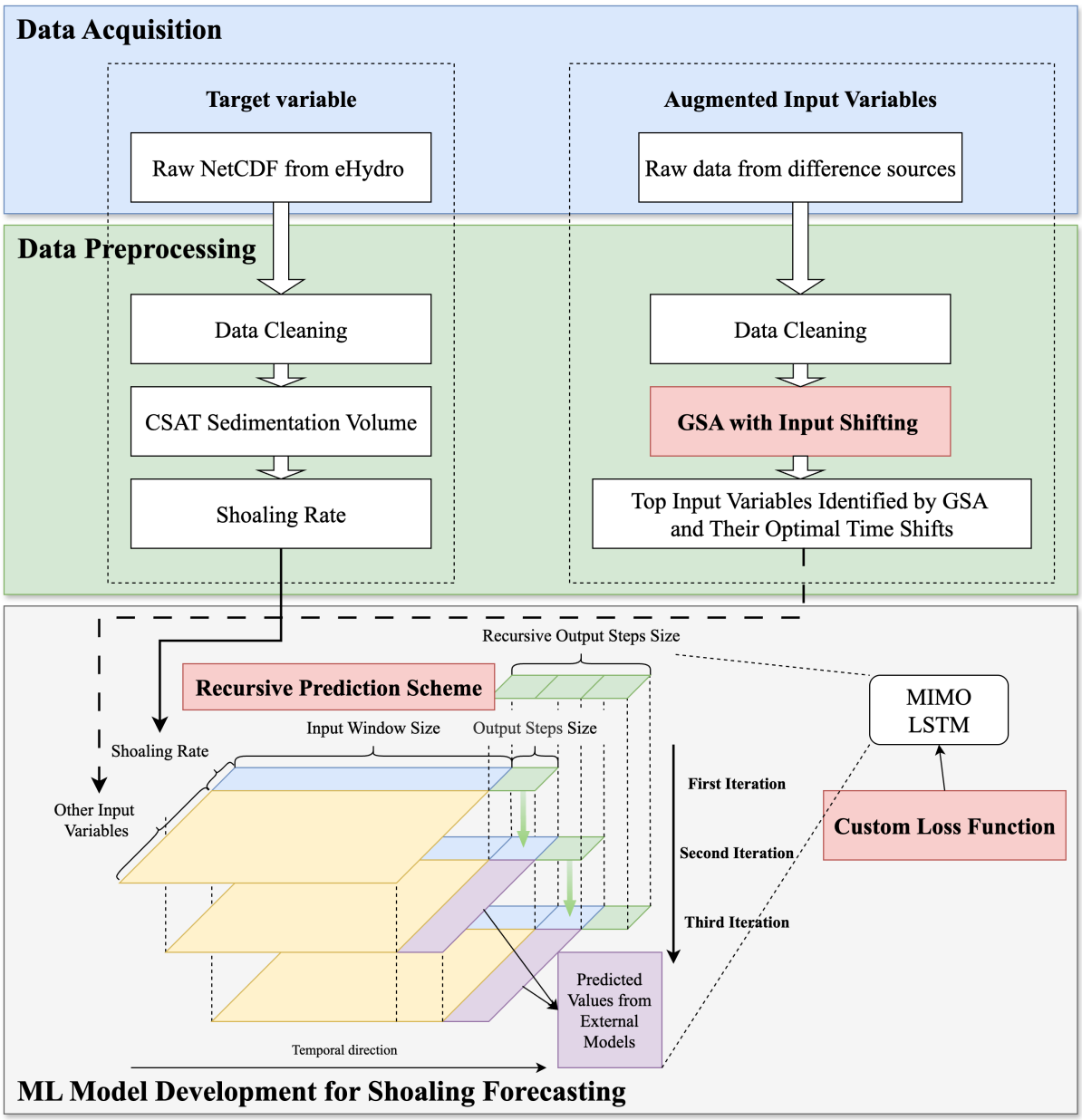


Figure 6.2: Overview of the proposed framework.

6.3.2 Introduction and Pre-processing of the Target Variable for the ML model

The CIR target variable is based on daily sedimentation volume, which originally stems from the hydrographic surveys conducted at time intervals of varied frequency on the SWP. Surveys are stored in the eHydro open-source database, by NCF reach. The surveys of each reach are ordinarily retrieved from eHydro, compiled, and stored in the CSAT package, as part of its Input component. They come in the form of NetCDF files, one for each reach. The established process to create a NetCDF file involves generating a bathymetry raster of uniform grid points with a cell size of 10 feet by 10 feet through interpolating the scattered XYZ points in a hydrographic survey using a Triangulated Irregular Network (TIN) object. The center of each grid point designates its latitude, longitude, and elevation. Moreover, the NCF polygon is used as a clipping template to solely retain the bathymetric data falling within the federally authorized navigation channel. The uniform grid points of multiple surveys, belonging to a particular reach, are all saved in a NetCDF file. After downloading these NetCDF files for a total of 13 reaches (i.e., for the SWP region), the data undergoes several pre-processing steps. As a first step, large elevations, generally corresponding to outside channel grid points, are often replaced with NaNs. After that, a linear interpolation is conducted to estimate the elevation of these NaNs; if a grid point has no reported elevation (NaN) across all historical surveys, it is excluded from subsequent sediment volume calculations. In the case of overlapping surveys conducted on the same day and reporting different elevations for the same grid point, the mean elevation value is computed to characterize that grid point.

In a final step, the elevation values for each grid are linearly interpolated to generate evenly stepped time series data, corresponding to one elevation every day. The sediment volume V for a given reach $r, \forall r = 1, \cdots, N_{\text{reach}}$ is then calculated by aggregating the contributions from all individual spatial cells within the reach. Denoting the total number of grid cells in the r-th reach as $N_{\text{cell,r}}$, and the sediment volume in cell i at time t as $v_i(t)$, the total raw sediment volume is given by

$$V_r^{\text{raw}}(t) = \sum_{i=1}^{N_{\text{cell,r}}} v_i(t). \tag{6.1}$$

In practical applications, evaluating the cumulative sediment volume across all reaches within a region is particularly valuable, as it offers a comprehensive assessment of sediment dynamics at a broader spatial scale. This aggregated perspective facilitates more effective regional planning and management decisions. Accordingly, this paper focuses exclusively on scenarios where the cumulative sediment volume for the entire region is considered. Let $\mathcal{C}$

represent a collection of reaches to be aggregated, comprising N_{reach} different reaches. The cumulative sediment volume for this composite set at time t can be computed as the sum of the corresponding segment volumes as

$$V_{\mathcal{C}}^{\text{raw}}(t) = \sum_{r=1}^{N_{\text{reach}}} V_r^{\text{raw}}(t). \tag{6.2}$$

To further improve the quality of the cumulative sediment volume time series and suppress unwanted short-term variability, a Savitzky–Golay filter (SGF) is utilized. This approach involves applying a sliding window, within which a low-order polynomial is fit to the observed data using the method of least squares [254]. Unlike a conventional moving average, this technique is designed to retain key features and trends by preserving the higher moments of the underlying signal, such as slope and curvature. By systematically smoothing the raw volume sequence with an SGF of window length W_{sg}, high-frequency disturbances—including those arising from measurement uncertainty or data processing artifacts—are effectively attenuated, yielding a cleaner representation of the reach-level sediment volume over time. Formally, this can be written as

$$V_{\mathcal{C}}(t) = \text{SGF}(V_{\mathcal{C}}^{\text{raw}}(t), W_{\text{sg}}). \tag{6.3}$$

As stated in Section 6.1, the objective of this paper is to assist engineers in predicting future sediment accumulation over time windows with improved accuracy in the SWP. Previous research has typically focused on directly forecasting the volume of accumulated shoaling sediment using ML models and other techniques. However, we compute the CIR of cumulative sediment volume and adopt it as the prediction target in this paper. This approach significantly mitigates issues arising from datum shifts or abrupt drops in sediment volume caused by dredging events, which can otherwise disrupt direct volume prediction. By focusing on the CIR, future sediment volumes can be estimated reliably, provided that current measurements (based on recent bathymetry surveys) are available. Therefore, the temporal CIR, evaluated with a delay interval of Δ days for the cumulative sediment volume, is defined as

$$R_{\Delta,\mathcal{C}}(t) = \frac{V_{\mathcal{C}}(t) - V_{\mathcal{C}}(t-\Delta)}{\Delta}. \tag{6.4}$$

To reconstruct the cumulative sediment volume from predicted CIR values during the forecasting phase, we employ a straightforward back-calculation approach. Specifically, once the future CIR sequence $\hat{R}_{\Delta,\mathcal{C}}(t)$ is predicted for a series of time steps, the corresponding

sediment volume $\hat{V}_{\mathcal{C}}(t)$ can be recursively computed using the following relationship

$$\hat{V}_{\mathcal{C}}(t+\Delta) = \hat{V}_{\mathcal{C}}(t) + \Delta \cdot \hat{R}_{\Delta,\mathcal{C}}(t+\Delta). \tag{6.5}$$

This iterative scheme is initialized from the most recent observed sediment volume and applies the model-predicted CIR values to yield a continuous volume forecast over the desired prediction window. As part of the data cleaning procedure, sections of the calculated CIR that fall significantly outside the expected range must be excluded. To systematically remove such extreme fluctuations—often arising from datum shifts in CSAT calculations or the effects of dredging—we apply the Interquartile Range (IQR) method as a robust statistical outlier filter [255]. To apply the IQR method for filtering the CIR, denoted by $R_{\Delta,\mathcal{C}}(t)$, we calculate the 25th (Q_{25}) and 75th (Q_{75}) percentiles values of the CIR series $\{R_{\Delta,\mathcal{C}}(t)\}_{t=1}^{T}$ firstly. Then the interquartile range is calculated by

$$\text{IQR} = Q_{75} - Q_{25}. \tag{6.6}$$

After that, the lower and upper bounds for identifying extreme values are defined as

$$\begin{aligned} Q_{\text{lower}} &= Q_{25} - q \cdot \text{IQR}, \\ Q_{\text{upper}} &= Q_{75} + q \cdot \text{IQR}, \end{aligned} \tag{6.7}$$

where q is a tunable threshold that controls the sensitivity of the outlier filter. We determined q through sensitivity testing across a range of candidate values. The goal was to balance sensitivity (eliminating abnormal patterns such as abrupt drops due to dredging) with robustness (retaining the natural variability of sediment dynamics). After testing multiple threshold levels, we found that the selected q value effectively removed most abnormal trends while preserving useful signal information. The specific threshold used in this study is introduced in a later section. Lastly, any value of $R_{\Delta,\mathcal{C}}(t)$ exceeding Q_{upper} or falling below Q_{lower} is flagged as an outlier and excluded from subsequent analysis. This results in the reduction of the total time from the original T to T_{filtered}, while the overall start and end dates of the dataset remain unchanged.

6.3.3 Introduction and Pre-processing of the Input Variables for the ML Model

As discussed in Section 2, several variables are essential for driving the sedimentation process in waterways and for constructing an ML model for CIR forecasting. Therefore,

these variables and their metrics were identified and retrieved from two main sources: (1) the official website of the United States Geological Survey (USGS) [256] and (2) the National Oceanic and Atmospheric Administration (NOAA) [257].

USGS is a science agency within the U.S. Department of the Interior, providing access to a wide range of products. These products include data collected from surface water monitoring stations, which are part of the National Water Information System (NWIS). These stations are distributed across thousands of sites, and they record time-series data that summarizes multiple attributes of water, including, but not limited to, gauge height (water level) and discharge, in addition to water-quality variables such as water temperature, salinity, specific conductance, turbidity, SSL, and SSC. The availability of these variables differs by station and by time: the combination of measured variables varies from one station to another, and the recording of a variable at a particular site may start at a later date and/or may be discontinued at an earlier date as compared to other sites and/or other variables within a site. Therefore, sites with discontinued data collection across all its variables are labeled as inactive, whereas those with continued data collection for at least one variable are labeled as active.

However, a key aspect characterizing this data source is its reliability. The majority of its values are labeled as "Approved", indicating that they have undergone a rigorous quality assurance process; the more recent values, pending rounds of revision to verify their accuracy, are distinguished from Approved values by being labeled as "Provisional". The quality assurance process and the separation between Approved and Provisional data allow for the integration of high-quality data into subsequent analyses. The data are generally recorded at fixed time intervals of 15-60 minutes. They are provided for retrieval in their continuous, real-time sampling frequency, and are also commonly provided in a daily resampled frequency.

NOAA is a science-based agency within the U.S. Department of Commerce. It provides access to a wide range of products, including those related to tides and currents; such products are made available and maintained by the Center for Operational Oceanographic Products and Services (CO-OPS). The products of interest to the study at hand include tides, water levels, currents, and meteorological observations across several stations. These are provided in the form of time-series variables describing water levels, wind speed and direction, air and water temperatures, salinity, barometric pressure, relative humidity, and visibility. In addition to that, NOAA provides historical measurements for tidal currents at several coastal locations. Other key data that can be retrieved include tide predictions. The data is typically presented in 1-minute or 6-minute intervals. Like USGS, a NOAA station does not necessarily have a full suite of sensors installed.

Although factors such as ship traffic and deforestation may also influence shoaling, they

were not included as direct predictors in this paper. The reason is that continuous, quality-controlled, and forecastable time-series records of these variables were not available for the full study period. That is also why this study focuses solely on variables retrieved from USGS and NOAA and that can be consistently observed, updated, and used in an operational forecasting framework.

Once the data sources had been established, stations were selected. For the SWP, the selection of target USGS sites began by locating all stations along the Mississippi River. Other stations that are on the tributaries of the Mississippi River, as considered in Asborno *et al.* [144], were also identified. The stations were then filtered to retain only those that have instantaneous and/or daily data for the variables of interest collected within a predetermined study time period, specified as 2012 to 2024. After identifying those stations, data were downloaded and pre-processed. Instantaneous data was resampled to daily measurements; this resampling process is crucial so that the frequency of input variables matches the frequency of the target variable, which is based on the daily sedimentation volume.

In a supplementary filtering process, all variables containing more than 10 percent of missing and/or anomalous (outside normal limits; e.g., negative stage or discharge) observations were removed. For the remaining variables that include missing data points, imputation by using the mean of adjacent observations (moving average) is used to replace these missing values. These steps are also crucial to increase the reliability of the input data and, therefore, to boost the performance of the data-driven forecasting model. In the present SWP case study, this filtering criterion did not limit the available feature set because sufficient USGS and NOAA observations remained after filtering; however, more flexible preprocessing strategies may be needed in data-scarce regions or under sensor-failure conditions.

As a result, a total of 79 variables corresponding to 56 stations were retained, such that one station possesses data for at least one variable. These variables include gauge height for 13 stations, discharge values for 43 stations, water temperature for 6 stations, salinity/ specific conductance for 11 stations, turbidity for 3 stations, and SSL or/and suspended sediment concentration for 3 stations. In a manner similar to selecting and filtering the USGS stations, three NOAA stations were identified. Two of these stations are equipped with sensors that measure water levels, wind speed and direction, air temperature, and water temperature in conjunction with presenting data for tide predictions. The third station has records for water temperature and water levels only. The variables were pre-processed and resampled to daily values.

After data cleaning, the remaining set of input variables—denoted as Z_{clean}, with N_{clean} indicating the number of features—consists of those with sufficient data quality for reliable modeling and is retained for subsequent analysis.

6.3.4 A MIMO LSTM-NARX Model with Shifted Delays for CIR Prediction

In this study, we develop a Multi-Input Multi-Output Long Short-Term Memory Nonlinear AutoRegressive with eXogenous inputs (LSTM-NARX) model to predict CIR and calculate sediment over multiple future time steps. This framework leverages the MIMO structure to produce direct multi-step forecasts while retaining the NARX formulation to capture delayed and nonlinear causal effects. The NARX framework allows for the integration of historical shoaling data along with external factors, such as river discharge, salinity, and other environmental influences. By modeling the time-lagged relationships between these variables and the target outcome, the NARX system effectively captures the delayed and non-linear causal effects inherent in geophysical systems. This structure enables the model to accurately reflect real-world dynamics, wherein changes in upstream conditions can affect shoaling patterns days or even weeks later, enhancing the model's predictive capabilities for complex, time-lagged interactions in coastal and riverine environments.

Furthermore, we note that synchronizing the timing of input variables with geographical pathways, by altering them by a few days, can enhance their impact on the model and boost explanatory power. For instance, river discharge requires time to travel from upstream origins to downstream areas of interest. By strategically shifting input variables, we strive to better capture the relationship between these variables and future CIR, thereby increasing the model's predictive accuracy. This adjustment ensures that the temporal and spatial dynamics of environmental factors are well-calibrated within the model's predictive framework. Therefore, we employed data shifting in conjunction with GSA to adjust the input data for lag-time, allowing a data-driven determination of statistically significant delays without imposing arbitrary assumptions.

6.3.4.1 LSTM Neural Networks

In terms of time-series forecasting, it is often uncertain how traditional neural networks can apply their understanding of previous events in a sequence to influence subsequent ones. Recurrent neural networks (RNNs) solve this problem. They feature looped connections that enable information to be retained and used in future processing steps. LSTM is a type of artificial RNN architecture used in the field of DL. It was designed to address some of the limitations of traditional RNNs, particularly the difficulty of learning long-term dependencies due to the problem of vanishing or exploding gradients [258]. This model is well-suited for predicting time series data related to CIR. The CIR exhibits temporal relationships that can be leveraged for prediction. LSTM models are beneficial for capturing these dependencies,

making them ideal for forecasting multiple steps ahead. This ability is especially beneficial for planning dredging operations with a longer lead time.

LSTM networks gain their advantage through unique components within each cell, which are designed to mimic biological memory units. These components allow LSTMs to capture short-term patterns and preserve long-term information, a feat that traditional RNNs struggle with due to the vanishing gradient problem [259]. There are four main components in an LSTM cell: Forget Gate vector f_t, Input Gate vector g_t, Cell State Update, and Output Gate vector o_t [260]. Equation 6.8 details the functioning of the four components within an LSTM cell [258]:

$$\begin{aligned} f_t &= \sigma\left(W_f\left[h_{t-1}, u_t\right] + b_f\right), \\ g_t &= \sigma\left(W_g\left[h_{t-1}, u_t\right] + b_g\right), \\ \tilde{C}_t &= \tanh\left(W_C\left[h_{t-1}, u_t\right] + b_C\right), \\ o_t &= \sigma\left(W_o\left[h_{t-1}, u_t\right] + b_o\right), \end{aligned} \tag{6.8}$$

where u_t is the input vector at time step t, h_{t-1} is the hidden state from the previous time step, $\tilde{C}_t$ is the candidate cell state, while C_t is the current cell state. h_t is the current hidden state, and W_f, W_g, W_C, W_o are the weight matrices for forget, input, cell candidate, and output gates; b_f, b_g, b_C, b_o are the corresponding bias vectors. σ is the sigmoid activation function, $\tanh(\cdot)$ is the hyperbolic tangent activation. These gates are then used to update the cell state and hidden state:

$$C_t = f_t \odot C_{t-1} + g_t \odot \tilde{C}_t, \quad h_t = o_t \odot \tanh\left(C_t\right). \tag{6.9}$$

Figure 6.3 illustrates the architecture of an LSTM cell and its role within the entire LSTM layer. This structure forms the basis for constructing the proposed LSTM-NARX model, whose design is presented in Figure 6.4. This figure presents the architecture of a multi-layer LSTM-NARX model, specifically tailored for predicting multiple steps ahead. The model inputs a window of historical time-series data, comprising both autoregressive elements and exogenous inputs. This data is processed through a stack of three LSTM layers, each capturing temporal dependencies at different levels of abstraction. To mitigate overfitting and improve generalization, a dropout layer with a rate of λ is applied between each LSTM layer. The stacked LSTM architecture enables the model to learn both short-term and long-term patterns in the sequential data. After the final LSTM block, dense layers are applied for regression. The output layer then generates predictions for multiple future time steps. This architecture is particularly effective for modeling systems where

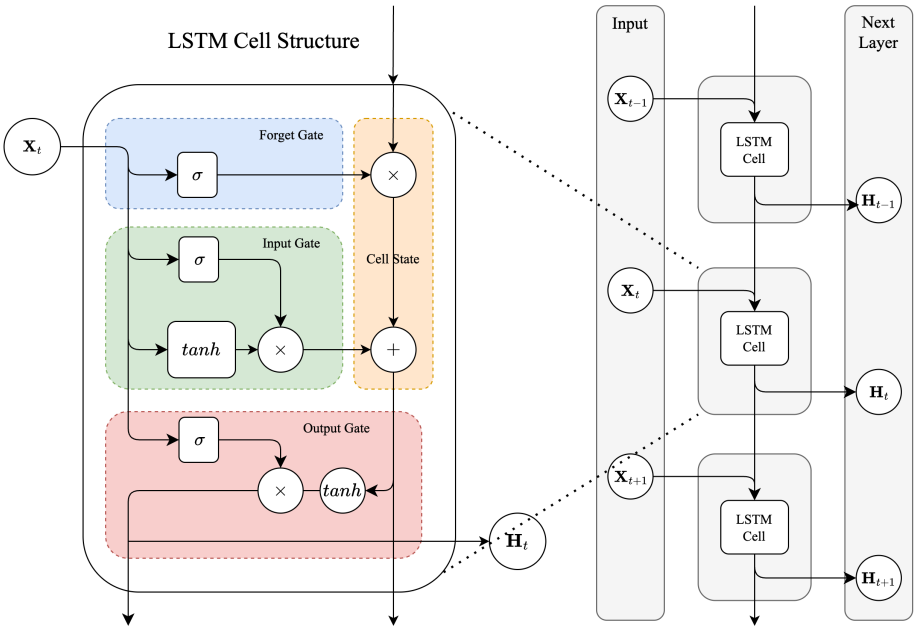


Figure 6.3: Structure of LSTM cell.

external inputs affect the target with potential time delays, characteristic of NARX modeling strategies. Suppose at each time step t, the cleaned input variables vector is represented as a row vector $\mathbf{Z}_{\text{clean},t} = \left[z_t^{(1)}, z_t^{(2)}, \ldots, z_t^{(N_{\text{clean}})}\right]^\top \in \mathbb{R}^{1\times N_{\text{clean}}}$. Let $\mathbf{y}_t$ denote the CIR value at time t. The LSTM-NARX model takes a specific window W_{past} of a time-series dataset as the input. Therefore, stacking the cleaned exogenous input variables vector over a window of length W_{past} forms the matrix $\mathbf{Z}_{\text{clean},t-W_{\text{past}}+1:t} = \left[\mathbf{z}_{t-W_{\text{past}}+1}, \ldots, \mathbf{z}_t\right] \in \mathbb{R}^{W_{\text{past}}\times N_{\text{clean}}}$, and let $\mathbf{y}_{t-W_{\text{past}}+1:t} = \left[y_{t-W_{\text{past}}+1}, \ldots, y_t\right]^\top \in \mathbb{R}^{W_{\text{past}}}$ denote the past W_{past} days values of the target variable.

The combined input to the LSTM at time t is the concatenation:

$$\mathbf{X}_{t-W_{\text{past}}+1:t} = \left[\mathbf{y}_{t-W_{\text{past}}+1:t} \,||\, \mathbf{Z}_{\text{clean},t-W_{\text{past}}+1:t}\right] \in \mathbb{R}^{W_{\text{past}}\times(1+N_{\text{clean}})}, \tag{6.10}$$

where $||$ denotes concatenation.

The LSTM model, denoted by $\mathcal{G}_{\text{LSTM}}(\cdot)$, predicts the future values as:

$$\hat{\mathbf{y}}_{t+1:t+W_{\text{pred}}} = \mathcal{G}_{\text{LSTM}}\left(\mathbf{X}_{t-W_{\text{past}}+1:t}\right), \tag{6.11}$$

where W_{pred} is the output window size and $\hat{\mathbf{y}}_{t+1:t+W_{\text{pred}}} \in \mathbb{R}^{1\times W_{\text{pred}}}$.

Overall, this LSTM-NARX model efficiently integrates autoregressive components and exogenous inputs through its structured input-output setup and adaptive use of predicted

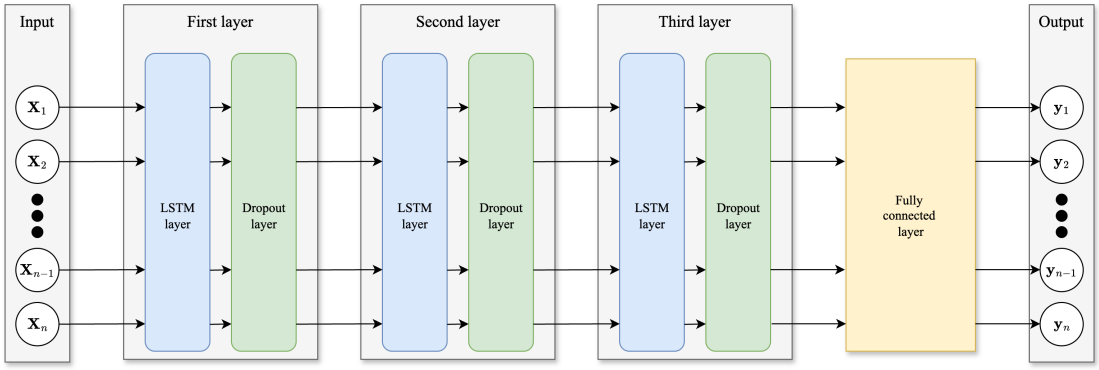


Figure 6.4: Structure of LSTM-NARX model.

variables, making it a powerful tool for capturing complex temporal relationships in time series data.

6.3.4.2 Global Sensitivity Analysis (GSA) for Identifying Important Input Features

Not all potential input variables provide useful information for learning CIR patterns; some may even degrade performance. Therefore, it is essential to carefully select a method that identifies the most sensitive and relevant inputs. The chosen approach should not only enhance the model's accuracy but also reduce input dimensionality, thereby mitigating the risk of overfitting in ML models. GSA is a technique for assessing how different sources of input uncertainty impact the uncertainty in a model's output [261]. A widely used approach in this domain employs Sobol indices, which offer a detailed assessment of each input variable's contribution to the overall variance in the model's output. The first-order Sobol index is a metric used to quantify the direct contribution of an individual input variable to the output variance of a model. It measures the proportion of the output variability that can be attributed solely to changes in one input, independent of any interactions with other inputs [262]. It is efficient to focus only on the first-order Sobol index because it not only identifies the direct impact of input variables on the output, but also helps prevent overfitting from variable interaction considerations. Equation 6.12 shows the definition of the first-order Sobol index

$$S_i = \frac{\mathrm{Var}_{\mathbf{x}^{(i)}}\left(\mathbb{E}_{\mathbf{X}^{(-i)}}\left[\mathbf{y} \mid \mathbf{x}^{(i)}\right]\right)}{\mathrm{Var}(\mathbf{y})}, \tag{6.12}$$

where i represents the i-th input variable, $\mathbf{x}^{(i)}$ is the vector of the model's i-th input, and $\mathbf{y}$ is the output of the model corresponding to $\mathbf{x}^{(i)}$. $\mathrm{Var}_{\mathbf{x}^{(i)}}(\cdot)$ denotes variance with respect to $\mathbf{x}^{(i)}$, $\mathbb{E}_{\mathbf{X}^{(-i)}}[\cdot]$ denotes an expectation over all inputs except $\mathbf{x}^{(i)}$, and $\mathrm{Var}(\mathbf{y})$ is the total

variance of $\mathbf{y}$.

One way to estimate the first-order Sobol index S_i is the "double loop" Monte Carlo method. For each sample in the dataset, the method fixes the value of the input variable of interest and estimates the expected output by averaging over all other input variables. This requires grouping or identifying samples with similar values for the chosen variable and computing the average model output within each group. After these conditional averages are calculated for all samples, the variability of these averages is assessed relative to the overall variance of the model output. The resulting ratio provides the sensitivity index for the variable. This approach is conceptually straightforward but computationally expensive since it involves repeated calculations of conditional expectations for every sample in the dataset. Another approach proposed by Azzini *et al.* [263] is a ML approach that uses a dynamic adaptive variance estimator to estimate Sobol indices based on existing Monte Carlo data. However, this method is constrained by the requirement for large datasets to effectively train the model. To avoid this computational burden, the double-loop Monte Carlo estimator and dynamic adaptive variance estimator are not used in the final implementation of this study. Instead, an alternative method is considered to estimate the Sobol indices in this paper. Li *et al.* introduced a robust, efficient, and modular way to estimate first-order Sobol indices directly from observational data—saving computational cost while broadening applicability in practical scenarios [264, 265]. It can be achieved by the following steps:

1. Partition the domain of $\mathbf{x}^{(i)}$ into D intervals (Φ_d, $d = 1, 2, \ldots, D$), each containing approximately the same number of samples.

2. For each interval Φ_d, compute the sample variance of the output:

$$\mathrm{Var}_{\Phi_d}(\mathbf{y}) = \frac{1}{n_d - 1} \sum_{j \in \Phi_d} \left(y_j - \bar{y}_{\Phi_d}\right)^2, \tag{6.13}$$

   where n_d is the number of samples in interval Φ_d, and $\bar{y}_{\Phi_d}$ is the mean output in Φ_d.

3. Average these sample variances over all intervals:

$$\mathbb{E}_{\Phi}\left[\mathrm{Var}_{\Phi_d}(\mathbf{y})\right] = \frac{1}{D} \sum_{d=1}^{D} \mathrm{Var}_{\Phi_d}(\mathbf{y}). \tag{6.14}$$

4. Estimate the total variance of the output across all samples:

$$\mathrm{Var}(\mathbf{y}) = \frac{1}{T_{\text{filtered}} - 1} \sum_{t=1}^{T_{\text{filtered}}} \left(y_t - \bar{y}\right)^2. \tag{6.15}$$

5. Calculate the first-order Sobol index using the following equation:

$$S_i = \text{Sobol}\left(\mathbf{x}^{(i)} \to \mathbf{y}\right) \approx 1 - \frac{\mathbb{E}_{\Phi}\left[V_{\Phi_d}(\mathbf{y})\right]}{\text{Var}(\mathbf{y})}. \tag{6.16}$$

This approach leverages stratified sampling and averages local variances, making it particularly advantageous for complex models or simulation-based systems where traditional methods would be prohibitively expensive. Traditional GSA using Sobol indices is typically formulated for models with a single-step input-output structure. In this setting, the input data are represented by a matrix $\mathbf{X}_{\text{GSA}} \in \mathbb{R}^{T_{\text{filtered}} \times (1+N_{\text{clean}})}$, where T_{filtered} is the number of available samples and $(1 + N_{\text{clean}})$ is the total number of input variables (with the first column corresponding to the target variable's own history). The output is denoted as $\mathbf{y}_{\text{GSA}} \in \mathbb{R}^{T_{\text{filtered}} \times 1}$, reflecting the model's prediction at a single time step. In this framework, the first-order Sobol index S_i quantifies the contribution of the i-th input variable to the variance of the model output at each time step.

However, directly assigning a single Sobol index to each input variable is challenging in the proposed LSTM sequence-to-sequence framework, because each variable appears at multiple historical positions in the input window and may affect multiple future steps in the output window. By contrast, the proposed LSTM-NARX model employs a sliding window approach for both inputs and outputs, resulting in a higher-dimensional input structure. Here, the input matrix for each training instance is represented as $\mathbf{X}_{t-W_{\text{past}}+1:t} \in \mathbb{R}^{W_{\text{past}} \times (1+N_{\text{clean}})}$, where W_{past} is the input window size and N_{clean} is the number of exogenous input variables. Each input matrix $\mathbf{X}_{t-W_{\text{past}}+1:t} \in \mathbb{R}^{W_{\text{past}} \times (1+N_{\text{clean}})}$ can be expressed as

$$\mathbf{X}_{t-W_{\text{past}}+1:t} = \begin{bmatrix} \overbrace{y_{t-W_{\text{past}}+1}}^{\text{CIR}} & \overbrace{z^{(1)}_{t-W_{\text{past}}+1} & z^{(2)}_{t-W_{\text{past}}+1} & \cdots & z^{(N_{\text{clean}})}_{t-W_{\text{past}}+1}}^{\text{Exogenous input variables}} \\ y_{t-W_{\text{past}}+2} & z^{(1)}_{t-W_{\text{past}}+2} & z^{(2)}_{t-W_{\text{past}}+2} & \cdots & z^{(N_{\text{clean}})}_{t-W_{\text{past}}+2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ y_t & z^{(1)}_t & z^{(2)}_t & \cdots & z^{(N_{\text{clean}})}_t \end{bmatrix}. \tag{6.17}$$

From this matrix above, over the full dataset, the collection of these windowed inputs forms a three-dimensional tensor of shape $\mathbb{R}^{T_{\text{filtered}} \times W_{\text{past}} \times (1+N_{\text{clean}})}$. The corresponding output for each windowed input is a sequence $\mathbf{y}_{t+1:t+W_{\text{pred}}}$, representing the target variable over a future prediction horizon of length W_{pred} . The vector of $\mathbf{y}_{t+1:t+W_{\text{pred}}}$ can be expressed as

$$\mathbf{y}_{t+1:t+W_{\text{pred}}} = \left[\, y_{t+1},\ y_{t+2},\ \ldots,\ y_{t+W_{\text{pred}}} \,\right]^{\top}. \tag{6.18}$$

As previously explained, directly calculating the Sobol index for windowed inputs and outputs is challenging since traditional Sobol GSA is designed for one-to-one input–output mappings. To adapt the traditional GSA to the proposed LSTM framework, we decompose the problem into per-time-step analyses and then aggregate the results. For each input variable $i \in \{1, \ldots, N_{\text{clean}}\}$, each input window time step $j \in \{1, \ldots, W_{\text{past}}\}$, and each output window time step $k \in \{1, \ldots, W_{\text{pred}}\}$, we define the local Sobol index:

$$S_{i,j,k} = \text{Sobol}\left(x^{(i)}_{t-W_{\text{past}}+j} \to y_{t+k}\right). \tag{6.19}$$

This definition captures the contribution of input i at delay j to the output at forecast step k. The dimensions W_{past} and W_{pred} may differ, and the indices are computed for all possible (j, k) combinations.

To obtain a global importance measure for each input variable, we aggregate the local indices over all input and output window positions

$$S_i^{(\text{agg})} = \sum_{j=1}^{W_{\text{past}}} \sum_{k=1}^{W_{\text{pred}}} S_{i,j,k}. \tag{6.20}$$

or, for a normalized measure, we define the following:

$$S_i^{(\text{mean})} = \frac{1}{W_{\text{past}} \cdot W_{\text{pred}}} \sum_{j=1}^{W_{\text{past}}} \sum_{k=1}^{W_{\text{pred}}} S_{i,j,k}. \tag{6.21}$$

This aggregated index captures the cumulative impact of each input variable across all temporal positions in both the input and output windows, thereby providing a comprehensive ranking of input importance in the context of sequence modeling. By summing and averaging the Sobol indices across all windowed positions, we account for both the temporal dynamics and the feature-specific effects within the LSTM's sequence-to-sequence architecture. Finally, the top N_{selected} input variables are selected for the following model training. Different values of N_{selected} were systematically tested during model training to determine the most suitable choice. The updated input to the LSTM at time t is then expressed as

$$\mathbf{X}^{\mathbf{GSA}}_{t-W_{\text{past}}+1:t} = \left[\mathbf{y}_{t-W_{\text{past}}+1:t} \,\middle\|\, \mathbf{z}^{(1)}_{t-W_{\text{past}}+1:t},\ \ldots,\ \mathbf{z}^{(N_{\text{selected}})}_{t-W_{\text{past}}+1:t} \right] \in \mathbb{R}^{W_{\text{past}} \times (1+N_{\text{selected}})}, \tag{6.22}$$

where $||$ denotes concatenation.

The updated LSTM predicts the future values as

$$\hat{\mathbf{y}}_{t+1:t+W_{\text{pred}}} = \mathcal{G}_{\text{LSTM}}\left(\mathbf{X}^{\mathbf{GSA}}_{t-W_{\text{past}}+1:t}\right). \tag{6.23}$$

This method is well-suited to the shoaling problem, as it can efficiently accommodate a large number of additional variables. Moreover, since GSA is a model-free approach to identifying sensitive features, it is applicable not only to the LSTM-based framework presented here but also to a wide range of other models in future applications.

6.3.4.3 Identifying the Optimal Shifted Delays for the River Discharge Data Using GSA

While employing GSA to identify the most influential input variables for the model, it is important to consider the fact that not all monitoring stations are located near the target area. This spatial separation can lead to time delays in the influence of particular input variables. In riverine and coastal environments, sediment deposition and transport designate dynamic processes involving upstream erosion, movement, and eventual downstream deposition. Sediments are often temporarily stored— such as behind levees or natural barriers— or redistributed due to changes in flow dynamics. These processes are governed by factors like flow velocity, sediment size, and riverbed morphology [266]. Moreover, in tidally influenced river systems such as the Hudson River, studies have shown that sediment transport can involve substantial time lags, ranging from months to years. These delays are closely linked to sediment grain-size characteristics and hydrodynamic conditions [267]. This means that downstream areas may experience changes in CIR within days, weeks, or even months after upstream events such as heavy rainfall or land-use changes. Understanding these time-lagged sediment responses is essential for building more accurate predictive models and for developing effective sediment management strategies that ensure navigational safety and ecological sustainability [268, 269]. Figure 6.5 illustrates the effect of travel time on sediment dynamics within a river system, highlighting the importance of accounting for temporal delays in input variables, particularly river discharge measurements from upstream gauge stations.

As mentioned before, in hydrological and environmental modeling some input variables– such as river discharge data collected from multiple upstream stations–may exhibit a physical time delay relative to the target variable due to transport and propagation processes in the system. To address this issue, we propose an alternative input scheme for the LSTM-NARX model that employs optimally shifted gauged river discharge data as input variables. In this approach, each discharge station is allowed to be shifted by its own optimal time delay, rather than being strictly aligned with the historical sequence of the target variable. Specifically:

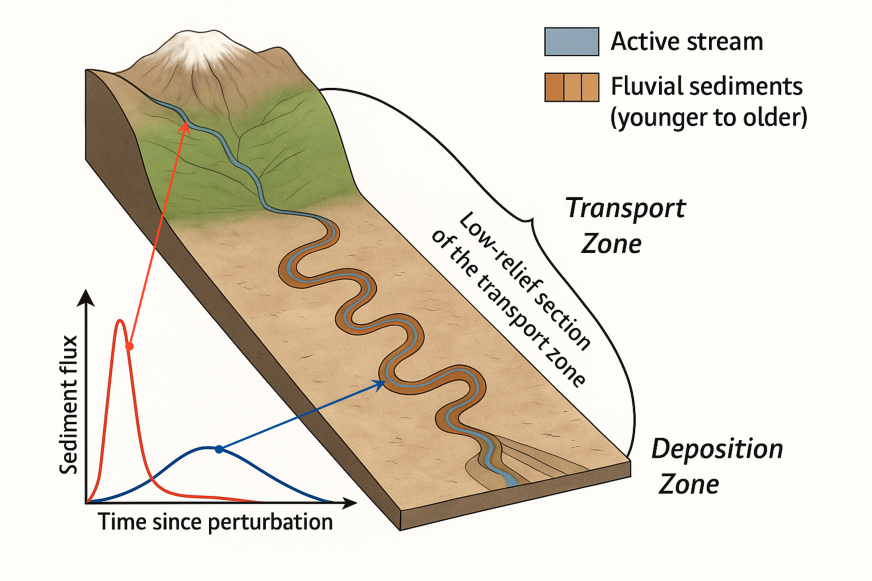


Figure 6.5: Conceptual sketch of sediment dynamics within a river system.

The output associated with each input window is

$$\mathbf{y}_{t+1:t+W_{\text{pred}}} = \left[\, y_{t+1},\ y_{t+2},\ \ldots,\ y_{t+W_{\text{pred}}} \,\right]^{\top}, \tag{6.25}$$

representing the values of the target variable over the prediction horizon.

When assessing the local Sobol indices for the shifted-input scheme, we impose the same time delay τ for all gauged river discharge input variables within each GSA evaluation. Specifically, for each candidate value of τ, all exogenous input variables are shifted by this

uniform lag before being subjected to sensitivity analysis. The local Sobol index for the i-th input variable at input window time step j and output window step k is then calculated as

$$S_{i,j,k}^{(\tau)} = \text{Sobol}\left(x_{t-W_{\text{past}}+j-\tau}^{(i)} \to y_{t+k}\right). \tag{6.26}$$

This procedure is repeated for a range of plausible τ values (e.g., spanning 0 to 365 days), with the process holding τ fixed across all input variables in each run. The aggregated Sobol index $S_{i,\tau}^{(\text{agg})}$ and its normalized form $S_{i,\tau}^{(\text{mean})}$ are calculated as

$$\begin{aligned} S_{i,\tau}^{(\text{agg})} &= \sum_{j=1}^{W_{\text{past}}} \sum_{k=1}^{W_{\text{pred}}} S_{i,j,k}^{(\tau)}, \\ S_{i,\tau}^{(\text{mean})} &= \frac{1}{W_{\text{past}} \cdot W_{\text{pred}}} \sum_{j=1}^{W_{\text{past}}} \sum_{k=1}^{W_{\text{pred}}} S_{i,j,k}^{(\tau)}, \end{aligned} \tag{6.27}$$

where $S_{i,j,k}^{(\tau)}$ denotes the Sobol index for input variable i at input window position j, output window position k, with a temporal shift of τ days applied to the input.

By collectively comparing the aggregated Sobol indices across the set of tested τ values, we identify the delay τ that yields the highest sensitivity for each input variable, thereby determining the optimal delay structure for the subsequent LSTM-NARX model input configuration. The set of optimal time shifts for the N_{selected} input variables is denoted as $\boldsymbol{\tau}^*$, where

$$\boldsymbol{\tau}^* = \left[\tau_1^*, \tau_2^*, \ldots, \tau_{N_{\text{selected}}}^*\right]^\top, \tag{6.28}$$

and each τ_i^* is defined by

$$\tau_i^* = \arg\max_{\tau} \{S_{i,\tau}^{(\text{agg})}\}. \tag{6.29}$$

Finally, similar to the previous scheme, the top N_{selected} input variables are selected for the following model training. The updated input to the LSTM at time t with the optimal shift τ_i for the i-th gauge input is then expressed as:

$$\begin{aligned} \mathbf{X}_{t-W_{\text{past}}+1:t}^{\textbf{shifted}} = \Big[&\mathbf{y}_{t-W_{\text{past}}+1:t} \,\Big|\Big|\, \mathbf{z}_{t-W_{\text{past}}+1-\tau_1^*:t-\tau_1^*}^{(1)}, \\ &\ldots, \mathbf{z}_{t-W_{\text{past}}+1-\tau_{N_{\text{selected}}}^*:t-\tau_{N_{\text{selected}}}^*}^{(N_{\text{selected}})} \Big] \\ &\in \mathbb{R}^{W_{\text{past}} \times (1+N_{\text{selected}})}. \end{aligned} \tag{6.30}$$

The updated LSTM then predicts the future values as

$$\hat{\mathbf{y}}_{t+1:t+W_{\text{pred}}} = \mathcal{G}_{\text{LSTM}}\left(\mathbf{X}_{t-W_{\text{past}}+1:t}^{\textbf{shifted}}\right). \tag{6.31}$$

6.3.5 Modified Loss Function for Training of the LSTM Model

In general, the loss function serves as the primary objective function used to optimize and guide model training. While the Mean Squared Error (MSE) loss is widely adopted for regression, certain application-specific considerations warrant a more tailored approach. For the CIR prediction task in sediment volume forecasting, it is particularly important to accurately capture transition points—where the CIR crosses zero—corresponding to local peaks and valleys in the sediment trend. These transitions are highly relevant for sediment management and operational decisions. As illustrated in Figure 6.6, CIR zero-crossing transition points are weighted more heavily (w_{high}), while regions away from such transitions are assigned a lower weight (w_{low}). This targeted weighting strategy ensures that the model places greater emphasis on learning the most critical segments of the time series.

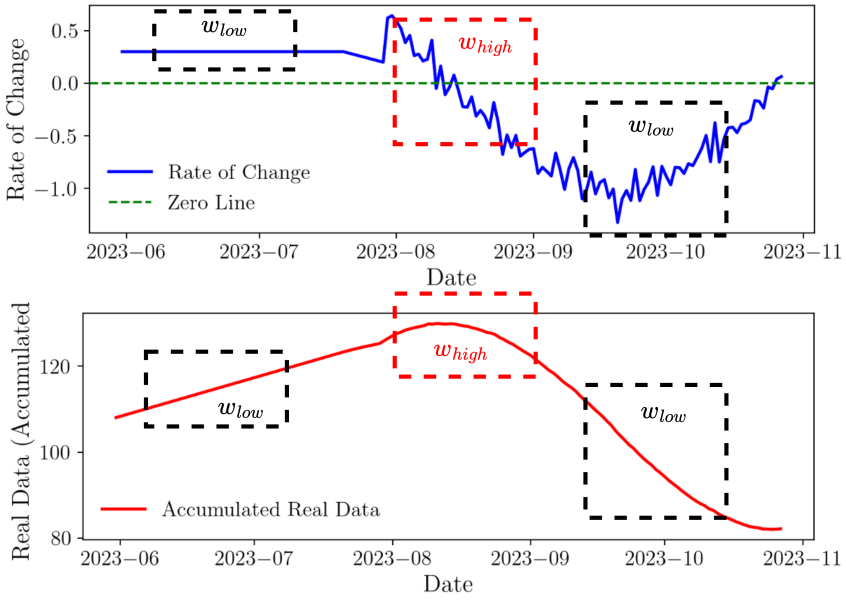


Figure 6.6: Illustration of low and high weighting scenarios.

To be more specific, let $\mathbf{Y}_{\text{true}} \in \mathbb{R}^{B\times W_{\text{pred}}}$ and $\mathbf{Y}_{\text{pred}} \in \mathbb{R}^{B\times W_{\text{pred}}}$ denote the true and predicted CIR values, respectively, for batch size B and prediction window W_{pred}. Define a binary indicator vector $\mathbf{f} \in \{0,1\}^B$ such that $f_b = 1$ if the corresponding sample in the batch exhibits a zero crossing, and $f_b = 0$ otherwise. The sample-wise weight $w(b), \forall b \in \{1, 2, \cdots, B\}$ is assigned according to

$$w(b) = \begin{cases} w_{\text{low}}, & \text{if no zero crossing in window} \\ w_{\text{high}}, & \text{if zero crossing occurs} \end{cases}. \tag{6.32}$$

This approach is consistent with the uniform weighting scheme proposed by Wang *et al.* [270], where "normal" events are assigned a weight of 1 and extreme or critical events (here, zero-crossing samples) are assigned an elevated weight.

The weighted mean squared error loss is thus given by

$$\mathcal{L}_{\mathrm{W-MSE}} = \frac{1}{B \cdot W_{\mathrm{pred}}} \sum_{b=1}^{B} \sum_{k=1}^{W_{\mathrm{pred}}} w(b) \left(Y_{\mathrm{pred}}^{(b,k)} - Y_{\mathrm{true}}^{(b,k)} \right)^2 . \tag{6.33}$$

A grid search procedure is employed to identify the optimal value of w_{high} for the proposed LSTM-NARX model.

6.3.6 A Recursive Prediction Scheme to Enable Monthly CIR Prediction

To forecast CIR effectively over long periods, it is important to choose a suitable method for predicting multiple time steps. The two most common approaches are the direct prediction method and the recursive prediction method. Each has its own benefits for managing time-related dependencies and forecast uncertainties. As shown in Figure 6.7, these approaches are fundamentally different in how they create future predictions based on past data and additional input. The direct prediction method uses one model to give a multi-step forecast at once using a fixed input window. This method is computationally efficient and avoids the compounding of errors that can occur when predictions are fed back into the model. It is particularly advantageous when the interdependence among future values is relatively simple or when prediction speed is a priority.

In contrast, the recursive prediction method breaks the forecasting period into smaller parts. It feeds previously predicted values, along with outputs from external models, back into the input window to gradually extend the forecast. This approach can better capture evolving temporal patterns and complex dependencies in the data, as the model dynamically updates its input sequence with its most recent predictions.

The 60-day input window and 30-day output window are selected based on systematic model testing and practical forecasting considerations. The 30-day output window represents an approximate forecasting horizon of one month, which is meaningful for dredging planning, while the 60-day input window provides sufficient recent hydrological and sedimentation history without introducing an unnecessarily long input sequence.

The following Section 6.3.6.1 and 6.3.6.2 will explain both methods in detail, showcasing their mechanics and roles in multi-step LSTM-based CIR prediction.

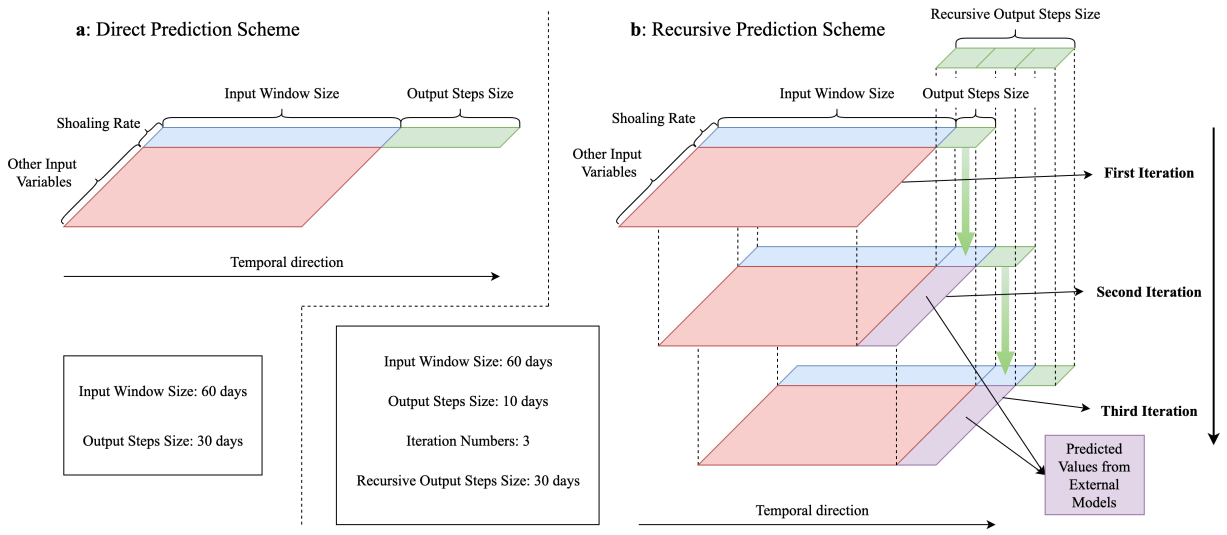


Figure 6.7: Illustration of direct and recursive prediction schemes for multi-step LSTM CIR forecasting.

6.3.6.1 Direct Prediction Scheme

The widely used direct multi-step forecasting method involves predicting future values in a single forward pass, generating multiple outputs simultaneously without relying on previously predicted outputs as inputs.

For an input window length of W_{past} and the corresponding input vector $\mathbf{X}_{t-W\text{past}+1:t}$, the direct multistep model produces a single predicted time series at time t, containing predicted values for future time steps of W_{direct}, denoted by $\hat{\mathbf{y}}_{t+1:t+W_{\text{direct}}}$, such as:

$$\hat{\mathbf{y}}_{t+1:t+W_{\text{direct}}} = \mathcal{G}_{\text{LSTM}}\left(\mathbf{X}_{t-W_{\text{past}}+1:t}\right). \tag{6.34}$$

Direct prediction techniques utilizing LSTM models offer an efficient approach for multi-step forecasting by producing future values in a single forward pass. However, a limitation of the direct prediction strategy is that it can become less accurate with longer forecasting windows, as the model does not adjust to evolving patterns in the data. This challenge transitions into the concept of recursive prediction, discussed in the next section, which addresses these accuracy issues by iteratively refining predictions.

6.3.6.2 Recursive Prediction Scheme

An alternative to predicting an entire sequence of future values at once is to use a recursive prediction strategy with the trained LSTM model for multi-step forecasting. This iterative

approach enhances prediction accuracy by progressively predicting each subsequent value. Specifically, the model iteratively predicts future sequences, extending the prediction horizon by recursively forecasting smaller windows of size p from a larger input window of size W_{past} (for example, $W_{\text{past}} > 3p$ if three iterations are needed). Each iteration leverages previously predicted values to form a dynamically evolving input window. The right portion of Figure 6.7 shows the structure of the recursive strategy used in this paper, and the detailed steps of this method is shown below.

Initially, suppose we have observed data up to time step t. The first prediction window (p steps ahead) is obtained directly from the observed historical inputs

$$\hat{\mathbf{y}}_{t+1:t+p} = \mathcal{G}_{\text{LSTM}}\left(\mathbf{X}_{t-W_{\text{past}}+1:t}\right), \tag{6.35}$$

where $\mathbf{X}_{t-W_{\text{past}}+1:t} \in \mathbb{R}^{W_{\text{past}} \times N_{\text{selected}}}$ represents the input window.

The next input window (to be used at iteration $s = 1$) is now updated as follows:

$$\begin{aligned}
&\tilde{\mathbf{X}}_{t+p-W_{\text{past}}+1:t+p} \\
&= \begin{bmatrix}
y_{t+p-W_{\text{past}}+1} & z^{(1)}_{t+p-W_{\text{past}}+1} & \cdots & z^{(N_{\text{selected}})}_{t+p-W_{\text{past}}+1} \\
y_{t+p-W_{\text{past}}+2} & z^{(1)}_{t+p-W_{\text{past}}+2} & \cdots & z^{(N_{\text{selected}})}_{t+p-W_{\text{past}}+2} \\
\vdots & \vdots & \ddots & \vdots \\
y_t & z^{(1)}_t & \cdots & z^{(N_{\text{selected}})}_t \\
\hat{y}_{t+1} & \tilde{z}^{(1)}_{t+1} & \cdots & \tilde{z}^{(N_{\text{selected}})}_{t+1} \\
\hat{y}_{t+2} & \tilde{z}^{(1)}_{t+2} & \cdots & \tilde{z}^{(N_{\text{selected}})}_{t+2} \\
\vdots & \vdots & \ddots & \vdots \\
\hat{y}_{t+p} & \tilde{z}^{(1)}_{t+p} & \cdots & \tilde{z}^{(N_{\text{selected}})}_{t+p}
\end{bmatrix},
\end{aligned} \tag{6.36}$$

where only the last p rows of the target variable are updated using the predictions from the LSTM. For each exogenous input variable with an associated time delay τ_i, the handling depends on the relationship between p and τ_i: if $p \leq \tau_i$, the original observed data are retained for that input; however, if $p > \tau_i$, we substitute the predicted values from an external model to fill in the missing future values. This way, all input variables are updated independently using either available observations or external predictions, assumed to be accessible in advance.

At iteration $s = 1$, predictions from $t + p + 1$ to $t + 2p$ are computed as

$$\hat{\mathbf{y}}_{t+p+1:t+2p} = \mathcal{G}_{\text{LSTM}}\left(\tilde{\mathbf{X}}_{t+p-W_{\text{past}}+1:t+p}\right). \tag{6.37}$$

Subsequently, the input window for iteration $s = 2$ now incorporates both the predictions

from the first and second iterations as follows

$$\tilde{\mathbf{X}}_{t+2p-W_{\text{past}}+1:t+2p} = \begin{bmatrix} y_{t+2p-W_{\text{past}}+1} & z^{(1)}_{t+2p-W_{\text{past}}+1} & \cdots & z^{(N_{\text{selected}})}_{t+2p-W_{\text{past}}+1} \\ y_{t+2p-W_{\text{past}}+2} & z^{(1)}_{t+2p-W_{\text{past}}+2} & \cdots & z^{(N_{\text{selected}})}_{t+2p-W_{\text{past}}+2} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{y}_{t+1} & \tilde{z}^{(1)}_{t+1} & \cdots & \tilde{z}^{(N_{\text{selected}})}_{t+1} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{y}_{t+p} & \tilde{z}^{(1)}_{t+p} & \cdots & \tilde{z}^{(N_{\text{selected}})}_{t+p} \\ \hat{y}_{t+p+1} & \tilde{z}^{(1)}_{t+p+1} & \cdots & \tilde{z}^{(N_{\text{selected}})}_{t+p+1} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{y}_{t+2p} & \tilde{z}^{(1)}_{t+2p} & \cdots & \tilde{z}^{(N_{\text{selected}})}_{t+2p} \end{bmatrix}, \tag{6.38}$$

where the last p rows ($t+p+1$ to $t+2p$) of the target variable are updated using the predictions from iteration $s=1$, and the previous p rows ($t+1$ to $t+p$) of the target variable are updated using predictions from the initial iteration $s=0$. The other input variables are updated following the same method mentioned before.

For the following iteration $s>2$, the target variable $\tilde{y}$ at each position is updated according to all previous prediction results using

$$\tilde{y}_{t+j} = \begin{cases} y_{t+j}, & t+j \le t, \\ & \text{(historical observed data)} \\ \hat{y}_{t+j}, & t+1 \le t+j \le t+p, \\ & \text{(initial prediction, iteration 0)} \\ \hat{y}_{t+j}, & t+p+1 \le t+j \le t+2p, \\ & \text{(iteration 1 prediction)} \\ \vdots & \\ \hat{y}_{t+j}, & t+(s-1)p+1 \le t+j \le t+sp, \\ & \text{(iteration } s-1 \text{ prediction)} \end{cases} \tag{6.39}$$

For all other input variables ($z^{(1)}, \ldots, z^{(N_{\text{selected}})}$), predictions at future time steps remain independently updated by external prediction models $g_{\psi_p}(\cdot)$:

$$\tilde{z}^{(n)}_{t+j} = g_{\psi_p}\left(\mathbf{E}_{t+j}\right), \quad n = 1, 2, \ldots, N_{\text{selected}}, \quad j > 0, \tag{6.40}$$

with $\mathbf{E}_{t+j}$ representing external data utilized by these specialized prediction models.

In summary, the input window structure for each iteration s is uniquely configured by integrating predictions from all preceding iterations. Specifically, only the first column, representing the target variable, is recursively updated based on the LSTM forecast. Regarding the exogenous input variables, the shifted-delay structure reduces the need for availability of their future forecasts. In case of relatively long delays resulting from applying the GSA-shifted delays, upstream hydrological variables such as river discharge can often use their readily available observed values from earlier dates. However, when variables have short or zero delay, or when the forecasting horizon extends beyond the available shifted observations, these inputs may still be supplemented by external, independent forecasting models, such as physics-based models. Consequently, while the equations defining the input windows vary at each iteration due to these dynamic updates, the fundamental strategy remains consistent: the target variable is recursively updated, and all other variables are supplemented with externally derived predictions. This framework ensures both flexibility in response to new data and reliability in maintaining a stable prediction strategy.

6.4 Case Study

6.4.1 The Southwest Pass (SWP) in the United States

The SWP serves as the principal deep-draft navigation channel in the Mississippi River Delta, providing the main conduit for maritime traffic between the Gulf of America and the Port of New Orleans. As one of the busiest commercial deep-draft waterways in the United States, SWP plays a critical role in connecting the U.S. Midwest to global markets. A zoomed-in view of the SWP is shown in the inset map at the bottom right of Figure 6.8. Spanning approximately 35 miles in length and ranging from 600 to 750 feet in width, the SWP is divided into 13 distinct NCF reaches. To support commercial navigation, the channel is maintained at a depth of 50 feet, with a recommended draft limit of 45 feet [271]. Unpredictable shoaling events in the SWP can disrupt navigation, impacting deep-draft vessel access to four of the nation's top fifteen ports by tonnage, which are Baton Rouge, Plaquemines, New Orleans, and South Louisiana. Collectively, these ports manage approximately 500 million short tons of cargo each year [150]. For example, from 2015 to 2019, navigation in the SWP was disrupted for an average of 93 days each year, resulting in increased shipping costs and delays [182]. A distinguishing feature of the SWP is its exceptionally high number of hydrographic surveys, averaging over 1,600 per year (number of entries) between 2012 and 2024. This extensive dataset provides a solid foundation for

applying advanced ML models to uncover the underlying patterns. However, the abundance of data also introduces high-frequency noise into the raw measurements CIR calculation, making it necessary to apply data filtering (SG filter mentioned before) prior to model configuration to ensure more reliable analyses.

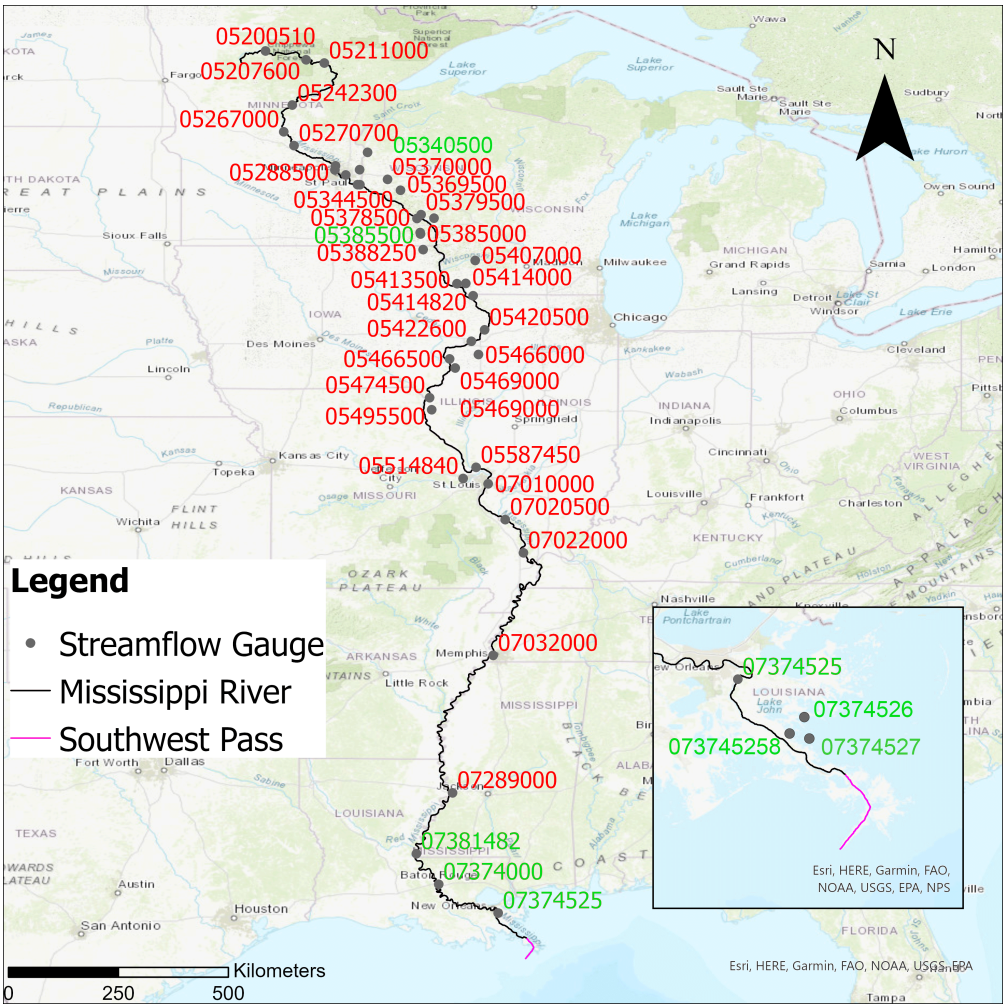


Figure 6.8: Geographical background of SWP and the selected gauge station locations (Station IDs in Table 6.3 are highlighted in green).

Figure 6.9 presents a comparison of the original sediment volume and CIR data with their IQR-filtered counterparts and the resulting sediment volume series. The threshold q is set to 2 in this case. This value was selected based on sensitivity testing over a range of candidate thresholds, including $q = 1, 1.5, 2, 2.5, \ldots, 10$. The choice of $q = 2$ provides a balance between sensitivity in detecting abnormal values and robustness in retaining meaningful sedimentation patterns. Specifically, it effectively removes most abnormal CIR trends, such as sudden artificial drops likely associated with dredging activities or spikes caused by

extreme events, while preserving the natural variability of sediment dynamics. As shown in Figure 6.9, several significant decreases in the original series were removed after filtering, as they are likely attributable to dredging events. The same threshold was also tested on another region of interest, the Houston Ship Channel (HSC), and could be applied without modification, suggesting reasonable cross-region robustness of the selected IQR threshold.

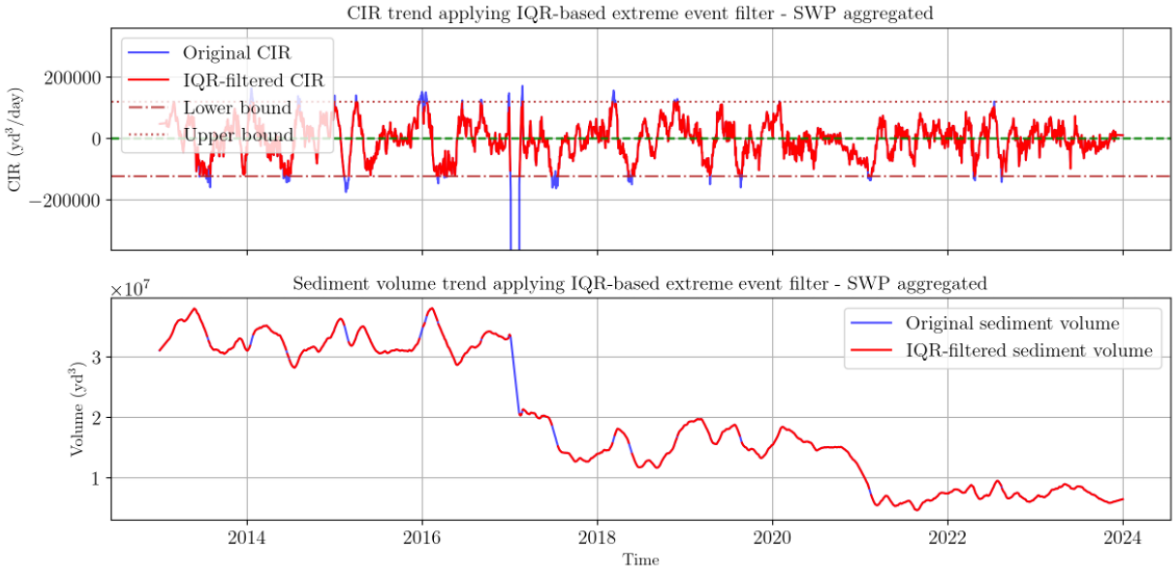


Figure 6.9: Impact of IQR-based outlier removal on CIR (top) and sediment volume (bottom) trends for the aggregated SWP data.

Since upstream river discharge measurements play a vital role in understanding shoaling processes, we incorporated selected upstream gauge data as key inputs for model training. In the United States, these measurements are primarily obtained from the NWIS managed by the USGS. In this case study, the selected stations primarily cover the Mississippi River and adjacent stations, providing river discharge measurements. After the data cleaning process, a total of 43 stations were retained for analysis. Figure 6.8 presents the locations of the selected gauges along with their corresponding station Ids. A total of 56 stations from USGS and NOAA—covering not only the aforementioned river discharge data but also other potential variables—are incorporated into this case study for comparison.

6.4.2 Proposed Framework Setup

6.4.2.1 GSA Setup

Aggregated first-order Sobol indices are employed to rank the significance of all input variables. For a given number of input–output samples, the only parameter requiring ad-

justment in the proposed sampling method is N_{domain}, which represents the number of equally probable intervals. Selecting the value of N_{domain} involves a trade-off: increasing N_{domain} enhances the accuracy of the outer loop of $\mathbb{E}_{\Phi}\left(V_{\Phi^l}(y)\right)$, but reduces it for the inner loop. Thus, choosing an appropriate N_{domain} is crucial. For this case study, $N_{\text{domain}} = 30$ was determined as optimal, based on both previous experience from earlier research [264] and the overall scale of the dataset.

To identify the most influential input variables using GSA across different time shifts, the analysis considers the number of shifted days for river discharge variables to range from 0 to 365, designating a maximum lag of one year. The outcomes of the GSA performed with these shifted inputs will be presented in Section 6.5.

6.4.2.2 LSTM-NARX Model Setup

As mentioned in Section 6.3.4, 6.3.5, and 6.3.6, an LSTM-NARX model with shifted inputs, modified loss function, and a recursive prediction approach is proposed in this paper. A comprehensive grid search was conducted to identify the optimal set of hyperparameters for the proposed LSTM-NARX model. The search space was defined based on a combination of prior experience and empirical studies, encompassing the following ranges for each hyperparameter (optimal hyperparameter is highlighted in bold):

- Number of LSTM Layers: 1, 2, **3**, 4.
- Units per LSTM Layer: 16, 32, **64**, 128.
- Batch Size: **16**, 32, 64.
- Epochs: 100, **300**, 600, 1000.
- Learning Rate: 10^{-2}, 10^{-3}, 10^{-4}.
- Dropout Rate: 0, 0.05, 0.1, **0.2**, 0.3.
- Optimizer: Adam, RMSprop, SGD.
- Return the Sequence: **True**, False.
- w_{high}: 2, **5**, 20, 50.
- Input Window Size: 30, **60**, 90, 120.

Each possible combination was evaluated using cross-validation on the training set to ensure robust model selection and avoid overfitting. Performance metrics were monitored on a validation split, with the configuration yielding the lowest validation error selected as optimal. After the grid search, the optimal LSTM-NARX architecture consists of three LSTM layers, each containing 64 units, a batch size of 16, and is trained for 300 epochs. The learning rate was set to 0.0001, with a dropout rate of 0.2 to prevent overfitting. The Adam optimizer was found to provide the best convergence, and the model was configured to return the full output sequence at each time step. When the modified loss function was incorporated, the optimal value of w_{high} was found to be 5. The comparison of different w_{high} will be shown in Section 6.5.2.3.

The optimal input window size was identified as 60 days preceding the prediction date. For the recursive prediction approach, various combinations of its settings— specifically, the number of iterations and the length of each prediction step— were systematically evaluated. The configuration of three iterations with 10 days per iteration was found to be optimal. The 3×10-day setup provides sufficient temporal granularity to capture evolving dynamics while maintaining robust forecast accuracy across the entire prediction window. For the direct prediction method, the output covers the next $W_{\text{pred}} = 30$ days. This window size was selected based on a grid search across candidate lengths of 30, 60, 90, and 120 days, with consideration given to practical factors such as seasonal variability. Adopting a 60-day input window achieves a balance between predictive accuracy, computational efficiency, and minimizing the risk of overfitting.

In this case study, aggregated sediment volume data were collected from January 1, 2013, to December 31, 2023. The dataset was divided as follows: data from 2013 to the beginning of 2020 forming the training set, data from 2020 to 2021 constituting the validation set, and data from 2021 through the end of 2023 serving as the test set. The model was trained to generate 30-day forecasts in total for each timestep within the test period. The full test set consists of 1006 filtered time steps, covering the period from January 1, 2021, to December 31, 2023.

6.4.3 Evaluation Metrics

In this case study, we employed three standard evaluation metrics— Root Mean Squared Error (RMSE), normalized RMSE (nRMSE), and Mean Absolute Percentage Error (MAPE)— to assess predictive performance from multiple perspectives. These metrics are widely used in regression and time-series forecasting tasks. RMSE was employed as one of the primary metrics to quantify the accuracy of $W_{\text{pred}} = 30$ days prediction windows. Let

$\mathbf{y}_{t+1:t+W_{\text{pred}}}$ denote the vector of observed true values and $\hat{\mathbf{y}}_{t+1:t+W_{\text{pred}}}$ the corresponding predicted values for each test instance t, with the total number of test samples denoted as T_{test}. The RMSE is defined as

$$\text{RMSE} = \frac{1}{T_{\text{test}}} \sum_{t=1}^{T_{\text{test}}} \sqrt{\frac{1}{W_{\text{pred}}} \sum_{k=1}^{W_{\text{pred}}} (y_{t+k} - \hat{y}_{t+k})^2}. \tag{6.41}$$

Since the observed sediment volumes can vary substantially in magnitude across the test windows, RMSE alone may not provide a fair basis for comparing predictive performance over the entire test set. To address this, nRMSE was used; it normalizes the RMSE by the mean observed value within each window, enabling consistent comparisons across different periods and varying data scales. For each prediction window that starts with $t+1$, nRMSE is computed as

$$\text{nRMSE}_t = \frac{\sqrt{\frac{1}{W_{\text{pred}}} \sum_{k=1}^{W_{\text{pred}}} (y_{t+k} - \hat{y}_{t+k})^2}}{\frac{1}{W_{\text{pred}}} \sum_{k=1}^{W_{\text{pred}}} y_{t+k}}. \tag{6.42}$$

To summarize model performance across all test windows, the overall nRMSE is given by

$$\text{nRMSE} = \frac{1}{T_{\text{test}}} \sum_{t=1}^{T_{\text{test}}} \text{nRMSE}_t. \tag{6.43}$$

In addition, MAPE was used to assess the relative accuracy of predictions across different periods with varying sediment volume scales. For each time step, MAPE is given by

$$\text{MAPE}_t = \frac{100}{W_{\text{pred}}} \sum_{k=1}^{W_{\text{pred}}} \left| \frac{y_{t+k} - \hat{y}_{t+k}}{y_{t+k}} \right|. \tag{6.44}$$

To summarize model performance across all T_{test}, the overall MAPE is computed as:

$$\text{MAPE} = \frac{1}{T_{\text{test}}} \sum_{t=1}^{T_{\text{test}}} \text{MAPE}_t. \tag{6.45}$$

6.5 Results and Discussion

6.5.1 GSA Results

As mentioned in Sections 6.3.4 and 6.4.2.1, for river discharge data, we explicitly considered time lags to account for travel time and propagation delays between upstream measurements and downstream shoaling responses. For non-discharge variables, no temporal shifting

was applied. The top ten input variables identified by the GSA are summarized in Table 6.3. The choice of using the top 10 input variables was made after systematically testing several candidate numbers of variables (e.g., 3, 5, 10, 20, 30). Through these experiments, we observed that smaller sets such as 3 or 5 variables often missed important contributing signals, while very large sets (e.g., 20 and 30) introduced redundant or noisy predictors that did not improve performance and, in some cases, reduced model robustness. Selecting 10 variables provided a balanced trade-off: it retained sufficient predictive information while avoiding overfitting or the inclusion of weakly relevant variables. Therefore, we selected the top 10 input variables as the final configuration for the model. The leading contributors are dominated by salinity and specific conductance measurements from stations near the lower Mississippi region, including salinity at the Mississippi River at Baton Rouge, LA, and Cow Bayou at American Bay, as well as specific conductance at Northeast Bay Gardene and Black Bay near Snake Island. These variables highlight the strong influence of salinity gradients and conductivity patterns on shoaling dynamics. Additional high-ranking variables include discharge and stage from the Mississippi River at Baton Rouge, LA, and discharge from the Root River at Houston, MN, which underscore the role of both upstream and downstream hydrodynamics in driving sediment transport. Other top contributors, such as water temperature at St. Croix River, WI, and stage observations at Belle Chasse, LA, and Old River Outflow Channel, further emphasize the interplay between seasonal environmental factors and river stage conditions in controlling sediment dynamics within the SWP system.

In this case study, several input variable strategies are evaluated:

- The first scheme considers only the top most sensitive variables selected from the full set of potential inputs (listed in Table 6.3).
- The second scheme also uses the top sensitive variables, but with river discharge data shifted according to their optimal shifted days.
- The third scheme restricts the inputs to river discharge data alone, without applying any shifting.
- Finally, the fourth scheme also considers only river discharge data, but with the inclusion of the optimal shifted days.

When calculating the aggregated GSA, shifted day counts ranging from 0 to 365 were considered for all available upstream gauged discharge data. In contrast, the aggregated GSA is calculated without applying any time shifting for all other exogenous variables. Figure 6.10 presents samples of the aggregated GSA index values and their corresponding optimal shifts over a 365-day period for four representative upstream gauges. The stations represented in

Table 6.3: Top 10 input variables ranked in descending order by aggregated GSA index.

Location	Measurement Type	Unit	Station I
Mississippi River at Baton Rouge, LA	Salinity	mg/L	07374000
Northeast Bay Gardene near Point-A-LA-Hache, LA	Specific Conductance	$\mu S\,cm^{-1}$ @ 25 °C	07374527
Cow Bayou at American Bay nr Pointe-A-La-Hache, LA	Salinity	mg/L	073745258
St. Croix River at St. Croix Falls, WI	Water Temperature	°C	05340500
Mississippi River at Baton Rouge, LA	Discharge	cfs	07374000
Mississippi River at Baton Rouge, LA	Stage	ft	07374000
Root River at Houston, MN	Discharge	cfs	05385500
Black Bay nr Snake Island nr Pointe-A-La-Hache, LA	Specific Conductance	$\mu S\,cm^{-1}$ @ 25 °C	07374526
Mississippi River at Belle Chasse, LA	Stage	ft	07374525
Old River Outflow Channel below Hydropower Channel	Stage	ft	07381482

the figure are situated in various locations; they correspond to stations along the Mississippi River and others in its vicinity, spanning from the nearest stations in Louisiana to the far north in Minnesota, which can be found in Figure 6.11. The figure clearly demonstrates that the GSA index patterns and the optimal number of shifted days vary depending on the station's location. In general, as discussed in Section 6.3.4.3, stations closer to the SWP exhibit shorter delays, while those farther away tend to have longer optimal shift periods. The results shown here are consistent with this spatial relationship in most cases.

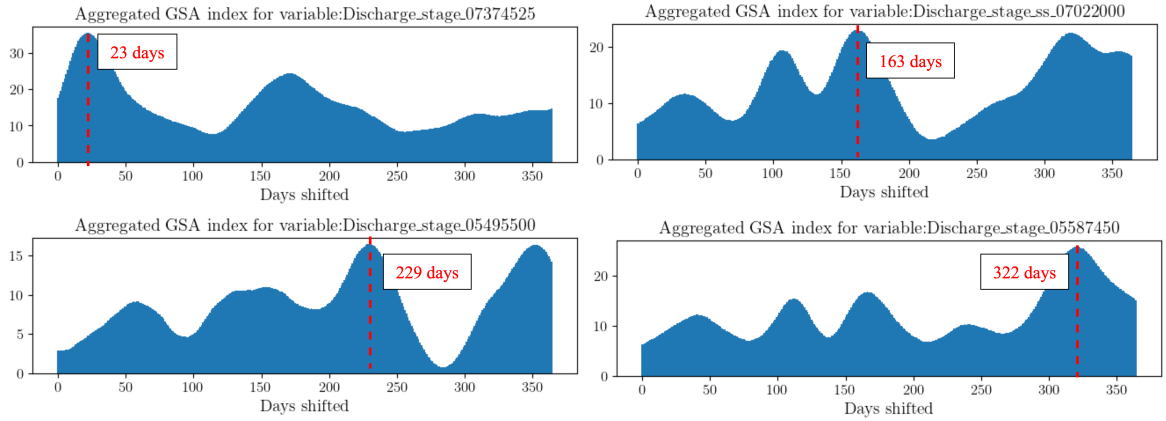


Figure 6.10: Samples of the aggregated GSA index values and corresponding optimal shifts.

To be more specific, Figure 6.11 illustrates the spatial distribution of the most influential river discharge gauge stations along the Mississippi River, with the height of each bar representing the optimal number of shifted days as determined by the GSA analysis. A distinct spatial pattern emerges: stations located farther upstream from the target site generally exhibit taller bars, corresponding to larger shifted-day values, whereas stations closer to the

target show shorter bars with smaller shifts. This pattern reflects the varying travel times of hydrological signals as they propagate through the river system. The zoomed-in panel on the right focuses on a pair of nearby stations outlined by the dotted square. Situated between them is the Melvin Price Lock and Dam, a hydraulic structure that likely influences flow travel time. This could account for the significant difference in the optimal number of shifted days between the two stations, even though they are close together along the river compared to other adjacent stations.

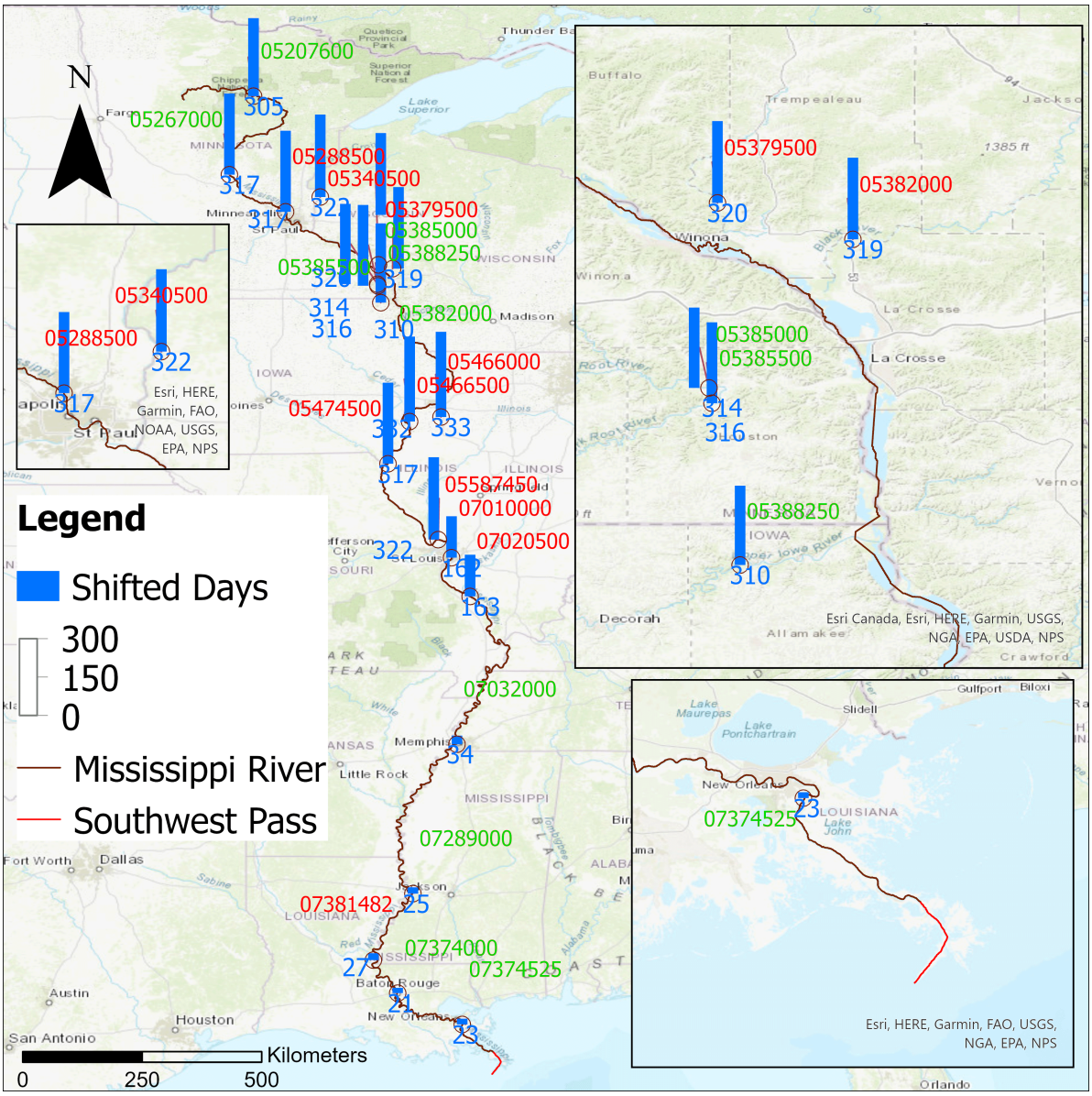


Figure 6.11: Map of the most influential station locations along the Mississippi River, annotated with their corresponding optimal number of shifted days as determined by GSA (Station IDs in Table 6.4 are highlighted in green).

For the proposed model configuration—recursive prediction utilizing only shifted-discharge input variables as additional features with a weighted loss function—the aggregated GSA is computed with an input window size of $W_{\text{past}} = 60$ and an output window size of $p = 10$. Based on the results of the GSA analysis, the top $N_{\text{selected}} = 10$ input variables are then selected for inclusion in the model inputs. Table 6.4 shows the most influential stations for the proposed model.

Table 6.4: Top 10 river discharge gauges ranked by aggregated GSA index, with corresponding optimal number of shifted days (ONSD).

Gauge ID	**Location**	**GSA**	**ONSD**
05385000	Root River at Houston, MN	41.37	314
07374000	Mississippi River at Baton Rouge, LA	37.18	21
05388250	Upper Iowa River at Dorchester, IA	36.40	310
07374525	Mississippi River at Belle Chasse, LA	35.41	23
07289000	Mississippi River at Vicksburg, MS	35.36	25
05382000	Black River near Galesville, WI	31.97	319
05207600	Mississippi River at Willow Beach at Ball Club, MN	30.83	305
05385500	South Fork Root River at Houston, MN	28.25	316
05267000	Mississippi River Near Royalton, MN	28.20	317
07032000	Mississippi River at Memphis, TN	27.63	34

6.5.2 LSTM-NARX Model Prediction Results Comparison

In this section, we analyze how different model design choices affect predictive performance. The comparisons are structured to move from basic modeling approaches to more advanced proposals, with each new step building on the optimal settings found in earlier ones. This progression helps highlight the specific impact of each improvement on the overall results.

6.5.2.1 Impact of Prediction Strategy: Direct vs. Recursive Forecasting

We first begin by evaluating the impact of the prediction strategy itself, comparing direct forecasting (30-day ahead prediction method) with recursive forecasting (predicting shorter segments sequentially and using each prediction as input for the next). The comparison is performed under a baseline configuration, ensuring all other modeling choices are held constant to isolate the effect of the prediction strategy.

Figure 6.12 compares the CIR and the back-calculated sediment volume predictions obtained from the proposed recursive prediction scheme and the direct multi-step approach

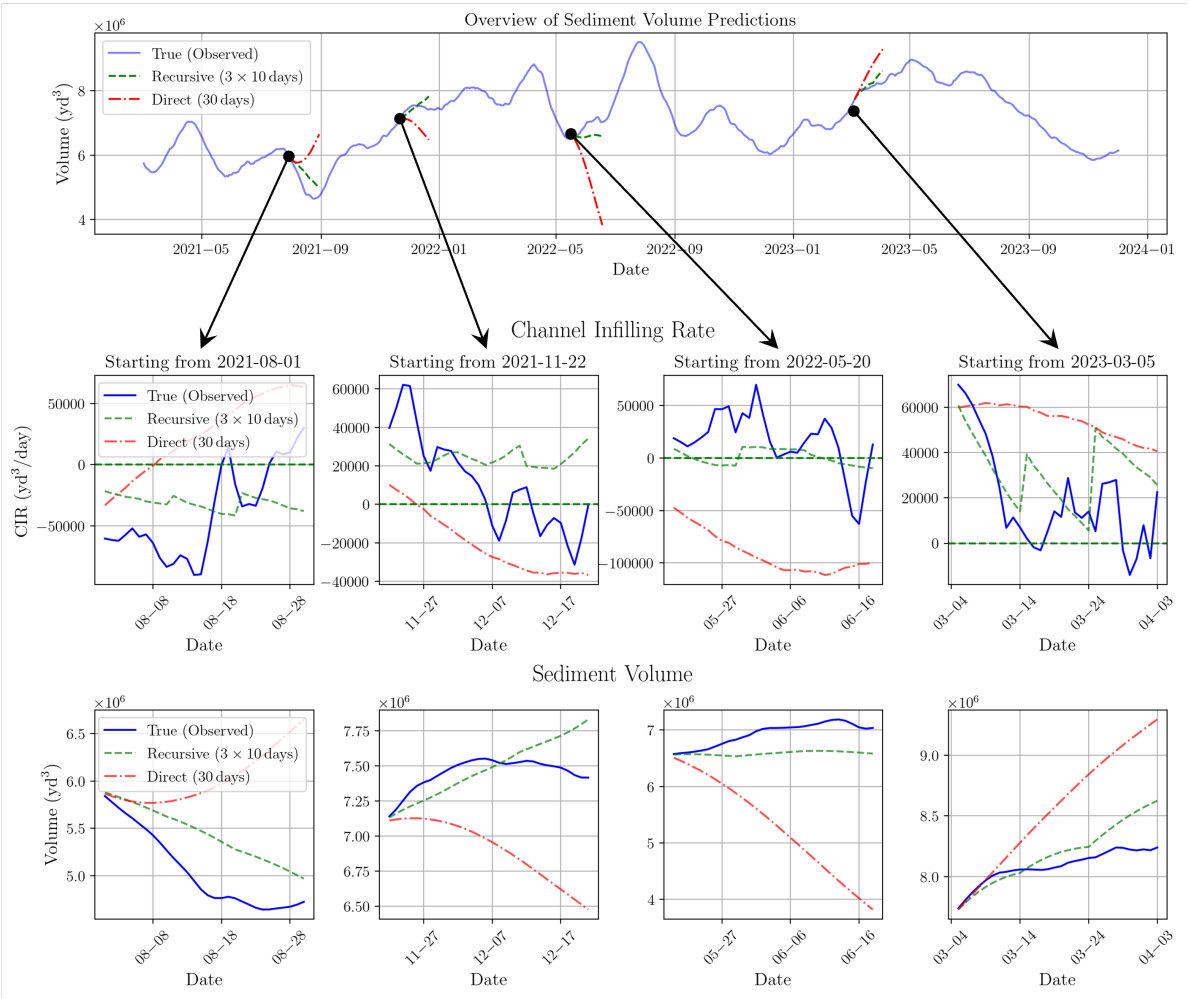


Figure 6.12: Comparison of CIR and back-calculated sediment volume predictions using recursive and direct predictions at selected periods.

against the actual values for selected periods. The top panel provides an overview of sediment volume variations across the full study period, highlighting four representative starting timesteps (dates) for detailed examination. In each case, the middle row presents CIR forecasts, while the bottom row projects the corresponding sediment volume trajectories reconstructed from CIR predictions. Across all selected periods, the recursive prediction method (green dashed lines) more closely follows the observed trends (blue solid lines) than the direct prediction method (red dashed lines), while particularly capturing short-term fluctuations and transition points in CIR. This improved fidelity translates into more accurate sediment volume reconstructions, as evident in the bottom panel; more specifically, the recursive scheme maintains closer alignment with measured sediment volumes, while the direct approach often exhibits systematic divergence over the forecast horizon. Notably, during periods of rapid change— such as the late 2022 and early 2023 intervals— the direct method tends to accumulate larger prediction errors, leading to substantial overestimation of sediment volumes.

Table 6.6 quantitatively reinforces the performance differences illustrated in Figure 6.11 by comparing RMSE, nRMSE, and MAPE values for the recursive and direct prediction methods across multiple forecast start dates and for the full test set. Across all evaluation periods ("full test set"), the recursive approach consistently achieves lower error metrics than the direct method.

For example, at the 2021-08-01 start date, the recursive prediction reduces RMSE by nearly 63 % and MAPE by more than half relative to the direct approach. Similar improvements are observed for other start dates, with the most pronounced gains occurring during periods of rapid sediment change (e.g., 2022-05-20), such that the direct method's RMSE exceeds 1.9 million yd^3 compared to 0.39 million yd^3 for the recursive scheme. On the full test set, the recursive model attains an RMSE of 716,345 yd^3 and an nRMSE of 0.0817, both lower than the 792,283 yd^3 and 0.0924 achieved by the direct approach, alongside a reduced MAPE (6.74 % vs. 7.64 %).

These results confirm that the recursive approach effectively mitigates error propagation in extended forecasts, yielding more reliable long-horizon predictions essential for operational dredging planning.

6.5.2.2 Influence of Shifted Input Variables on Prediction Performance

After identifying the recursive prediction scheme as the optimal strategy, we fix this setting and proceed to assess the role of shifted input variables—that is, exogenous variables with time delays optimized via GSA, applied only to discharge variables. By comparing results with and without input shifting, we quantify how aligning input variables with their

Table 6.5: Comparison of prediction metrics for recursive and direct prediction methods.

Prediction Method	Starting Time	RMSE	nRMSE	MAPE
Recursive prediction	2021-08-01	**430,879**	**0.0853**	**8.06**
	2021-11-22	**168,310**	**0.0226**	**1.82**
	2022-05-20	**385,678**	**0.0557**	**4.88**
	2023-03-05	**169,239**	**0.0209**	**1.57**
	Full test set	**716,345**	**0.0817**	**6.74**
Direct prediction	2021-08-01	1,157,532	0.2293	20.31
	2021-11-22	607,644	0.0815	7.29
	2022-05-20	1,930,787	0.2787	23.35
	2023-03-05	597,489	0.0739	5.92
	Full test set	792,283	0.0924	7.64

most relevant temporal relationships enhances the model's ability to capture the physical processes influencing the prediction target. All other aspects of the model, including the previously selected prediction strategy, remain unchanged for this comparison.

Figure 6.13 and Table 6.6 together present a comprehensive assessment of how different input configurations influence the accuracy of the 3×10-day recursive prediction scheme for CIR and sediment volume forecasting. The figure provides a visual comparison across four representative starting dates, while the table quantifies performance in terms of RMSE, nRMSE, and MAPE.

In Figure 6.13, predictions generated using shifted discharge-only inputs (green dashed lines) consistently align more closely with observed CIR and sediment volume than those from the unshifted discharge-only configuration (red dashed lines). This improvement is especially evident during periods of rapid change, such as May 2021 and September 2021; during these time periods, the unshifted series exhibits pronounced phase lags and cumulative biases, while the shifted approach captures both the timing and magnitude of observed variations. A similar pattern emerges for the top-input configurations: without shifting any variable, the GSA-selected variables (cyan dashed lines) fail to fully replicate the observed dynamics, whereas incorporating optimal discharge shifts (magenta dashed lines) produces predictions that better match both the shorter-term CIR fluctuations and the longer-term sediment volume trajectories.

The statistical results in Table 6.6 reinforce the outlined visual trends. For example, for the time instance with the start time of 2021-05-01, applying discharge shifts reduces RMSE from 562,793 yd^3 to 74,431 yd^3 and MAPE from 7.26 % to 0.81 % in the discharge-only setting. Similarly, for September 2021, discharge shifting within the top-input set reduces RMSE from 1,050,776 yd^3 to 135,772 yd^3 and cuts MAPE by more than 13 percentage

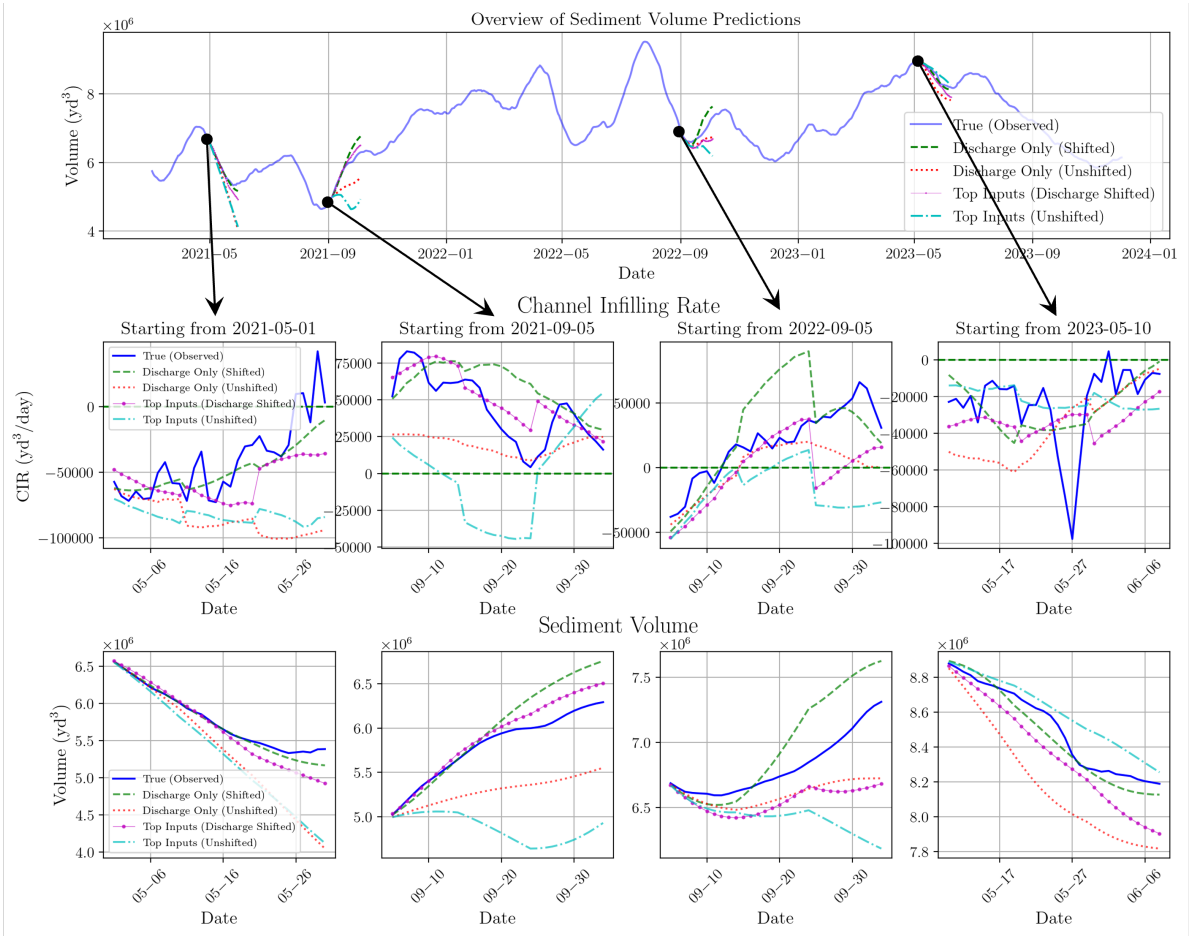


Figure 6.13: Comparison of CIR and back-calculated sediment volume predictions using different input combinations under the 3×10-day recursive prediction scheme at selected periods.

Table 6.6: Comparison of prediction metrics for different input configurations.

Extra Input	**Starting Time**	**RMSE**	**nRMSE**	**MAPE**
Discharge Only (Shifted)	2021-05-01	**74,431**	**0.0129**	**0.81**
	2021-09-05	273,442	0.0471	3.36
	2022-09-05	**259,024**	**0.0379**	**2.94**
	2023-05-10	**65,017**	**0.0076**	**0.66**
	Full test set	**658,567**	**0.0769**	**6.49**
Discharge Only (Unshifted)	2021-05-01	562,793	0.0973	7.26
	2021-09-05	558,026	0.0962	8.66
	2022-09-05	259,446	0.0381	2.98
	2023-05-10	335,891	0.0395	3.71
	Full test set	681,714	0.0805	7.02
Top Inputs (Discharge Shifted)	2021-05-01	180,925	0.0313	2.33
	2021-09-05	**135,772**	**0.0234**	**1.78**
	2022-09-05	305,767	0.0449	3.69
	2023-05-10	164,205	0.0193	1.71
	Full test set	668,444	0.0775	6.84
Top Inputs (Unshifted)	2021-05-01	559,323	0.0967	7.79
	2021-09-05	1,050,776	0.1811	15.57
	2022-09-05	501,169	0.0735	5.51
	2023-05-10	118,100	0.0139	1.16
	Full test set	716,345	0.0817	6.74

points. Across the full test set, "Discharge Only (Shifted)" achieves the lowest nRMSE (0.0769) among all configurations, while "Top Inputs (Discharge Shifted)" delivers competitive accuracy (nRMSE 0.0775) and benefits from the broader feature set when combined with optimal temporal alignment.

Combining their respective findings, the figure and the table demonstrate that the advantages of expanding the input feature set are fully realized only when critical upstream discharge variables are temporally aligned with downstream shoaling responses. Physically consistent time-shifting not only improves phase matching in CIR predictions but also prevents long-horizon drifts in reconstructed sediment volumes, ultimately enhancing the operational value of the recursive forecasting framework.

6.5.2.3 Evaluation of Loss Functions: Standard MSE vs. Modified Loss

Finally, with the optimal prediction strategy and input configuration established, we investigate the impact of the loss function used during training. The previous combination—using recursive predictions and shifted discharge variables within the standard loss function—shows strong performance in forecasting. However, its accuracy declines in certain portions of the test set, especially at peaks and valleys. The modified loss function can address this issue, especially where the CIR crosses zero. For example, as illustrated in Figure 6.14, the standard MSE loss fails to capture the peaks and valleys in sediment volume prediction for the four sample periods shown. In contrast, the weighted MSE loss more accurately tracks the CIR as it crosses zero, resulting in improved predictions during these critical intervals. To be more specific, Table 6.7 demonstrates that the Weighted MSE loss function ($w_{\text{high}} = 5$) consistently outperforms the standard MSE loss function in sediment volume prediction accuracy. Notably, the weighted approach achieves lower RMSE, nRMSE, and MAPE values across the full test set. For example, the RMSE and nRMSE for the full test set are reduced from 658,567 and 0.0769 (standard MSE) to 523,351 and 0.0645 (weighted MSE), respectively. This performance gain is especially pronounced for the challenging periods, such as the windows starting from 2022-04-05 and 2022-04-29, during which the weighted loss yields significant improvements in all metrics. These results highlight the effectiveness of the weighted MSE approach in capturing critical CIR transitions and improving overall predictive performance.

To systematically evaluate the effect of the weighting parameter w_{high} on model performance, three distinct values were selected for analysis. Figure 6.15 presents a comparison of CIR predictions and back-calculated sediment volumes obtained using a weighted MSE loss function, with w_{high} set to 5, 20, and 50. The results clearly demonstrate the influence of weighting on prediction performance, particularly in balancing the accuracy of CIR dynam-

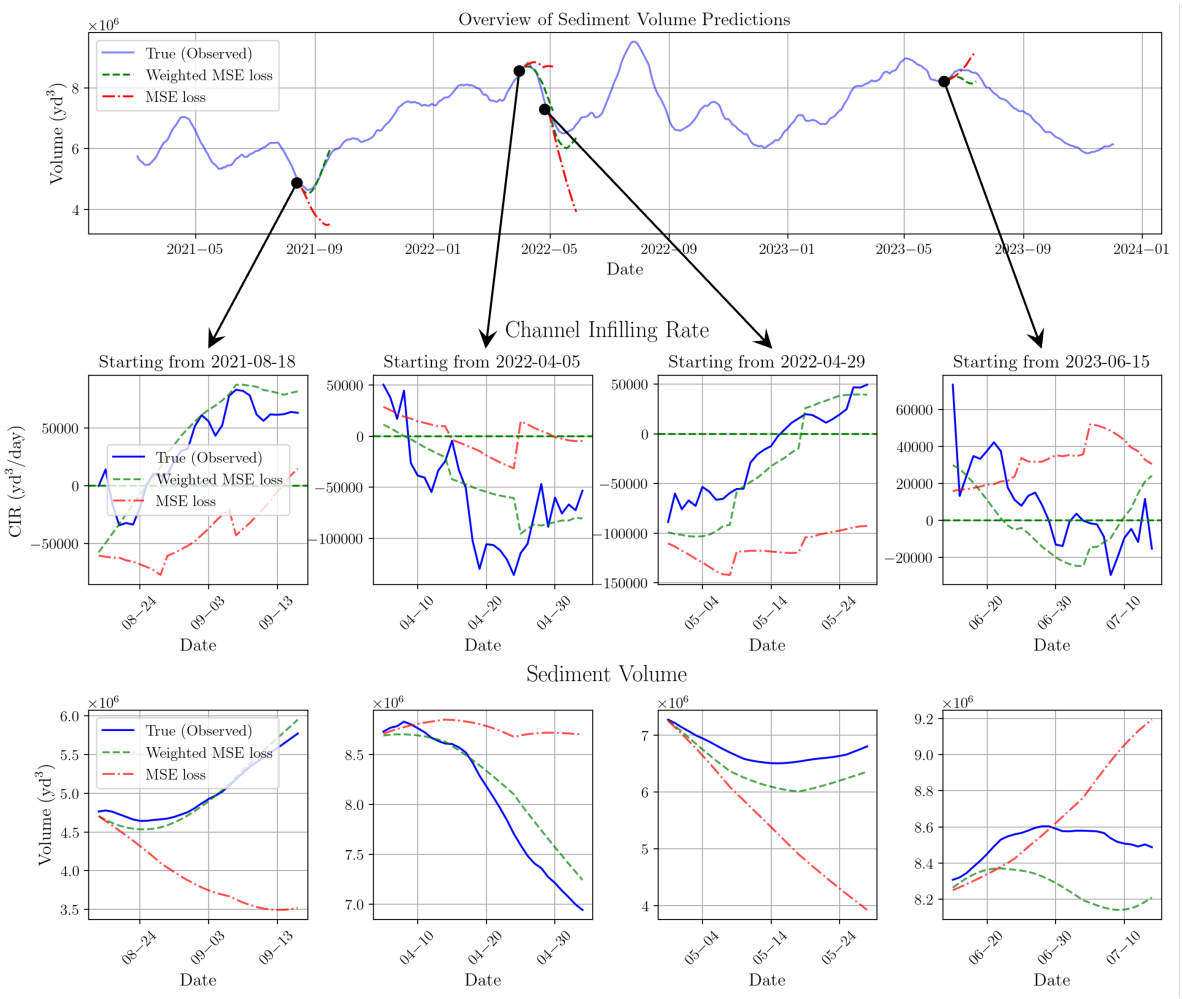


Figure 6.14: Comparison of CIR and back-calculated sediment volume predictions using standard and weighted MSE loss functions under the 3×10-day recursive prediction scheme with shifted gauge-only input variables at selected periods.

Table 6.7: Comparison of prediction metrics for standard MSE loss and weighted MSE loss with different w_{high} values.

Method / Weights	**Starting Time**	**RMSE**	**nRMSE**	**MAPE**
Standard MSE	2021-08-18	1,304,198	0.2601	20.91
	2022-04-05	924,193	0.1145	9.41
	2022-04-29	1,508,101	0.2243	18.50
	2023-06-15	310,636	0.0365	2.68
	Full test set	658,567	0.0769	6.49
Weighted MSE ($w_{\text{high}} = 5$)	2021-08-18	**97,684**	**0.0195**	**1.69**
	2022-04-05	**246,768**	**0.0306**	**2.55**
	2022-04-29	**378,202**	**0.0563**	**5.25**
	2023-06-15	280,324	0.0329	2.93
	Full test set	**523,351**	**0.0645**	**5.52**
Weighted MSE ($w_{\text{high}} = 20$)	2021-08-18	1,304,198	0.2601	20.91
	2022-04-05	924,194	0.1146	9.41
	2022-04-29	1,508,101	0.2243	18.50
	2023-06-15	310,636	0.0365	2.68
	Full test set	617,783	0.0739	6.19
Weighted MSE ($w_{\text{high}} = 50$)	2021-08-18	1,070,697	0.2135	18.04
	2022-04-05	1,889,344	0.2343	20.31
	2022-04-29	907,900	0.1350	10.49
	2023-06-15	**96,121**	**0.0113**	**0.95**
	Full test set	650,020	0.0749	6.27

ics against long-term sediment volume estimates. When $w_{\text{high}} = 5$, the model provides the best overall performance, with consistently lower errors across evaluation metrics (RMSE, nRMSE, and MAPE) shown in Table 6.7. For example, at the starting time of 2023-06-15, the prediction error reaches its minimum, with RMSE = 96,121, nRMSE = 0.0113, and MAPE = 0.95. These values highlight the model's ability to capture both short-term CIR fluctuations and long-term volume changes with high fidelity. This weight setting appears to strike the most effective balance between penalizing high CIR deviations and preserving the underlying temporal trends in sediment accumulation. In contrast, higher weights ($w_{\text{high}} = 20$ and 50) introduce noticeable trade-offs. While CIR predictions under these settings sometimes exhibit closer alignment with observed short-term variations, they generally lead to significant deterioration in overall accuracy, as reflected in elevated RMSE and nRMSE values. This indicates that over-weighting high CIR values can cause the model to overfit short-term noise, ultimately compromising long-term predictive accuracy.

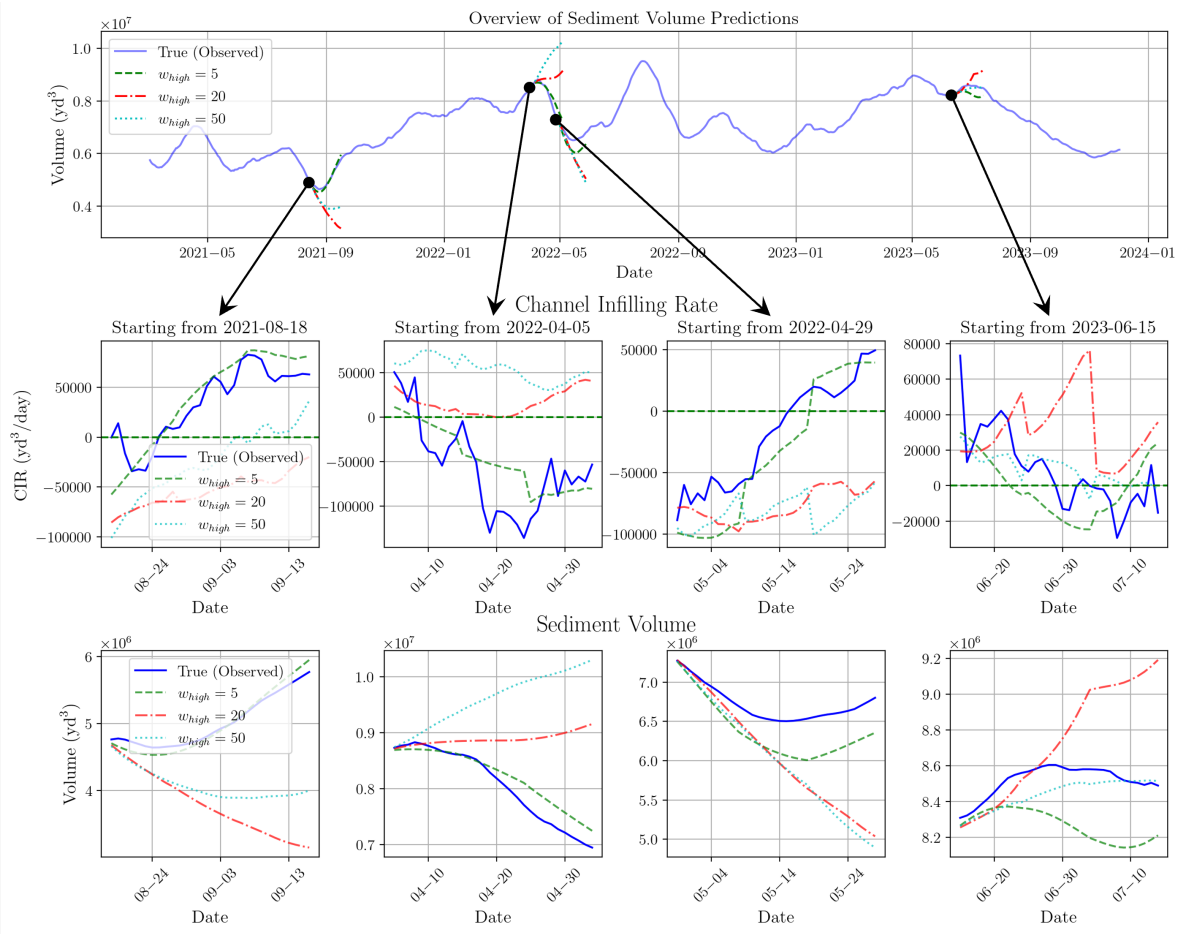


Figure 6.15: Comparison of CIR and back-calculated sediment volume predictions using weighted MSE loss function with $w_{\text{high}} = 5, 20, 50$.

6.6 Conclusions and Future Work

This study introduces a comprehensive forecasting framework for shoaling, demonstrated on predictions in the SWP, that combines three key novelties. First, the proposed recursive prediction method enhances 30-day horizon forecasts of the Channel Infilling Rate (CIR), which can be back-calculated to sediment volume. By iteratively updating inputs with recent predictions, the model better captures evolving temporal patterns and reduces error accumulation. Second, the use of GSA for input feature selection, combined with the identification of each variable's optimal number of shifted days, ensures that upstream discharge is temporally aligned with downstream shoaling responses. This alignment captures the delayed effects in sediment transport and improves predictive accuracy. Third, a weighted loss function is designed to place greater emphasis on zero-crossing points in the CIR, improving the model's ability to detect critical transitions in sediment accumulation trends.

Integrating these components results in sediment volume forecasts that are both more accurate and more dependable than those produced by traditional methods, delivering substantial benefits for navigation channel management. By incorporating delayed upstream influences, selecting the most informative variables, and concentrating on events with operational significance, this framework serves as a robust decision-support tool for optimizing dredging schedules and minimizing maintenance costs.

One limitation of the current implementation is that it relies on a systematic hyperparameter search over the LSTM architecture, learning rate, dropout rate, and the loss-function weight w_{high}. While this search was useful for validating the proposed framework in the SWP case study, repeating the same process in each new geographic region may limit rapid deployment. Future work will focus on enhancing deployment efficiency through Bayesian optimization, transfer learning, warm-starting from hydrologically similar regions, and physically informed parameter ranges for shifted delays and loss-function weights. At the same time, future research will also apply this methodology to other complex waterways and investigate the inclusion of additional physical and operational factors to further enhance predictive performance.

Another limitation is that the recursive prediction scheme assumes future exogenous variables are available or can be estimated by external forecasting models. In real-time deployment, errors in these exogenous variable forecasts may propagate through the recursive prediction process and affect long-horizon sediment-volume predictions. Future work will incorporate uncertainty propagation from exogenous variable forecasts, such as ensemble discharge forecasts or scenario-based inputs, to improve the operational robustness of the framework.

Lastly, although the proposed framework is designed to be transferable, the present study validates it only in the SWP region. Therefore, its generalizability to waterways with different sediment characteristics, hydrodynamic forcing, tidal asymmetry, and monitoring data availability requires further evaluation. Future work will apply the framework to additional waterways to assess its robustness, transferability, and region-specific calibration needs.

Acknowledgment

This work was supported by the United States Army Corps of Engineers through the U.S. Army Engineer Research and Development Center Research Cooperative Agreement W912HZ24C0044. The support is gratefully acknowledged.

CHAPTER 7

Conclusions and Future Work

7.1 Conclusions

This dissertation developed ML and hybrid modeling frameworks to support data-driven asset management for inland waterway networks. The work was motivated by the coupled nature of inland waterway operation, where navigation infrastructure, hydrologic conditions, and sediment dynamics jointly influence system reliability. Navigation locks and miter gates must be monitored and maintained under uncertain loading and limited inspection access; river discharge and water levels affect both structural responses and sediment transport; and shoaling changes channel geometry, reduces navigability, and influences dredging needs. Addressing these challenges requires predictive tools that can use sparse and heterogeneous data, incorporate physical knowledge, quantify uncertainty, and support operational decision-making. As mentioned in Section 1, following this motivation, the dissertation progressed from miter gate data augmentation to probabilistic damage diagnostics, to streamflow prediction with uncertainty quantification, and finally to shoaling and sedimentation forecasting.

The first part of the dissertation addressed data scarcity in miter gate SHM through a CycleGAN-based data augmentation framework. The results showed that monitoring information from a source gate can be translated into synthetic observations for a target gate while preserving meaningful relationships between hydraulic inputs and structural strain responses. The proposed accuracy verification strategy provided a practical way to screen generated observations before they were used in Bayesian inference. After incorporating the verified synthetic data into Bayesian updating, the estimated gap length became more accurate and the posterior uncertainty was reduced compared with the case using limited real observations alone. These findings demonstrate that generative data augmentation can improve probabilistic damage diagnostics when field monitoring data are sparse, provided that the generated data are carefully verified before being used for inference.

The second part extended this idea into a more comprehensive Bayesian diagnostic framework enhanced through domain translation. The proposed BiEDT framework combined translated observations, conditional invertible neural networks, physics-based Bayesian inference, and Bayesian model averaging to estimate miter gate damage with quantified uncertainty. The results showed that integrating domain-translated information with physics-based inference can improve diagnostic robustness compared with relying on a single source of information or a single diagnostic model. The comparison between CycleGAN-based and DANN-based domain translation also provided insight into how different transfer strategies affect damage estimation. In addition, Bayesian model averaging offered a principled way to combine probabilistic predictions and reduce dependence on one selected model. Overall, this study demonstrated that domain translation can be embedded within a Bayesian diagnostic framework to support more reliable and uncertainty-aware damage assessment for miter gates.

The third part focused on streamflow prediction with UQ. The hybrid framework combined physics-based river-routing information with ML models to improve discharge prediction in ungauged or sparsely monitored river reaches. The results indicated that incorporating physical-model outputs and selected hydrologic features can improve predictive performance compared with purely data-driven modeling under limited-data conditions. UQ further provided prediction intervals and reliability information, allowing the streamflow forecasts to be interpreted not only in terms of accuracy but also in terms of confidence. This study showed that hybrid modeling can use physics-based simulations as informative guidance while retaining the flexibility of ML, making it useful for risk-informed waterway operation and planning.

The fourth part developed a data-driven framework for shoaling and sedimentation forecasting. By using hydrographic survey data, hydrologic variables, and ML models, the study predicted CIR and cumulative sedimentation volume for the Houston Ship Channel and the Southwest Pass. The results showed that data-driven models can capture useful relationships between hydrologic conditions and sedimentation behavior, although performance varied across regions, modeling strategies, and input configurations. The comparison between reach-level and aggregated sedimentation predictions clarified how spatial aggregation affects forecasting performance, while the uncertainty analysis provided additional information on the reliability of the forecasts. These findings demonstrate the feasibility of using ML-based sedimentation forecasting to support dredging planning and navigation channel maintenance.

The fifth part further improved shoaling prediction by introducing shifted hydrologic delays, a customized loss function, and recursive forecasting. Sensitivity analysis identified

influential upstream discharge gauges and showed that delayed hydrologic effects are important for explaining downstream sedimentation behavior in the Southwest Pass. Incorporating these shifted inputs into a multi-input multi-output LSTM-NARX model improved the representation of temporal dependence between upstream discharge and downstream channel infilling. The recursive forecasting strategy enabled monthly prediction of channel infilling rate and back-calculation of sediment volume, making the framework more suitable for operational sediment management. This study demonstrated that accounting for delayed hydrologic forcing can enhance shoaling forecast performance and provide a more physically meaningful connection between streamflow dynamics and sediment accumulation.

Overall, the dissertation contributes to inland waterway asset management in three connected domains: infrastructure monitoring, hydrologic forecasting, and sediment management. Across these domains, the work demonstrates that ML methods are most effective when they are designed with physical context, uncertainty, and decision needs in mind. For miter gate monitoring, domain translation and Bayesian inference improve damage diagnostics under sparse data. For streamflow prediction, hybrid modeling combines physical routing information with data-driven adaptability. For shoaling forecasting, ML models and shifted-delay recursive prediction improve the ability to anticipate sedimentation and support dredging decisions. Together, these studies provide a connected framework for improving data utilization, predictive reliability, and uncertainty-aware decision-making in inland waterway networks.

7.2 Future Work

Although this dissertation advances ML-enabled asset management for inland waterway networks, several research directions remain. For miter gate SHM, systematic criteria for evaluating structural similarity between source and target gates should also be developed to reduce the risk of negative transfer. In addition, physics-informed constraints could be incorporated into generative models to ensure that synthetic observations remain consistent with structural mechanics and hydraulic loading conditions. For Bayesian diagnostics with domain translation, extending the BiEDT framework to other hydraulic infrastructure systems, such as tainter gates, spillways, dams, or other lock components, would also help evaluate its generalizability beyond miter gate gap detection.

For streamflow forecasting, similarly, future work should validate the hybrid modeling framework across broader geographic regions with different hydrologic regimes, climate conditions, and monitoring densities. Incorporating additional meteorological, hydrologic, and remote sensing variables may improve prediction in ungauged or data-scarce basins.

For shoaling and sedimentation forecasting, future studies should explore more flexible ways to learn delayed hydrologic effects, such as time-varying delays or distributed lag relationships between upstream discharge and downstream sedimentation. Testing the framework in other navigation channels would help evaluate its transferability, while coupling recursive ML forecasts with physics-based sediment transport models may improve long-horizon prediction and reduce error accumulation.

Overall, future research should continue to integrate ML methods, physics-based modeling, and domain knowledge to develop interpretable and uncertainty-aware tools for inland waterway management. Such tools can support more informed decisions in inspection planning, infrastructure maintenance, streamflow forecasting, navigation management, and dredging operations.

BIBLIOGRAPHY

[1] Yichao Zeng, Jice Zeng, Michael Todd, and Zhen Hu. Data augmentation based on image translation for bayesian inference-based damage diagnostics of miter gates. ASCE-ASME J Risk and Uncert in Engrg Sys Part B Mech Engrg, pages 1–62, 2024.

[2] Yichao Zeng, Jice Zeng, Michael Todd, and Zhen Hu. Augmenting bayesian inference-based damage diagnostics of miter gates based on image translation. In International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. American Society of Mechanical Engineers, 2024.

[3] Jice Zeng, Michael D Todd, and Zhen Hu. Probabilistic damage detection using a new likelihood-free bayesian inference method. Journal of Civil Structural Health Monitoring, 13(2):319–341, 2023.

[4] U.S. Army Corps of Engineers. Inland waterway navigation. Brochure (archived), 2005. Archived version from November 25, 2005.

[5] Waterways Council, Inc. Us inland waterways' role in a global marketplace. In the News article, Waterways Council, Inc., May 2024.

[6] Waterways Council, Inc. Waterways system: Learn about the future of our waterways. Waterways Council, Inc. website, 2025.

[7] Stuart D. Foltz. Investigation of mechanical breakdowns leading to lock closuresInvestigation of mechanical breakdowns leading to lock closures. Construction Engineering Research Laboratory (U.S.) Engineer Research and Development Center (U.S.), 2017.

[8] Smart gate 2007.

[9] Dongming Feng and Maria Q. Feng. Model updating of railway bridge using in situ dynamic displacement measurement under trainloads. Journal of Bridge Engineering, 20:04015019, 2015.

[10] Tong Guo, Aiqun Li, and Hao Wang. Influence of ambient temperature on the fatigue damage of welded bridge decks. International Journal of Fatigue, 30:1092–1102, 6 2008.

[11] Robert B. Randall. Vibration-based Condition Monitoring: Industrial, Aerospace and Automotive Applications. Wiley, Hoboken, 1. aufl.;1; edition, 2010.

[12] F. Necati Catbas, Melih Susoy, and Dan M. Frangopol. Structural health monitoring and reliability estimation: Long span truss bridge application with environmental monitoring data. Engineering Structures, 30:2347–2359, 9 2008.

[13] C. J. Morgan, B. F. Sparling, and L. D. Wegner. Use of structural health monitoring to extend the service life of the diefenbaker bridge. Journal of Civil Structural Health Monitoring, 12:913–929, 8 2022.

[14] Mukesh K Ramancha, Manuel A Vega, Joel P Conte, Michael D Todd, and Zhen Hu. Bayesian model updating with finite element vs surrogate models: Application to a miter gate structural system. Engineering Structures, 272:114901, 2022.

[15] Marcos E. Orchard and George J. Vachtsevanos. A particle filtering approach for online failure prognosis in a planetary carrier plate. International Journal of Fuzzy Logic and Intelligent Systems, 7:221–227, 12 2007.

[16] Matthew Daigle and Kai Goebel. Multiple damage progression paths in model-based prognostics. In 2011 Aerospace Conference, pages 1–10, 2011.

[17] Dawn An, Joo Ho Choi, Tony L. Schmitz, and Nam H. Kim. In situ monitoring and prediction of progressive joint wear using bayesian statistics. Wear, 270:828–838, 5 2011.

[18] Linxia Liao and Felix Köttig. Review of hybrid prognostics approaches for remaining useful life prediction of engineered systems, and an application to battery life prediction. IEEE Transactions on Reliability, 63(1):191–207, 2014.

[19] Andre Jesus, Peter Brommer, Robert Westgate, Ki Koo, James Brownjohn, and Irwanda Laory. Modular bayesian damage detection for complex civil infrastructure. Journal of civil structural health monitoring, 9(2):201–215, 2019.

[20] Jice Zeng and Young Hoon Kim. Identification of structural stiffness and mass using bayesian model updating approach with known added mass: Numerical investigation. International Journal of Structural Stability and Dynamics, 20, 10 2020.

[21] Michael J. Zyphur and Frederick L. Oswald. Bayesian estimation and inference: A user's guide. Journal of management, 41(2):390–420, 2015.

[22] Dongming Feng and Maria Q. Feng. Computer vision for shm of civil infrastructure: From dynamic response measurement to damage detection – a review. Engineering structures, 156:105–117, 2018.

[23] Babak Moaveni, Xianfei He, Joel P. Conte, and Jose I. Restrepo. Damage identification study of a seven-story full-scale building slice tested on the ucsd-nees shake table. Structural Safety, 32(5):347–356, 2010. Probabilistic Methods for Modeling, Simulation and Optimization of Engineering Structures under Uncertainty in honor of Jim Beck's 60th Birthday.

[24] Rodrigo Astroza, Hamed Ebrahimian, Joel P. Conte, José I. Restrepo, and Tara C. Hutchinson. System identification of a full-scale five-story reinforced concrete building tested on the nees-ucsd shake table. Structural Control and Health Monitoring, 23:535–559, 3 2016.

[25] Xin Gao, Gengxin Duan, and Chunguang Lan. Bayesian updates for an extreme value distribution model of bridge traffic load effect based on shm data. Sustainability, 13(15), 2021.

[26] Manuel A Vega, Zhen Hu, and Michael D Todd. Optimal maintenance decisions for deteriorating quoin blocks in miter gates subject to uncertainty in the condition rating protocol. Reliability Engineering & System Safety, 204:107147, 2020.

[27] Yichao Yang, Mayank Chadha, Zhen Hu, and Michael D Todd. An optimal sensor design framework accounting for sensor reliability over the structural life cycle. Mechanical Systems and Signal Processing, 202:110673, 2023.

[28] Suorong Yang, Weikang Xiao, Mengchen Zhang, Suhan Guo, Jian Zhao, and Furao Shen. Image data augmentation for deep learning: A survey. 4 2022.

[29] Brian Kenji Iwana and Seiichi Uchida. An empirical survey of data augmentation for time series classification with neural networks. PLoS ONE, 16, 7 2021.

[30] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. A comprehensive survey on transfer learning. Proceedings of the IEEE, 109(1):43–76, 2021.

[31] Xufeng Huang, Tingli Xie, Zhuo Wang, Lei Chen, Qi Zhou, and Zhen Hu. A transfer learning-based multi-fidelity point-cloud neural network approach for melt pool modeling in additive manufacturing. ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part B: Mechanical Engineering, 8(1):011104, 2022.

[32] Adam Thelen, Xiaoge Zhang, Olga Fink, Yan Lu, Sayan Ghosh, Byeng D Youn, Michael D Todd, Sankaran Mahadevan, Chao Hu, and Zhen Hu. A comprehensive review of digital twin—part 2: roles of uncertainty quantification and optimization, a battery digital twin, and perspectives. Structural and multidisciplinary optimization, 66(1):1, 2023.

[33] Chen Jiang, Manuel A Vega, Mukesh K Ramancha, Michael D Todd, Joel P Conte, Matthew Parno, and Zhen Hu. Bayesian calibration of multi-level model with unobservable distributed response and application to miter gates. Mechanical Systems and Signal Processing, 170:108852, 2022.

[34] Adam Thelen, Xiaoge Zhang, Olga Fink, Yan Lu, Sayan Ghosh, Byeng D Youn, Michael D Todd, Sankaran Mahadevan, Chao Hu, and Zhen Hu. A comprehensive review of digital twin—part 1: modeling and twinning enabling technologies. Structural and Multidisciplinary Optimization, 65(12):354, 2022.

[35] Marcos Orchard, Gregory Kacprzynski, Kai Goebel, Bhaskar Saha, and George Vachtsevanos. Advances in Uncertainty Representation and Management for Particle Filtering Applied to Prognostics. Springer Netherlands, Dordrecht, 2009.

[36] Jice Zeng, Michael D Todd, and Zhen Hu. A recursive inference method based on invertible neural network for multi-level model updating using video monitoring data. Mechanical Systems and Signal Processing, 203:110736, 2023.

[37] Zihan Wu, Travis B Fillmore, Manuel A Vega, Zhen Hu, and Michael D Todd. Diagnostics and prognostics of multi-mode failure scenarios in miter gates using multiple data sources and a dynamic bayesian network. Structural and Multidisciplinary Optimization, 65(9):270, 2022.

[38] Yichao Yang, Mayank Chadha, Zhen Hu, and Michael D Todd. An optimal sensor placement design framework for structural health monitoring using bayes risk. Mechanical Systems and Signal Processing, 168:108618, 2022.

[39] Ankan Dash, Junyi Ye, and Guiling Wang. A review of generative adversarial networks (gans) and its applications in a wide variety of disciplines - from medical to remote sensing. CoRR, abs/2110.01442, 2021.

[40] Veit Sandfort, Ke Yan, Perry J. Pickhardt, and Ronald M. Summers. Data augmentation using generative adversarial networks (cyclegan) to improve generalizability in ct segmentation tasks. Scientific Reports, 9:16884, 2019.

[41] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014.

[42] Hyoun-Joong Kong, Jin Youp Kim, Hye-Min Moon, Hae Chan Park, Jeong-Whun Kim, Ruth Lim, Jonghye Woo, Georges El Fakhri, Dae Woo Kim, and Sungwan Kim. Automation of generative adversarial network-based synthetic data-augmentation for maximizing the diagnostic performance with paranasal imaging. Scientific Reports, 12, 2022.

[43] T Zhang and Y Wang. Deep learning algorithms for structural condition identification with limited monitoring data. In International Conference on Smart Infrastructure and Construction 2019 (ICSIC) Driving data-informed decision-making, pages 421–426. ICE Publishing, 2019.

[44] Mateusz Heesch, Krzysztof Mendrok, and Ziemowit Dworakowski. Time-domain signal synthesis with style-based generative adversarial networks applied to guided waves. In Artificial Intelligence and Soft Computing: 20th International Conference, ICAISC 2021, Virtual Event, June 21–23, 2021, Proceedings, Part I 20, pages 78–88. Springer, 2021.

[45] Furkan Luleci, F Necati Catbas, and Onur Avci. Generative adversarial networks for labeled acceleration data augmentation for structural damage detection. Journal of Civil Structural Health Monitoring, 13(1):181–198, 2023.

[46] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In Proceedings of the IEEE international conference on computer vision, pages 2223–2232, 2017.

[47] Furkan Luleci, F Necati Catbas, and Onur Avci. A literature review: Generative adversarial networks for civil structural health monitoring. Frontiers in Built Environment, 8:1027379, 2022.

[48] Naftali Cohen, Srijan Sood, Zhen Zeng, Tucker Balch, and Manuela Veloso. Visual forecasting of time series with image-to-image regression. CoRR, abs/2011.09052, 2020.

[49] O. Ronneberger, P.Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In Medical Image Computing and Computer-Assisted Intervention (MICCAI), volume 9351 of LNCS, pages 234–241. Springer, 2015.

[50] Serkan Kiranyaz, Junaid Malik, Habib Ben Abdallah, Turker Ince, Alexandros Iosifidis, and Moncef Gabbouj. Self-organized operational neural networks with generative neurons, 2020.

[51] Dominique Brunet, Edward R Vrscay, and Zhou Wang. On the mathematical properties of the structural similarity index. IEEE Transactions on Image Processing, 21(4):1488–1499, 2011.

[52] Min Jin Chong and David Forsyth. Effectively unbiased fid and inception score and where to find them. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 6070–6079, 2020.

[53] Dan Ao, Zhen Hu, and Sankaran Mahadevan. Dynamics model validation using time-domain metrics. Journal of Verification, Validation and Uncertainty Quantification, 2(1):011004, 2017.

[54] Brian A Eick, Zachary R Treece, Billie F Spencer Jr, Matthew D Smith, Steven C Sweeney, Quincy G Alexander, and Stuart D Foltz. Automated damage detection in miter gates of navigation locks. Structural Control and Health Monitoring, 25(1):e2053, 2018.

[55] Zhonglai Wang, Hong-Zhong Huang, and Xiaoping Du. Optimal design accounting for reliability, maintenance, and warranty. Journal of Mechanical Design, 132(1):011007–011015, 2010.

[56] Manuel A. Vega, Zhen Hu, Travis B. Fillmore, Matthew D. Smith, and Michael D. Todd. A novel framework for integration of abstracted inspection data and structural health monitoring for damage prognosis of miter gates. Reliability Engineering & System Safety, 211:107561, 2021.

[57] Allen C. Estes, Dan M. Frangopol, and Stuart D. Foltz. Updating reliability of steel miter gates on locks and dams using visual inspection results. Engineering Structures, 26(3):319–333, 2004.

[58] Xinyang Liu and Pingfeng Wang. Valuation of continuous monitoring systems for engineering system design in recurrent maintenance decision scenarios. Journal of Mechanical Design, 144(9):091702, 2022.

[59] Mayank Chadha, Zhen Hu, Charles R Farrar, and Michael D Todd. Risk and value informed structural health monitoring system design for the miter gates. eJNDT, 29(7), July 2024.

[60] Venkat Nemani, Adam Thelen, Chao Hu, and Steve Daining. Degradation-aware ensemble of diverse predictors for remaining useful life prediction. Journal of Mechanical Design, 145(3):031706, 2023.

[61] Farhad Ansari Vistasp M. Karbhari. Structural Health Monitoring of Civil Infrastructure Systems. Woodhead Publishing Series in Civil and Structural Engineering, 2009.

[62] Nathaniel Levine, Thomas Golecki, Fernando Gomez, Brian Eick, and Billie F Spencer. Bayesian model updating of concrete-embedded miter gate anchorages and implications for design. Structural and Multidisciplinary Optimization, 66(3):60, 2023.

[63] Guofeng Qian, Zihan Wu, Zhen Hu, and Michael D Todd. Pitting corrosion diagnostics and prognostics for miter gates using multiscale simulation and image inspection data. Structural Health Monitoring, page 14759217241264291, 2024.

[64] Guofeng Qian, Jice Zeng, Zhen Hu, and Michael Todd. Bayesian Model Updating of Multiscale Simulations Informing Corrosion Prognostics Using Conditional Invertible Neural Networks. ASCE-ASME J Risk and Uncert in Engrg Sys Part B Mech Engrg, pages 1–33, 07 2024.

[65] Lawrence A Bull, Paul A Gardner, Julian Gosliga, Timothy J Rogers, Nikolaos Dervilis, Elizabeth J Cross, Evangelos Papatheou, AE Maguire, Carles Campos, and Keith Worden. Foundations of population-based shm, part i: Homogeneous populations and forms. Mechanical systems and signal processing, 148:107141, 2021.

[66] J Gosliga, PA Gardner, LA Bull, N Dervilis, and K Worden. Foundations of population-based shm, part ii: Heterogeneous populations–graphs, networks, and communities. Mechanical Systems and Signal Processing, 148:107144, 2021.

[67] P Gardner, LA Bull, J Gosliga, N Dervilis, and K Worden. Foundations of population-based shm, part iii: Heterogeneous populations–mapping and transfer. Mechanical Systems and Signal Processing, 149:107142, 2021.

[68] Eamon Whalen and Caitlin Mueller. Toward reusable surrogate models: Graph-based transfer learning on trusses. Journal of Mechanical Design, 144(2):021704, 2022.

[69] Hemanth Venkateswara and Sethuraman Panchanathan. Introduction to Domain Adaptation, pages 3–21. Springer International Publishing, Cham, 2020.

[70] Pang-jo Chun and Toshiya Kikuta. Self-training with bayesian neural networks and spatial priors for unsupervised domain adaptation in crack segmentation. Computer-Aided Civil and Infrastructure Engineering, 39(17):2642–2661, 2024.

[71] Zuoyi Chen, Chao Wang, Jun Wu, Chao Deng, and Yuanhang Wang. Deep convolutional transfer learning-based structural damage detection with domain adaptation. Applied Intelligence, 53(5):5085–5099, Mar 2023.

[72] Cheng-Geng Huang, Jun Zhu, Yu Han, and Weiwen Peng. A novel bayesian deep dual network with unsupervised domain adaptation for transfer fault prognosis across different machines. IEEE Sensors Journal, 22(8):7855–7867, 2022.

[73] Jialin Li and David He. A bayesian optimization adabn-dcnn method with self-optimized structure and hyperparameters for domain adaptation remaining useful life prediction. IEEE Access, 8:41482–41501, 2020.

[74] Myeongsun Kwak and Jongsoo Lee. Diagnosis-based domain-adaptive design using designable data augmentation and bayesian transfer learning: Target design estimation and validation. Applied Soft Computing, 143:110459, 2023.

[75] Zhimin Xi. Model-based reliability analysis with both model uncertainty and parameter uncertainty. Journal of Mechanical Design, 141(5):051404, 2019.

[76] Zhen Hu, Zissimos P Mourelatos, David Gorsich, Paramsothy Jayakumar, and Monica Majcher. Testing design optimization for uncertainty reduction in generating off-road mobility map using a bayesian approach. Journal of Mechanical Design, 142(2):021402, 2020.

[77] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks, 2016.

[78] George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. Journal of Machine Learning Research, 22(57):1–64, 2021.

[79] Ivan Kobyzev, Simon JD Prince, and Marcus A Brubaker. Normalizing flows: An introduction and review of current methods. IEEE transactions on pattern analysis and machine intelligence, 43(11):3964–3979, 2020.

[80] Mohammad Noori, Fuh-Gwo Yuan, and Ehsan Noroozinejad Farsangi. Data-Centric Structural Health Monitoring: Mechanical, Aerospace and Complex Infrastructure Systems. Walter de Gruyter GmbH & Co KG, 2023.

[81] Lynton Ardizzone, Jakob Kruse, Sebastian Wirkert, Daniel Rahner, Eric W Pellegrini, Ralf S Klessen, Lena Maier-Hein, Carsten Rother, and Ullrich Köthe. Analyzing inverse problems with invertible neural networks. arXiv preprint arXiv:1808.04730, 2018.

[82] Stefan T Radev, Ulf K Mertens, Andreas Voss, Lynton Ardizzone, and Ullrich Köthe. Bayesflow: Learning complex stochastic models with invertible neural networks. IEEE transactions on neural networks and learning systems, 33(4):1452–1466, 2020.

[83] Lynton Ardizzone, Carsten Lüth, Jakob Kruse, Carsten Rother, and Ullrich Köthe. Guided image generation with conditional invertible neural networks. arXiv preprint arXiv:1907.02392, 2019.

[84] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. arXiv preprint arXiv:1605.08803, 2016.

[85] Lynton Ardizzone, Jakob Kruse, Carsten Lüth, Niels Bracher, Carsten Rother, and Ullrich Köthe. Conditional invertible neural networks for diverse image-to-image translation. In Pattern Recognition: 42nd DAGM German Conference, DAGM GCPR 2020, Tübingen, Germany, September 28–October 1, 2020, Proceedings 42, pages 373–387. Springer, 2021.

[86] Zhen Hu, Chao Hu, Zissimos P Mourelatos, and Sankaran Mahadevan. Model discrepancy quantification in simulation-based design of dynamical systems. Journal of Mechanical Design, 141(1):011401, 2019.

[87] Zhen Hu, Chao Hu, and Weifei Hu. A tutorial on digital twins for predictive maintenance. Structural Health Monitoring/Management (SHM) in Aerospace Structures, pages 453–501, 2024.

[88] Zhen Jiang, Wei Li, Daniel W Apley, and Wei Chen. A spatial-random-process based multidisciplinary system uncertainty propagation approach with model uncertainty. Journal of Mechanical Design, 137(10):101402, 2015.

[89] Chen Jiang, Zhen Hu, Yixuan Liu, Zissimos P Mourelatos, David Gorsich, and Paramsothy Jayakumar. A sequential calibration and validation framework for model uncertainty quantification and reduction. Computer Methods in Applied Mechanics and Engineering, 368:113172, 2020.

[90] Daniel W Apley, Jun Liu, and Wei Chen. Understanding the effects of model uncertainty in robust design with computer experiments. Journal of Mechanical Design, 128(4):945–958, 2006.

[91] Stefan T Radev, Marvin Schmitt, Valentin Pratz, Umberto Picchini, Ullrich Köthe, and Paul-Christian Bürkner. Jana: Jointly amortized neural approximation of complex bayesian models. In Uncertainty in Artificial Intelligence, pages 1695–1706. PMLR, 2023.

[92] Stefan T Radev, Marvin Schmitt, Lukas Schumacher, Lasse Elsemüller, Valentin Pratz, Yannik Schälte, Ullrich Köthe, and Paul-Christian Bürkner. Bayesflow: Amortized bayesian workflows with neural networks. arXiv preprint arXiv:2306.16015, 2023.

[93] Wen Zhang, Lingfei Deng, Lei Zhang, and Dongrui Wu. A survey on negative transfer. IEEE/CAA Journal of Automatica Sinica, 10(2):305–329, February 2023.

[94] Venkat Nemani, Luca Biggio, Xun Huan, Zhen Hu, Olga Fink, Anh Tran, Yan Wang, Xiaoge Zhang, and Chao Hu. Uncertainty quantification in machine learning for engineering design and health prognostics: A tutorial. Mechanical Systems and Signal Processing, 205:110796, 2023.

[95] Dan Lu, Goutam Konapala, Scott L. Painter, Shih-Chieh Kao, and Sudershan Gangrade. Streamflow simulation in data-scarce basins using bayesian and physics-informed machine learning models. Journal of Hydrometeorology, 22(6):1421 – 1438, 2021.

[96] Ying Zhao, Mayank Chadha, Dakota Barthlow, Elissa Yeates, Charles J Mcknight, Natalie P Memarsadeghi, Guga Gugaratshan, Michael D Todd, and Zhen Hu. Physics-enhanced machine learning models for streamflow discharge forecasting. Journal of Hydroinformatics, 26(10):2506–2537, 2024.

[97] F.H.S. Chiew, S.L. Zhou, and T.A. McMahon. Use of seasonal streamflow forecasts in water resources management. Journal of Hydrology, 270(1):135–144, 2003.

[98] Cédric H. David, David R. Maidment, Guo-Yue Niu, Zong-Liang Yang, Florence Habets, and Victor Eijkhout. River network routing on the nhdplus dataset. Journal of Hydrometeorology, 12(5):913 – 934, 2011.

[99] Cédric H. David, Florence Habets, David R. Maidment, and Zong-Liang Yang. Rapid applied to the sim-france model. Hydrological Processes, 25(22):3412–3425, 2011.

[100] Gerald T. McCarthy. The Unit Hydrograph and Flood Routing. Army Engineer District, Providence, 1939.

[101] Ying Zhao, Mayank Chadha, Nicholas Olsen, Elissa Yeates, Josh Turner, Guga Gugaratshan, Guofeng Qian, Michael D Todd, and Zhen Hu. Machine learning-enabled calibration of river routing model parameters. Journal of Hydroinformatics, 25(5):1799–1821, 2023.

[102] Chloe Qin, Yichao Zeng, Ying Zhao, Guga Gugaratshan, and Zhen Hu. Recalibration of neural networks using transfer learning for streamflow forecasting. In International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, volume 88360, page V03AT03A038. American Society of Mechanical Engineers, 2024.

[103] Tianfang Xu and Feng Liang. Machine learning for hydrologic sciences: An introductory overview. WIREs Water, 8(5):e1533, 2021.

[104] Hana Salimi Saeed Samadianfard Esmaeil Asadi Sadra Shadkani Katayoun Kargar Amir Mosavi Narjes Nabipour Shahabbodin Shamshirband, Sajjad Hashemi and Kwok-Wing Chau. Predicting standardized streamflow index for hydrological drought using machine learning models. Engineering Applications of Computational Fluid Mechanics, 14(1):339–350, 2020.

[105] Salah Difi, Salim Heddam, Bilel Zerouali, Sungwon Kim, Yamina Elmeddahi, Nadjem Bailek, Celso Augusto Guimarães Santos, and Habib Abida. Improved daily streamflow forecasting for semi-arid environments using hybrid machine learning and multi-scale analysis techniques. J. Hydroinformatics, 26(12):3266–3286, December 2024.

[106] Michele Magni, Edwin H. Sutanudjaja, Youchen Shen, and Derek Karssenberg. Global streamflow modelling using process-informed machine learning. Journal of Hydroinformatics, 25(5):1648–1666, 08 2023.

[107] Lingling Ni, Dong Wang, Vijay P. Singh, Jianfeng Wu, Yuankun Wang, Yuwei Tao, and Jianyun Zhang. Streamflow and rainfall forecasting by two long short-term memory-based models. Journal of Hydrology, 583:124296, 2020.

[108] Sudan Pokharel and Tirthankar Roy. A parsimonious setup for streamflow forecasting using cnn-lstm. Journal of Hydroinformatics, 26(11):2751–2761, 10 2024.

[109] Woon Yang Tan, Sai Hin Lai, Kumar Pavitra, Fang Yenn Teo, and Ahmed El-Shafie. Deep learning model on rates of change for multi-step ahead streamflow forecasting. Journal of Hydroinformatics, 25(5):1667–1689, 08 2023.

[110] Zain Syed, Prince Mahmood, Sajjad Haider, Shakil Ahmad, Khan Zaib Jadoon, Rashid Farooq, Sibtain Syed, and Khalil Ahmad. Short–long-term streamflow forecasting using a coupled wavelet transform–artificial neural network (wt–ann) model at the gilgit river basin, pakistan. Journal of Hydroinformatics, 25(3):881–894, 03 2023.

[111] Maryam Rahimzad, Alireza Moghaddam Nia, Hosam Zolfonoon, Jaber Soltani, Ali Danandeh Mehr, and Hyun-Han Kwon. Performance comparison of an LSTM-based deep learning model versus conventional machine learning algorithms for streamflow forecasting. Water Resour. Manage., 35(12):4167–4187, September 2021.

[112] F. Kratzert, D. Klotz, C. Brenner, K. Schulz, and M. Herrnegger. Rainfall–runoff modelling using long short-term memory (lstm) networks. Hydrology and Earth System Sciences, 22(11):6005–6022, 2018.

[113] Dapeng Feng, Kuai Fang, and Chaopeng Shen. Enhancing streamflow forecast and extracting insights using long-short term memory networks with data integration at continental scales. Water Resources Research, 56(9):e2019WR026793, 2020. e2019WR026793 2019WR026793.

[114] Grey Nearing, Deborah Cohen, Vusumuzi Dube, Martin Gauch, Oren Gilon, Shaun Harrigan, Avinatan Hassidim, Daniel Klotz, Frederik Kratzert, Asher Metzger, Sella Nevo, Florian Pappenberger, Christel Prudhomme, Guy Shalev, Shlomo Shenzis, Tadele Yednkachw Tekalign, Dana Weitzner, and Yossi Matias. Global prediction of extreme floods in ungauged watersheds. Nature, 627(8004):559–563, March 2024.

[115] Christiaan Schutte, Michael van der Laan, and Barend van der Merwe. Leveraging historic streamflow and weather data with deep learning for enhanced streamflow predictions. Journal of Hydroinformatics, 26(4):835–852, 02 2024.

[116] Newsha K. Ajami, Qingyun Duan, and Soroosh Sorooshian. An integrated hydrologic bayesian multimodel combination framework: Confronting input, parameter, and model structural uncertainty in hydrologic prediction. Water Resources Research, 43(1), 2007.

[117] Abhinav Gupta and Rao S. Govindaraju. Uncertainty quantification in watershed hydrology: Which method to use? Journal of Hydrology, 616:128749, 2023.

[118] K S Kasiviswanathan and K P Sudheer. Quantification of the predictive uncertainty of artificial neural network based river flow forecast models. Stoch. Environ. Res. Risk Assess., 27(1):137–146, January 2013.

[119] Tongtiegang Zhao, Q.J. Wang, James C. Bennett, David E. Robertson, Quanxi Shao, and Jianshi Zhao. Quantifying predictive uncertainty of streamflow forecasts based on a bayesian joint probability model. Journal of Hydrology, 528:329–340, 2015.

[120] Martyn P Clark, Robert L Wilby, Ethan D Gutmann, Julie A Vano, Subhrendu Gangopadhyay, Andrew W Wood, Hayley J Fowler, Christel Prudhomme, Jeffrey R Arnold, and Levi D Brekke. Characterizing uncertainty of the hydrologic impacts of climate change. Curr. Clim. Change Rep., 2(2):55–64, June 2016.

[121] F. Kratzert, D. Klotz, G. Shalev, G. Klambauer, S. Hochreiter, and G. Nearing. Towards learning universal, regional, and local hydrological behaviors via machine learning applied to large-sample datasets. Hydrology and Earth System Sciences, 23(12):5089–5110, 2019.

[122] Siyan Liu, Dan Lu, Scott L. Painter, Natalie A. Griffiths, and Eric M. Pierce. Uncertainty quantification of machine learning models to improve streamflow prediction under changing climate and environmental conditions. Frontiers in Water, 5, 2023.

[123] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In Maria Florina Balcan and Kilian Q. Weinberger, editors, Proceedings of The 33rd International Conference on Machine Learning, volume 48 of Proceedings of Machine Learning Research, pages 1050–1059, New York, New York, USA, 20–22 Jun 2016. PMLR.

[124] Grey S. Nearing, Frederik Kratzert, Alden Keefe Sampson, Craig S. Pelissier, Daniel Klotz, Jonathan M. Frame, Cristina Prieto, and Hoshin V. Gupta. What role does hydrological science play in the age of machine learning? Water Resources Research, 57(3):e2020WR028091, 2021.

[125] Dayang Li, Lucy Marshall, Zhongmin Liang, Ashish Sharma, and Yan Zhou. Bayesian lstm with stochastic variational inference for estimating model uncertainty in process-based hydrological models. Water Resources Research, 57(9):e2021WR029772, 2021.

[126] D. Klotz, F. Kratzert, M. Gauch, A. Keefe Sampson, J. Brandstetter, G. Klambauer, S. Hochreiter, and G. Nearing. Uncertainty estimation with deep learning for rainfall–runoff modeling. Hydrology and Earth System Sciences, 26(6):1673–1693, 2022.

[127] Ali Sattari, Ehsan Foroumandi, Keyhan Gavahi, and Hamid Moradkhani. A probabilistic machine learning framework for daily extreme events forecasting. Expert Systems with Applications, 265:126004, 2025.

[128] Yongen Lin, Dagang Wang, Guiling Wang, Jianxiu Qiu, Kaihao Long, Yi Du, Hehai Xie, Zhongwang Wei, Wei Shangguan, and Yongjiu Dai. A hybrid deep learning algorithm and its application to streamflow prediction. Journal of Hydrology, 601:126636, 2021.

[129] Huseyin Cagan Kilinc and Bulent Haznedar. A hybrid model for streamflow forecasting in the basin of euphrates. Water, 14(1), 2022.

[130] Ming Zhong, Hongrui Zhang, Tao Jiang, Jun Guo, Jinxin Zhu, Dagang Wang, and Xiaohong Chen. A hybrid model combining the cama-flood model and deep learning methods for streamflow prediction. Water Resour. Manage., 37(12):4841–4859, September 2023.

[131] J. A. Cunge. On the subject of a flood propagation computation method (muskingum method). Journal of Hydraulic Research, 7:205–230, 1969.

[132] Jundong Li, Kewei Cheng, Suhang Wang, Fred Morstatter, Robert P. Trevino, Jiliang Tang, and Huan Liu. Feature selection: A data perspective. ACM Comput. Surv., 50(6), December 2017.

[133] Guo-Yue Niu, Zong-Liang Yang, Kenneth E. Mitchell, Fei Chen, Michael B. Ek, Michael Barlage, Anil Kumar, Kevin Manning, Dev Niyogi, Enrique Rosero, Mukul Tewari, and Youlong Xia. The community noah land surface model with multiparameterization options (noah-mp): 1. model description and evaluation with local-scale measurements. Journal of Geophysical Research: Atmospheres, 116(D12), 2011.

[134] Zhen Hu, Dan Ao, and Sankaran Mahadevan. Calibration experimental design considering field response and model uncertainty. Computer Methods in Applied Mechanics and Engineering, 318:92–119, 2017.

[135] Yongjie Wang, Tong Zhang, Xu Guo, and Zhiqi Shen. Gradient based feature attribution in explainable ai: A technical review. arXiv preprint arXiv:2403.10415, 2024.

[136] Chenzhao Li and Sankaran Mahadevan. An efficient modularized sample-based method to estimate the first-order sobol index. Reliability Engineering System Safety, 153:110–121, 2016.

[137] Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. Neural Computation, 9(8):1735–1780, 11 1997.

[138] Jeremiah Liu, Zi Lin, Shreyas Padhy, Dustin Tran, Tania Bedrax Weiss, and Balaji Lakshminarayanan. Simple and principled uncertainty estimation with deterministic deep learning via distance awareness. Advances in neural information processing systems, 33:7498–7512, 2020.

[139] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770–778, 2016.

[140] Peter L Bartlett, Steven N Evans, and Philip M Long. Representing smooth functions as compositions of near-identity functions with implications for deep network optimization. arXiv preprint arXiv:1804.05012, 2018.

[141] Bianca Zadrozny and Charles Elkan. Transforming classifier scores into accurate multiclass probability estimates. In Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '02, page 694–699, New York, NY, USA, 2002. Association for Computing Machinery.

[142] Abbas Khosravi, Saeid Nahavandi, Doug Creighton, and Amir F. Atiya. Comprehensive review of neural network-based prediction intervals and new advances. IEEE Transactions on Neural Networks, 22(9):1341–1356, 2011.

[143] Benjamin Renard, Dmitri Kavetski, George Kuczera, Mark Thyer, and Stewart W. Franks. Understanding predictive uncertainty in hydrologic modeling: The challenge of identifying input and structural errors. Water Resources Research, 46(5), 2010.

[144] Magdalena Asborno, Jacob Broders, Kenneth N Mitchell, Michael A Hartman, and Lauren D Dunkin. Forecasting sediment accumulation in the southwest pass with machine-learning models. Journal of Waterway, Port, Coastal, and Ocean Engineering, 150(2):04023022, 2024.

[145] Elissa M Yeates, Ahmad A Tavakoly, Kenneth Ned Mitchell, Gregory W Dreaper, and Shahab Afshari. Utilizing stream flows to forecast dredging requirements. 2020.

[146] Vanesa Mateo-Pérez, Marina Corral-Bobadilla, Francisco Ortega-Fernández, and Vicente Rodríguez-Montequín. Determination of water depth in ports using satellite data based on machine learning algorithms. Energies, 14(9):2486, 2021.

[147] Haitham Abdulmohsin Afan, Ahmed El-shafie, Wan Hanna Melini Wan Mohtar, and Zaher Mundher Yaseen. Past, present and prospect of an artificial intelligence (ai) based model for sediment transport prediction. Journal of Hydrology, 541:902–913, 2016.

[148] Biswanath Bhattacharya and Dimitri P Solomatine. Machine learning in sedimentation modelling. Neural Networks, 19(2):208–214, 2006.

[149] Jui-Sheng Chou and Ji-Wei Lin. Risk-informed prediction of dredging project duration using stochastic machine learning. Water, 12(6):1643, 2020.

[150] Magdalena Asborno, Jacob Broders, Kenneth Ned Mitchell, Michael A Hartman, and Lauren M Dunkin. Selection of hydrological variables to forecast shoaling with xgb. Available at SSRN 4641922, 2023.

[151] Sharad Kumar Jain. Development of integrated sediment rating curves using anns. Journal of hydraulic engineering, 127(1):30–37, 2001.

[152] HM Nagy, KAND Watanabe, and M Hirano. Prediction of sediment load concentration in rivers using artificial neural network model. Journal of Hydraulic Engineering, 128(6):588–595, 2002.

[153] Frank Engelund, Hansen Anker, et al. Monograph on sediment transport in alluvial streams. (No Title), 1967.

[154] Hikmet Kerem Cigizoglu and Murat Alp. Generalized regression neural network in modelling river sediment yield. Advances in Engineering Software, 37(2):63–68, 2006.

[155] Murat Alp and H Kerem Cigizoglu. Suspended sediment load simulation by two artificial neural network methods using hydrometeorological data. Environmental Modelling & Software, 22(1):2–13, 2007.

[156] Vinayakam Jothiprakash and Vaibhav Garg. Reservoir sedimentation estimation using artificial neural network. Journal of Hydrologic Engineering, 14(9):1035–1040, 2009.

[157] AM Melesse, Sajjad Ahmad, ME McClain, X Wang, and YH Lim. Suspended sediment load prediction of river systems: An artificial neural network approach. Agricultural Water Management, 98(5):855–866, 2011.

[158] Özgür Kisi. Constructing neural network sediment estimation models using a data-driven algorithm. Mathematics and Computers in Simulation, 79(1):94–103, 2008.

[159] J-SR Jang. Anfis: adaptive-network-based fuzzy inference system. IEEE transactions on systems, man, and cybernetics, 23(3):665–685, 1993.

[160] Ozgur Kisi. Suspended sediment estimation using neuro-fuzzy and neural network approaches/estimation des matières en suspension par des approches neurofloues et à base de réseau de neurones. Hydrological sciences journal, 50(4), 2005.

[161] AK Lohani, Narendra Kumar Goel, and KK S Bhatia. Deriving stage–discharge–sediment concentration relationships using fuzzy logic. Hydrological Sciences Journal, 52(4):793–807, 2007.

[162] Mahmut Firat, Mustafa Erkan Turan, and Mehmet Ali Yurdusev. Comparative analysis of fuzzy inference systems for water consumption time series prediction. Journal of hydrology, 374(3-4):235–241, 2009.

[163] Mahmut Firat and Mahmud Güngör. Monthly total sediment forecasting using adaptive neuro fuzzy inference system. Stochastic Environmental Research and Risk Assessment, 24:259–270, 2010.

[164] Ozgur Kisi. Modeling discharge-suspended sediment relationship using least square support vector machine. Journal of hydrology, 456:110–120, 2012.

[165] Aytac Guven and Özgür Kişi. Estimation of suspended sediment yield in natural rivers using machine-coded linear genetic programming. Water resources management, 25:691–704, 2011.

[166] Ozgur Kisi and Jalal Shiri. River suspended sediment estimation by climatic variables implication: Comparative study among soft computing techniques. Computers & Geosciences, 43:73–82, 2012.

[167] Turgay Partal and H Kerem Cigizoglu. Estimation and forecasting of daily suspended sediment data using wavelet–neural networks. Journal of hydrology, 358(3-4):317–331, 2008.

[168] Vahid Nourani, Aida Hosseini Baghanam, Jan Adamowski, and Ozgur Kisi. Applications of hybrid wavelet–artificial intelligence models in hydrology: a review. Journal of Hydrology, 514:358–377, 2014.

[169] Taher Rajaee. Wavelet and neuro-fuzzy conjunction approach for suspended sediment prediction. CLEAN–Soil, Air, Water, 38(3):275–286, 2010.

[170] JH Graw, WT Wood, and BJ Phrampus. Predicting global marine sediment density using the random forest regressor machine learning algorithm. Journal of Geophysical Research: Solid Earth, 126(1):e2020JB020135, 2021.

[171] Lauren M Dunkin, Christina L Saltus, K Ned Mitchell, and Brandan M Scully. Improved channel evaluation using automatic identification system and hydrographic survey data. Proc., Western Dredging Association and Texas A&M University Center for Dredging Studies "Dredging Summit and Expo, 2015.

[172] Paul Paris and Fazil Erdogan. A critical analysis of crack propagation laws. 1963.

[173] Mayank Chadha and Michael D Todd. A generalized approach for reconstructing the three-dimensional shape of slender structures including the effects of curvature, shear, torsion, and elongation. Journal of Applied Mechanics, 84(4):041003, 2017.

[174] Julie Dean Rosati and Nicholas C Kraus. Rapid methods for estimating navigation channel shoaling. In WEDA XXIX Technical Conference and 40th Texas A&M Dredging Seminar, Tempe, Arizona, 2009.

[175] Mdmasharul Kabir, Masood Jafari Kang, Xing Wu, and Maryam Hamidi. Study on u-turn behavior of vessels in narrow waterways based on ais data. Ocean Engineering, 246:110608, 2022.

[176] U.S. Army Corps of Engineers, Mobile District. eHydro, 2025. Accessed: 2025-06-26.

[177] Port Houston. The Houston Ship Channel – Maintaining a National Economic Engine. https://porthouston.com/the-houston-ship-channel-maintaining-a-national-economic-engine/, August 2024. Accessed: 2025-06-26.

[178] U.S. Army Corps of Engineers, Galveston District. Houston Ship Channel Improvement Project. https://www.swg.usace.army.mil/Projects/Houston-Ship-Channel-Improvement-Project/, 2025. Accessed: 2025-06-26.

[179] Port of Houston. Project Overview - Port Houston: Houston Ship Channel Expansion. https://www.expandthehoustonshipchannel.com/project-overview/, May 2025. Accessed: 2025-06-26.

[180] Steven K. Ayres, F. Pinkard, B. Kleiss, and Mississippi Valley Division. Southwest pass outlets bathymetry and flow distribution assessment. Mississippi river geomorphology and potamology report no. 5, U.S. Army Corps of Engineers, Mississippi Valley Division, 2014. https://www.mvd.usace.army.mil/Portals/52/docs/SandT/Morphology-Potamology/MRGand%20P (Accessed: 2025-06-26).

[181] U.S. Department of Commerce, National Oceanic and Atmospheric Administration, and National Ocean Service. US Coast Pilot 5 (2025), Gulf of Mexico, Puerto Rico, Virgin Islands. NOAA Coast Survey, 2025. https://nauticalcharts.noaa.gov/publications/coast-pilot/files/cp5/CPB5_WEB.pdf (Accessed: 2025-06-26).

[182] Michael A Hartman, Kenneth N Mitchell, Lauren M Dunkin, James Lewis, Benjamin Emery, Nicholas F Lenssen, and Ronald Copeland. Southwest pass sedimentation and dredging data analysis. Journal of Waterway, Port, Coastal, and Ocean Engineering, 148(2):05021017, 2022.

[183] M. Kress. Hydrosurvey data archive for federal navigation projects. https://data.gov/maritime/hydrosurvey-data-archive-for-federal-navigation-projects/index.html, 2021.

[184] USACE. National channel framework. https://geospatial-usace.opendata.arcgis.com/maps/9227967a 2022. Accessed: 2022-10-19.

[185] Jennifer L Irish and Thomas E White. Coastal engineering applications of high-resolution lidar bathymetry. Coastal engineering, 35(1-2):47–71, 1998.

[186] Mark Richard Byrnes, Jessica L Baker, Feng Li, et al. Quantifying potential measurement errors and uncertainties associated with bathymetric change analysis. 2002.

[187] U.S. Army Corps of Engineers. Corps Shoaling Analysis Tool (CSAT). https://cirp.usace.army.mil/products/csat.php, 2024.

[188] Rachel Bain, Michael Hartman, Anna Dominique Godfrey, Charlene Sylvester, and Kathryn Smith. Corps shoaling analysis tool (csat) user guide. https://cirp.usace.army.mil/products/files/CSAT/CSAT_UserGuide_Draft.pdf, January 2025. Draft version accessed on: 2025-06-26.

[189] L. Dunkin, L. Coe, and J. Ratcliff. Corps shoaling analysis tool: Predicting channel shoaling. Erdc/chl technical report no. erdc/chl tr-18-16, U.S. Army Engineer Research and Development Center, Coastal and Hydraulics Laboratory, 2018. https://apps.dtic.mil/sti/trecms/pdf/AD1081215.pdf (Accessed: 2025-06-26).

[190] Ganesh R Ghimire, Carly Hansen, Sudershan Gangrade, Shih-Chieh Kao, Peter E Thornton, and Debjani Singh. Insights from dayflow: A historical streamflow reanalysis dataset for the conterminous united states. Water Resources Research, 59(2):e2022WR032312, 2023.

[191] Charlotte FK Uphues, Graziela Miot da Silva, Arnold van Rooijen, and Patrick A Hesp. Modelling sediment dynamics in a high energy coastal embayment. Coastal Engineering, page 104924, 2025.

[192] Ali Pourzangbar and Maurizio Brocchini. A new process-based, wave-resolving, 2dh circulation model for the evolution of natural sand bars: The role of nearbed dynamics and suspended sediment transport. Coastal Engineering, 177:104192, 2022.

[193] N Dodd, AM Stoker, R Garnier, G Vittori, F De Los Santos, Maurizio Brocchini, Luciano Soldini, and MA Losada. Use of numerical models to study land-based sedimentation and subsequent nearshore morphological evolution. Coastal engineering, 55(7-8):601–621, 2008.

[194] Tara N Sainath, Oriol Vinyals, Andrew Senior, and Haşim Sak. Convolutional, long short-term memory, fully connected deep neural networks. In 2015 IEEE international conference on acoustics, speech and signal processing (ICASSP), pages 4580–4584. Ieee, 2015.

[195] Shaojie Bai. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. arXiv preprint arXiv:1803.01271, 2018.

[196] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In international conference on machine learning, pages 1050–1059. PMLR, 2016.

[197] USACE. Operations and dredging endangered species system. https://odess.usace.army.mil/, 2025. Accessed: 2025-08-27.

[198] Institute for Water Resources. Value to the nation: Navigation. https://www.iwr.usace.army.mil/Missions/Value-to-the-Nation/Navigation/tmp1-Navigation/, 2025. Accessed: 2025-08-27.

[199] Benjamin E Emery. Fair and reasonable: A conceptual insight into usace dredge estimating. Journal of Waterway, Port, Coastal, and Ocean Engineering, 150(5):05024001, 2024.

[200] R. L. Bain, K. A. McPherran, M. A. Hartman, A. M. Godfrey, and K. N. Mitchell. Improving the accuracy of navigation channel maintenance forecasts using dredge location data. In Proceedings of the Western Dredging Association Dredging Summit & Expo '24, Tampa, FL, USA, June 24–27 2024.

[201] AR Niles. ehydro navigation channel condition reporting. US Army Engineer Research and Development Center. Coastal and Hydraulics Laboratory, 2013.

[202] Lauren Dunkin, Lauren Coe, and Jay Ratcliff. Corps shoaling analysis tool : predicting channel shoaling. Technical report, November 2018.

[203] Charlene Sylvester, Michael Hartman, and Sean McGill. Integration of regional lidar to expand shoaling rate analytics beyond navigation channels. In Coastal Sediments 2023: The Proceedings of the Coastal Sediments 2023, pages 2749–2762. World Scientific, 2023.

[204] Mohammed Falah Allawi, Sadeq Oleiwi Sulaiman, Khamis Naba Sayl, Mohsen Sherif, and Ahmed El-Shafie. Suspended sediment load prediction modelling based on artificial intelligence methods: The tropical region as a case study. Heliyon, 9(8), 2023.

[205] Yiping Zhang, Xinyu Yao, Qiang Wu, Yongbin Huang, Zhixu Zhou, Jun Yang, and Xiaowei Liu. Turbidity prediction of lake-type raw water using random forest model based on meteorological data: A case study of tai lake, china. Journal of Environmental Management, 290:112657, 2021.

[206] Wenchong Tian, Zhenliang Liao, and Jin Zhang. An optimization of artificial neural network model for predicting chlorophyll dynamics. Ecological Modelling, 364:42–52, 2017.

[207] Wenxiang Ding, Caiyun Zhang, Shaoping Shang, and Xueding Li. Optimization of deep learning model for coastal chlorophyll a dynamic forecast. Ecological Modelling, 467:109913, 2022.

[208] Woon Yang Tan, Sai Hin Lai, Kumar Pavitra, Fang Yenn Teo, and Ahmed El-Shafie. Deep learning model on rates of change for multi-step ahead streamflow forecasting. Journal of Hydroinformatics, 25(5):1667–1689, 2023.

[209] Woon Yang Tan, Sai Hin Lai, Fang Yenn Teo, Danial Jahed Armaghani, Kumar Pavitra, and Ahmed El-Shafie. Three steps towards better forecasting for streamflow deep learning. Applied Sciences, 12(24):12567, 2022.

[210] Juan Wang, Jinquan Xu, and Dongxian Kong. Time-lagged response of streamflow in the lower yellow river to the water regulation by xiaolangdi reservoir: Implication for efficient water supply. Water, 16(1):78, 2024.

[211] Lee H MacDonald and Drew Coe. Influence of headwater streams on downstream reaches in forested areas. Forest Science, 53(2):148–168, 2007.

[212] Kai Ma, Daming He, Shiyin Liu, Xuan Ji, Yungang Li, and Huiru Jiang. Novel time-lag informed deep learning framework for enhanced streamflow prediction and flood early warning in large-scale catchments. Journal of Hydrology, 631:130841, 2024.

[213] Efrain Noa-Yarasca, Javier M Osorio Leyton, and Jay P Angerer. Extending multi-output methods for long-term aboveground biomass time series forecasting using convolutional neural networks. Machine Learning and Knowledge Extraction, 6(3):1633–1652, 2024.

[214] Souhaib Ben Taieb, Gianluca Bontempi, Amir F Atiya, and Antti Sorjamaa. A review and comparison of strategies for multi-step ahead time series forecasting based on the nn5 forecasting competition. Expert systems with applications, 39(8):7067–7083, 2012.

[215] Daizong Ding, Mi Zhang, Xudong Pan, Min Yang, and Xiangnan He. Modeling extreme events in time series prediction. In Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining, pages 1114–1122, 2019.

[216] Yuanyuan Man, Qinli Yang, Junming Shao, Guoqing Wang, Linlong Bai, and Yunhong Xue. Enhanced lstm model for daily runoff prediction in the upper huai river basin, china. Engineering, 24:229–238, 2023.

[217] L. Lin, Z. Demirbilek, and M. S. Islam. Numerical modeling of channel shoaling and scour in corpus christi ship channel, texas. In L. Lin, Z. Demirbilek, and M. S. Islam, editors, Proceedings of the 10th International Conference on Scour and Erosion. International Society for Soil Mechanics and Geotechnical Engineering, 2021.

[218] Stefan D Kalev and Gurpal S Toor. The composition of soils and sediments. In Green chemistry, pages 339–357. Elsevier, 2018.

[219] Yani Li, Zhihao Xu, Xiaoyun Zhan, and Tibin Zhang. Summary of experiments and influencing factors of sediment settling velocity in still water. Water, 16(7):938, 2024.

[220] Marine Jusiane Bastos da Silva, Iran Carlos Stalliviere Correa, Jose Antonio Scotti Fontoura, and Christian Garcia Serpa. Sedimentation processes in the navigation channel of patos lagoon estuary, southern brazil. Regional Studies in Marine Science, 61:102931, 2023.

[221] Stephen Michael Sheppard. A shoaling analysis of the houston ship channel. 1986.

[222] S Earle. Physical geology–2nd edition. victoria, bc: Bc campus, 2019.

[223] Filip Hjulström. Studies of the morphological activity of rivers as illustrated by the River Fyris. PhD thesis, The Geological institution of the University of Upsala, 1935.

[224] NOAA. Tides and currents. https://oceanservice.noaa.gov/navigation/tidesandcurrents/, 2025. National Ocean Service website.

[225] AD Short. Coastal processes and beaches. nature education knowledge, 3 (10), 2012.

[226] Ruby Vallarino Castillo, Vicente Negro Valdecantos, and Jose Maria del Campo. Understanding the impact of hydrodynamics on coastal erosion in latin america: a systematic review. Frontiers in Environmental Science, 11:1267402, 2023.

[227] HE De Swart and JTF Zimmerman. Morphodynamics of tidal inlet systems. Annual review of fluid mechanics, 41(1):203–229, 2009.

[228] MC Christie, KR Dyer, and P Turner. Sediment flux and bed level measurements from a macro tidal mudflat. Estuarine, Coastal and Shelf Science, 49(5):667–688, 1999.

[229] Andreas Brand, Jessica R Lacy, Kevin Hsu, Daniel Hoover, Steve Gladding, and Mark T Stacey. Wind-enhanced resuspension in the shallow waters of south san francisco bay: Mechanisms and potential implications for cohesive sediment transport. Journal of Geophysical Research: Oceans, 115(C11), 2010.

[230] XH Wang and N Pinardi. Modeling the dynamics of sediment transport and resuspension in the northern adriatic sea. Journal of Geophysical Research: Oceans, 107(C12):18–1, 2002.

[231] David H Schoellhamer. Sediment resuspension mechanisms in old tampa bay, florida. Estuarine, Coastal and Shelf Science, 40(6):603–620, 1995.

[232] Aimei Wang, Xiao Wu, Naishuang Bi, David K Ralston, Chenghao Wang, and Houjie Wang. Combined effects of waves and tides on bottom sediment resuspension in the southern yellow sea. Marine Geology, 452:106892, 2022.

[233] Jennifer Tate, Brittany Gunkel, Julie Rosati, Alejandro Sanchez, Naveen Ganesh, Thad Pratt, Eric Wood, and Rob Thomas. Houston-galveston navigation channel shoaling study. Technical report, 2014.

[234] Yuanyi Su, Yang Zhang, Huanyuan Wang, and Tingyu Zhang. Effects of vegetation spatial pattern on erosion and sediment particle sorting in the loess convex hillslope. Scientific Reports, 12(1):14187, 2022.

[235] R. Bain, M. Hartman, A. Godfrey, and K. McPherran. Csat-derived insights into the relationship between navigation channel deepening and future maintenance requirements. https://cirp.usace.army.mil/techtransfer/discussions/FY24/CIRP_TD_CSAT-derived-insights202405 2024.

[236] Zeki Demirbilek and Julie Rosati. Verification and validation of the coastal modeling system. report 1: Summary report. Technical report, 2011.

[237] Tetra Tech, Northwest Hydraulic Consultants, ESA Edolfson, and Shannon Wilson. Lower puyallup river flood protection investigation: Without project analysis. Technical report, Pierce County, June 2008.

[238] Sokchhay Heng and Tadashi Suetsugi. Using artificial neural network to estimate sediment load in ungauged catchments of the tonle sap river basin, cambodia. J. Water Resource Prot., 2013.

[239] Birendra Bharti, Ashish Pandey, SK Tripathi, and Dheeraj Kumar. Modelling of runoff and sediment yield using ann, ls-svr, reptree and m5 models. Hydrology Research, 48(6):1489–1507, 2017.

[240] Adem Bayram, Murat Kankal, Gökmen Tayfur, and Hızır Önsoy. Prediction of suspended sediment concentration from water quality variables. Neural Computing and Applications, 24(5):1079–1087, 2014.

[241] Haitham Abdulmohsin Afan, Ahmed El-Shafie, Zaher Mundher Yaseen, Mohammed Majeed Hameed, Wan Hanna Melini Wan Mohtar, and Aini Hussain. Ann based sediment prediction model utilizing different input scenarios. Water resources management, 29(4):1231–1245, 2015.

[242] Ajai Singh, Mohd Imtiyaz, RK Isaac, and DM Denis. Comparison of artificial neural network models for sediment yield prediction at single gauging station of watershed in eastern india. Journal of Hydrologic Engineering, 18(1):115–120, 2013.

[243] Bidhan Bhattacharya, RK Price, and DP Solomatine. Machine learning approach to modeling sediment transport. Journal of Hydraulic Engineering, 133(4):440–450, 2007.

[244] Shicheng Li, Qiancheng Xie, and James Yang. Daily suspended sediment forecast by an integrated dynamic neural network. Journal of Hydrology, 604:127258, 2022.

[245] DV Luu, TNC Doan, and ND Vo. Suspended sediment concentration prediction using long short-term memory (lstm) networks at the vu gia thu bon catchment, viet nam. In IOP Conference Series: Earth and Environmental Science, volume 1143, page 012010. IOP Publishing, 2023.

[246] Arvind Yadav, Devendra Joshi, Vinod Kumar, Hitesh Mohapatra, Celestine Iwendi, and Thippa Reddy Gadekallu. Capability and robustness of novel hybridized artificial intelligence technique for sediment yield modeling in godavari river, india. Water, 14(12):1917, 2022.

[247] Barend van Maanen, Giovanni Coco, Karin R Bryan, and BG Ruessink. The use of artificial neural networks to analyze and predict alongshore sediment transport. Nonlinear Processes in Geophysics, 17(5):395–404, 2010.

[248] Mohamed Abdellah Ezzaouini, Gil Mahé, Ilias Kacimi, Ali El Bilali, Abdelaziz Zerouali, and Ayoub Nafii. Predicting daily suspended sediment load using machine learning and narx hydro-climatic inputs in semi-arid environment. Water, 14(6):862, 2022.

[249] Yaseen A Hamaamin, A Pouyan Nejadhashemi, Zhen Zhang, Subhasis Giri, Umesh Adhikari, and Matthew R Herman. Evaluation of neuro-fuzzy and bayesian techniques

in estimating suspended sediment loads. Sustainable Water Resources Management, 5(2):639–654, 2019.

[250] Fatemeh Barzegari Banadkooki, Mohammad Ehteram, Ali Najah Ahmed, Fang Yenn Teo, Mahboube Ebrahimi, Chow Ming Fai, Yuk Feng Huang, and Ahmed El-Shafie. Suspended sediment load prediction using artificial neural network and ant lion optimization algorithm. Environmental Science and Pollution Research, 27(30):38094–38116, 2020.

[251] Adem Bayram, Murat Kankal, and Hizir Önsoy. Estimation of suspended sediment concentration from turbidity measurements using artificial neural networks. Environmental monitoring and assessment, 184(7):4355–4365, 2012.

[252] B Bhattacharya, S Sewagudde, T van Kessel, and DP Solomatine. A hybrid approach in combining numerical and data-driven models in modelling fine sediment transport. In Proceedings of the 34th World Congress of the International Association for Hydro-Environment Research and Engineering: 33rd Hydrology and Water Resources Symposium and 10th Conference on Hydraulics in Water Engineering, pages 4368–4375. Engineers Australia Barton, ACT, 2011.

[253] Mohammad Abdullah Abid Almubaidin, Sarmad Dashti Latif, Kalaiarasan Balan, Ali Najah Ahmed, and Ahmed El-Shafie. Enhancing sediment transport predictions through machine learning-based multi-scenario regression models. Results in Engineering, 20:101585, 2023.

[254] Ronald W Schafer. What is a savitzky-golay filter?[lecture notes]. IEEE Signal processing magazine, 28(4):111–117, 2011.

[255] Alejandro C. Frery. Interquartile Range, pages 664–666. Springer International Publishing, Cham, 2023.

[256] United States Geological Survey. Usgs current water data for the nation. https://waterdata.usgs.gov/nwis/rt, 2025.

[257] National Oceanic and Atmospheric Administration. Noaa tides currents. https://tidesandcurrents.noaa.gov/, 2025.

[258] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. Neural computation, 9(8):1735–1780, 1997.

[259] Alex Sherstinsky. Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network. Physica D: Nonlinear Phenomena, 404:132306, 2020.

[260] Ralf C Staudemeyer and Eric Rothstein Morris. Understanding lstm–a tutorial into long short-term memory recurrent neural networks. arXiv preprint arXiv:1909.09586, 2019.

[261] Bertrand Iooss and Paul Lemaître. A review on global sensitivity analysis methods. Uncertainty management in simulation-optimization of complex systems: algorithms and applications, pages 101–122, 2015.

[262] Ilya M Sobol. Global sensitivity indices for nonlinear mathematical models and their monte carlo estimates. Mathematics and computers in simulation, 55(1-3):271–280, 2001.

[263] Ivano Azzini and Rossana Rosati. Sobol'sensitivity indices- a machine learning approach using the dynamic adaptive variances estimator with given data. International Journal for Uncertainty Quantification, 15, 2025.

[264] Chenzhao Li and Sankaran Mahadevan. An efficient modularized sample-based method to estimate the first-order sobol index. Reliability Engineering & System Safety, 153:110–121, 2016.

[265] Zhen Hu and Sankaran Mahadevan. Probability models for data-driven global sensitivity analysis. Reliability Engineering & System Safety, 187:40–57, 2019.

[266] Manoj Kumar Jain and Debjyoti Das. Estimation of sediment yield and areas of soil erosion and deposition for watershed prioritization using gis and remote sensing. Water Resources Management, 24(10):2091–2112, 2010.

[267] David K Ralston and W Rockwell Geyer. Episodic and long-term sediment transport capacity in the hudson river estuary. Estuaries and Coasts, 32(6):1130–1151, 2009.

[268] Thomas Hoffmann. Sediment residence time and connectivity in non-equilibrium and transient geomorphic systems. Earth-Science Reviews, 150:609–627, 2015.

[269] Ellen E Wohl and Daniel A Cenderelli. Sediment deposition and transport patterns following a reservoir sediment release. Water Resources Research, 36(1):319–333, 2000.

[270] Zhi-wei Wang, Xiao-fan Lu, Wen-ming Zhang, Vasileios C Fragkoulis, Yu-feng Zhang, and Michael Beer. Deep learning-based prediction of wind-induced lateral displacement response of suspension bridge decks for structural health monitoring. Journal of Wind Engineering and Industrial Aerodynamics, 247:105679, 2024.

[271] U.S. Department of Commerce, National Oceanic and Atmospheric Administration, and National Ocean Service. US Coast Pilot 5 (2025), Gulf of Mexico, Puerto Rico, Virgin Islands. NOAA Coast Survey, 2025. https://nauticalcharts.noaa.gov/publications/coast-pilot/files/cp5/CPB5_WEB.pdf (Accessed: 2025-06-26).