

Inferencing with sparse spatio-temporal data in biological systems

by

Aabila Tharzeen

B. Tech., University of Kerala, 2016

M. Tech., Indian Institute of Technology (IIT) Gandhinagar, 2021

AN ABSTRACT OF A DISSERTATION

submitted in partial fulfillment of the
requirements for the degree

DOCTOR OF PHILOSOPHY

Mike Wieggers Department of Electrical and Computer Engineering
Carl R. Ice College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2025

Abstract

Spatio-temporal data analysis plays a crucial role in many scientific domains, including biological systems, earth sciences, autonomous vehicles, and many others, providing critical insights into how spatially coherent entities evolve over time. Particularly in biological systems, accurately understanding the underlying cause-effect relationships requires systematic exploitation of both spatial and temporal variations. One of the major challenges in biological systems is that available data can be limited due to high experimental costs, ethical considerations, or logistical constraints and can hinder accurate modeling of the underlying complex spatio-temporal phenomena. Classical statistical approaches, while interpretable, frequently struggle to model complex nonlinear dependencies, whereas purely data-driven machine learning (ML) methods risk overfitting and poor generalization when dealing with limited data. Thus, addressing challenges associated with sparse spatio-temporal data is crucial to reliably inferring meaningful insights from biological systems. This dissertation addresses challenges associated with sparse spatio-temporal inference by systematically integrating uncertainty quantification (UQ) into ML frameworks specifically tailored for biological systems.

Missing data is often handled by imputation based on available observations. However, the missingness itself can contain critical information. The imputation can introduce bias and information loss, while failing to effectively capture the underlying spatio-temporal relationships. To address this “imputation dilemma,” this dissertation proposes and theoretically analyzes a novel Informative Missing Indicator Method (IMIM) specifically designed for neural networks. IMIM helps decide when imputation should occur without introducing bias or loss of information in the data. Furthermore, a graph neural network combined with a recurrent neural network-based spatio-temporal imputation framework is developed to systematically capture spatio-temporal relationships, significantly enhancing predictive

capabilities in downstream tasks by effectively increasing the amount of informative data available. Additionally, learning ML models from limited data can also be challenging due to the high risk of overfitting and poor generalizability. To mitigate these issues, this dissertation introduces innovative methods to integrate prior knowledge that represents high-level abstractions of natural phenomena into ML frameworks either as observational bias or learning bias. This ensures that model predictions conform to known scientific principles. The framework also facilitates inference of uncertain parameters not directly observable from data using the uncertainty quantified on the model predictions.

In safety-critical biological applications, confidence in predictions is as crucial as prediction accuracy itself. Therefore, recognizing the critical importance of uncertainty quantification, this dissertation presents a generic, task-agnostic UQ framework utilizing neural stochastic differential equations (Neural SDEs). This framework analytically captures epistemic uncertainty in both traditional neural networks and graph neural networks, thereby enhancing model reliability and interoperability. Additionally, this dissertation proposes an uncertainty-guided active learning framework that analytically propagates spatio-temporal measurement uncertainty to strategically select the most informative samples. This approach effectively reduces overall prediction uncertainty, optimizing resource usage and improving predictive accuracy.

The methods developed in this dissertation are highly beneficial for healthcare and other data-scarce, safety-critical biological applications where reliability, accuracy, and informed decision-making are essential.

Inferencing with sparse spatio-temporal data in biological systems

by

Aabila Tharzeen

B. Tech., University of Kerala, 2016

M. Tech., Indian Institute of Technology (IIT) Gandhinagar, 2021

A DISSERTATION

submitted in partial fulfillment of the
requirements for the degree

DOCTOR OF PHILOSOPHY

Mike Wieggers Department of Electrical and Computer Engineering
Carl R. Ice College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2025

Approved by:

Major Professor
Dr. Balasubramaniam Natarajan

Copyright

© Aabila Tharzeen 2025.

Abstract

Spatio-temporal data analysis plays a crucial role in many scientific domains, including biological systems, earth sciences, autonomous vehicles, and many others, providing critical insights into how spatially coherent entities evolve over time. Particularly in biological systems, accurately understanding the underlying cause-effect relationships requires systematic exploitation of both spatial and temporal variations. One of the major challenges in biological systems is that available data can be limited due to high experimental costs, ethical considerations, or logistical constraints and can hinder accurate modeling of the underlying complex spatio-temporal phenomena. Classical statistical approaches, while interpretable, frequently struggle to model complex nonlinear dependencies, whereas purely data-driven machine learning (ML) methods risk overfitting and poor generalization when dealing with limited data. Thus, addressing challenges associated with sparse spatio-temporal data is crucial to reliably inferring meaningful insights from biological systems. This dissertation addresses challenges associated with sparse spatio-temporal inference by systematically integrating uncertainty quantification (UQ) into ML frameworks specifically tailored for biological systems.

Missing data is often handled by imputation based on available observations. However, the missingness itself can contain critical information. The imputation can introduce bias and information loss, while failing to effectively capture the underlying spatio-temporal relationships. To address this “imputation dilemma,” this dissertation proposes and theoretically analyzes a novel Informative Missing Indicator Method (IMIM) specifically designed for neural networks. IMIM helps decide when imputation should occur without introducing bias or loss of information in the data. Furthermore, a graph neural network combined with a recurrent neural network-based spatio-temporal imputation framework is developed to systematically capture spatio-temporal relationships, significantly enhancing predictive

capabilities in downstream tasks by effectively increasing the amount of informative data available. Additionally, learning ML models from limited data can also be challenging due to the high risk of overfitting and poor generalizability. To mitigate these issues, this dissertation introduces innovative methods to integrate prior knowledge that represents high-level abstractions of natural phenomena into ML frameworks either as observational bias or learning bias. This ensures that model predictions conform to known scientific principles. The framework also facilitates inference of uncertain parameters not directly observable from data using the uncertainty quantified on the model predictions.

In safety-critical biological applications, confidence in predictions is as crucial as prediction accuracy itself. Therefore, recognizing the critical importance of uncertainty quantification, this dissertation presents a generic, task-agnostic UQ framework utilizing neural stochastic differential equations (Neural SDEs). This framework analytically captures epistemic uncertainty in both traditional neural networks and graph neural networks, thereby enhancing model reliability and interoperability. Additionally, this dissertation proposes an uncertainty-guided active learning framework that analytically propagates spatio-temporal measurement uncertainty to strategically select the most informative samples. This approach effectively reduces overall prediction uncertainty, optimizing resource usage and improving predictive accuracy.

The methods developed in this dissertation are highly beneficial for healthcare and other data-scarce, safety-critical biological applications where reliability, accuracy, and informed decision-making are essential.

Contents

List of Figures	xii
List of Tables	xiv
List of Nomenclature	xvi
Acknowledgements	xix
Dedication	xxi
Preface	xxii
1 Introduction	1
1.1 Research Questions	5
1.2 Contributions	6
1.3 Organization of the Dissertation	9
2 Sparse data: Should we impute or not?	11
2.1 Related work	11
2.1.1 Contributions	14
2.2 Background	15
2.2.1 Missing data mechanisms	15
2.2.2 Missing indicator method	16
2.3 Proposed Informative Missing Indicator Method	17
2.3.1 Input features without noise	18
2.3.2 Input features with noise	19

2.4	Results	21
2.4.1	Synthetic datasets	22
2.4.2	Real-world datasets	26
2.5	Summary	27
3	Multivariate Spatio-temporal Imputation	30
3.1	Related work	30
3.1.1	Contributions	31
3.2	Problem Formulation	32
3.3	Proposed G-LSTM Framework	33
3.3.1	Spatial Module	33
3.3.2	Temporal Module	34
3.3.3	Complexity Analysis	36
3.4	Results	36
3.5	Summary	41
4	Uncertainty Quantification in Machine Learning Models	43
4.1	UQ in Deep Learning	43
4.1.1	Related work	43
4.1.2	Contributions	45
4.1.3	Proposed UQ-Net approach	45
4.1.4	Results	52
4.2	UQ in GNN	56
4.2.1	Related Work	56
4.2.2	Contribution	58
4.2.3	Proposed GNSDE-UQ approach	60
4.2.4	Results	65
4.3	Summary	67

5	Scientific Machine Learning with Uncertainty Quantification	68
5.1	Case Study 1: Bayesian Physics-Informed Machine Learning for Thermal Ab-	
	lation	69
5.1.1	Related Work	69
5.1.2	Contributions	71
5.1.3	Background	72
5.1.4	Problem Formulation	76
5.1.5	Proposed approach	76
5.1.6	Results	81
5.1.7	Discussion	89
5.2	Case Study 2: Graph-Based Machine Learning for Biomarker Identification .	90
5.2.1	Related Work	90
5.2.2	Contributions	90
5.2.3	Background	91
5.2.4	Experimental Set up	91
5.2.5	Proposed approach	92
5.2.6	Results	93
5.3	Summary	95
6	Uncertainty Guided Active Sampling of Spatio-Temporal Data	96
6.1	Related Work	96
6.1.1	Contributions	99
6.2	Background	100
6.2.1	Spatio-Temporal Graphs	100
6.2.2	Dilated Causal Convolutional Neural Networks	101
6.2.3	Graph Neural Networks	102
6.2.4	Active Learning	103
6.3	Problem Formulation	104

6.4	Proposed Framework	106
6.5	Results	109
6.6	Summary	113
7	Conclusions and Future Work	114
7.1	Conclusions	114
7.2	Future work	116
	Bibliography	118

List of Figures

1.1	Challenges Associated with Sparse Spatio-temporal Data Inference	5
2.1	Summary of the Informative Missing Indicator Method	17
2.2	Performance evaluation of IMIM on synthetic regression dataset	23
2.3	Performance evaluation of IMIM on synthetic classification dataset	24
2.4	Percentage of missing data imputed with change in informativeness parameter for synthetic dataset	25
2.5	Regression performance of IMIM on voting dataset	27
2.6	Classification performance of IMIM on wine dataset	28
2.7	Percentage of missing data imputed with change in informativeness parameter for real world dataset	29
3.1	Structure of proposed spatiotemporal imputation network (G-LSTM)	36
3.2	Sensor monitor distribution in PEMS7 traffic dataset	37
3.3	Training loss curve for G-LSTM	39
3.4	Speed values estimated for each node for a single time instant using G-LSTM framework	41
3.5	Speed values estimated for each node for a single time instant using CosTCO	42
4.1	Neural network with uncertainty quantification module (UQ-Net)	46
4.2	UQ-Net	47
4.3	Block diagram of GNSDE-UQ Framework	60
5.1	Architecture for Bayesian physics informed neural network	77

5.2	Heat map for true, prediction of temperature profile (T in $^{\circ}C$) at the colocation points (x, y in mm), and the error 10 sec outside of training window for simulation data	85
5.3	Heat map for true, prediction of temperature profile (T in $^{\circ}C$) at the colocation points (x, y in mm), and the error 20 sec outside of training window for simulation data	86
5.4	Heat map for true, prediction of temperature profile (T in $^{\circ}C$) at the colocation points (x, y in mm), and the error 30 sec outside of training window for simulation data	86
5.5	Heat map for true and prediction of temperature profile (T in $^{\circ}C$) at the colocation points (x, y in mm), 294.4 seconds after start of heating for MRI data	86
5.6	Heat map for true and prediction of temperature profile (T in $^{\circ}C$) at the colocation points (x, y in mm), 300.8 seconds after start of heating for MRI data	87
5.7	Heat map for true and prediction of temperature profile (T in $^{\circ}C$) at the colocation points (x, y in mm), 307.2 seconds after start of heating for MRI data	87
5.8	Variation of the estimated and actual electromagnetic energy absorbed from the microwave applicator along the x-axis for MRI data	88
5.9	The 1D plot along x direction for the BPINN predictions on MRI data along with one std deviation obtained using assumed density filtering	88
5.10	Potential Biomarker identified for Traumatic Brain Injury at 3 hours post injury	94
6.1	Proposed framework for uncertainty-guided AL on spatio-temporal graphs. The following result formalizes the propagation of uncertainty across DC-CNN and GNN layers.	107

List of Tables

3.1	Performance Evaluation for Missing Data Imputation using G-LSTM and SOTA methods	40
4.1	ENCE scores obtained using two methods for regression and classification task for synthetic dataset	54
4.2	ENCE scores obtained using two methods for regression and classification task for real-world dataset	55
4.3	Computation time in seconds required for regression and classification task for synthetic dataset	55
4.4	Computation time in seconds required for regression and classification task for real-world dataset	55
4.5	ENCE scores for GNSDE-UQ, BGNSDE and MC Dropout across varying dropout probabilities p for regression	67
4.6	Performance of proposed approach and comparison with MC dropout for classification	67
5.1	Prediction error across various time instances using different frameworks on simulation data	83
5.2	Values of the uncertain parameter obtained using BPINN for simulated data	83
5.3	Prediction error at different time points using various frameworks on MRI data, evaluated at 320.0 s, 326.4 s, and 332.8 s after the start of heating . . .	85
5.4	Values of the uncertain parameter obtained using BPINN for MRI data . . .	87
5.5	UQ for the biomarker identification using GNSDE-UQ and MC dropout framework	94

6.1	Percentage of node labels required to achieve 90% predictive performance. . .	111
6.2	Performance Evaluation at different stages of Active Learning.	111
6.3	Uncertainty Evaluation at different stages of Active Learning (considering 0.10 at input).	112
6.4	Longitudinal Studies for specific nodes: Prediction of TR based on VWR (uncertainty measured by considering 0.10 at input).	112

Nomenclature

KNN	K-Nearest Neighbor
SICE	Single Center Imputation from Multiple Chained Equations
GAN	Generative Adversarial Networks
MIM	Missing Indicator Method
MCAR	Missing Completely at Random
IMIM	Informative Missing Indicator Method
MAR	Missing at Random
MNAR	Missing Not at Random
CoSTCo	Convolutional Neural Network-based Tensor Completion
CNN	Convolutional Neural Networks
RNN	Recurrent Neural Networks
LSTM	Long Short-Term Memory
SVD	Singular Value Decomposition
GCN	Graph Convolutional Networks
DSAE	Denoising Stacked Autoencoder
GNN	Graph Neural Networks
PeMS	Performance Measurement System
GCRINT	Graph Convolutional Recurrent Neural Network
SOTA	State of the Art
RMSE	Root Mean Square Error
MAE	Mean Absolute Error

TR	Temperature Reading
VWR	Volumetric Water Reading
PP	PP
AL	Active Learning
MLP	Multi-Layer Perceptron
DC-CNN	Dilated Causal Convolutional Neural Network
MWA	Microwave Ablation
MRT	Magnetic Resonance Thermometry
MRI	Magnetic Resonance Imaging
PDE	Partial Differential Equation
ADF	Assumed Density Filtering
DNN	Deep Neural Networks
MCMC	Monte Carlo Markov Chain
ODE	Ordinary Differential Equation
SDE	Stochastic Differential Equation
OOD	Out-of-Distribution
UQ	Uncertainty Quantification
ENCE	Expected Normalized Calibration Error
RMV	Root Mean Variance
EV	Extracellular Vesicles
LFQ	Label-free quantification
PPI	Protein-Protein Interaction
hpi	hours post injury
dpi	days post injury

TBI	Traumatic Brain Injury
KL	Kullback–Leibler
GNSDE	Graph Neural Stochastic Differential Equation
AUROC	Area Under the Receiver Operating Characteristic Curve
AUPOC	Area Under the Precision-Recall Curve

Acknowledgments

First and foremost, I am deeply grateful to my advisor, Dr. Bala Natarajan, for his excellent guidance and unwavering support without which this work wouldn't have been possible. He always encouraged and inspired me to be a better version of myself both as a researcher and as a human being. He was always patient, kind and understanding and helped me find peace during all the hard times. I would also like to thank Dr. Punit Prakash for his valuable time and feedback during my PhD. He welcomed all my questions and was always a source of positive energy. My heartfelt thanks to my committee members, Dr. Doina Caragea and Dr. Pietro Poggi-Corradini, for their time, insights, and constructive feedback, which helped improve the quality of my research.

I'm grateful to my collaborators—Dr. Anna Bottiglieri, Dr. Jungkwun Kim, Dr. Rahul Sheth, Dr. Jun Yin Tan, Dr. Zifeng (Tony) Tang, and Dr. Bartosz Szczesny—for their valuable support in various aspects of my research.

I am forever thankful to my parents for the countless sacrifices they made and their endless love and affection. Their prayers and encouragement keep me going. To my brother Aarif—thank you for always reminding me that everything would turn out fine. My heartfelt gratitude to my husband, Ajmal, who, despite being a PhD student himself, was always there for me through thick and thin. His constant care and support mean the world to me. I am also thankful to my in-laws and extended family for checking in on me from time to time and their unconditional affection.

I would like to thank all my friends and colleagues from the CPSWin research group—Dr. Deepesh Agarwal, Dr. Rahul Madbhavi, Dr. Sai Munikoti, Dr. Amulya Sreejith, Dr. Dylan Wheeler, Dr. Mohammad Abujubbeh, Dr. James Ranjith, Dr. Shweta Dahale, Dr. Biswajeet Rout, Prudhviraaj Dhanapala, Priyanka Gautam, Rakhul Kumar, Adwaitha Pradeep, Reyhane Zare, and Tameema Rashid—for their support and encouragement. To my friends at K-State—Niranjana, Arya, Sowmya, Sriha, Prakalya, and Bulime—thank you

for the wonderful memories that helped me through this journey. A warm thank you also to my pet companions, Diamond and Jerry, for their comforting presence.

Last but not the least, I am thankful to God Almighty for giving me the strength and blessings to complete this work.

This material is based upon work supported by National Science Foundation (NSF) CNS 2039014

Dedication

To my loving family for their constant support and encouragement.

Preface

This dissertation, “Inferencing with sparse spatio-temporal data in biological systems,” is submitted for the degree of Doctor of Philosophy in the Mike Weigers Department of Electrical and Computer Engineering at Kansas State University. The research was conducted under the supervision of Professor Balasubramaniam Natarajan. This work is original to the best of my knowledge, except where acknowledgments and references are indicated. Most of the work has been presented in the following published peer-reviewed journals and conferences:

1. Tharzeen, A., Munikoti, S., Prakash, P., Kim, J., & Natarajan, B. (2023, June). A General Spatiotemporal Imputation Framework for Missing Sensor Data. In 2023 IEEE Conference on Artificial Intelligence (CAI) (pp. 55-58). IEEE.
2. Tharzeen, A., Dahale, S., Natarajan, B. (2024). Neural SDE-Based Epistemic Uncertainty Quantification in Deep Neural Networks. In: Iliadis, L., Maglogiannis, I., Papaleonidas, A., Pimenidis, E., Jayne, C. (eds) Engineering Applications of Neural Networks. EANN 2024. Communications in Computer and Information Science, vol 2141. Springer, Cham. https://doi.org/10.1007/978-3-031-62495-7_19
3. Tang, Z.T., DeJesus, J.E., Agarwal, D., Tharzeen, A., Zhang, Y., Patrikeev, I., Natarajan, B., Bucchieri, F., Zhao, Y., Micci, M.A., Bossmann, S.H., Motamedi, M., Szczesny, B. Circulating cell-free DNA and novel protein biomarkers within extracellular vesicles for the identification of traumatic brain injury severity. *Journal of Neurotrauma*, (Abstract, 2023).
4. Tang, T. Z., Zhao, Y., Agarwal, D., Tharzeen, A., Patrikeev, I., Zhang, Y., DeJesus, J., Bossmann, S. H., Natarajan, B., Motamedi, M. & Szczesny, B. (2024). Serum Amyloid A and Mitochondrial DNA in Extracellular Vesicles are Novel Markers for Detecting Traumatic Brain Injury in a Mouse Model. *Isience*.

5. Agarwal, D., Tharzeen, A., & Natarajan, B. (2024, July). Uncertainty-guided Active Learning for Measurement Prediction in Sparsely Observed Agricultural Fields. 16th International Conference on Precision Agriculture.

Chapter 1

Introduction

Spatio-temporal processes are fundamental to many scientific fields, including biology, social network traffic forecasting, and earth science [1]. Central to these processes is spatio-temporal data, which captures information distributed across both spatial dimensions (location) and temporal dimensions (time). Given that understanding causality is considered the “holy grail” of science, answering cause-effect relationships (“why” and “how”) requires “when” and “where” the data were collected [2]. This necessitates the analysis of spatio-temporal data, as interactions across space and time are often critical to explaining system behavior. For instance, the behavior at a spatial location at one time point can influence nearby locations at subsequent times. Thus, analyzing spatial snapshots or time-series profiles in isolation may not be appropriate; by integrating both dimensions can we uncover how and why spatially coherent entities evolve over time.

Several statistical approaches have been proposed to handle spatio-temporal data. Kriging models are widely used in spatio-temporal geostatistics to make predictions [3]. The interpolated values are estimated using an optimal linear weighted combination of neighbors. The optimal weights are obtained by considering and modeling a spatio-temporal covariance function, which describes the covariance structure present in the random field that is expected to generate the data [4]. However, when the time is taken into account, the number of observations multiplies and the underlying mathematical mechanism necessitates the in-

version of a square matrix whose size is determined by the number of observations. Thus, kriging may become infeasible [2]. Although assuming separability in the spatio-temporal covariance function can partially address this issue, traditional statistical approaches may still fall short in adequately characterizing complex, nonlinear spatio-temporal dependencies [4].

Recent advances in machine learning (ML) have revolutionized fields such as healthcare, computer vision, and autonomous systems. Their capability as universal function approximators, i.e., the ability to approximate complex non-linear functions, has shown remarkable success. Their popularity stems from their high statistical power in recognizing patterns in larger datasets [5]. While large datasets are becoming more common in some domains, traditional hypothesis-driven research in fields such as genomics, neuroimaging, and cancer treatment often has to rely on small sample sizes. This is because, in such applications, the associated costs can be high, and there are complexities associated with data collection in vivo [6]. This is particularly true in biological systems, where the complexity, ethical considerations and cost of experiments limit the number of samples available, resulting in sparse spatio-temporal data [7].

To effectively address the “why” and “how” questions within biological systems, it is essential to systematically exploit both spatial and temporal variations present in the data [8]. Statistical and machine learning-based approaches have been used in an attempt to uncover hidden relationships, identify critical events, and predict future states within biological systems [8]. However, limited data makes it difficult to accurately capture intricate spatio-temporal patterns, constraining our understanding of underlying biological mechanisms [6]. Both ML and statistical approaches have their own advantages and disadvantages.

For complex scenarios, as in the case of biological systems, one statistical approach is Berliner’s Bayesian Hierarchical Model (BHM). BHM provides a unified probabilistic framework that allows for uncertainty quantification (UQ) across data, models, and parameters. BHMs typically decompose the problem into three distinct levels [2]:

- **Data model:** Describes the distribution of observed data given the underlying process

and parameters.

- **Process model:** Captures the hidden process and its uncertainty, often incorporating domain knowledge.
- **Parameter model:** Accounts for uncertainty in the parameters governing the process.

The fundamental difference between parametric statistical and machine learning-based frameworks is that statistical modeling (data modeling) assumes a predefined stochastic data model (e.g., linear or logistic regression), with the goal of estimating parameters, validated through goodness-of-fit tests and residual checks. Conversely, machine learning (algorithmic modeling) treats the model as an unknown, complex system, emphasizing predictive accuracy through algorithms such as neural networks and decision trees, validated primarily by predictive performance [9]. Although interpretable, the disadvantage in using statistical methods over ML is that strong assumptions about how the system works are required, and if the assumptions are inaccurate, they can lead to questionable scientific conclusions [9].

Compared to classical statistical methods, ML frameworks for spatio-temporal data modeling can more easily incorporate additional spatially and temporally referenced features, as well as external variables influencing spatio-temporal processes. However, in scenarios involving sparse datasets, existing ML approaches may struggle with generalizability and risk overfitting, thereby hindering robust inference [10]. Existing ML-based approaches to address limited-data problems include few-shot learning [11], which leverages related datasets through meta-learning, transfer learning, or metric learning; semi-supervised learning [12], which predicts additional labels from a limited amount of labeled data; self-supervised learning [13], which extracts patterns directly from unlabeled data; and data augmentation, which fills data gaps by simulating real-world events and synthesizing additional data samples to create a sufficiently large training dataset. However, understanding the underlying mechanism causing the missingness is crucial. Particularly in biological datasets, the absence of data itself can carry valuable information. Imputing missing values solely based on observed data may lead to information loss when making inferences. In certain cases, imputing values can introduce bias and is often considered non-recoverable [14].

In applications and decision support systems such as healthcare, where biological data is extensively used, knowing the confidence in ML predictions is crucial. In such cases, the uncertainty quantification (UQ) can be extremely useful. As George Box stated, “All models are wrong, but some are useful,”. In healthcare applications, quantifying uncertainty can guide decisions regarding model reliability and suitability, ultimately influencing clinical outcomes. UQ allows researchers to critically evaluate the influence of data on the model and its parameters, thereby providing meaningful insights into the reliability of predictions. In this context, statistical approaches are advantageous over ML frameworks as uncertainty quantification is deeply rooted in statistical methodologies. The specific focus of this thesis is to develop machine learning-based frameworks that address the challenges associated with inferencing sparse spatio-temporal data from biological systems.

Broadly, there are three primary goals associated with the modeling and analysis of spatio-temporal data according to [15]:

- **Goal 1:** Prediction in space and time: Estimating values at unobserved locations and times.
- **Goal 2:** Inference on parameters: Understanding the underlying mechanisms and parameters governing the system.
- **Goal 3:** Forecasting in time: Predicting future states of the system.

A critical aspect accompanying these goals is the quantification of uncertainty, which ensures the reliability and interpretability of predictions specifically for biological systems, as we discussed earlier.

To summarize, the challenges associated with inferencing from sparse spatio-temporal data can be categorized into two groups, as illustrated in Figure 1.1. The first category is associated with the quantity of informative data. Due to high acquisition costs, the amount of available data is often limited, and the existing data can be noisy or might not always be representative of the system being studied. This can result in inaccurate predictions and increased uncertainty. The second category involves the limitations of learning frameworks

in effectively capturing complex relationships inherent in biological systems. This can lead to poor generalization and predictions with increased uncertainty, even if the available data are informative.

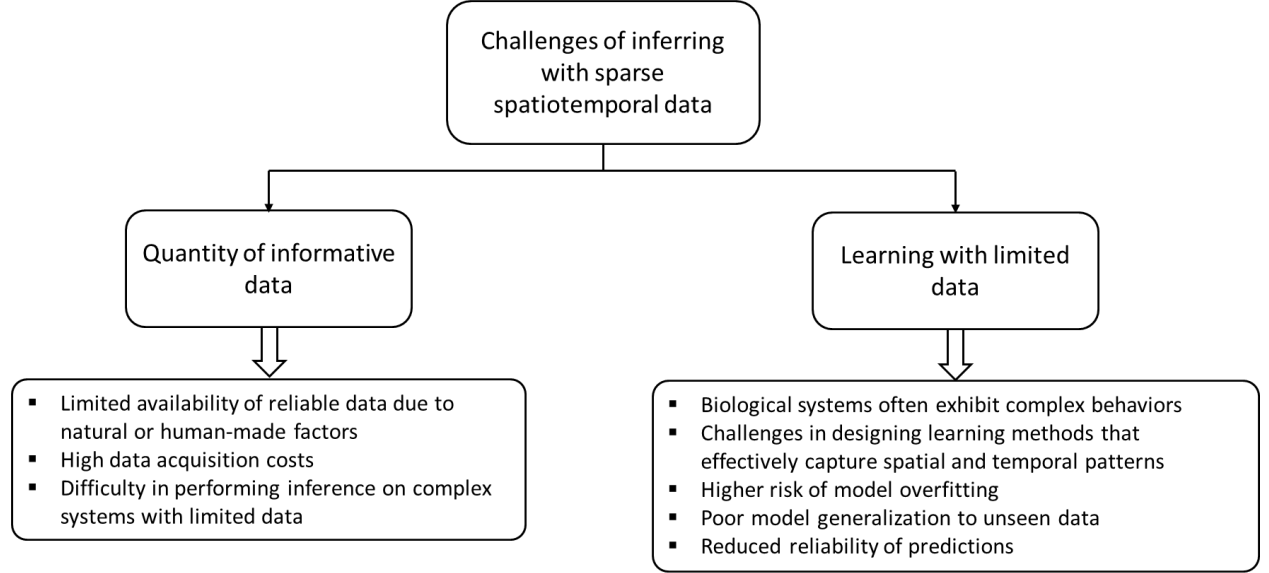


Figure 1.1: Challenges Associated with Sparse Spatio-temporal Data Inference

Based upon these perspectives, this dissertation seeks to address a few fundamental research questions. These research questions and contributions of the dissertation that aim to address them are discussed in the following sections.

1.1 Research Questions

In order to address the inferencing challenges, this thesis explores the modeling and inferencing of sparse spatio-temporal data by integrating simpler models with uncertainty quantified machine learning techniques. Specifically, we address the following key questions.

Question 1: *Should we always impute missing data? If so, how can we effectively capture underlying spatio-temporal relationships to achieve accurate predictions?*

Question 2: *How can we quantify uncertainties in machine learning models to improve their reliability?*

Question 3: *Can uncertainty quantification help in the inferencing process with limited*

data?

Question 4: *How can we enhance the quality of spatio-temporal data collection to reduce overall uncertainty?*

By addressing these questions, this thesis aims to develop robust machine learning-based frameworks for inferencing with sparse spatio-temporal data, particularly in biological systems. The proposed frameworks will not only improve prediction accuracy but also provide interpretable and reliable insights using the quantification of uncertainty, which is especially important in safety-critical domains like healthcare.

1.2 Contributions

We address Question 1 in Chapters 2 and 3 of the dissertation. As discussed earlier, one way to handle limited data is by imputing missing values using available observations. However, this approach can lead to information loss, as the missingness itself can be informative. In Chapter 2, we address the dilemma of whether or not to impute by proposing an Informative Missing Indicator Method (IMIM), specifically tailored for universal function approximators such as neural networks, accompanied by theoretical analyses considering both noisy and noiseless input features. IMIM is task-agnostic, applicable to both classification and regression, and can be combined with other imputation methods. Although various approaches exist for imputing missing data, they often fail to effectively capture underlying spatio-temporal relationships or the inherent trends in the data, resulting in inaccurate imputations and degraded performance on downstream tasks. In Chapter 3, we systematically capture the spatio-temporal relationships using a graph neural network-based model in conjunction with a recurrent neural network to impute missing values. We demonstrate the effectiveness of this approach on downstream tasks. Using the methodologies described in Chapters 2 and 3, we can determine when imputation is appropriate and employ effective spatio-temporal imputation methods accordingly. More details on the methodologies and experimental evaluations of these approaches can be found in the following articles:

- Tharzeen, A. & Natarajan, B. Imputation Dilemma: Addressing Missing Data in Neural Networks. (*Submitted to ACM Transactions on Knowledge Discovery from Data*)
- Tharzeen, A., Munikoti, S., Prakash, P., Kim, J., & Natarajan, B. (2023, June). A General Spatiotemporal Imputation Framework for Missing Sensor Data. In 2023 IEEE Conference on Artificial Intelligence (CAI) (pp. 55-58). IEEE.

To address Research Question 2, Chapter 4 proposes generic frameworks to quantify uncertainties associated with both deep neural networks as well as graph neural networks. Uncertainty quantification is essential for ensuring reliability and robustness, particularly in safety-critical applications. However, accurately quantifying model/epistemic uncertainty in machine learning-based regression and classification tasks is challenging. While Monte Carlo Dropout is a valuable tool for uncertainty quantification in deep learning, relying solely on it can be limiting, especially when dealing with complex models or situations where more principled approaches are needed. In Chapter 4, we propose a novel task-agnostic framework that can analytically quantify the epistemic uncertainty of deep neural networks and graph neural networks. Specifically, we use a surrogate stochastic differential equation-based framework, that can capture and measure epistemic uncertainty of target neural networks effectively. To extend this framework to graph neural networks, the surrogate stochastic differential equations are modified to encode graph structures. This improved understanding of uncertainty will contribute to improved decision-making and foster trust in the application of neural networks in various domains. Additional details on the developed framework for epistemic uncertainty quantification in deep neural networks and graph neural networks can be found in the following articles:

- Tharzeen, A., Dahale, S., Natarajan, B. (2024). Neural SDE-Based Epistemic Uncertainty Quantification in Deep Neural Networks. In: Iliadis, L., Maglogiannis, I., Papaleonidas, A., Pimenidis, E., Jayne, C. (eds) Engineering Applications of Neural Networks. EANN 2024. Communications in Computer and Information Science, vol 2141. Springer, Cham. https://doi.org/10.1007/978-3-031-62495-7_19

- Tharzeen, A., & Natarajan, B. Graph Neural SDE-Based Epistemic Uncertainty Quantification for Graph Neural Networks, (to be submitted to IEEE Transactions on Artificial Intelligence).

To address Research Question 4, we present two case studies in Chapter 5. We demonstrate how prior information can be integrated into machine learning-based frameworks to improve the accuracy of the predictions. In the first case study, we illustrate how a Bayesian physics-informed machine learning framework can predict temperature during thermal ablation, particularly when data availability is limited. Incorporating prior information, such as the governing computational equations that describe heat transfer in tissues during heating, enhances the generalizability of the machine-learning model. The proposed framework also includes an analytical method to quantify uncertainty in model predictions. We also demonstrate how the uncertainty can be leveraged to analytically estimate some uncertain parameter values that cannot be directly observed/measured (solving the inverse problem). In the second case study, we incorporate prior information represented as a graph derived from an existing database to identify potential biomarkers for traumatic brain injury. Additionally, uncertainty quantification is employed to assess confidence in the prediction, facilitating more efficient decision support. Further details regarding these case studies can be found in the following articles:

- Tang, Z.T., DeJesus, J.E., Agarwal, D., Tharzeen, A., Zhang, Y., Patrikeev, I., Natarajan, B., Bucchieri, F., Zhao, Y., Micci, M.A., Bossmann, S.H., Motamedi, M., Szczesny, B. Circulating cell-free DNA and novel protein biomarkers within extracellular vesicles for the identification of traumatic brain injury severity. *Journal of Neurotrauma*, (Abstract, 2023).
- Tang, T. Z., Zhao, Y., Agarwal, D., Tharzeen, A., Patrikeev, I., Zhang, Y., DeJesus, J., Bossmann, S. H., Natarajan, B., Motamedi, M. & Szczesny, B. (2024). Serum Amyloid A and Mitochondrial DNA in Extracellular Vesicles are Novel Markers for Detecting Traumatic Brain Injury in a Mouse Model. *Iscience*.

- Tharzeen, A., Bottiglieri, A., Tan, J., Prakash, P., Sheth, R., Kim, J. & Natarajan, B., Bayesian physicsinformed neural network for temperature profile prediction during thermal ablation, IEEE Journal on Multiscale and Multiphysics Computational Techniques, (under 2nd review)

Research question 4 is addressed in Chapter 6. As previously discussed, in many health-care applications, acquiring data, especially from sensors can become laborious or costly, resulting in limited data. While using such limited data for machine learning predictions, it is crucial that the data is informative, enabling accurate predictions with reduced uncertainty. In Chapter 6, we demonstrate how, while capturing spatial and temporal relationships in the data, we can analytically propagate measurement uncertainty to model predictions and subsequently use this uncertainty knowledge to guide the sampling process. Thus, the framework proposed in Chapter 6 helps guide sampling when there is a limited budget for data collection, leveraging uncertainty associated with machine learning predictions and thereby reducing the model’s overall uncertainty. Further details of this work can be found in the following article:

- Agarwal, D., Tharzeen, A., & Natarajan, B. (2024, July). Uncertainty-guided Active Learning for Measurement Prediction in Sparsely Observed Agricultural Fields. 16th International Conference on Precision Agriculture.

1.3 Organization of the Dissertation

The remainder of this dissertation is organized as follows. Chapter 2 presents a theoretical approach addressing the dilemma of whether or not to impute missing data in the context of deep learning methods. Chapter 3 introduces an enhanced imputation framework specifically tailored for spatio-temporal data. Chapter 4 proposes a novel, generic framework for epistemic uncertainty quantification in neural networks and graph neural networks. Chapter 5 demonstrates two case studies, through which we demonstrate how quantified uncertainty combined with prior information can enhance machine learning frameworks and

solve inverse problems (estimating uncertain parameters that are not directly measurable). Chapter 6 illustrates how measurement uncertainty can be leveraged to improve the quantity of informative data. Finally, Chapter 7 summarizes the dissertation and outlines potential directions for future research.

Chapter 2

Sparse data: Should we impute or not?

As discussed in Chapter 1, one common approach to handle sparse data is to fill in gaps using available observations, typically by capturing underlying relationships from existing data or leveraging data homogeneity. However, when the pattern of missingness itself carries information, relying solely on observed data for imputation can lead to information loss. To address this, we introduce the Informative Missing Indicator Method (IMIM), a strategy specifically tailored for supervised learning settings. IMIM helps determine whether imputation is necessary under both noisy and noiseless conditions. This chapter begins with a review of conventional imputation techniques and their limitations, then details the development of IMIM, followed by a theoretical analysis addressing the imputation dilemma.

2.1 Related work

Missing data is an unavoidable factor in analytics associated with many practical applications such as healthcare, business, agriculture and so on. Missing data can occur due to multiple reasons like errors made by human and machine, damage of equipment, refusal to answer questions by clients, etc. The quality of data is of utmost importance when it comes

to real world data analysis. The data with poor quality due to missing information can result in performance degradation and biased outcomes. Additionally, statistical analysis and machine learning methods work poorly on data with missing information. In such cases, data preprocessing is required.

In general, preprocessing data with missing information can be classified into two categories. One approach is to only work with samples that do not have missing information [16]. For example, in the case of tabular data collected from healthcare domains, where each row of the table would be a sample, the rows with missing observations are discarded completely from the dataset. The drawback with such an approach is that it can bias the results as the partially observed data samples can contain important information and dropping out samples can be catastrophic if all the samples contain missing information [17]. The second approach is to use an imputation method that requires utilizing the observed data to estimate the values at missing locations. However, considering the missing pattern before imputing is important as the missingness can be informative, i.e, the missing pattern may encode information about the response variable and imputing them can result in biased outcomes during data analysis.

The data can be missing due to identifiable or unidentifiable reasons. Imputing the values can add bias in the first scenario and hence can be distinguished as non-recoverable, and in the second case the data might be missing due to a random cause and can be recovered [14]. As explained in [18], the missing pattern is categorized into three mechanisms: (i) missing at random (MAR), (ii) missing not at random (MNAR), (iii) missing completely at random (MCAR) based on the relationship between the pattern of the available and missing data. This pattern also tells us if the missingness is informative or not. For example, if missingness is informative, MAR assumption does not hold. A motivating example for informative missingness in electronic health record data is the following: while building a predictive model for cardiovascular risk of patients who visit their general practitioner, we might not have cholesterol measurements in the health record. This is because it may not have been deemed necessary at earlier consultations since the cardiovascular risk was less for the patient. In this situation, the absence of cholesterol measurement is in fact informative

and can be a favorable prognosis with missing measurement than if it had been measured [19].

An alternative strategy to handle missing data is the Missing Indicator Method (MIM), which is used when missing values are informative. This method introduces an indicator variable for partially observed features, signifying whether a feature is missing. Though MIM is a common method used in scikit-learn [20] a python library, theoretical analysis of MIM for linear regressors and linear classifiers has only recently been conducted in [21].

Some existing methods address it through preprocessing steps using statistical inference[22] or deep learning [23] to fill gaps, while others focus on improving task performance with available data. The most widely used method falls into three main categories: (i) deletion methods, which delete records with missing values, (ii) single imputation methods, which substitute missing values with mean, mode, or linear interpolation, and (iii) model-based methods, such as regression, multiple imputation, or k-nearest neighbors (KNN). These methods are simple but can result in significant information loss if many observations are missing. There are also hybrid approaches such as Single Center Imputation from Multiple Chained Equations (SICE) [24].

Machine learning approaches such as decision trees, KNN, clustering, and random forests, have also been used for missing data imputation. Unsupervised learning techniques like clustering groups similar objects to aid imputation [25], while KNN uses distance metrics to find neighbors for imputing values [26]. In machine learning settings, in one case, the complete data represents the input features, and the missing values represent the predictions. Decision trees can test attributes, where the outcome is denoted by the branch of the tree [27]. Multiple decision trees are constructed based on bootstrapping, and the average of the final predictions is used in random forests [28]. Other imputation models utilize either or both spatial and temporal gradient information for imputation [23],[29] [30]. [31] employs a GAN-based imputation framework that recursively averages the input to reconstruct missing values in industrial time series data. Task-agnostic approaches that can be used with any imputation method are less explored in the literature. The missing indicator method (MIM) that was proposed initially for missing confounder data in etiologic research has received

much attention in medical literature.

The missing indicator method adds a dummy variable (1/0) to the covariate, with missing locations filled with zero, indicating whether a value is missing [19]. This in turn helps to include each observation even in the presence of missing data and improves the statistical power. Clinical datasets often contain missing values that tend to be informative. [32] shows how choices about handling missing data under different scenarios can be guided by the expectations about the missing data mechanism. Due to the ubiquity of the missing data in tabular clinical datasets, MIM is widely explored in the medical literature [19]. While MIM is used in practice in Python libraries such as sklearn [20], its statistical properties are understudied in literature. Although MIM has been employed and studied theoretically in supervised learning using linear regression, multilayer perceptron and random forests, it remains limited to linear setups[21]. Furthermore, existing analyses of MIM do not account for noise in the input features, which might reduce the reliability of the indicator parameter.

2.1.1 Contributions

In this chapter, we propose, for the first time, an Informative Missing Indicator Method (IMIM), suitable for deep learning frameworks with missing input data. IMIM identifies locations to impute by preserving the informativeness of the missingness pattern, even with noisy input data. The main contributions of this chapter can be summarized as follows:

- We introduce an Informative Missing Indicator Method (IMIM), tailored for universal function approximators such as neural networks. This approach helps identify informative missingness and determines the missing locations when the missingness is uninformative. We also provide a theoretical analysis of IMIM within a deep learning framework.
- IMIM is task-agnostic and can be used for both classification and regression. It can also be combined with any other imputation methods. Extensive empirical studies on synthetic and real-world data across different missing percentages show a 10% increase

in performance compared to imputation frameworks without an informative missing indicator, highlighting its superiority.

- We address noisy input data scenarios with a Bayesian approach to identify informative missing patterns and use significance tests to select reliable informative indicators, resulting in an 8% performance increase compared to the baseline approach.

2.2 Background

Consider a supervised learning setup with complete input feature matrix $\mathbf{X} \in \mathbb{R}^{n,d}$, where there are n samples and d features, an incomplete input feature matrix $\mathbf{Z} \in \mathbb{R}^{n,d}$ and a response variable $\mathbf{Y} \in \mathbb{R}^n$ which can be continuous or discrete based on the task to be performed. We also denote a missingness indicator $\mathbf{M} \in \mathbb{R}^{n,d}$ that indicates if the entries in $\mathbf{Z}$ are missing or not. Specifically, if $\mathbf{X}_{i,j}$ is missing, then $\mathbf{M}_{i,j} = 1$ and $\mathbf{Z}_{i,j} = 0$; if $\mathbf{X}_{i,j}$ is observed, then $\mathbf{M}_{i,j} = 0$ and $\mathbf{Z}_{i,j} = \mathbf{X}_{i,j}$. Given a loss function $\mathcal{L} = (\mathbf{Y} - \mathbf{f}(\mathbf{Z}))^2$, the goal is to find the function $\mathbf{f}^*$, characterized by a neural network with parameter vector $\boldsymbol{\theta}$ that minimizes the expected loss, i.e., $\mathbf{f}^* = \operatorname{argmin}_{\mathbf{f}} \mathbf{E}[\mathcal{L}(\mathbf{f}(\mathbf{Z}, \boldsymbol{\theta}), \mathbf{Y})]$. Neural networks are known as universal function approximators, capable of approximating any continuous function with high accuracy under suitable conditions. Therefore, they can effectively capture the functional relationship between input features and the response variable.

2.2.1 Missing data mechanisms

As explained in [18], missing data follows patterns based on underlying assumptions:

- **Missing completely at random (MCAR):** The missing pattern is independent of both observed and unobserved data, i.e., $P(\mathbf{M}|\mathbf{X}) = P(\mathbf{M})$.
- **Missing at random (MAR):** The missing pattern depends only on the observed data, i.e., $P(\mathbf{M}|\mathbf{X}) = P(\mathbf{M}|\mathbf{X}_{\text{obs}})$. This is common in healthcare data.

- **Missing not at random (MNAR):** The missing pattern depends on both missing and observed values, i.e., $P(\mathbf{M}|\mathbf{X}_{\text{obs}}, \mathbf{X}_{\text{miss}})$.

We provides theoretical guarantees to determine whether imputation is necessary, such as when missingness is uninformative (in the case of MCAR, where the missing pattern is independent of the data) or informative (as in the case of MAR, where the pattern depends on the observations).

2.2.2 Missing indicator method

As discussed previously, missing values and the patterns of missingness in datasets can be informative when making inferences, and this is prevalent in real-world datasets. The Missing Indicator Method (MIM) is an effective strategy that incorporates indicator variables to denote the missing pattern, and it is already used in Python libraries. However, from the perspective of supervised learning using neural network, it has not been explored. Existing works utilize MIM as part of a supervised learning framework for optimal prediction accuracy but are limited to simple linear regressor, multilayer perceptrons and linear classifier models and decision trees [21]. The MIM optimizes prediction accuracy in a supervised setting by estimating the best linear predictors, i.e.,

$$\boldsymbol{\beta}_{MIM}^*, \boldsymbol{\gamma}_{MIM}^* = \underset{\boldsymbol{\beta}, \boldsymbol{\gamma}}{\operatorname{argmin}} \mathbf{E} \left[\left(\mathbf{Y} - \mathbf{Z}\boldsymbol{\beta} - \mathbf{M}\boldsymbol{\gamma} \right)^2 \right] \quad (2.1)$$

Here $\boldsymbol{\beta}, \boldsymbol{\gamma}$ are the standard ordinary least square estimates. In this work, for the first time, we develop the theory of MIM specifically for use with deep neural networks. We also consider cases where the features are noisy and provide statistical tests to determine whether the missingness is informative under uncertainties.

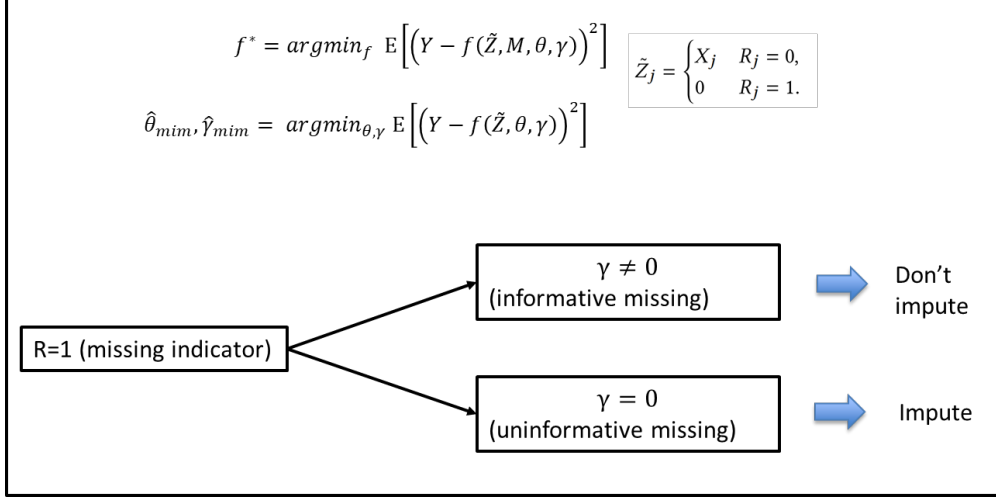


Figure 2.1: Summary of the Informative Missing Indicator Method

2.3 Proposed Informative Missing Indicator Method

In this section, we introduce the informative missing indicator method (IMIM) that optimizes prediction accuracy for deep learning architectures in the presence of incomplete input data. Given an incomplete input $\mathbf{Z}$, response $\mathbf{Y}$ and missing indicator $\mathbf{M}$, our goal is to find the nonlinear function $\mathbf{f}$ parameterized by the weights $\boldsymbol{\theta}$ of the neural network. We also estimate γ which determines if the missing pattern or the missing indicator $\mathbf{M}$ is informative. Thus the loss minimization of a neural network can be modified as

$$\mathbf{f}^* = \underset{\mathbf{f}}{\operatorname{argmin}} \mathbb{E}[\mathcal{L}(\mathbf{f}(\mathbf{Z}, \mathbf{M}, \boldsymbol{\theta}, \gamma), \mathbf{Y})] \quad (2.2)$$

We use the value of the informative indicator γ to decide if the missing indicator $\mathbf{M}$ is informative and to decide whether or not to impute the missing locations and the summary of this process is shown in Figure 2.1. In the real world, the available data can be noisy. Therefore, in this work, we consider two cases: (i) noiseless input features and (ii) noisy input features.

2.3.1 Input features without noise

In this case, the input features $\mathbf{Z}$ do not contain any noise. We optimize the neural network as per equation (4.6) to obtain the weights $\boldsymbol{\theta}$ and the informative indicator γ . The value of the informative indicator helps determine if and where imputation should be performed. The following paragraph describes the missing indicator method for a deep learning architecture.

Let $\mathbf{Y}$ be the output of the neural network layer with nonlinear transformation $\mathbf{g}(\cdot)$ of the intermediate activations, network layer parameter $\boldsymbol{\theta}$, and the informative indicator γ . Assume a single-layer neural network:

$$\mathbf{Y} = \mathbf{g}(\underbrace{\mathbf{Z}\boldsymbol{\theta} + \mathbf{M}\gamma}_{\mathbf{h}}) \quad (2.3)$$

The input to the neural network consists of $[\mathbf{Z} \quad \mathbf{M}]$ with trainable parameters $[\boldsymbol{\theta} \quad \gamma]$. The nonlinear function learned by the neural network can be linearized using Taylor series [33] at points $(\hat{\boldsymbol{\theta}}, \hat{\gamma})$ as shown in equation (2.4)

$$\begin{aligned} \mathbf{Y} &= \mathbf{g}(h) + \frac{\partial \mathbf{g}}{\partial \boldsymbol{\theta}} \bigg|_{\boldsymbol{\theta} = \hat{\boldsymbol{\theta}}} + \frac{\partial \mathbf{g}}{\partial \gamma} \bigg|_{\gamma = \hat{\gamma}} \\ &= \underbrace{\mathbf{g}(h)(1 - \mathbf{g}(h)) \odot \mathbf{Z}\hat{\boldsymbol{\theta}}}_{\mathbf{A}} + \underbrace{\mathbf{g}(h)(1 - \mathbf{g}(h)) \odot \mathbf{M}\hat{\gamma}}_{\mathbf{B}} + \mathbf{C} \end{aligned} \quad (2.4)$$

where, $\mathbf{C} = \mathbf{g}(h) - \mathbf{g}(h)(1 - \mathbf{g}(h)) \odot \mathbf{Z}\hat{\boldsymbol{\theta}} - \mathbf{g}(h)(1 - \mathbf{g}(h)) \odot \mathbf{M}\hat{\gamma}$ is a constant value since it is evaluated at a point. Equation (2.4) can be simplified as follows:

$$\mathbf{Y} = \begin{bmatrix} \mathbf{A} & \mathbf{B} & \mathbf{C} \end{bmatrix} \begin{bmatrix} \boldsymbol{\theta} \\ \gamma \\ 1 \end{bmatrix} \quad (2.5)$$

$$\text{Let } \mathbf{D} = \begin{bmatrix} \mathbf{A}^T & \mathbf{B}^T \end{bmatrix}^T$$

$$\begin{aligned}
\begin{bmatrix} \boldsymbol{\theta}_{MIM}^* \\ \boldsymbol{\gamma}_{MIM}^* \end{bmatrix} &= \mathbf{E}[\mathbf{D}\mathbf{D}^T]^{-1} \mathbf{E}[\mathbf{D}\mathbf{Y}] \\
&= \begin{bmatrix} \mathbf{E}[\mathbf{A}^2]^{-1} & \mathbf{E}[\mathbf{A}\mathbf{B}^T] \\ \mathbf{E}[\mathbf{B}\mathbf{A}^T] & \mathbf{E}[\mathbf{B}^2]^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{E}[\mathbf{A}\mathbf{Y}] \\ \mathbf{E}[\mathbf{B}\mathbf{Y}] \end{bmatrix} \\
&= \begin{bmatrix} \boldsymbol{\theta}^* \\ \mathbf{E}[\mathbf{B}^2]^{-1} \mathbf{E}[\mathbf{B}\mathbf{Y}] \end{bmatrix}
\end{aligned} \tag{2.6}$$

Under the MCAR assumption and that $\mathbf{M}$ is uninformative, $\mathbf{E}[\mathbf{D}\mathbf{D}^T]$ becomes a 2×2 block diagonal matrix. This is due to the fact that, under the MCAR assumption, when $j \neq i$, $\mathbf{E}[\mathbf{Z}_i \mathbf{M}_j] = 0$ as $\mathbf{Z}_i$ and $\mathbf{M}_j$ are independent. This results in $\mathbf{E}[\mathbf{A}_i \mathbf{B}_j] = 0$. For $j = i$, $\mathbf{Z}_i$ and $\mathbf{M}_j$ are not independent (since they are directly dependent by construction). Let $p_i = P(\mathbf{M} = 1)$, $\mathbf{E}[\mathbf{Z}_i \mathbf{M}_i] = \mathbf{E}[\mathbf{Z}_i \mathbf{M}_i | \mathbf{M}_i = 1]p_i + \mathbf{E}[\mathbf{Z}_i \mathbf{M}_i | \mathbf{M}_i = 0](1 - p_i) = 0$, which results in $\mathbf{E}[\mathbf{A}_i \mathbf{B}_i] = 0$

If the missing indicator $\boldsymbol{\gamma}^*$ is uninformative and $\mathbf{Y}$ is centered, $\mathbf{E}[\mathbf{B}_j \mathbf{Y}] = \mathbf{E}[\mathbf{B}_i] \mathbf{E}[\mathbf{Y}] = 0$. Thus, if the missingness is uninformative, $\boldsymbol{\gamma}^* = 0$, which helps identify the missing locations where imputation can be performed.

2.3.2 Input features with noise

Most real-world datasets consist of noisy features that impact the performance of predictions made by the deep learning models. These noisy features result in aleatoric uncertainty [34] which must be considered when working with machine learning models. Thus, we also consider a case where the input features $\mathbf{Z}$ are noisy. As the noise in the input features increases, the estimate of the informative indicator will also be erroneous. We propose a Bayesian setup to estimate the informative indicator in the presence of noisy input features. We assume a normal prior on the weights and the informative indicator and use Bayes' theorem to find the posterior of the informative indicator.

Theorem 1. Consider a neural network with noisy input features and network layer parameters $\Theta = \begin{bmatrix} \theta & \gamma \end{bmatrix}$, where θ represents the neural network weights and γ represents the informative indicator. The mean (μ_λ) and variance (v_λ) of the informative missing indicator under input uncertainty, given the linearized approximation of the neural network output as $Y = \Lambda\Theta + C$, are:

$$\begin{aligned}\mu_\lambda &= \Phi^{-1}\Gamma^T, \\ v_\lambda &= \Phi^{-1},\end{aligned}\tag{2.7}$$

where, $\Phi = \Lambda^T \Sigma_y^{-1} \Lambda + \Sigma_\Theta^{-1}$ and $\Gamma = \mu_\Theta^T \Sigma_\Theta^{-1} + Y^T \Sigma_y^{-1} \Lambda$. Here, Σ_y is the variance associated with the output Y , and μ_Θ and Σ_Θ are the prior mean and variance of the network layer parameters.

Proof. Let $\mathbf{Z} \sim \mathcal{N}(\mu_z, \Sigma_z)$ be the input for a neural network with the parameters $\Theta = \begin{bmatrix} \theta & \gamma \end{bmatrix}$. Assume normal priors on the parameters $\Theta \sim \mathcal{N}(\mu_\Theta, \Sigma_\Theta)$. Using Bayes' theorem, the posterior on the parameters can be estimated as follows

$$\underbrace{P(\Theta|\mathbf{Z}, \mathbf{Y})}_{\text{posterior}} \propto \underbrace{P(\mathbf{Y}|\mathbf{Z}, \Theta)}_{\text{likelihood}} \underbrace{P(\Theta)}_{\text{prior}}\tag{2.8}$$

The linearized approximation of the predictions of the neural network from equation (2.5) can be rewritten as $\mathbf{Y} = \Lambda\Theta + \mathbf{C}$ where $\Lambda = \begin{bmatrix} \mathbf{A} & \mathbf{B} \end{bmatrix}$ and can be used to calculate the likelihood in equation (5.7) using the propagation of uncertainty for linear functions [35]. For a neural network with nonlinear transformation $\mathbf{f}$ of intermediate activations, the network layer parameter Θ and the output of the layer given by $\mathbf{Y} = \mathbf{f}(\mathbf{Z}, \Theta)$, the estimation of expected value of response variable $\hat{\mathbf{Y}}$ and the associated covariance can be calculated as follows

$$\begin{aligned}\hat{\mathbf{Y}} &= \mathbf{f}(\hat{\mathbf{Z}}, \mathbf{M}, \Theta), \\ \Sigma_Y &= (\nabla_Z \mathbf{f}) \Sigma_Z (\nabla_Z \mathbf{f})^\top + (\nabla_\Theta \mathbf{f}) \Sigma_\Theta (\nabla_\Theta \mathbf{f})^\top\end{aligned}\tag{2.9}$$

Here, the expected value of the input is $\mathbb{E}[\mathbf{Z}] = \hat{\mathbf{Z}}$ and Σ_Y is the covariance associated with the output of the neural network layer. Thus, the likelihood $P(\mathbf{Y}|\mathbf{Z}, \boldsymbol{\Theta}) \sim \mathcal{N}(\boldsymbol{\mu}_Y, \Sigma_Y)$ can be obtained from the equation (5.7) and the posterior is calculated as follows,

$$P(\boldsymbol{\Theta}|\mathbf{Z}, \mathbf{Y}) \sim \mathcal{N}(\Phi^{-1}\Gamma^T, \Phi^{-1}) \quad (2.10)$$

where $\Phi = \mathbf{A}^T \Sigma_y^{-1} \mathbf{A} + \Sigma_{\Theta}^{-1}$ and $\Gamma = \boldsymbol{\mu}_{\Theta}^T \Sigma_{\Theta}^{-1} + \mathbf{Y}^T \Sigma_y^{-1} \mathbf{A}$.

□

Thus, the updated posterior from Theorem 1 gives an estimate for an informative indicator. However, due to the noise in the input, the estimated informative indicator could also be erroneous. To address this, we statistically test the relationship between the response variable and the informative indicator. We use a t-test to compare $\mathbf{E}[\mathbf{Y}|\gamma = 0]$ and $\mathbf{E}[\mathbf{Y}|\gamma \neq 0]$ with a significance level α . Given that we have a simple linear model from equation (6), let (X, Y) be the set of points that satisfy equation (6). Let $t = \frac{\hat{\gamma} - \gamma'}{\frac{s}{\sqrt{\sum (x - \bar{x})^2}}}$, where s is the sample standard deviation. To test $H_0 : \gamma = \gamma'$ vs $H_1 : \gamma \neq \gamma'$ at the significance level α , reject H_0 if t is either less than or equal to $-t_{\alpha/2, n-2}$ or greater than or equal to $t_{\alpha/2, n-2}$. The steps for estimating the informative indicator with noisy input features are given in Algorithm 1.

Algorithm 1 Algorithm for Estimating Informative Indicator with Noisy input features

Require: Input: $\mathbf{Z}, \mathbf{M}, \mathbf{Y}$ significance level α Output: γ^*

- 1: (i) Estimate $\boldsymbol{\theta}^*$ and γ^* by training the neural network,
 - 2: (ii) Calculate the mean and covariance of response variable Y using equation (2.9),
 - 3: (iii) Find the posterior of γ using equation (5.9),
 - 4: (iv) t-test($\mathbf{Y}|\gamma^* = 0, \mathbf{Y}|\gamma^* \neq 0$) with significance level α
 - 5: Output: Return if informative indicator is zero or not
-

2.4 Results

We empirically evaluate the performance of the proposed IMIM for both noiseless and noisy cases. We create missing patterns similar to those in [21]. The dataset consists of $\mathbf{X}$, the complete input feature matrix, and we assume that the missing pattern follows the MCAR

mechanism. A missing mask, based on equation (2.11) is used to control the informativeness in the missing pattern:

$$P(\mathbf{M}_j = 0|\mathbf{X}) = \frac{1}{1 + \exp(-\lambda_j \mathbf{X}_j)} \quad (2.11)$$

The parameter λ_j is the informativeness parameter, used to control the informativeness of the missingness. It should be noted that the missingness is uninformative when $\lambda = 0$, and becomes more informative as the value of λ increases.

The proposed method is evaluated on synthetic as well as real-world datasets for both regression and classification tasks. For each of these tasks, we mask the input features and create missingness using equation (2.11).

2.4.1 Synthetic datasets

We create synthetic data for regression (with 1000 datapoints and 5 features) and classification (with 4000 datapoints and 3 classes). Mean squared error (MSE) was used as a metric to evaluate the performance in regression, while area under the curve (AUC) was used for binary classification, and accuracy was used for multiclass classification. The results are shown in Fig. 2.2–2.3. For both the classification as well as regression tasks, the informative parameter from equation (2.11) is varied from 0 to 5 to control the informative missingness. The proposed approach is compared against the baseline, where imputation is performed at all the missing locations. It can be seen from the Fig. 2.2–2.3 that the proposed IMIM (imputation performed at selected missing locations) method outperforms the baseline NoIMIM (imputation performed at all the missing locations) for both regression and classification tasks. As the informativeness parameter increases (indicating that missingness is informative), performing imputation at these locations leads to information loss and reduced accuracy in the NoIMIM scenario. In contrast, IMIM selectively imputes, focusing on non-informative locations, thereby preserving information. This is also evident from Fig. 2.4, where we plot the percentage of imputations, i.e., the number of imputations performed relative to the total missing locations. Intuitively, as the informativeness

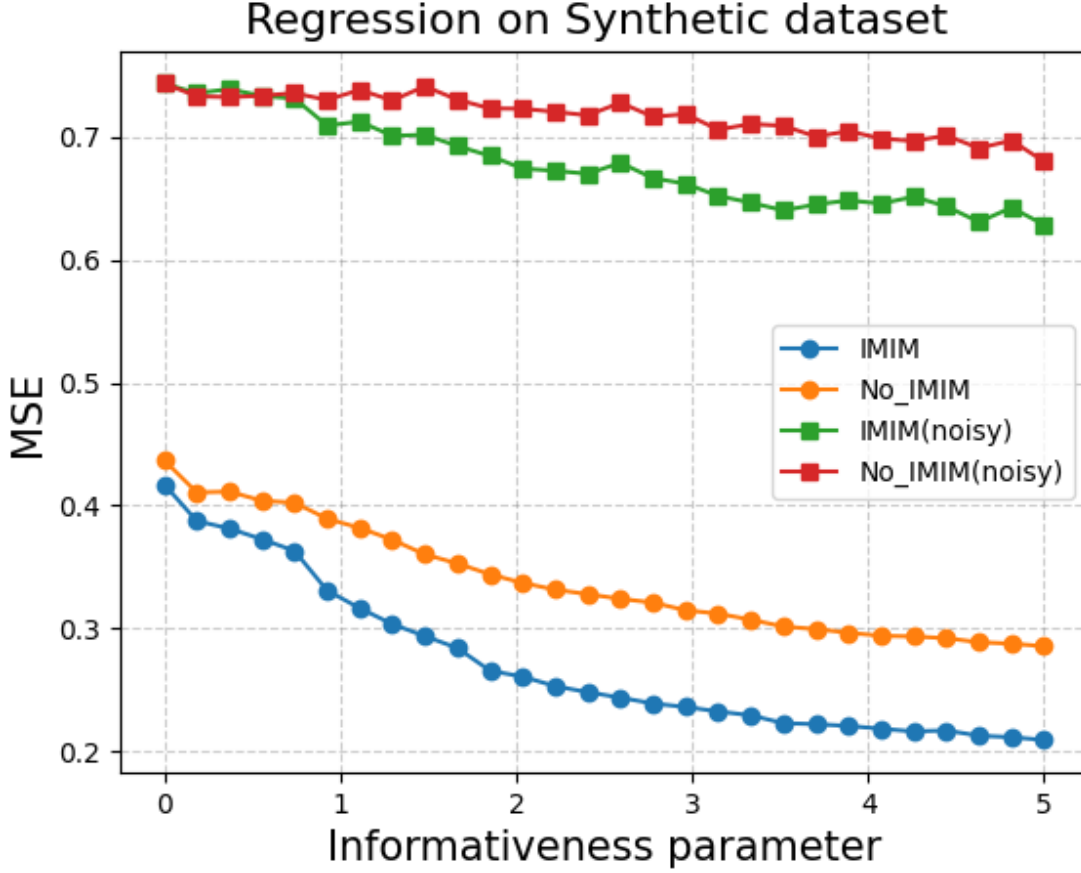


Figure 2.2: Performance evaluation of IMIM on synthetic regression dataset

parameter increases, the missingness becomes more informative and should not be imputed. The algorithm successfully identifies this trend, resulting in a decrease in the percentage of imputations as the informativeness parameter increases.

We also consider the noisy case by adding Gaussian noise to the input feature matrix. The results are shown in Fig. 2.2–2.3. In this scenario, the uncertainty in the input features results in noisy estimates of the informative indicator. The uncertainty in the estimates is quantified using equation (5.9). This quantified uncertainty helps determine whether to “trust” the informative indicator. We check if this informative indicator is reliable and decide whether to impute the missing values. The results shown in Fig. 2.2–2.3 are obtained by setting a significance level $\alpha = 0.05$. It is evident from the figures that even in the noisy case, the proposed IMIM method outperforms the baseline method of imputing at all locations.

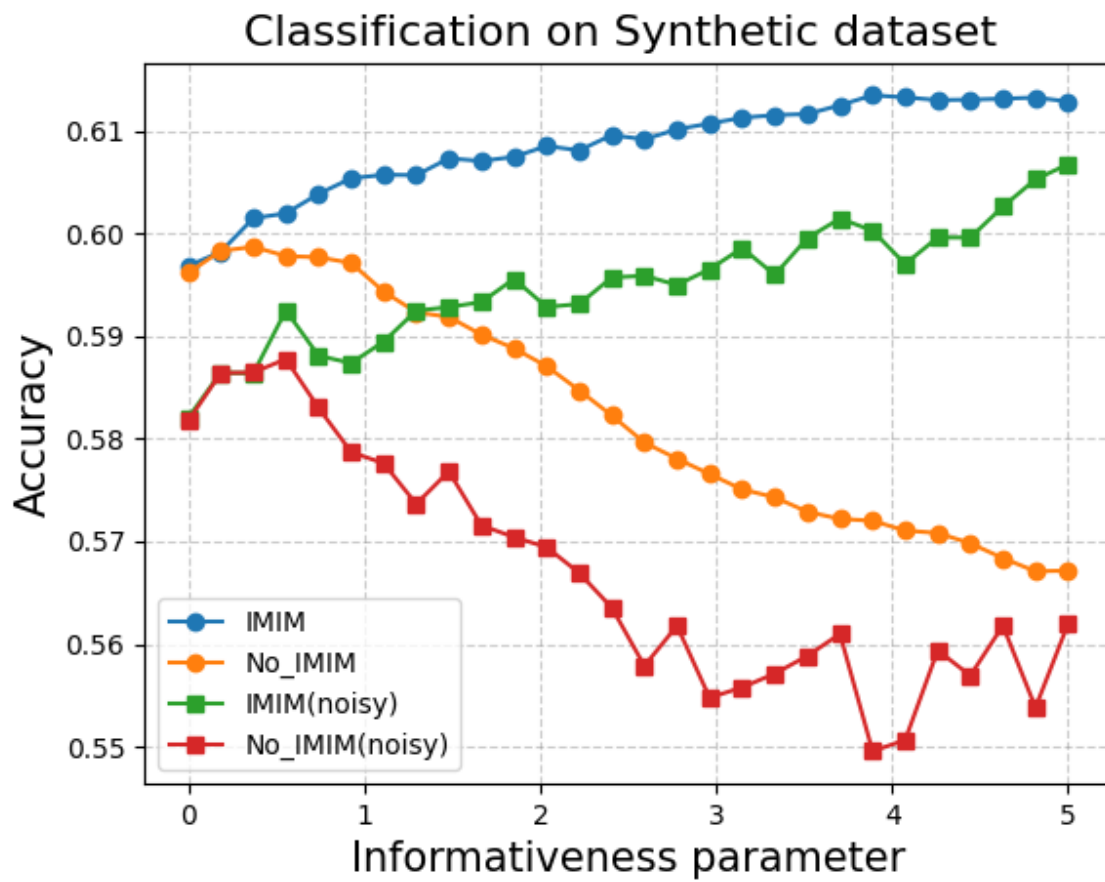


Figure 2.3: Performance evaluation of IMIM on synthetic classification dataset

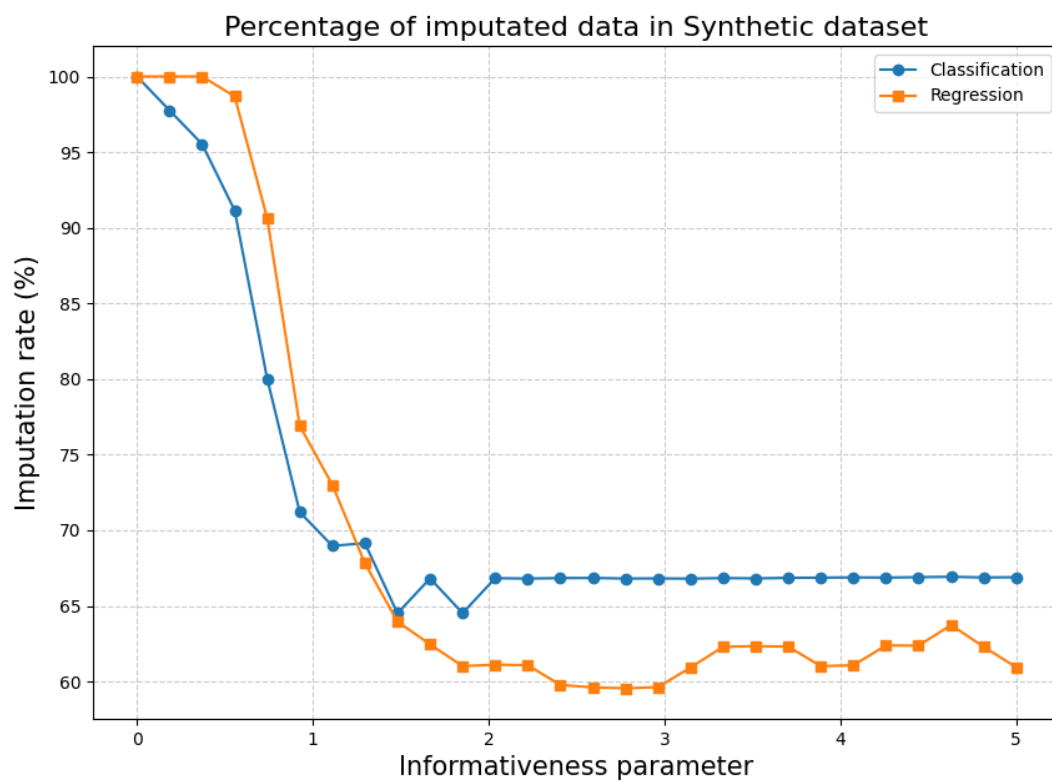


Figure 2.4: Percentage of missing data imputed with change in informativeness parameter for synthetic dataset

Although we consider only mean imputation for the missing locations in this work, other statistical measures can also be used.

2.4.2 Real-world datasets

We evaluate the performance of our proposed IMIM method on two real-world datasets for regression and classification from OpenML [36]. The details of the datasets are as follows:

- Geographical Analysis Spatial Data for regression: Contains 3,107 observations on U.S. county votes in the 1980 presidential election, including votes (VOTES), population over 18 (POP), educated population (EDUCATION), owner-occupied housing units (HOUSES), aggregate income (INCOME), and spatial coordinates.
- Wine Quality for classification: Contains data on red and white wine samples, with inputs from objective tests (e.g., PH values) and outputs based on sensory data from wine experts grading quality between 0 and 10.

We control the degree of missingness by manipulating the parameter λ similar to our approach with synthetic datasets. Fig. 2.5–2.6 illustrate the comparative performance of IMIM and the baseline across different values of λ . As mentioned previously, as the value of λ increases, the missingness becomes informative. In such scenarios, imputing the missing locations results in a loss of information rather than gaining information as it is evident from the results shown in Fig. 2.5–2.6. The figures demonstrate that higher values of λ correspond to more informative missingness, leading to reduced mean squared error in regression and improved accuracy in classification tasks when using IMIM. By selectively imputing only at locations where missingness is uninformative, IMIM consistently outperforms the baseline strategy of imputing at all locations. On average, our method achieves up to a 5.97% reduction in error for regression tasks and 15.02% improvement in accuracy for classification tasks when the informativeness parameter is set to 5. Fig. 2.7 illustrates the percentage of missing locations imputed as the informativeness parameter increases. Similar to the case

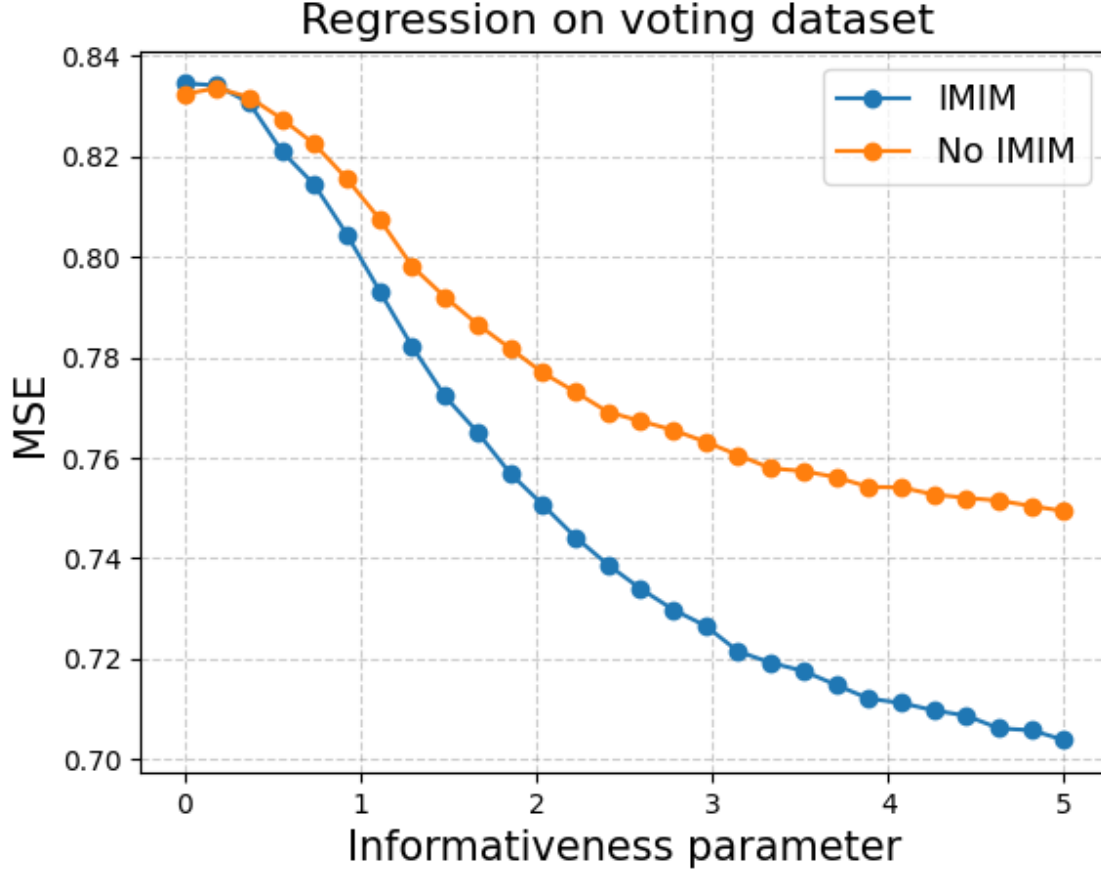


Figure 2.5: Regression performance of IMIM on voting dataset

with the synthetic dataset, the trend shows that the percentage of imputed missing locations decreases as the informativeness parameter increases.

2.5 Summary

In this chapter, we introduced the Informative Missing Indicator Method (IMIM), designed for nonlinear function approximators such as neural networks, which identifies informative missingness and guides imputation when missingness is uninformative. We incorporated statistical tests to determine whether imputation is necessary in the presence of noisy input features, ensuring the reliability of the informative indicator even under uncertainty caused by measurement noise. Experimental evaluations conducted on synthetic and real-world

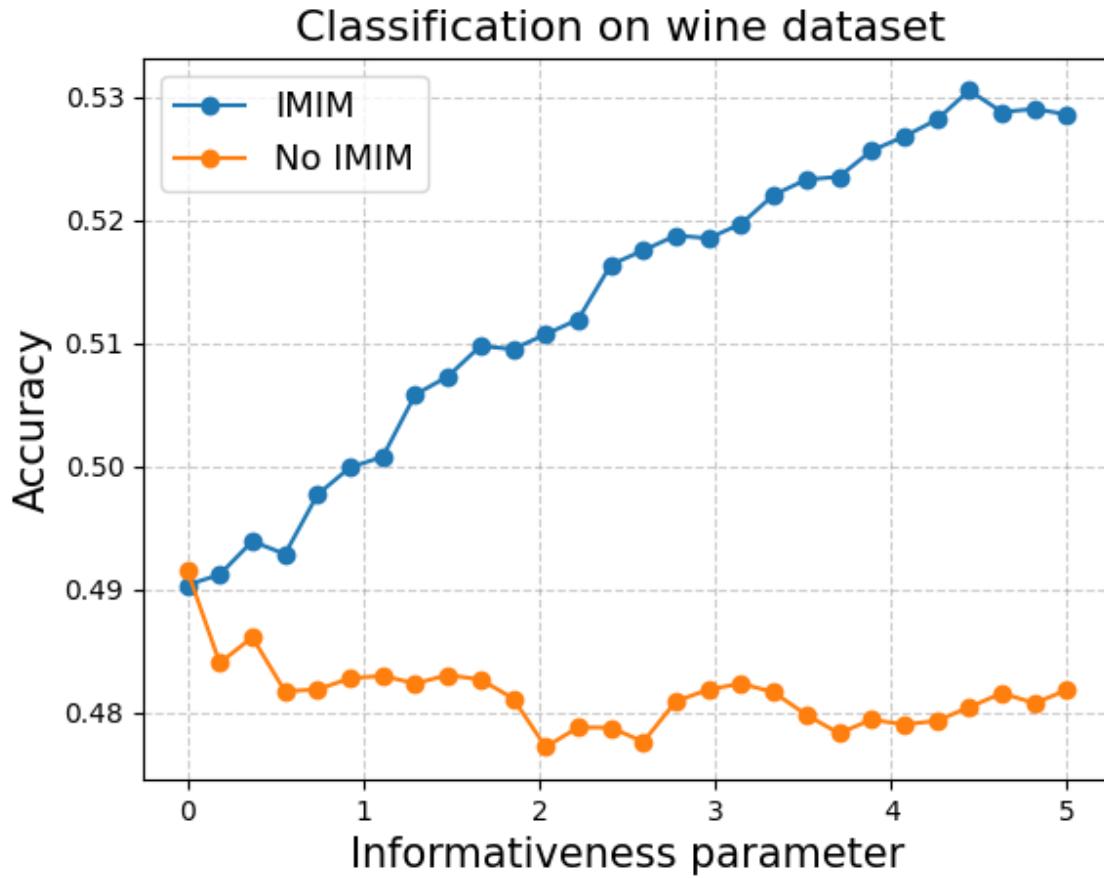


Figure 2.6: Classification performance of IMIM on wine dataset

datasets for classification and regression tasks demonstrated that selectively imputing data using IMIM significantly outperforms methods that fully impute missing values. In the next chapter, we will explore how to further leverage underlying spatio-temporal relationships in the data to enhance imputation techniques.

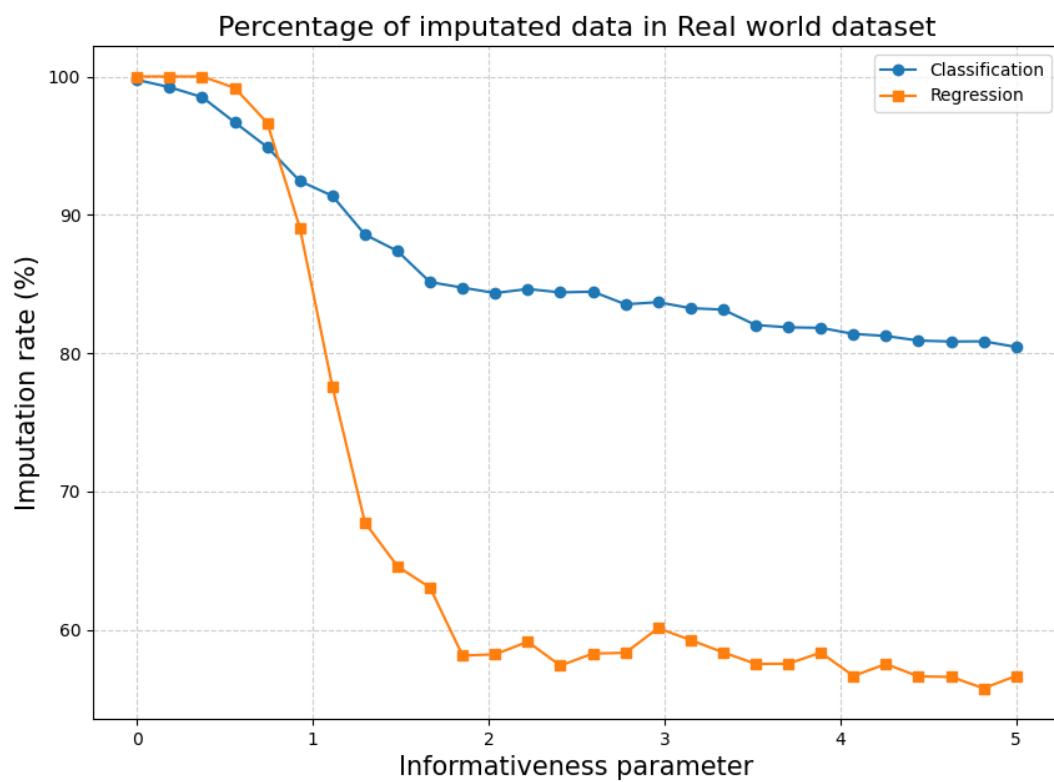


Figure 2.7: Percentage of missing data imputed with change in informativeness parameter for real world dataset

Chapter 3

Multivariate Spatio-temporal Imputation

As discussed in Chapter 2, imputing the entire dataset may lead to the loss of critical information, resulting in lower accuracy in downstream tasks. However, selective imputation using effective methods can enrich the dataset without sacrificing important details. In this chapter, we introduce an imputation framework that leverages both spatial and temporal relationships inherent in the data, effectively filling data gaps and enhancing the overall informativeness of sparse spatio-temporal datasets.

3.1 Related work

A variety of spatiotemporal imputation models have been developed to address missing data in spatiotemporal datasets. Some of the classic statistical methods involve interpolation-based methods that use linear interpolation to estimate missing values based on the values from the neighboring time/spatial points [37], [38]. However, these classical methods rely on the assumption that the underlying data follows a smooth trend and fail to provide accurate estimates when there is a large number of missing points in the data. Yet another method to impute the missing data is by estimating/assuming the correlation between multiple variables

in the dataset [22]. This approach requires a high degree of statistical expertise and can be computationally expensive [39].

Multilinear tensor completion that uses a low-rank tensor approximation based on observed entries to reconstruct the missing values is proposed in [40]. While the approach in [40] effectively captures multi-dimensional structural dependencies, it is unsuitable for complex interactions and diverse data missing patterns. A convolutional neural network based tensor completion (CoSTCo) method was proposed in [41]. While CoSTCo captures the non-linear relationships in the dataset, the transductive nature makes the algorithm less scalable. Imputation techniques based on machine learning algorithms use k-nearest neighbors [42] or support vector machines to estimate missing values based on patterns in the data [43]. These methods do not attempt to capture the complex relationships inherent in the spatiotemporal data and only rely on data similarity metrics.

Recently, deep learning-based approaches have been proposed to impute missing data. Denoising stacked autoencoder (DSAE) [44], is a typical deep learning model that combines denoising and autoencoders for imputation. However, DSAE does not account for the underlying spatial correlations. To leverage the spatial correlations, [45] proposes a multi-range convolutional neural network (CNN) to model spatial correlations and impute missing information. Though the method proposed in [45] effectively handles correlations in Euclidean space, they are inefficient in modeling relationships in non-Euclidean spaces. Recently, graph structures for relational reasoning have been utilized in Graph Convolutional Networks (GCN) [46], [47]. Though GCN is effective in modeling topological relationships, it is not tailored to capture temporal dependencies. Alternatively, recurrent neural network-based models can impute time series with missing values. Specifically, long short-term memory (LSTM) networks can capture and maintain long-term dependencies [48].

3.1.1 Contributions

In this chapter, we propose a novel inductive framework (G-LSTM) for missing data imputation that integrates a graph neural network with LSTMs to effectively capture both spatial

and temporal dependencies. We use GraphSAGE [49] as a GNN module which is an inductive method and computationally efficient compared to GCN. Furthermore, GraphSAGE allows for incorporating node features in the embedding generation process, which can be useful for tasks where node attributes are essential. The proposed general framework can be used for both data imputation and prediction.

The performance of the proposed framework is highlighted using comprehensive case studies on real-world traffic datasets. The case studies include missing rates ranging from 10 % to 90%. Experimental results demonstrate that the proposed GNN integrated with the LSTM framework achieves improved imputation and maintains steady performance even when there are extreme missing conditions in comparison with the state-of-the-art imputation framework CoSTCo. The simulation results on the traffic network show up to 69% reduction in mean absolute error and 61% reduction in root mean square error when compared to state-of-the-art imputation framework.

3.2 Problem Formulation

Spatio-temporal data from a sensor network can be defined as data that varies with respect to time and space and can be represented by a tensor $X \in \mathbb{R}^{N \times M \times T}$ where, N represents the number of sensors, T is the number of time steps in the observation window of interest and M is the number of measured features. Sensor data may have missing information due to sensor failure and/or communication disruptions. The objective is to impute the missing values by utilizing spatial and temporal relationships in the data. Due to complex underlying patterns, the analysis of spatiotemporal data becomes difficult. Adopting a network structure aids in explicitly modeling and capturing the underlying complicated spatio-temporal interactions.

We represent the spatiotemporal dataset as an undirected graph $\mathbf{G} = (\mathbf{A}, \mathbf{E}, \mathbf{X}_t)$, where $\mathbf{A} \in \mathbb{R}^{N \times N}$ denotes the spatial adjacency matrix; N denotes the number of sensors and the element $\mathbf{a}_{i,j} \in \mathbf{A}$ is 1 if the sensors i and j are adjacent; Edge $\mathbf{E}$ represents the spatial relationship between the sensors, and $\mathbf{X}_t \in \mathbb{R}^{N \times \mathbb{D}}$ is the set of $\mathbb{D}$ dimensional dynamic features measured at each sensor placed at various locations at time t . Given the incomplete

feature values from each sensor at different time instances, the goal is to impute the missing feature values by considering spatial and temporal dependencies. The observability of the feature values/measurements is captured by a binary mask matrix $\mathbf{M}_t \in \{0, 1\}^{N \times D}$ with elements as 1 if $\mathbf{X}_t^{(i,j)}$ is observed and 0 otherwise. Given the incomplete sensor measurements $\mathbf{Y}_t = \mathbf{M}_t \odot \mathbf{X}_t$, the objective is to reconstruct the missing sensor measurements and estimate the complete matrix $\hat{\mathbf{Y}}_t$.

3.3 Proposed G-LSTM Framework

The proposed general spatio-temporal imputation framework is depicted in figure 3.1. The framework consists of a spatial module integrated with a temporal module to estimate $\hat{\mathbf{Y}}_t$. These modules are described next.

3.3.1 Spatial Module

The spatial module in our framework captures the spatial relationships within the data. The module consists of multiple stacked layers of GraphSAGE (SAmple and aggreGatE) [49]. Each node in the graph is represented by a node embedding vector that helps capture the structural information. GraphSAGE is an inductive approach in which the algorithm learns a mapping (aggregator) function instead of the node embedding vector. The two primary steps associated with GraphSAGE are aggregate and update.

- **Aggregate:** Each node in the graph is uniquely represented by a feature vector. The aggregate step aggregates the neighboring node representations (feature vectors) for our target node. The aggregate function is a placeholder for any differentiable function and it could be something as simple as an averaging function or something as complex as a neural network. The corresponding aggregation function is:

$$a_v = f_{agg}(h_u | u \in N(v)) \quad (3.1)$$

Here, h_u s the feature values associated with the neighbor u of node v

- Update: After obtaining an aggregated representation for node v based on its neighbors, the feature representation of the current node v is updated using a combination of its previous representation and the aggregated representation. The update function serves as a placeholder for any differentiable function which can once again, be as basic as an averaging function or as sophisticated as a neural network and can be represented as,

$$h_v^k = f_u(a_v, h_v^{k-1}) \quad (3.2)$$

Here, k is the number of hops considered for neighborhood aggregation. The feature values of the nodes at the k^{th} hop are updated based on the feature values from the $(k - 1)^{th}$ layer.

GraphSAGE’s inductive nature allows it to infer the node embedding vector for nodes not encountered during training, making it suitable for our imputation framework. The GNN module takes the spatial adjacency matrix and the observed feature values at each time instance. It captures the spatial information using the message-passing mechanism of a Graph Neural Network. The output of GNN is then passed on to the temporal module described next.

3.3.2 Temporal Module

Temporal modules are used for capturing the temporal relations in the network. The temporal module is driven by recurrent neural networks (RNN)s. RNNs are neural networks with loops in them that allow past information to be retained from a sequential dataset. Among the various available RNN architectures, we have adopted Long Short Term Memory (LSTM) memory networks [50] for our imputation framework. LSTMs have been shown to perform well on sequence-based tasks with long-term dependencies, assisting in capturing temporal associations. At time instant t , LSTM accepts as input the current variable vector x_t and previous cell state vector c_{t-1} and previous hidden state vector h_{t-1} . It also includes

three gate units, input gate i_t , forget gate f_t , output gate o_t , and hidden memory state C_t . These gates make LSTM an efficient, scalable sequential data learning technique. The cell state and hidden state can be represented in terms of the gates as:

$$c_t = f_t \odot c_{t-1} + i_t \odot C_t \quad (3.3)$$

$$h_t = o_t \odot \tanh(c_t) \quad (3.4)$$

where $\odot$ represents the element-wise vector/matrix multiplication operator. The output of the LSTM layer can be a set of outputs from all T time steps, resulting in a one-to-one architecture, where for each input at time t , there would be a corresponding output.

The output of the LSTM is then passed to a fully connected neural network. The reconstruction loss calculated during the training is on the entire observation and helps to learn the inherent relationships among the entire spatio-temporal dataset. The patterns learned and propagated help update/ impute data at the missing location and time instants.

The overall training workflow of the framework is shown in figure 4.1. The spatio-temporal data is loaded and punctured to create missing values (the missing locations are given zero values). Then, the data is split temporally into training and testing segments. Finally, the ST block is trained and the recovered values are used for evaluation by comparison with the original input data. The loss function used for evaluation corresponds to the following:

$$\mathbb{L}(\mathcal{Z}) = \sqrt{\sum_{v=1}^N \sum_{t=1}^T (x_t^v - \hat{x}_t^v)^2} \quad (3.5)$$

where $\mathbb{L}(\mathcal{Z})$ is the reconstruction error on both observed and missing data. x_t^v and $\hat{x}_t^v$ are the actual and estimated values respectively for node v and time t .

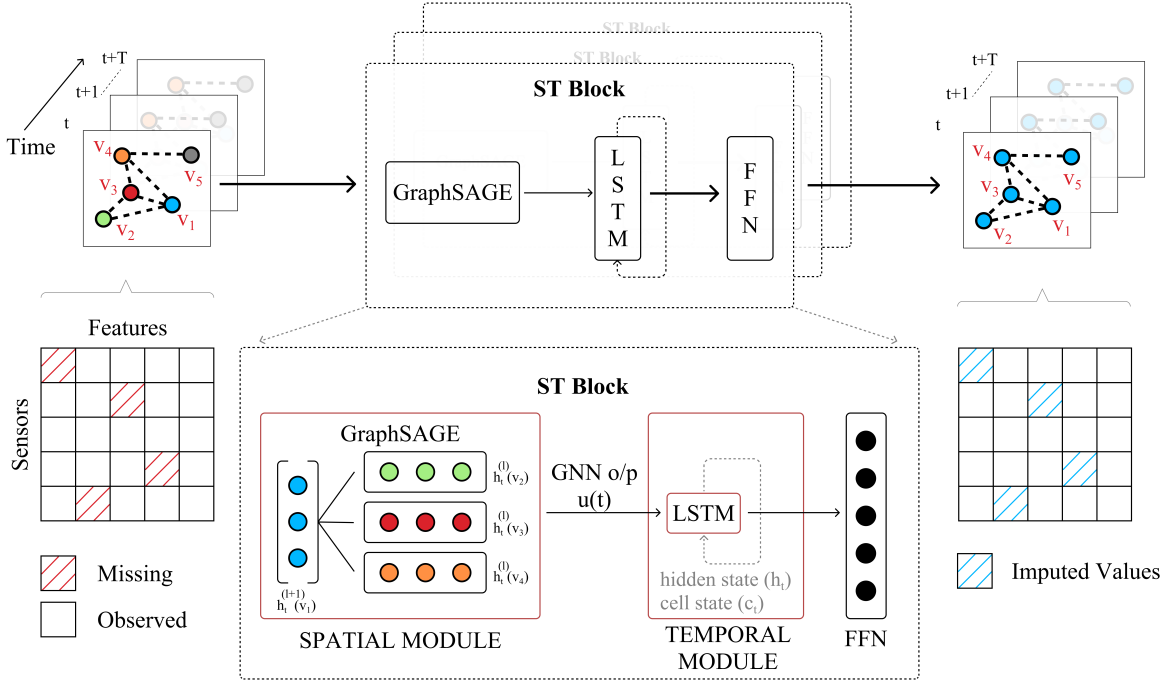


Figure 3.1: Structure of proposed spatiotemporal imputation network (G-LSTM)

3.3.3 Complexity Analysis

In this subsection, we discuss the computational complexities of the proposed algorithm. The computing cost of the G-LSTM is concentrated on the GraphSAGE layers and the LSTM layers. Suppose for a graph that is being considered there are n number of nodes, r number of neighbors being sampled for each node and m the total number of edges and K number of layers. The computational complexity of the GraphSAGE can be represented as $\mathcal{O}(r^K nd^2)$. For an LSTM with h as the hidden layer dimension and b as the input feature, the computational complexity is $\mathcal{O}(4h(b + 2 + h))$. Thus the computational complexity of the G-LSTM framework can be approximated as $\mathcal{O}(r^K nd^2 + 4h(b + 2 + h))$.

3.4 Results

We evaluate the performance of the proposed imputation method on a real-world traffic dataset. We introduce the dataset and explain the experimental settings first. Then various

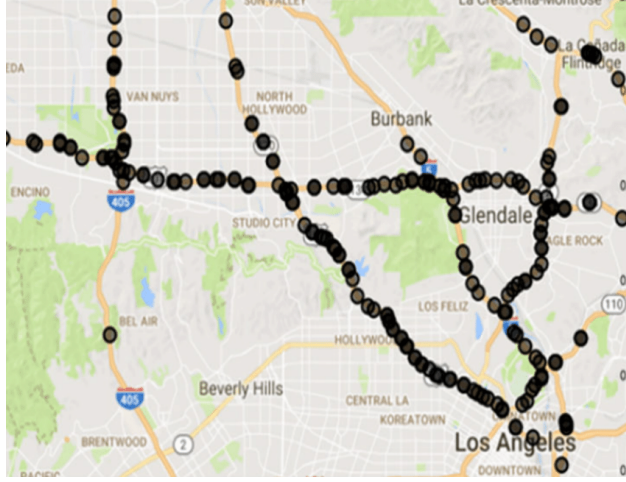


Figure 3.2: Sensor monitor distribution in PEMS7 traffic dataset

missing data scenarios are considered. The imputation performance of the proposed method is compared with Costco, a neural tensor completion-based imputation framework as the baseline.

Traffic-State dataset: PeMSD7 is a traffic dataset collected from Caltrans Performance Measurement System (PeMS) and describes the speed detectors covering the freeway system in all major California urban centers. Among the 1000 sensors placed on the arterial roads of District 7 in California figure 3.2, we chose 228 sensors for our study, similar to [51]. The dataset consists of 5-min average traffic speed data collected from May 1, 2012, to June 30, 2012, and has 11232 time points. The graphical representation of the PEMS dataset was created in a manner similar to [51]. The weighted adjacency matrix of the road is computed based on the distances between the sensor stations. The weights are calculated using as:

$$w_{ij} = \begin{cases} \exp(-\frac{d_{ij}^2}{\sigma^2}), i \neq j \text{ and } \exp(-\frac{d_{ij}^2}{\sigma^2}) \geq \delta \\ 0, \text{otherwise} \end{cases} \quad (3.6)$$

The stations i and j are assumed to be connected in the equivalent graph if the $w_{ij} \geq \delta$, where σ^2 and δ are thresholds chosen to determine whether or not there is a connection in the graph, with values of 10 and 0.5 chosen respectively, d_{ij} is the distance between two stations. The nodes correspond to each sensor in the network and the initial node features are the

speed values concatenated with the singular value decomposition (SVD) of the weighted adjacency matrix. The SVD values concatenated with the speed values aid in uniquely identifying the nodes [52]. For computational purposes, the data is normalized.

The imputation performance is evaluated using two metrics, root mean square error (RMSE) and mean absolute error (MAE) defined as,

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \hat{x}_i)^2} \quad (3.7)$$

$$MAE = \frac{1}{n} \sum_{i=1}^n |x_i - \hat{x}_i| \quad (3.8)$$

where, n is the number of missing values, x_i is the i^{th} missing value and $\hat{x}_i$ is the imputed value of x_i .

Experimental Results:

For training and testing purposes, the dataset is split temporally. The initial 80% of time instances are used for training purposes and the remaining 20% is used for testing. The data is scaled to be between 0 and 1 before imputation and the output of the proposed approach is rescaled back to the original values. It is assumed that the sensors at specific locations were absent across all time instances to simulate the missing values. The proposed imputation framework model is trained by Adam optimizer [53]. The learning rate is set at 0.001 with the ReLu activation function after the GNN layers. The time window for the LSTM is set as one in order to maintain an end-to-end training framework and the temporal information is captured by the hidden and the cell states. The model is iterated for 150 epochs and the training loss curve is shown in figure 3.3. All experiments are implemented with PyTorch and conducted on an NVIDIA GeForce RTX 3070 GPU.

To evaluate the performance of the proposed approach, we compare its performance with CoSTCo [41], a state-of-the-art convolution neural network-based tensor completion approach. CoSTCo tries to address the inability of multilinear models to generate a low-

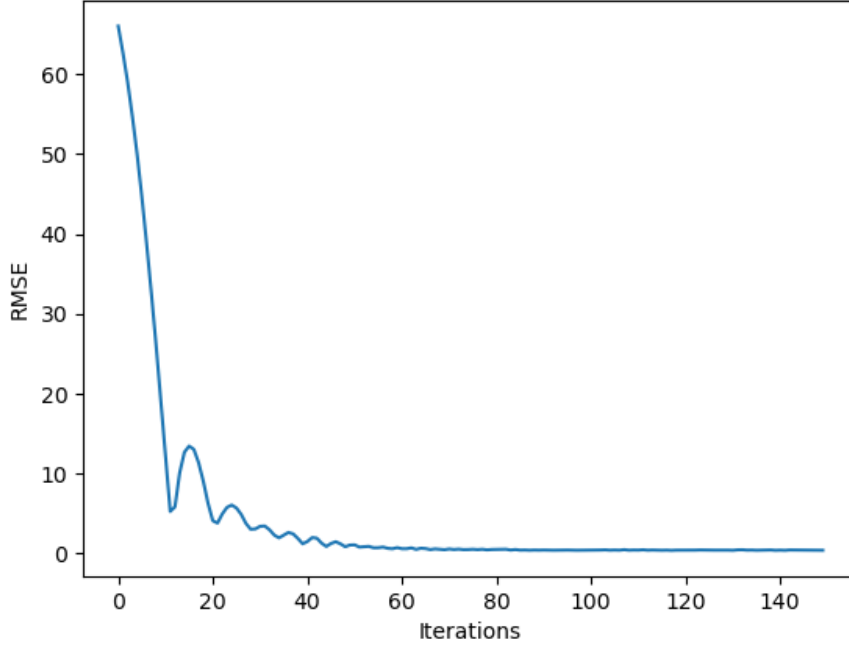


Figure 3.3: Training loss curve for G-LSTM

rank representation of non-linear data. It utilizes the activation functions in a convolutional neural network to capture non-linear relationships. Table 3.1 summarizes the experimental results using our proposed approach compared with CoSTCo as the baseline. The imputation errors are calculated with various missing ratios for a single time instant. The results are averaged over five random missing scenarios for each of the missing ratios. It can be seen that the proposed approach (G-LSTM) outperforms the baseline and as the percentage of missing information increases, the imputation error metrics remain steady and sometimes show improvement. This surprising observation can be attributed to the possibility that at a lower percentage of missing data, the GNN may be overfitting. As fewer data are available (or the missing percentage increases), the overfitting behavior is less likely. The G-LSTM is trained with 50% of missing data and the pre-trained model is used for imputing the data with 10%, 30%, 70% and 90% of missing information. From the results shown in table 3.1, it can be seen that for missing percentages lesser than 50%, the pre-trained G-LSTM model gives improved results compared to the corresponding G-LSTM model and CoSTCo. For missing

percentages greater than 50%, the pre-trained G-LSTM outperforms CoSTCo. However, the pre-trained model shows slightly inferior performance with respect to G-LSTM. The results indicate that the model trained for higher missing percentages can impute the data with lower missing information more efficiently and thus possess the inductive capability. Relative to the baseline CoSTCo method, the proposed framework offers approximately 69% reduced MAE and 61% reduced RMSE on the imputed values. We also compare our framework with the Graph Convolutional Recurrent Neural Network (GCRINT). GCRINT uses a combination of a Recurrent Neural Network (RNN) and a Graph Convolutional Neural Network to fill in the missing values of network traffic data. The performance of GCRINT has been compared with our framework, and the MAE and RMSE losses are tabulated. Unlike our approach, which uses an explicit graph to represent the relationships among traffic flows, GCRINT adopts a self-adaptive adjacency matrix to learn these relationships from the input data. This approach might not always capture the precise spatial relationships. Possibly for this reason, the performance of GCRINT is inferior to that of G-LSTM for some percentages of missing data, as shown in Table 3.1. The pretrained version of G-LSTM outperforms GCRINT for all percentages of missing data.

Table 3.1: Performance Evaluation for Missing Data Imputation using G-LSTM and SOTA methods

% Missing	10%		30%		50%		70%		90%	
Model	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
G-LSTM (ours)	1.61	1.90	1.67	2.09	0.74	1.19	1.16	1.50	0.83	1.23
G-LSTM (pre-trained)	0.72	1.36	0.92	1.23	NA	NA	1.33	1.59	1.65	1.85
GCRINT	0.96	1.78	0.97	1.85	1.02	1.90	1.05	1.85	–	–
CoSTCo	2.26	2.27	2.60	2.61	4.66	4.67	6.96	6.97	7.90	7.91

The speed values at each sensor for a single time step as predicted by G-LSTM and CoSTCo when 50% of information is missing are shown in figures 3.4 and 3.5, respectively. It can be seen that our framework outperforms the baseline in prediction with missing information.

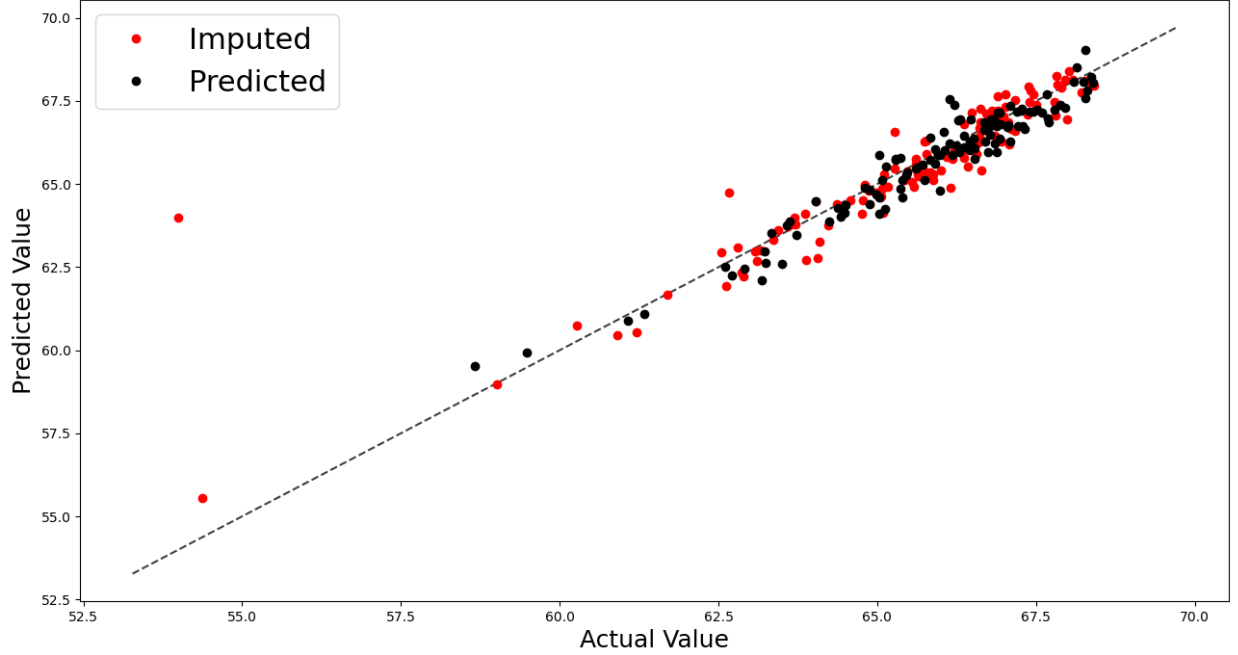


Figure 3.4: Speed values estimated for each node for a single time instant using G-LSTM framework

3.5 Summary

In this chapter, we propose a general imputation framework that leverages both spatial and temporal relationships to effectively fill in missing values. The framework, which comprises spatial and temporal modules, was validated on a traffic dataset where intersections (nodes in the graph) record the average speed of vehicles at various times. The results clearly demonstrate that the spatio-temporal module successfully captures the inherent relationships among neighboring intersections, enabling more accurate imputation. The performance evaluations for forecasting tasks indicate that the G-LSTM-based imputation framework outperforms state-of-the-art methods. Despite these promising results, uncertainties inherent in the machine learning models and the data persist. In the following chapters, we will explore how the uncertainty associated with the learning frameworks can be quantified.

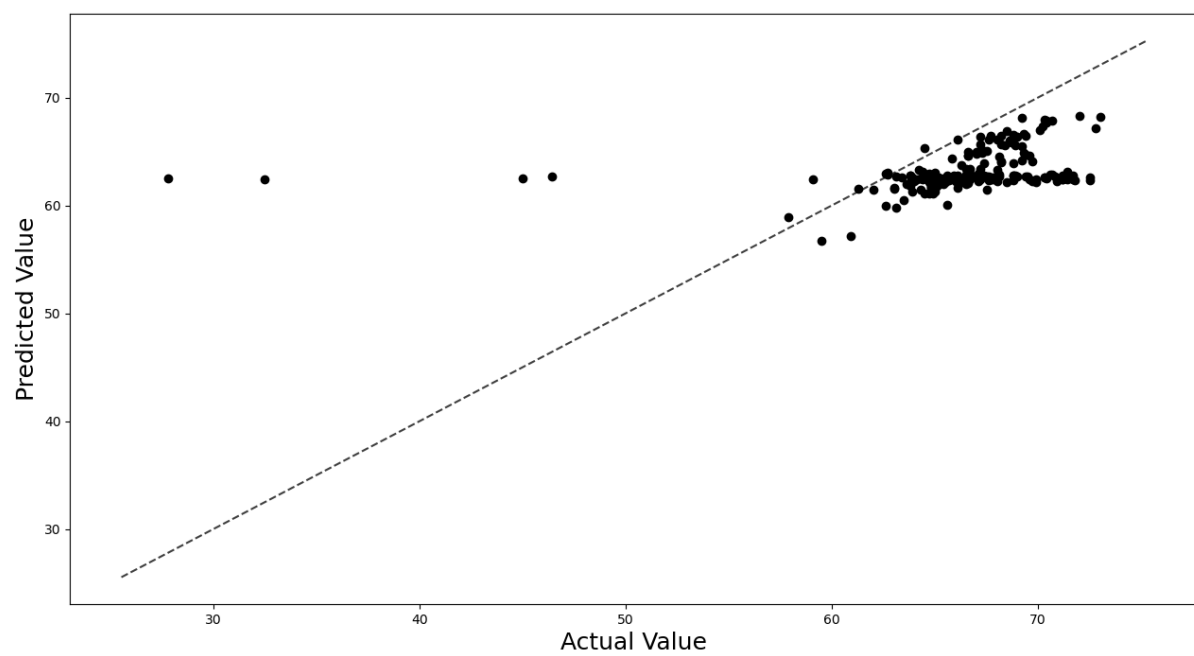


Figure 3.5: Speed values estimated for each node for a single time instant using CosTCO

Chapter 4

Uncertainty Quantification in Machine Learning Models

Traditional black-box machine learning models typically provide only point predictions, with no indication of the model’s confidence in its outputs. However, in safety-critical applications such as healthcare, point estimates alone are insufficient. It is crucial to determine when a model’s prediction can be trusted, and one way to achieve this is by assessing the uncertainty in those predictions. In this chapter, we discuss how uncertainty arises in machine learning models and introduce a novel approach in which surrogate models can be used to analytically quantify the uncertainty of the target neural networks, specifically for deep neural networks and graph neural networks.

4.1 UQ in Deep Learning

4.1.1 Related work

Deep neural networks (DNNs) are a type of artificial neural network trained using large datasets, and optimization algorithms to learn the parameters that map input data to output predictions. However, a significant challenge in deploying DNNs lies in dealing with uncertainty. Various methods have been proposed to quantify uncertainty in DNNs. Exist-

ing uncertainty quantification techniques can be divided into (1) Bayesian approaches or (2) sampling-based techniques. A classic neural network model with input x and output y and network parameter θ can be written probabilistically as $p(y|x, \theta)$. The posterior distribution obtained by applying Bayes' theorem can be written as follows:

$$p(\theta|x, y) = \frac{p(y|x, \theta)p(\theta)}{\int p(y|x, \theta)p(\theta)d\theta} \quad (4.1)$$

Existing approaches use Bayesian inference to estimate the posterior distribution over model parameters instead of treating the model parameters as fixed [54]. However, finding the posterior distribution over all possible model parameters can be intractable. There exists variation inference-based techniques that approximate the posterior probabilities rather than finding the exact value [55]. The process involves finding a simple distribution that is as close to the actual distribution. Using this technique training takes longer and inference is slower as approximating posterior probability requires several samples to be drawn from the posterior distribution. Laplace approximation is also used in Bayesian inference, focusing on the region around the maximum of the posterior distribution. By leveraging a Taylor-series expansion, it provides an approximation of the distribution in the vicinity of this maximum [56]. This method can only capture the local behavior of the distribution, which implies that the approximation may significantly differ from the true distribution in other areas. Among sampling-based techniques, another efficient technique for approximating inference is the Monte Carlo Markov chain (MCMC) method, which involves applying a stochastic transition to a random draw from a distribution. [57]. Despite the success of the MCMC method, the method converges slowly. Monte Carlo dropout is yet another technique that uses dropout as a Bayesian Approximation to calculate the intractable posterior distribution [58]. Monte Carlo (MC) dropout, which involves running inference multiple times with different dropout masks during testing can be used to estimate the epistemic uncertainty. Although dropout is a widely used technique, it is an empirical approach that gets computationally expensive for larger networks. The Deep Ensembles method proposed in [59] involves training multiple DNNs on the same dataset and combining their predictions to obtain the mean and variance

of the predictions and can be used as a metric for the uncertainty. However, the existing methods are time-consuming and require maintaining many copies of the model parameters to quantify uncertainty, which can be costly for large NNs. In [60], stochastic differential equations (SDEs)-based techniques were employed to train the SDE model using out-of-distribution (OOD) data for training in order to quantify total uncertainty. However, the method falls short in its ability to assess uncertainty in a rigorous and principled manner.

4.1.2 Contributions

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

- We propose a generic framework that quantifies the epistemic uncertainty of deep neural networks with the help of a neural SDE framework as a surrogate model applicable across various tasks and domains.
- The framework allows for uncertainty propagation through all the layers of the target neural network model, and the analytical results capturing the uncertainty (output mean and covariance) along each neural network layer are derived.
- It can be applied to pre-trained networks using the analytical method, which eliminates the need for any computationally demanding uncertainty quantification procedures.
- The effectiveness of the proposed framework is demonstrated using synthetic data for classification and regression tasks and the results show up to 88.45% reduction in UQ computational complexity relative to the state-of-the-art Monte Carlo dropout method.

4.1.3 Proposed UQ-Net approach

This section presents a detailed outline of the problem statement along with an introduction to the main elements of our suggested structure.

Given a training dataset $D = (x_j, y_j)_{j=1}^N$, where x_j represents the input data and y_j corresponds to the corresponding ground truth labels, we can train a model $\mathbf{M}$ parameterized

by θ to make predictions on new test cases. The predictive uncertainty originates from two sources as described earlier. Our goal is to analytically quantify the epistemic uncertainty of the neural network model with reduced computational time in comparison to the existing methods.

Uncertainty quantification with UQ-Net for deep neural networks

Given a target neural network model that performs a regression or classification task on a set of data, the UQ-Net works as a surrogate network and helps to measure how uncertain the neural network model's predictions are. As shown in figure 4.1, the UQ-Net, which has an input, output, and a hidden layer, helps in propagating the uncertainty in the corresponding layers of the target neural network. As shown in figure 4.2, the UQ-Net architecture consists of input and output linear layers of deep neural networks at both the input and output stages. The neural SDE layer within UQ-Net corresponds to the hidden layer of the deep neural network. Since neural SDEs preserve dimensionality, the input and output layers of the network serve as reshaping layers before and after the hidden layer. These reshaping layers facilitate the seamless integration of the neural SDE layer into the overall network structure and help UQ-Net propagate the mean and covariance analytically from input to output. Each component of UQ-Net is further elaborated in the following sections, along with the analytical derivation of the output mean and covariance.

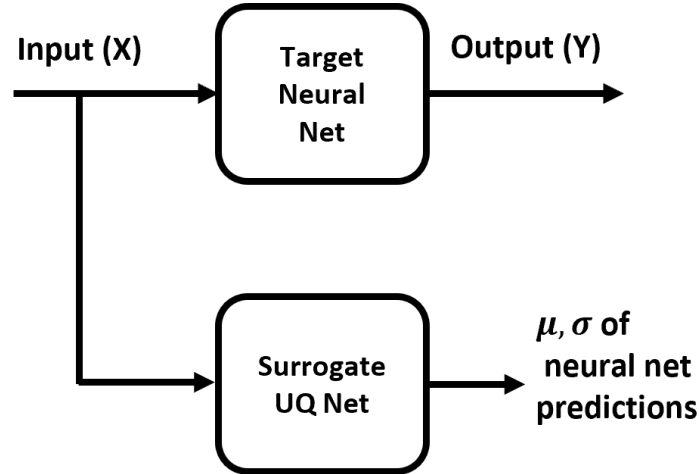


Figure 4.1: Neural network with uncertainty quantification module (UQ-Net)

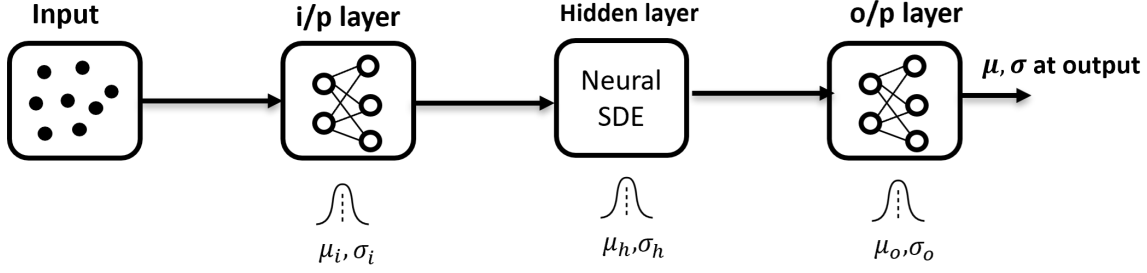


Figure 4.2: UQ-Net

Neural SDE

Traditional neural networks typically consist of multiple stacked hidden layers that map input $\mathbf{x}$ to output $\mathbf{y}$. However, recent research has shown that these hidden representations can be interpreted as the states of a continuous dynamical system rather than discrete layers [61]. For instance, in the case of a residual neural network (a deep learning model in which each layer learns residual functions with respect to their input)[62], the transformation between layers can be expressed as

$$\mathbf{h}_{t+1} = \mathbf{h}_t + \mathbf{f}(\mathbf{h}_t, \theta) \quad (4.2)$$

Here, $\mathbf{h}_t$ represents the hidden features at depth t , and $\mathbf{f}$ is a neural network function approximation. The continuous limit of the residual neural network structure is expressed as

$$\mathbf{h}_{t+1} = \mathbf{h}_t + \int_t^{t+1} \mathbf{f}(\mathbf{h}_\tau, \tau, \theta) d\tau \quad (4.3)$$

The equation (4.3) is the continuous approximation of the transformations in residual neural network architecture and also the solution to an ordinary differential equation (ODE) problem. The neural ordinary differential equation (Neural ODE) method [61] parameterizes $\mathbf{f}(\mathbf{h}_\tau, \tau, \theta)$ with a neural network and leverages an ODE solver to evaluate the hidden states during the continuous transformation. However, the deterministic nature of neural ODEs is not suitable to model epistemic uncertainty. To address this, we employ the neural SDE model [63], which augments a neural ODE with a stochastic term. Thus, neural SDE incor-

porates a diffusion term to model randomness using the Brownian motion component[63]. Thus equation (4.3) can be rewritten as follows:

$$d\mathbf{h}_t = \underbrace{\mathbf{f}(\mathbf{h}_t, t; w)dt}_{\text{drift}} + \underbrace{\mathbf{g}(\mathbf{h}_t, t; v)d\mathbf{B}_t}_{\text{diffusion}} \quad (4.4)$$

Here, $\mathbf{g}(\mathbf{h}_t, t; v)d\mathbf{B}_t$ represents the Brownian motion variance, which helps to capture the epistemic uncertainty of the hidden state $\mathbf{h}_t$ at depth t . The drift and diffusion terms from the equation are functions that are approximated via neural networks with w and v as the corresponding network parameters. Equation (4.4) is a general form that can be modified to incorporate existing randomness models, such as dropout in neural networks.

Dropout using neural SDE: The dropout term randomly deactivates some neurons in the neural network and can be modeled using neural SDE [64]. Mathematically, the dropout rate can be incorporated into equation (4.3) as $\mathbf{h}_{t+1} = \mathbf{h}_t + \mathbf{f}(\mathbf{h}_t, t; w) \odot \frac{\gamma_n}{p}$, where γ_n is drawn from i.i.d Bernoulli distribution with parameter p and $\odot$ indicates the Hadamard product. Furthermore, the above hidden state dynamic can be simplified as,

$$\begin{aligned} \mathbf{h}_{t+1} &= \mathbf{h}_t + \mathbf{f}(\mathbf{h}_t, t; w) + \mathbf{f}(\mathbf{h}_t, t; w) \odot \left(\frac{\gamma_n}{p} - \mathbf{I}\right), \\ &= \mathbf{h}_t + \mathbf{f}(\mathbf{h}_t, t; w) + \mathbf{f}(\mathbf{h}_t, t; w) \odot \sqrt{\frac{1-p}{p}} \mathbf{z}_n. \end{aligned} \quad (4.5)$$

where $\mathbf{z}_n \stackrel{\text{i.i.d.}}{\sim} \mathcal{N}(0, 1)$. To compare the quantified epistemic uncertainty with the state-of-the-art Monte Carlo dropout method, we utilize equation (4.15), which unifies dropouts under the neural SDE framework. Here, $\sqrt{\frac{1-p}{p}}\mathbf{f}(\mathbf{h}_t, t; w)$ represents the diffusion term and $q = \sqrt{\frac{1-p}{p}}$ controls the strength of regularization. It determines the scaling factor of the diffusion term and affects the magnitude of the injected noise or uncertainty. By adjusting the value of q , one can balance the trade-off between regularization and the expressiveness of the model. q can be treated similarly to the dropout probability term in the MC dropout method [58].

UQ-Net

As discussed above, the mean and covariance are computed from the input layer and subsequently propagated through the hidden layer, effectively carrying the uncertainty measures to the output layer with the help of UQ-Net. The input and output layers can be linear or nonlinear functions depending on the presence of an activation function.

Uncertainty estimation in input/output layer: Quantification of uncertainty for a non-linear input/output layer can be achieved by computing the statistics of the linearized approximation of the nonlinear function [35]. Suppose that for a nonlinear function $\mathbf{f}$ and a random vector $\mathbf{x}$ with expected value $\hat{\mathbf{x}}$ and covariance $\mathbf{C}_{xx}$, the expected value $\hat{y}$ of the output $\mathbf{y} = \mathbf{f}(\mathbf{x})$ can be written as a Taylor series expansion around the mean as $\hat{\mathbf{y}} \approx \mathbf{f}(\hat{\mathbf{x}})$

Here $\nabla_x \mathbf{f}$ is the Jacobian of the function $\mathbf{f}$ at the operating point $\mathbf{x}$. Theorem 2 below can be used to evaluate the expected value of the mean and covariance of the output from the input/output layers of the neural network.

Theorem 2. *Let $\mathbf{y} = f(\mathbf{x}, \theta)$ denote the output of a neural network layer f parameterized by weights θ , with input $\mathbf{x}$ with expected value $\hat{\mathbf{x}}$ and covariance $\mathbf{C}_{xx}$. Then, under a first-order linearization of f around $\hat{\mathbf{x}}$, the predicted output mean $\hat{\mathbf{y}}$ and covariance $\mathbf{C}_{yy}$ are given by:*

$$\begin{aligned}\hat{\mathbf{y}} &= f(\hat{\mathbf{x}}, \theta), \\ \mathbf{C}_{yy} &= (\nabla_{\mathbf{x}} f(\hat{\mathbf{x}})) \mathbf{C}_{xx} (\nabla_{\mathbf{x}} f(\hat{\mathbf{x}}))^{\top},\end{aligned}$$

where $\nabla_{\mathbf{x}} f(\hat{\mathbf{x}})$ is the Jacobian of the network layer with respect to the input, evaluated at $\hat{\mathbf{x}}$.

Proof. Assume the input $\mathbf{x}$ with expected value $\hat{\mathbf{x}}$ and covariance $\mathbf{C}_{xx}$. Linearizing the network function f using a first-order Taylor expansion at $\hat{\mathbf{x}}$ gives:

$$f(\mathbf{x}) \approx f(\hat{\mathbf{x}}) + \nabla_{\mathbf{x}} f(\hat{\mathbf{x}})(\mathbf{x} - \hat{\mathbf{x}}).$$

Taking the expectation:

$$\mathbb{E}[f(\mathbf{x})] \approx f(\hat{\mathbf{x}}) = \hat{\mathbf{y}}.$$

And the covariance:

$$\text{Cov}(f(\mathbf{x})) \approx \nabla_{\mathbf{x}} f(\hat{\mathbf{x}}) \mathbf{C}_{xx} \nabla_{\mathbf{x}} f(\hat{\mathbf{x}})^\top = \mathbf{C}_{yy}.$$

Here, the expected value of the input is $\mathbb{E}[\mathbf{x}] = \hat{\mathbf{x}}$ and its associated covariance is $\mathbf{C}_{xx}$ and $\mathbf{C}_{yy}$ is the covariance associated with the output of the neural network layer. Thus, the output uncertainty is obtained by propagating input uncertainty through the Jacobian of the layer. $\square$

The uncertainty estimates obtained using Theorem 2 are propagated through the input layer and are passed to the hidden layer. Theorem 2 can also be applied to uncertainty estimation at the output layer.

Uncertainty estimation in hidden layer: Uncertainty quantification of the hidden layer of the neural network is achieved using Gaussian assumed density approximation of Neural SDE [65]. By utilizing linearization techniques, we can compute the statistics of nonlinear neural SDE. Specifically, the nonlinear neural SDE can be linearized via Taylor series approximation. The drift term can be approximated by linearizing around the mean $\mathbf{m}$ and at depth t as follows:

$$f(\mathbf{h}, t) \approx f(\mathbf{m}, t) + \mathbf{F}_{\mathbf{h}}(\mathbf{h}, t)(\mathbf{h} - \mathbf{m}) \quad (4.6)$$

And the diffusion term can be linearized as:

$$g(\mathbf{h}, t) \approx g(\mathbf{m}, t) + \mathbf{G}_{\mathbf{h}}(\mathbf{h}, t)(\mathbf{h} - \mathbf{m}) \quad (4.7)$$

Here $\mathbf{F}_{\mathbf{h}}$ and $\mathbf{G}_{\mathbf{h}}$ are the Jacobian of f and g with respect to the hidden state $\mathbf{h}$ and the statistics of the hidden state can be calculated using Theorem 2.

Theorem 3. *Let the dynamics of the hidden state $\mathbf{H}_t$ of a neural stochastic differential equation be:*

$$d\mathbf{H}_t = f(\mathbf{H}_t, t) dt + G(\mathbf{H}_t, t) d\mathbf{B}_t,$$

where f is the drift function, G is the diffusion term, and $\mathbf{B}_t$ is standard Brownian motion. Then the evolution of the mean $\mathbf{m}(t)$ and covariance $\mathbf{C}(t)$ is governed by the following system of ordinary differential equations:

$$\frac{d\mathbf{m}}{dt} = f(\mathbf{m}, t), \tag{4.8}$$

$$\frac{d\mathbf{C}}{dt} = \mathbf{C}F_h^\top(\mathbf{m}, t) + F_h(\mathbf{m}, t)\mathbf{C} + G_h(\mathbf{m}, t)\mathbf{Q}G_h^\top(\mathbf{m}, t), \tag{4.9}$$

where $F_h = \nabla_{\mathbf{H}} f$ is the Jacobian of the drift with respect to $\mathbf{H}_t$, $G_h = \nabla_{\mathbf{H}} G$, and $\mathbf{Q}$ is the diffusion matrix.

Incorporating randomness similar to dropout, the diffusion term is defined as:

$$G(\mathbf{H}_t, t) = \sqrt{\frac{1-p}{p}} f(\mathbf{H}_t, t),$$

where p is the dropout probability. Substituting this into equation (4.9) yields a closed-form differential equation for the covariance evolution under randomness similar to dropout.

Proof. The derivation follows from the Gaussian process approximation for SDEs. Linearizing the drift and diffusion terms around the mean trajectory, we obtain the moment equations for nonlinear Neural SDEs:

$$\frac{d\mathbf{m}}{dt} = \mathbb{E}[f(\mathbf{H}_t, t)] \approx f(\mathbb{E}[\mathbf{H}_t], t),$$

$$\frac{d\mathbf{C}}{dt} = \mathbf{C}F_h^\top + F_h\mathbf{C} + G_h\mathbf{Q}G_h^\top.$$

In the case where dropout noise is interpreted as Brownian uncertainty with scaling

$G = \sqrt{\frac{1-p}{p}}f$, we can substitute this into the covariance equation to analytically quantify the output uncertainty across network layers. $\square$

Equations (4.8) and (4.9) enable the analytical determination of the mean and covariance of the hidden layer output in the neural network. It is to be noted that the diffusion term, denoted as g , is proportional to the function $\mathbf{f}$ in our particular context and is defined as $g = \sqrt{\frac{1-p}{p}}f(h_t, t; w)$. Thus, the uncertainty quantification of a deep neural network can be achieved analytically using UQ-Net.

4.1.4 Results

In this section, we conduct a comprehensive evaluation of our proposed uncertainty quantification method through experiments on two synthetic datasets and two real-world datasets: two designed for regression tasks and the other two for classification tasks. Detailed descriptions of these datasets are provided below:

Synthetic Regression dataset: This dataset is generated synthetically and is intended for regression tasks. It comprises of a single feature $\mathbf{x}$ and consists of randomly generated homoscedastic data.

Half Moon dataset: This synthetic dataset consists of two features and is specifically designed for performing classification tasks. Each data point in the dataset can be classified into one of two distinct classes.

Housing dataset: The dataset is drawn from the 1990 U.S. Census reflecting real-world data. It consists 8 distinct features and is utilized to forecast the median house value within California districts [66]

Ionosphere dataset: This real-world dataset consists of 34 feature which classifies radar returns from the ionosphere into three different categories [67].

By utilizing these datasets, we aim to assess the performance and effectiveness of our proposed uncertainty quantification method. To assess the effectiveness of our proposed uncertainty quantification method, we compare its performance against the widely adopted Monte Carlo dropout technique. All training and evaluation experiments are performed on

a computer with an Intel i7 processor running at 2.80 GHz with 12GB memory and 12GB RAM. The learning rate set was 0.001 and the Adam optimizer was used.

Regression task

The regression task is performed on the single feature synthetic regression dataset. The uncertainty estimates obtained using UQ-Net is compared with the MC dropout method using the expected normalized calibration error (ENCE) metric [68]. The ENCE metric serves as an indicator of the reliability of the confidence scores provided by the methods. The ENCE metric is utilized to calibrate the regressor by aligning the mean square error (MSE), representing the expected error, with the predicted uncertainty, denoted by the standard deviation σ . To assess the calibration of the regressor, the standard deviation axis is divided into bins, and the ENCE metric is calculated using the following formula:

$$\text{ENCE} = \frac{1}{N} \sum_{j=1}^N \frac{|\text{RMV}(j) - \text{RMSE}(j)|}{\text{RMV}(j)} \quad (4.10)$$

Here, B_j is a bin that represents the standard deviation axis interval, root mean variance(RMV) and root mean square error (RMSE) are obtained using the following equations:

$$\text{RMV}(j) = \frac{1}{|B_j|} \sum_{t \in B_j} \sigma_t^2 \quad (4.11)$$

$$\text{RMSE}(j) = \frac{1}{|B_j|} \sum_{t \in B_j} (y_t - \hat{y}_t)^2 \quad (4.12)$$

A lower ENCE value indicates a better-calibrated model. To assess the performance of UQ-Net, we calculate the ENCE score for various values of p , and the results are shown in Table 4.1. Similar experiments are conducted for the MC Dropout method, and the corresponding results are also presented in Table 4.1. From the table, it becomes evident that by adjusting the values of p , both methods effectively quantify the epistemic uncertainty and exhibit a similar trend. However, the advantage of UQ-Net lies in its analytical nature, resulting in significantly reduced computational time required to obtain uncertainty estimates

Table 4.1: ENCE scores obtained using two methods for regression and classification task for synthetic dataset

P	Regression		Classification	
	UQ-Net	MC dropout	UQ-Net	MC dropout
0.1	0.76	0.804	0.779	2.685
0.2	0.758	1.608	0.632	1.981
0.3	0.767	0.656	0.693	0.977
0.4	0.747	2.423	0.916	0.992
0.5	0.77	0.3	0.99	0.426
0.6	0.718	0.401	0.607	0.439
0.7	0.246	0.184	0.692	0.638
0.8	0.15	0.492	0.755	0.338
0.9	0.607	0.516	1.343	0.476
Avg	0.613	0.820	0.823	0.994

compared to the empirical MC Dropout approach. The computational time required for both methods is summarized in Table 4.3.

For achieving optimal performance, the commonly chosen dropout value for hidden layers using MC Dropout is 0.5. However, the optimal dropout value may vary depending on various factors such as data size, model architecture, etc. [69]. Similarly, in the case of UQ-Net, there can be an optimal value of q that can be chosen to yield a lower ENCE score, making the choice of hyperparameters crucial for obtaining reliable uncertainty quantification results. We further asses the performance of our framework using real-world data from the California housing dataset. The results are detailed in Table 4.2. The trend observed in Table 4.1 for regression tasks is similarly evident in the evaluation using real-world datasets shown in Table 4.2. The computational time needed for assessing both UQ-Net and MC dropout is illustrated in Table 4.4.

Table 4.2: ENCE scores obtained using two methods for regression and classification task for real-world dataset

P	Regression		Classification	
	UQ-Net	MC dropout	UQ-Net	MC dropout
0.1	0.43	6.14	0.48	12.08
0.2	0.60	4.35	0.72	9.64
0.3	0.57	3.60	0.51	6.75
0.4	0.59	3.02	0.59	4.70
0.5	0.62	2.65	0.50	4.14
0.6	0.63	2.09	0.64	6.21
0.7	0.67	1.72	0.93	3.82
0.8	0.65	1.13	0.38	1.81
0.9	0.61	0.76	1.50	1.29

Table 4.3: Computation time in seconds required for regression and classification task for synthetic dataset

Regression		Classification	
UQ-Net	MC dropout	UQ-Net	MC dropout
0.83 sec	4.88 sec	0.41 sec	6.47 sec

Table 4.4: Computation time in seconds required for regression and classification task for real-world dataset

Regression		Classification	
UQ-Net	MC dropout	UQ-Net	MC dropout
8.79 sec	79.07 sec	0.37 sec	3.46 sec

Classification task

In the classification task, we utilize the half-moon dataset with two features. Just like in the regression experiments, we compare the uncertainty estimates obtained using UQ-Net with those from the MC Dropout method. While the ENCE score is used for evaluating uncertainty in regression tasks, it cannot be applied directly to classification tasks, as the RMSE loss is not relevant for classification. However, classifiers can predict continuous scores that are often transformed into class labels through a thresholding process during the final step of classification. To evaluate uncertainty estimates in our specific framework, we utilize the continuous output prior to this final step. Thus, the ENCE score is suitable for our specific case in evaluating uncertainty estimates for classification tasks.

Similar to the regression task, ENCE scores are calculated for various values of p for both UQ-Net and MC Dropout, and the results are depicted in Table 4.1. As previously discussed, an optimal value of p can be selected based on a lower ENCE score. Similarly, an optimal value of q can also be chosen. The computational time required for both methods is presented in Table 4.3 and similar to the case of regression task there is up to 94% reduction in computation time compared to MC dropout method.

We also evaluate our framework’s performance for classification using real-world data from the ionosphere dataset. The results are presented in Table 4.2. The trend observed in Table 4.1 for classification tasks is also evident in the evaluation using real-world datasets, as shown in Table 4.2. The computational time required for assessing both UQ-Net and MC dropout is provided in Table 4.4.

4.2 UQ in GNN

4.2.1 Related Work

Graph Neural Networks (GNNs) have gained widespread popularity in many applications where data can be represented in a graph structure, such as social networks, transportation networks, and biological networks, proving effective in tasks such as node classification and

regression [70]. Despite their widespread adoption, GNNs inherently exhibit uncertainty in their predictions due to factors such as the randomness in the data, commonly referred to as aleatoric uncertainty and due to limited knowledge about the problem being learned or limited training data, known as epistemic uncertainty [71]. Epistemic uncertainty can be further divided into model uncertainty and distributional uncertainty. Model uncertainty arises from choices related to model structure and hyperparameters, e.g., shallow networks may lead to under confidence in GNNs [72]. Distributional uncertainty stems from incorrect assumptions about the problem setting or data-generating process [72], and is most commonly encountered in inductive inference scenarios. Properly quantifying and managing uncertainty in GNNs is crucial, especially in safety-critical applications such as finance [73], healthcare [74], and security [75], where incorrect predictions can lead to severe consequences. Although several works have addressed the quantification of aleatoric uncertainty in GNNs, our focus in this section is on quantifying epistemic uncertainty. This type of uncertainty is particularly important in applications where trustworthiness and prediction confidence are critical, such as medical diagnosis and financial modeling or in scenarios involving sparse data [76].

Among the most common approach to handle epistemic uncertainty is the Bayesian approach. Bayesian approaches attempt to model uncertainty in parameters by assuming probability distributions over weights or activations. These methods provide a principled approach to uncertainty estimation through posterior inference [77]. While they help mitigate overfitting in small labeled datasets and are effective in detecting out-of-distribution nodes [78], they can become computationally expensive, especially when full Bayesian inference is carried out using Markov Chain Monte Carlo (MCMC) methods [79], which involve iterative posterior sampling and are often infeasible for large-scale graphs [80]. Moreover, while they may be accurate on smaller datasets, their performance can degrade for deep GNN architectures [81]. Bayesian approximation methods, such as Monte Carlo Dropout [82] and variational inference [83], provide uncertainty quantification without maintaining full posterior distributions. These approaches are less computationally intensive and moderately scalable, but they may be unreliable in high-stakes applications due to imprecise

epistemic uncertainty estimates, often leading to overconfident predictions in critical tasks [82]. Ensemble-based methods aggregate predictions from multiple GNN models trained on the same or varied data distributions to generate uncertainty estimates. While they tend to produce consistent and highly reliable estimates of epistemic uncertainty [83], they are computationally demanding, particularly for large graphs and lack the interpretability and theoretical grounding found in Bayesian methods [83]. Conformalized GNNs extend conformal prediction techniques to graphs, offering statistically valid uncertainty intervals for tasks like node classification. They guarantee valid coverage probabilities for uncertainty estimates, making them theoretically sound. However, these methods are heavily reliant on data assumptions. Specifically, the requirement of exchangeability, which may not hold in dynamic graph settings [84]. Recently, Stochastic Differential Equation (SDE)-based models have been used for uncertainty modeling in GNNs. These models track stochastic motion (e.g., Brownian processes) in the latent space of GNNs, enhancing robustness. SDE-based models are mathematically grounded and offer theoretical uncertainty bounds tied to the stability of the underlying dynamical system [85]. However, they still rely on sampling based techniques to estimate the posterior. In safety-critical applications, it is vital to have epistemic uncertainty frameworks that are both theoretically grounded and well-calibrated. In this context, we propose a Graph Neural SDE-based epistemic uncertainty framework that enables tracking of uncertainty in the predictions of a target graph neural network.

4.2.2 Contribution

The contributions of the proposed UQ framework for GNN are as follows:

- We propose a surrogate method based on a Graph Neural Stochastic Differential Equation (SDE) that incorporates randomness, similar to dropout—into Graph Neural Networks (GNNs).
- The approach analytically quantifies uncertainty through estimation of the mean and variance of the target GNN’s predictions.

- We also derive a Bayesian approximation of the proposed approach, providing analytically tractable framework for modeling epistemic uncertainty in GNNs
- The proposed method outperforms state-of-the-art approaches such as Monte Carlo (MC) Dropout in uncertainty estimation, as demonstrated through various evaluation metrics.
- The framework is task-agnostic and can be applied to both regression and classification tasks.

Background

Stochastic Differential Equations (SDEs) are mathematical models that describe continuous-time dynamic systems influenced by stochastic noise. Unlike deterministic differential equations, SDEs introduce random perturbations (e.g., Brownian motion) to model uncertainties or unknown influences in a system. In machine learning, SDE-based formulations have gained interest for modeling uncertainty [86], capturing stochastic evolution in data, and enabling probabilistic reasoning. Notably, Neural SDEs extend ordinary differential equation-based neural networks (ODE-Nets) by incorporating stochastic diffusion terms, offering a principled way to estimate uncertainty [86], [87]. Graph Neural Stochastic Differential Equations (GN-SDEs) [85] integrate the uncertainty modeling capabilities of SDEs with the representational power of GNNs, leading to the deterministic propagation of features through the network (e.g., message passing between nodes). Diffusion term, that introduces random noise into feature transformations (e.g., through Brownian motion), allows for epistemic uncertainty quantification. This formulation enables GN-SDEs to capture uncertainty in both feature space and graph topology, making them particularly useful for epistemic uncertainty quantification. For convenience, consider a typical Graph Convolutional Network (GCN) layer, which can be defined as:

$$\mathbf{H}^{(t)} = \sigma \left(\hat{\mathbf{A}} \mathbf{H}^{(t-1)} \mathbf{W}^{(t-1)} \right), \quad (4.13)$$

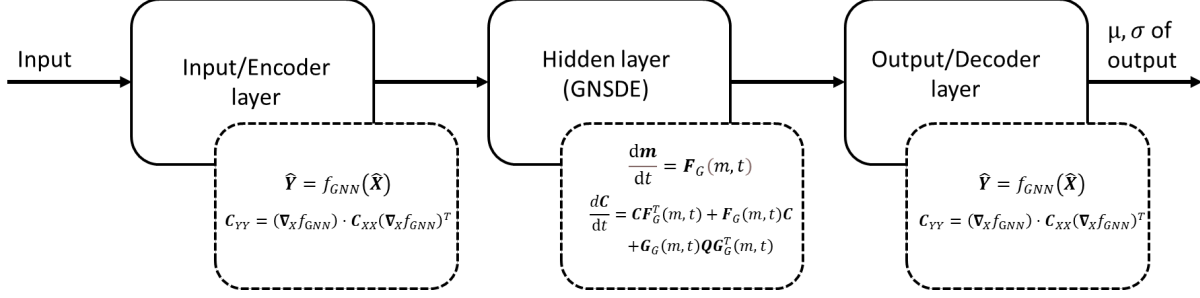


Figure 4.3: Block diagram of GNSDE-UQ Framework

Where, $\mathbf{H}^{(t)}$ is the node feature matrix at depth/layer t (with $\mathbf{H}^{(0)} = \mathbf{X}$ as the input), $\hat{\mathbf{A}}$ is the normalized adjacency matrix (often computed as $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$ with self-loops, followed by a symmetric normalization), $\mathbf{W}^{(t-1)}$ is the learnable weight matrix for depth/layer $t-1$, σ is an elementwise nonlinearity (activation) (e.g., ReLU, Tanh). From [88], the stochastic latent evolution of the node features in a graph structured data can be represented as:

$$d\mathbf{H}(t) = \mathbf{F}(\mathbf{H}(t), t)dt + \mathbf{G}(\mathbf{H}(t), t)d\mathbf{W}(t) \quad (4.14)$$

Here $\mathbf{F}$ and $\mathbf{G}$ are the drift and diffusion functions parameterized by the neural network and $d\mathbf{W}(t)$ is the Wiener process. Given a training dataset $D = (x_j, y_j)_{j=1}^N$, where x_j represents the input data and y_j corresponds to the corresponding ground truth labels, we train a GCN model $\mathbf{M}$ parameterized by θ to make predictions on new test cases. The predictive uncertainty in the GNN originates from two sources as described earlier. Our goal is to analytically quantify the epistemic uncertainty of the GNN model with the help of the surrogate GN-SDE framework that will be explained further.

4.2.3 Proposed GNSDE-UQ approach

The steps involved in the proposed approach for UQ in GNN is illustrated in Figure 4.3.

Uncertainty quantification in GNN SDE with dropout

We start by extending the idea of incorporating dropout in neural SDEs to graph neural SDE. The dropout term randomly deactivates some neurons in the graph neural network and can be modeled using graph neural SDE, similar to neural SDE [64]. Mathematically, the dropout rate can be incorporated into equation (4.14) as $\mathbf{H}(t) = \mathbf{H}(t-1) + \mathbf{F}_G(\mathbf{H}(t), t)dt \odot \frac{\gamma_n}{p}$, where γ_n is drawn from i.i.d Bernoulli distribution with parameter p and $\odot$ indicates the Hadamard product. Furthermore, the above hidden state dynamic can be simplified as,

$$\begin{aligned}\mathbf{H}(t) &= \mathbf{H}(t-1) + \mathbf{F}_G(\mathbf{H}(t), t)dt + \mathbf{F}_G(\mathbf{H}(t), t) \odot \left(\frac{\gamma_n}{p} - \mathbf{I}\right), \\ &= \mathbf{H}(t-1) + \mathbf{F}_G(\mathbf{H}(t), t) + \mathbf{F}_G(\mathbf{H}(t), t) \odot \sqrt{\frac{1-p}{p}} \mathbf{z}_n.\end{aligned}\tag{4.15}$$

Propagating variance through the network

To estimate uncertainty in the predictions of a GNN, we employ a surrogate GNN-SDE model that preserves the structural and dimensional properties of the original GNN. Since SDE layers maintain the same dimensionality, encoder and decoder layers (input/ output layers) are introduced to match the input and output dimensions of the target network. The overall variance is propagated analytically from the input through the hidden GN-SDE layers and finally to the output.

Linear Transformation of Variance

Following Theorem 2 from the UQ-Net section, we extend the linear transformation of variance to graph convolutional layers (the input/output layers) . Given a graph convolutional function f_{GNN} and an input $\mathbf{X} \sim \mathcal{N}(\hat{\mathbf{X}}, \mathbf{C}_{\mathbf{X}\mathbf{X}})$, the output statistics are computed as:

$$\hat{\mathbf{Y}} = f_{\text{GNN}}(\hat{\mathbf{X}}),\tag{4.16}$$

$$\mathbf{C}_{\mathbf{Y}\mathbf{Y}} = (\nabla_{\mathbf{x}} f_{\text{GNN}}) \cdot \mathbf{C}_{\mathbf{x}\mathbf{x}} \cdot (\nabla_{\mathbf{x}} f_{\text{GNN}})^\top \quad (4.17)$$

The above equation allows for covariance propagation through encoder/decoder layers.

Gaussian Assumed Density Filtering in GNN-SDEs

Within the GNN-SDE layers, variance is propagated using Gaussian assumed density filtering, which linearizes the drift and diffusion functions. Extending Theorem 2:

$$\frac{d\mathbf{m}}{dt} = \mathbf{F}_G(\mathbf{m}, t) \quad (4.18)$$

$$\frac{d\mathbf{C}}{dt} = \mathbf{C}\mathbf{F}_G^\top(\mathbf{m}, t) + \mathbf{F}_G(\mathbf{m}, t)\mathbf{C} + \mathbf{G}_G(\mathbf{m}, t)\mathbf{Q}\mathbf{G}_G^\top(\mathbf{m}, t) \quad (4.19)$$

Here, $\mathbf{m}$ and $\mathbf{C}$ represent the mean and covariance of the hidden state matrix $\mathbf{H}(t)$ across all nodes in the graph. $\mathbf{Q}$ is the diffusion matrix determined by the dropout noise scale. The linearized terms $\mathbf{F}_G$ and $\mathbf{G}_G$ are computed per graph convolutional block.

This formulation provides a principled, analytical framework for estimating epistemic uncertainty in GNNs using surrogate GNN-SDE models. It avoids the need for expensive Monte Carlo sampling and integrates seamlessly with the GCN architectures.

Bayesian Approximation of Graph Neural SDE

In this section, we present a Bayesian approximation of the Graph Neural SDE (GNSDE) framework with dropout-induced noise. In the context of deep neural networks, [58] interpreted dropout as approximate Bayesian inference in deep Gaussian processes. This leads to a loss function of the form:

$$\mathcal{L}_{\text{GP-MC}} \propto \frac{1}{N} \sum_{n=1}^N \frac{-\log P(y_n | x_n, \hat{\omega}_n)}{\tau} + \sum_{i=1}^L \left(\frac{P_i l^2 \|M_i\|_2^2 + \|m_i\|_2^2}{2\tau N} \right) \quad (4.20)$$

Here, $\hat{\omega}_n$ denotes the sampled model weights under dropout, τ is a precision-related

parameter, and M_i , m_i are the weight matrices and biases for the i -th layer respectively. The loss combines a negative log-likelihood data term and a regularization term that resembles a prior over the model parameters. For the specific case of a single-hidden-layer neural network, this approximation corresponds to a loss function of the form:

$$L_{\text{dropout}} = \frac{1}{N} \sum_{i=1}^N E(y_i, \hat{y}_i) + \lambda \sum_{i=1}^L (\|w_i\|_2^2 + \|b_i\|_2^2) \quad (4.21)$$

In this expression, $E(y_i, \hat{y}_i)$ denotes the error between the true and predicted outputs and is given by:

$$E(y_n, \hat{y}(x_n, \hat{\omega}_n)) = \frac{-\log P(y_n | x_n, \hat{\omega}_n)}{\tau}$$

The regularization strength λ is a function of the dropout rate and controls the model complexity through an L_2 penalty on weights and biases. We now extend this concept to the GNSDE framework, where node features in a graph evolve over a continuous depth dimension via stochastic differential equations using Theorem 4

Theorem 4. *Let $\mathbf{H}_t$ denote the evolving hidden node representations governed by a Graph Neural Stochastic Differential Equation. Suppose dropout is applied with drop probability rate p to the drift function $f(\mathbf{H}_t, t; w)$. Then, the approximate posterior over the node features $\mathbf{H}_t$ under a variational Gaussian assumption is given by:*

$$q(\mathbf{H}_t) \sim \mathcal{N}(\mu_q, \Sigma_q)$$

and the variational loss objective becomes:

$$\mathcal{L}_{\text{GN-SDE}} = \mathbb{E}_q[\log P(D | \mathbf{H}_t)] + \frac{1}{2} \left(\frac{p}{1-p} \right) \int_0^T (\mu_p - \mu_q)^\top (ff^\top)^{-1} (\mu_p - \mu_q) dt$$

where μ_q is the posterior mean, μ_p is the prior mean, and $\frac{1-p}{p} ff^\top$ is the dropout-induced covariance.

Proof. We begin with the stochastic differential equation:

$$d\mathbf{H}_t = f(\mathbf{H}_t, t; w) dt + G(\mathbf{H}_t, t; w) d\mathbf{B}_t$$

Under dropout-based noise, we approximate $G(\cdot)$ as:

$$G(\mathbf{H}_t, t) = \sqrt{\frac{1-p}{p}} f(\mathbf{H}_t, t)$$

Assume a variational posterior over $\mathbf{H}_t$ of the form $q(\mathbf{H}_t) \sim \mathcal{N}(\mu_q, \Sigma_q)$ and prior $P(\mathbf{H}_t) \sim \mathcal{N}(\mu_p, \Sigma_p)$. The KL divergence becomes:

$$\text{KL}(q\|p) = \frac{1}{2} \int_0^T \left[\log \frac{|\Sigma_p|}{|\Sigma_q|} - d + \text{tr}(\Sigma_p^{-1} \Sigma_q) + (\mu_p - \mu_q)^\top \Sigma_p^{-1} (\mu_p - \mu_q) \right] dt$$

Using the dropout-induced approximation:

$$\Sigma_q \approx \frac{1-p}{p} f f^\top$$

and assuming $\Sigma_q = \Sigma_p$ (covariance matching for bounded KL), the KL simplifies to:

$$\text{KL}(q\|p) = \frac{1}{2} \left(\frac{p}{1-p} \right) \int_0^T (\mu_p - \mu_q)^\top (f f^\top)^{-1} (\mu_p - \mu_q) dt$$

To avoid divergence in the KL term, we align the prior and posterior covariances by setting $\Sigma_p = \Sigma_q$, which simplifies the loss to:

$$L = \mathbb{E}_q[\log P(D | \mathbf{H}_t)] + \frac{1}{2} \int_0^T \left(\frac{p}{1-p} \right) (\mu_p - \mu_q)^\top (f f^\top)^{-1} (\mu_p - \mu_q) dt$$

This final expression provides a Bayesian objective that balances data loss with a regularization term. The regularization is scaled by the inverse of the diffusion term, which itself is governed by the dropout rate. This formulation offers an analytically tractable framework for modeling epistemic uncertainty in GNNs via GNSDEs.

□

4.2.4 Results

In this section, we evaluate the effectiveness of our proposed uncertainty quantification method using GNSDE-UQ (Graph Neural Stochastic Differential Equation based Uncertainty Quantification) framework. We assess performance on both regression and classification tasks using two real-world datasets and compare our method against the widely used Monte Carlo (MC) dropout approach. For regression and classification we use graph datasets from PyTorch Geometric library [89].

To evaluate uncertainty calibration for regression, we use the Expected Normalized Calibration Error (ENCE). For classification tasks, we evaluate predictive performance under uncertainty using the Area Under the Receiver Operating Characteristic curve (AUROC) and the Area Under the Precision-Recall curve (AUPR).

The ENCE metric quantifies how well the predicted variance corresponds to the observed errors in a regression task. A lower ENCE value indicates better calibration and more reliable uncertainty estimates. ENCE is computed as:

$$\text{ENCE} = \frac{1}{N} \sum_{j=1}^N \frac{|\text{RMV}(j) - \text{RMSE}(j)|}{\text{RMV}(j)} \quad (4.22)$$

Here, $\text{RMV}(j)$ and $\text{RMSE}(j)$ denote the root mean variance and root mean square error within bin B_j , respectively:

$$\text{RMV}(j) = \frac{1}{|B_j|} \sum_{t \in B_j} \sigma_t^2, \quad \text{RMSE}(j) = \frac{1}{|B_j|} \sum_{t \in B_j} (y_t - \hat{y}_t)^2 \quad (4.23)$$

For classification, we use AUROC and AUPR to evaluate the quality of uncertainty estimates in distinguishing between correct and incorrect predictions. These metrics assess how well the model’s predictive confidence separates high-certainty (correct) predictions from low-certainty (incorrect or uncertain) predictions. A higher AUROC or AUPR indicates better discriminative power and thus better-calibrated uncertainty [85].

Comparison with MC Dropout: We compare the GNSDE-based uncertainty quantification method against the MC Dropout approach. While both methods aim to cap-

ture epistemic uncertainty, the key difference lies in their computation: GNSDE-UQ offers an analytical uncertainty propagation framework, whereas MC Dropout relies on repeated stochastic forward passes to empirically estimate uncertainty. Additionally, we evaluate the performance of the Bayesian approximation of the GNSDE framework (BGNSDE).

Table 4.5 reports the ENCE scores and test and train losses for both methods on the regression and classification tasks. It can be observed that BGNSDE, followed by GNSDE-UQ consistently achieves lower ENCE values compared to MC Dropout across all tested values of p , indicating superior uncertainty calibration. It is important to note that the dropout probability parameter p has different interpretations in MC Dropout and GNSDE-UQ contexts, and a value of $p = 0.1$ in one does not correspond directly to the other. Nevertheless, across the entire set of evaluated p values, GNSDE-UQ and BGNSDE never underperform relative to MC Dropout, further validating the efficiency of our proposed method.

Table 4.6 presents the AUROC and AUPR scores for classification tasks using the GNSDE, MC Dropout, and Bayesian GNSDE (BGNSDE) frameworks. All models were trained on the Cornell graph from the PyTorch Geometric WebKB dataset and evaluated on the Texas graph from the same dataset. This experimental setup is designed to assess model performance under both in-distribution (Cornell) and out-of-distribution (Texas) conditions. It should be noted that both graphs have an equal number of nodes.

Across all tested values of dropout probability p , GNSDE consistently achieves the highest AUROC and AUPR scores, indicating superior capability in capturing predictive uncertainty, in the presence of epistemic uncertainty (uncertainty due to out of distribution data used here). BGNSDE follows closely, outperforming the MC Dropout baseline in most cases. These results highlight the effectiveness of proposed methods in modeling epistemic uncertainty in classification tasks.

	GNSDE-UQ			MC dropout			BGNSDE		
p	Train Loss	Test Loss	ENCE	Train Loss	Test Loss	ENCE	Train Loss	Test Loss	ENCE
0.1	0.0213	0.2345	0.4311	0.0306	0.1401	2.331	0.0727	0.0664	0.2863
0.3	0.0454	0.1903	0.2988	0.0328	0.1457	2.701	0.0634	0.0858	0.1439
0.5	0.0523	0.1723	0.3934	0.034	0.145	2.8351	0.0646	0.0853	0.0234
0.7	0.0386	0.174	0.3301	0.0405	0.1686	3.798	0.0741	0.0702	0.0764
0.9	0.0476	0.1792	0.4389	0.0607	0.0892	0.9277	0.073	0.0668	0.188
Avg	0.378			2.518			0.143		

Table 4.5: ENCE scores for GNSDE-UQ, BGNSDE and MC Dropout across varying dropout probabilities p for regression

	GNSDE-UQ		MC dropout		BGNSDE	
p	AUROC	AUPR	AUROC	AUPR	AUROC	AUPR
0.1	0.5748	0.5571	0.5096	0.4705	0.4644	0.41
0.3	0.5136	0.4482	0.4519	0.3882	0.521	0.4465
0.5	0.481	0.5385	0.4496	0.4044	0.5511	0.6119
0.7	0.4906	0.5884	0.4477	0.3833	0.511	0.4487
0.9	0.5549	0.5918	0.4938	0.5198	0.4956	0.4437
Avg	0.522	0.544	0.470	0.433	0.508	0.472

Table 4.6: Performance of proposed approach and comparison with MC dropout for classification

4.3 Summary

In this chapter, we propose a novel approach to quantifying epistemic uncertainty in target neural networks, including both deep neural networks and graph neural networks. The framework is generic, task-agnostic, and leverages neural stochastic differential equations as a surrogate model which is a continuous representation of the neural network, that enables the analytical quantification of epistemic uncertainty. While Monte Carlo Dropout provides a useful approximation of uncertainty, we demonstrate that the proposed approach produces better-calibrated probability scores, improving the reliability of model correctness.

In the next chapter, we explore how machine learning can be applied in limited-data settings and how uncertainty can be leveraged to make meaningful predictions.

Chapter 5

Scientific Machine Learning with Uncertainty Quantification

In this chapter, we explore strategies to prevent overfitting and enhance the generalizability of machine learning models in scenarios where only limited labeled data is available. We present two case studies demonstrating how incorporating prior information in the form of learning and observational biases can improve the inferencing capabilities of these frameworks. Additionally, we integrate uncertainty quantification methods to ensure model reliability, making them suitable for safety-critical applications, such as those in biological systems. By embedding domain knowledge and uncertainty metrics into the learning process, our approach not only enhances performance but also increases confidence in model predictions under challenging data conditions. Finally, we demonstrate how the knowledge of uncertainty can be leveraged to analytically estimate certain parameters that cannot be directly measured.

5.1 Case Study 1: Bayesian Physics-Informed Machine Learning for Thermal Ablation

5.1.1 Related Work

Thermal ablation is a clinically accepted therapeutic technique for treating unresectable tumors located in the liver, lungs, and other organs. During an ablation procedure, a device (known as an applicator) is inserted into the tumor under image-guidance, and once appropriately positioned, delivers power from an external generator and transfers it to tissue. The power is locally absorbed in tissue, leading to heating. Cell death by coagulative necrosis is a function of time and temperature, and occurs relatively quickly at temperatures exceeding $\sim 55^{\circ}\text{C}$ [90]. Among the available energy sources for ablation, microwave ablation (MWA) delivered at 915 MHz or 2.45 GHz is a commonly used energy modality [91]. Ablation durations are typically on the order of 5-15 mins depending on size of the targeted tumor. The goal of the procedure is to cover both the tumor and a 5-10 mm margin [92] of surrounding tissue with an ablative thermal dose to ensure the technical success of the ablation procedure [93],[94].

A significant challenge during ablation is the lack of accessible, reliable methods for real-time monitoring of ablation growth, which is informative for clinicians to decide when to terminate the treatment [95]. The temperature monitoring devices that can potentially help in making the above decisions fall into two categories. The first are point-based methods, which use needle-based sensors to measure temperature or other parameters at selected points within the tissue [96]. While point-based sensors are affordable and straightforward to use, their placement might be challenging and increase invasiveness, providing data in a limited number of measurement points, rather than a 3D distribution. The second category comprises imaging-based techniques, such as Magnetic Resonance Thermometry (MRT), Ultrasound-based thermometry and CT thermometry which provides spatially detailed measurements [97], though spatial coverage is often limited to select 2D planes (resolution of $1\text{mm} \times 1\text{mm}$)[98]. Despite its potential, MRT is costly, can be complex to set up, and

requires ablation instrumentation compatible with the MRI environment, resulting in significant practical limitations.

Recently, computational modeling of the microwave ablation have been instrumental in guiding the design and evaluation of the ablation devices performance. These models help assess how varying energy delivery parameters, such as applied power, impact electromagnetic power absorptions and thermal damage within tissues during microwave ablation. This typically involves solving the Helmholtz equation to determine the electric field within tissue and the bioheat transfer equation to map transient temperature profiles. The fully coupled model estimates how electromagnetic and thermal profiles evolve dynamically during ablation, by updating the tissue electrical and thermal properties and recalculation of the electric field at each time step of the transient heat transfer solver [99],[100].

One key strength of these models is their potential to incorporate patient-specific data such as tumor type (which can influence tissue properties) [101]. Tissue properties, both electrical and thermal, can theoretically be inferred from imaging data or other measurements. However, this has not yet been demonstrated in the literature. Spatial variations in these properties can also be estimated from imaging, making it feasible to adapt models for individualized treatment planning [102],[103, 104]. However, there exist several barriers to use of modeling approaches for guiding ablation procedures in the clinical setting. Nonlinearities in tissue properties pose a significant challenge, as these properties change considerably during heating, varying as a function of both temperature and the rate of heating [95]. Additionally, at high temperatures, other physical phenomena such as water vaporization, bubble formation, change in blood viscosity and tissue shrinkage may occur, but current models present limitations in fully describing these biophysical modifications of the tissue. Finally, computational models for microwave ablation are computationally expensive, particularly when fully coupled approaches are employed to capture the intricate physical processes involved [105].

Recently, machine learning based frameworks have been introduced to address such challenges [106],[107]. However, these data-driven approaches often require large volumes of training data and lack generalizability, particularly for tasks like modeling temperature dy-

namics in thermal ablation and capturing complex underlying processes. Physics-informed machine learning integrates physical laws or constraints into the learning process, combining the strengths of data-driven models with the theoretical rigor of computational physics [108],[109]. This approach guides the neural network to learn solutions that not only fit the limited observed data but also comply with fundamental physical principles. However, challenges remain while incorporating physics equations into machine learning models, particularly when there are uncertainties associated with the parameters of the physics equation. Estimating these parameters, known as solving the inverse problem, is a complex task. One approach for solving the inverse problem falls under data-driven methods, where both the predictive model and unknown parameters are learned simultaneously [110]. However, this approach requires large volumes of training data, which may not be available in real-world scenarios. Another method falls under the Bayesian approach, where a prior distribution is assumed for the uncertain parameters, and the posterior distribution is estimated based on predictions from the forward predictive model [111]. While effective, this method can be computationally expensive and may require sampling-based techniques such as Markov chain Monte Carlo method [112].

5.1.2 Contributions

In this work, we propose a Bayesian machine learning framework that aims to achieve the following: given a subsample of temperature measurements obtained during microwave ablation across a 2D plane at different time points, the framework learns the temperature distribution as a function of time and 2D space. This approach can then be used to forecast the temperature profile of the tissue within the specified region of interest and at desired time points (e.g., up to 332.8 seconds after the start of heating), achieving a 19.70% reduction in mean absolute error compared to the baseline classic PINN models. Thus, the framework is suitable for scenarios where data acquisition for monitoring tissue heating profiles is challenging. Moreover, it is crucial to estimate the values of uncertain parameters within the physics equation incorporated into the neural network that influences the heating

process with the available data. This estimation can possibly serve as an initial step toward developing patient-specific treatment planning. The key contributions of our work include:

- We propose a Bayesian physics-informed neural network (BPINN) framework that captures temperature dynamics in tissues during thermal ablation and can be used to make spatio-temporal predictions.
- The proposed framework enhances reliability by quantifying the uncertainty associated with model predictions using analytical approaches.
- We introduce an analytical method for estimating, any underlying uncertain parameters associated with the biophysics model and are crucial for understanding the heating dynamics.
- We demonstrate the generalizability of our proposed framework on a real-world dataset, achieving a performance improvement of 23% over the baseline (vanilla neural network and classic PINN).
- Our approach can also be extended to similar problems in thermal ablation studies or other applications, with adjustments to the physics equations and datasets used.

5.1.3 Background

Computational modeling of Microwave ablation in tissues

MWA uses electromagnetic power (typically at 915 MHz or 2.45 GHz) delivered to the tissue via an applicator to generate heat and induce coagulative necrosis in target tissues [99], [113]. Thus, it is typically modeled as a two-part process involving both electromagnetic wave propagation and heat transfer dynamics within tissue. The propagation of microwave energy in tissues is described by the Helmholtz equation, which is derived from Maxwell's equations and can be expressed as:

$$\nabla \times (\nabla \times \mathbf{E}) - k^2 \mathbf{E} = 0 \quad (5.1)$$

where $\mathbf{E}$ is the electric field vector, and k is the complex wave number. For tissue, k is defined as: $k = \omega\sqrt{\mu\varepsilon_c}$ where ω is the angular frequency, μ is the magnetic permeability, and $\varepsilon_c = \varepsilon - j\frac{\sigma}{\omega}$ is the complex permittivity that accounts for both the permittivity ε and the conductivity σ of the medium. The power absorbed in tissue is proportional to $|\mathbf{E}|^2$, which serves as the heat source in the next step of the modeling. The thermal response of tissue to the absorbed microwave energy is then described by the bioheat transfer equation. In unperfused ex-vivo (the setting considered in this study), the thermal response of tissue to the absorbed microwave power is then described by the heat equation:

$$\rho c_p \frac{\partial T}{\partial t} = \nabla \cdot (k \nabla T) + Q, \quad (5.2)$$

where ρ is the tissue density, c_p is the specific heat capacity, T is the temperature, k the thermal conductivity $Q = \frac{1}{2}\sigma|\mathbf{E}|^2$ is the heat generated by microwave absorption and σ is the electrical conductivity of the tissue. Since both electrical and thermal properties of tissue are temperature-dependent, this problem is coupled and nonlinear. A common approach to solving this system is to iteratively alternate between the two parts: first, solve for the electric field distribution (1), then update the transient temperature profile by solving the bioheat equation (2)) over a short time step, typically 5–10 s. After this, both the electrical and thermal properties of the tissue are updated according to the new temperature distribution, and the process is continued iteratively by resolving (1) and (2).

Physics informed neural networks (PINNs)

Physics informed machine learning is a class of deep learning architectures that seamlessly integrate prior knowledge into the neural network framework. Real-world data may be imperfect due to noise, missing information, and other challenges. Learning and making inferences with such imperfect data is challenging, as it makes the machine learning model less generalizable and less interpretable. A solution to this challenge is to integrate prior knowledge that create bias into the machine learning framework, ensuring that the model's predictions follow certain physical constraints. Physics informed machine learning was first

introduced as a surrogate for partial differential equations and has since expanded to various fields of study such as bioengineering, fluid mechanics and so on [109]. The major components of a learning model are the data, the model, and the loss function, and accordingly, the biases can be of three types [114]:

- **Observational Bias:** The data can be modified to embody the underlying physics.
- **Inductive Bias:** The model can be partitioned and trained to satisfy the underlying physics, incorporating prior assumptions and physical laws expressed as mathematical constraints.
- **Learning Bias:** The loss function can be modified to incorporate the physics constraints, thus integrating data and physical models in partially understood contexts, making it suitable for handling missing data or noisy boundary conditions, and scaling to large problems.

Incorporating these biases into the machine learning model in the form of observational bias, inductive bias, or learning bias can result in predictions that are physically consistent. In this work, our focus is on developing physics-informed machine learning that incorporates learning bias into the model. As opposed to the traditional machine learning models, where the loss function only includes the discrepancy between the true values and the predictions made by the machine learning models, the loss term here also includes physical or thermodynamic constraints. The combined loss associated with the data and the constraints is optimized while learning the neural network parameters.

Bayesian inverse problems

The inverse problem, also known as calibration, involves estimating parameters of a system from observed effects or data. In a forward problem, the goal is to determine the outcome from known parameter or causes, mathematically represented as $y = F(\theta)$, where y is the observed data, θ represents the parameters, and F is the forward model describing the relationship between the parameters and the data. The inverse problem is the reverse of the

forward problem where given the observed data y and the forward model F , the goal is to estimate the unknown parameter θ , i.e. $\theta = F^{-1}(y)$. In practice, the inverse problems are complex to solve for various reasons. These include (1) insufficient information to obtain a unique solution; (2) noisy and sparse nature of real-world data, and (3) computationally expensive as the forward modeling must be carried out thousands of times [115]. Inverse problems are generally formulated as search problems and are traditionally solved using the least square method [116]. Recently, neural networks have been leveraged as an inversion technique to solve the inverse problems, where uncertain parameters are updated to satisfy both the governing laws and the boundary conditions. As mentioned previously, existing literature on handling inverse problems using neural networks can be broadly classified into two categories: (i) Learning-based [117],[109], [118] and (ii) Bayesian approaches [111],[119],[120]. The learning-based approach involves treating the uncertain physics parameters as a trainable neural network parameter. However, this technique requires a large number of training points and can result in incorrect fits. Conversely, the Bayesian approach assumes prior information about the uncertain parameters and estimates the posterior distribution of the parameters conditioned on the observation. Although this method can be used with sparse observational data, calculating the posterior distribution can sometimes become intractable. It relies on sampling-based techniques like Markov Chain Monte Carlo to approximate the posterior distribution. However, the sampling process becomes computationally expensive [121]. Though various sampling techniques reduce the computational burden, it still becomes challenging as the solution space grows. Currently, there are no analytical approach for approximating the posterior to further reduce the computational burden in calculating the posterior. In this chapter, we propose a novel approach where we derive an analytical method that quantifies the uncertainty in the output, use this information to update the posterior, and improve the performance of the neural network prediction.

5.1.4 Problem Formulation

We considered microwave ablation in ex vivo bovine liver tissue with 30 W power applied at 2.45 GHz for 10 min. MR thermometry data were available at a temperature resolution of 6.4 s with 1.0156 mm x 1.0156 mm in-plane resolution and 1.5 mm slice thickness. Further experimental details are provided in [122]. We extract the change in temperature at a specific time instance t and the corresponding x, y, z locations. Thus, our dataset consists of the spatio-temporal points (x, y, z, t) and the prediction will be ΔT , the change in temperature at the corresponding spatio-temporal points. This will serve as the dataset for our machine learning model. The computational model used in [123] will serve as the learning bias and is given in equation (5.2). According to the experimental data from previous studies, density and thermal conductivity have little variation from the baseline as function of the temperature. Thus, it is expected that, compared to other biophysical changes such as those in electrical conductivity, permittivity are less impactful on the ultimate distribution of the temperature. It is important to note that the exact values of some coefficients in equation (5.2) are uncertain (in equation (5.2) the exact value of the electromagnetic energy absorbed by the tissue Q , volumetric heat capacity ρc and the tissue conductivity k is generally unknown and might vary for different types of tissues). Therefore, given the limited data from the MRI and the computational physics model with some uncertain parameters, the goal is to develop a predictive model that can accurately predict the temperature profile of the tissue during thermal ablation over the course of 10 minutes long ablation procedure.

5.1.5 Proposed approach

In this chapter, our goal is to learn the temperature profile of tissues using sparse MRI data and underlying physics information represented by a partial differential equation (PDE) with some uncertain parameters. As illustrated in Figure 5.1, given the MRI images of the tissue section being thermally ablated and the computational physics equation governing the temperature variations in the tissue, our objective is to train a neural network model that predicts the temperature at any given spatial location, time instance, and the values

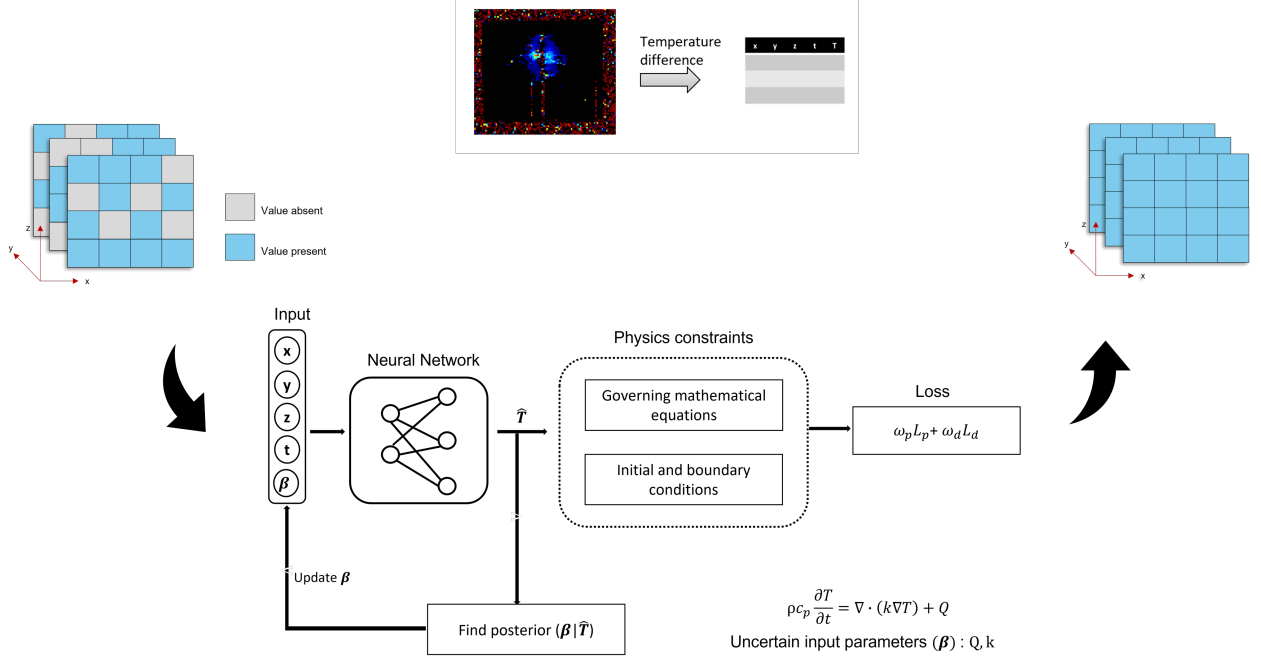


Figure 5.1: Architecture for Bayesian physics informed neural network

of the PDE coefficients as given in equation (5.2). Thus, there are two key problems to address in this setup: (i) the forward problem, where we aim to predict the temperature profile given input values (which include the co-location points and some uncertain physics parameters), and (ii) the inverse problem, where we estimate the values of the uncertain physics parameters based on the given temperature values and the predictive model. The solution of the PDE, denoted as $\hat{\mathbf{u}}(\boldsymbol{\alpha}, \boldsymbol{\beta})$, is approximated by a neural network as $\hat{\mathbf{u}}(\boldsymbol{\alpha}, \boldsymbol{\beta}; \boldsymbol{\Theta})$, where $\boldsymbol{\alpha}$ is the colocation points $(\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{t})$, $\boldsymbol{\beta}$ is the uncertain physics parameters and $\boldsymbol{\Theta}$ represents the neural network parameters obtained by minimizing the following loss function:

$$\boldsymbol{\Theta}^* = \underset{\boldsymbol{\Theta}}{\operatorname{argmin}} w_p \mathbf{L}_p(\boldsymbol{\Theta}, \boldsymbol{\alpha}, \boldsymbol{\beta}) + w_d \mathbf{L}_d(\boldsymbol{\Theta}, \boldsymbol{\alpha}, \boldsymbol{\beta}) \quad (5.3)$$

where, $\mathbf{L}_p$ is the loss associated with the physics model captured via the PDE in equation (5.2) and $\mathbf{L}_d$ is the loss associated with the data and are calculated as follows:

$$\mathbf{L}_p(\boldsymbol{\Theta}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{1}{N} \sum_{i=1}^N \|PDE(\boldsymbol{\alpha}, \boldsymbol{\beta}, \hat{\mathbf{u}}_i(\boldsymbol{\alpha}, \boldsymbol{\beta}))\|_2^2 \quad (5.4)$$

$$\mathbf{L}_d(\boldsymbol{\Theta}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{1}{N} \sum_{i=1}^N \|\mathbf{u}_i(\boldsymbol{\alpha}, \boldsymbol{\beta}) - \hat{\mathbf{u}}_i(\boldsymbol{\alpha}, \boldsymbol{\beta})\|_2^2 \quad (5.5)$$

w_d and w_p are the weights associated with each of the loss terms and N is the number of co-location points.

Our proposed BPINN framework is outlined in Algorithm 2. The input to the BPINN consists of $\boldsymbol{\alpha}$, which represents the colocation points, and $\boldsymbol{\beta}$, which represents the uncertain parameters associated with the PDE. For example, parameters such as Q and k from the PDE equation 5.2 are often uncertain and are denoted by $\boldsymbol{\beta}$ in our scenario. We assume a prior distribution on these uncertain parameters, which are then fed into the neural network. The uncertainty in the input is propagated to the output using the assumed density filtering (ADF) approach [124]. The ADF method allows the propagation of a random distribution through the neural network, specifically using Gaussian distributions as the ADF approximation. The weights of the BPINN are learned by optimizing the loss function provided in equation (5.3). This loss function consists of a weighted sum of both the PDE residue and the mean square error between the actual temperature values and the BPINN predictions. The predictions made by BPINN will be the possible temperature values at the regions of interest, which corresponds to solving the forward problem by learning the neural network weights.

Once the weights are learned, the posterior distribution of the uncertain parameters is estimated using Bayes' theorem. The likelihood, obtained using the ADF approach, along with the prior assumptions on $\boldsymbol{\beta}$, is used to estimate the posterior distribution of $\boldsymbol{\beta}$. This process addresses the inverse problem. As we mentioned previously, while existing methods typically approximate the posterior using sampling-based techniques, in this chapter, we propose an analytic approach to estimate the posterior, which is detailed in Theorem 5.

Theorem 5. *For an inverse problem with observations $\mathbf{Y}$ and associated variance $\boldsymbol{\Sigma}_y$, where the forward model is parameterized by the weights $\boldsymbol{\theta}$ of a Bayesian physics-informed neural network and linearized as $\mathbf{Y} = \mathbf{A}\boldsymbol{\beta} + \mathbf{C}$, the minimum mean square estimate of the uncertain input parameters $\boldsymbol{\beta}$ is given by $\hat{\boldsymbol{\beta}}$ with variance $\mathbf{v}_\beta$, where:*

$$\begin{aligned}\hat{\boldsymbol{\beta}} &= (\mathbf{A}^\top \boldsymbol{\Sigma}_y^{-1} \mathbf{A} + \boldsymbol{\Sigma}_\beta^{-1})^{-1} (\boldsymbol{\Sigma}_\beta^{-1} \boldsymbol{\mu}_\beta + \mathbf{A}^\top \boldsymbol{\Sigma}_y^{-1} (\mathbf{Y} - \mathbf{C})) \\ \mathbf{v}_\beta &= (\mathbf{A}^\top \boldsymbol{\Sigma}_y^{-1} \mathbf{A} + \boldsymbol{\Sigma}_\beta^{-1})^{-1}\end{aligned}\tag{5.6}$$

Proof. Consider a neural network with a nonlinear transformation $\mathbf{f}$ applied to intermediate activations. Let the network layer parameters be denoted by $\boldsymbol{\theta}$, and assume an output of the form:

$$\mathbf{Y} = \mathbf{f}(\boldsymbol{\beta}\boldsymbol{\theta} + \mathbf{b}),$$

where $\mathbf{f}(\cdot)$ is a nonlinear activation function, and $\boldsymbol{\beta}$ represents the uncertain input parameters. Assume a Gaussian prior on $\boldsymbol{\beta}$:

$$\boldsymbol{\beta} \sim \mathcal{N}(\boldsymbol{\mu}_\beta, \boldsymbol{\Sigma}_\beta).$$

Using Bayes' theorem, the posterior distribution over the parameters is:

$$p(\boldsymbol{\beta} \mid \mathbf{Y}) \propto p(\mathbf{Y} \mid \boldsymbol{\beta}) \cdot p(\boldsymbol{\beta})\tag{5.7}$$

Assume the likelihood $p(\mathbf{Y} \mid \boldsymbol{\beta}) \sim \mathcal{N}(\boldsymbol{\mu}_Y, \boldsymbol{\Sigma}_Y)$, which can be obtained using the Assumed Density Filtering (ADF) approach.

To linearize the nonlinear function learned by the neural network, we perform a first-order Taylor expansion [33] around a point $\bar{\boldsymbol{\beta}}$:

$$\begin{aligned}\mathbf{Y} &\approx \mathbf{f}(\bar{\boldsymbol{\beta}}\boldsymbol{\theta} + \mathbf{b}) + \left. \frac{\partial \mathbf{f}}{\partial \boldsymbol{\beta}} \right|_{\boldsymbol{\beta}=\bar{\boldsymbol{\beta}}} (\boldsymbol{\beta} - \bar{\boldsymbol{\beta}}) \\ &\approx \left. \frac{\partial \mathbf{f}}{\partial \boldsymbol{\beta}} \right|_{\boldsymbol{\beta}=\bar{\boldsymbol{\beta}}} \boldsymbol{\beta} + \mathbf{f}(\bar{\boldsymbol{\beta}}\boldsymbol{\theta} + \mathbf{b}) - \left. \frac{\partial \mathbf{f}}{\partial \boldsymbol{\beta}} \right|_{\boldsymbol{\beta}=\bar{\boldsymbol{\beta}}} \bar{\boldsymbol{\beta}} \\ &\approx \mathbf{A}\boldsymbol{\beta} + \mathbf{C}\end{aligned}\tag{5.8}$$

Thus, the network's output can be approximated as a linear function of $\boldsymbol{\beta}$:

$$\mathbf{Y} \approx \mathbf{A}\boldsymbol{\beta} + \mathbf{C}$$

Substituting this into the Bayesian formulation in Equation 5.7, the posterior becomes:

$$p(\boldsymbol{\beta} | \mathbf{Y}) \propto \exp \left(-\frac{1}{2}(\mathbf{Y} - \mathbf{A}\boldsymbol{\beta} - \mathbf{C})^\top \boldsymbol{\Sigma}_y^{-1}(\mathbf{Y} - \mathbf{A}\boldsymbol{\beta} - \mathbf{C}) - \frac{1}{2}(\boldsymbol{\beta} - \boldsymbol{\mu}_\beta)^\top \boldsymbol{\Sigma}_\beta^{-1}(\boldsymbol{\beta} - \boldsymbol{\mu}_\beta) \right)$$

Taking the logarithm of the posterior:

$$\begin{aligned} \log p(\boldsymbol{\beta} | \mathbf{Y}) = & -\frac{1}{2}\boldsymbol{\beta}^\top (\mathbf{A}^\top \boldsymbol{\Sigma}_y^{-1} \mathbf{A} + \boldsymbol{\Sigma}_\beta^{-1}) \boldsymbol{\beta} \\ & + \boldsymbol{\beta}^\top (\mathbf{A}^\top \boldsymbol{\Sigma}_y^{-1}(\mathbf{Y} - \mathbf{C}) + \boldsymbol{\Sigma}_\beta^{-1} \boldsymbol{\mu}_\beta) + \text{const} \end{aligned}$$

Thus, the posterior over $\boldsymbol{\beta}$ is:

$$p(\boldsymbol{\beta} | \mathbf{Y}) \sim \mathcal{N}(\Phi^{-1}\Gamma, \Phi^{-1}) \quad (5.9)$$

where:

$$\Phi = \mathbf{A}^\top \boldsymbol{\Sigma}_y^{-1} \mathbf{A} + \boldsymbol{\Sigma}_\beta^{-1}, \quad \Gamma = \mathbf{A}^\top \boldsymbol{\Sigma}_y^{-1}(\mathbf{Y} - \mathbf{C}) + \boldsymbol{\Sigma}_\beta^{-1} \boldsymbol{\mu}_\beta$$

□

Thus, after the BPINN model is trained using the initial set of priors, the posterior is estimated using the learned BPINN weights. The newly estimated $\boldsymbol{\beta}$ values using equation (5.9) are then used as updated inputs to the BPINN, and the model is retrained. This iterative process of model training and posterior estimation is continued until convergence, solving both the forward and the inverse problems.

Algorithm 2 BPINN Training and Inference

- 1: **Input:** Spatio-temporal points $\{(x_i, y_i, t_i)\}_{i=1}^N$, initial priors on uncertain PDE coefficients μ_β, Σ_β , sparse MRI data $\{T_i\}_{i=1}^N$
- 2: **Output:** Trained neural network parameters Θ^* , updated priors $\mu_\beta^*, \Sigma_\beta^*$
- 3: **repeat**
- 4: Compute PDE loss $L_p(\Theta, \beta, \alpha)$ and data loss $L_d(\Theta, \beta, \alpha)$
- 5: Compute total loss:
$$\text{Loss} = w_p L_p(\Theta, \beta, \alpha) + w_d L_d(\Theta, \beta, \alpha)$$
- 6: Update neural network parameters Θ using optimizer
- 7: **until** convergence
- 8: Obtain the final neural network parameters Θ^*
- 9: **repeat**
- 10: Compute posterior distribution of PDE coefficients using Bayes' theorem:

$$P(\beta|\mathbf{Y}) \sim \mathcal{N}(\Phi^{-1}\Gamma, \Phi^{-1})$$

- 11: Use the updated priors $\mu_\beta^*, \Sigma_\beta^*$ to retrain the neural network and repeat steps 4-8
 - 12: **until** input parameter values converge
-

5.1.6 Results

Experimental Setup

Our proposed framework was tested on both simulated temperature profiles and an experimental MRI thermometry dataset. The simulated data were generated using finite element model in COMSOL Multiphysics, which solves the coupled electromagnetic and transient heat transfer equations. The electromagnetic field distribution was computed by solving the time-harmonic Helmholtz equation based on tissue-specific dielectric properties, including temperature-dependent relative permittivity ε_r and effective conductivity σ . The energy absorption Q was calculated as $Q = \sigma|\mathbf{E}|^2$, where $\mathbf{E}$ is the complex electric field. This electromagnetic power density served as the heat source in the bioheat equation, which incorporated temperature-dependent thermal conductivity k , density ρ , and specific heat capacity c . The experimentally acquired dataset consisted of MR thermometry measurements during microwave ablation of ex-vivo bovine liver tissue, performed under 3T MRI guidance using a custom 2.45 GHz water-cooled monopole applicator. The applicator fea-

tured a coaxial antenna with water flow to dissipate heat and enhance energy delivery. MR thermometry data were acquired every 6.4 s using a FLASH sequence (TR/TE = 50/12.3 ms) in one coronal plane and two axial planes to monitor temperature changes. Additional details about the dataset can be found in [122]. The temperature changes at each (x, y, z, t) point obtained from the MRI serves as the input to the BPINN module. Theoretically, symmetry exists in the heating patterns. Therefore, to simplify the analysis, we consider only a single quadrant of the spatial domain at a fixed time point, assuming symmetry for prediction. The electromagnetic energy absorbed by the tissue, denoted by Q , was approximated to decay exponentially from the center of the microwave applicator, i.e, it follows the equation $Q = Q_0 e^{-ad}$, where d is the radial distance from the center of the antenna, Q_0 is the maximum value of the absorbed electromagnetic energy, and a is the decay coefficient and was set to $a = 0.20$. It is important to note that the value of Q_0 was determined numerically from the MRI data, under the assumption that the specific absorption rate during the initial 30 s of heating [125] is equal to $\rho c_p \frac{\partial T}{\partial t} = Q_0$. Thus the training data consists of time shots from 0 to 40 s, with 10 s intervals. The volumetric heat capacity ρc_p can range from 3.7×10^6 to 4.3×10^6 [122] and in this chapter we assumed $\rho c_p = 4 \times 10^6$.

For the BPINN framework, the ADAM optimizer with a learning rate of 0.001 was used for training. After the training process, which included updating the uncertain physics parameters Q_0 and k , the model was evaluated using root mean square error (RMSE) and mean absolute error (MAE). The performance of the BPINN was compared to that of a traditional neural network (NN) and a physics-informed neural network without Bayesian updates (PINN) in terms of predictive accuracy and generalizability and is presented in the following section.

Numerical tests on simulated dataset

In this section, we evaluate the predictive performance of the proposed BPINN framework and compare it with the classic neural network and physics informed neural network frameworks. Figure 5.2 shows the heat maps of the actual temperature at 50 s, the corresponding

Metric	Time=100sec			Time=120sec			Time=140sec		
	NN	PINN	BPINN	NN	PINN	BPINN	NN	PINN	BPINN
RMSE	3.89	3.19	2.53	10.50	4.40	3.88	16.58	6.45	5.99
MAE	3.28	2.01	1.56	9.57	3.30	2.77	15.65	5.42	4.80

Table 5.1: Prediction error across various time instances using different frameworks on simulation data

Parameter	True	Mean	Variance
$\mathbf{Q}_0(W/m^3)$	11.17×10^7	11.912×10^7	0.00106
$\mathbf{k}(W/m.K)$	0.49	0.496	4.42×10^{-5}

Table 5.2: Values of the uncertain parameter obtained using BPINN for simulated data

predictions made by BPINN and the difference between the true and predicted values. Similarly, predictions and errors for 60 s and 70 s are shown in Figures 5.3 and 5.4, respectively. It can be observed that BPINN provides reasonable temperature predictions, with errors remaining below 7°C for the predictions at 70 s. The predictive performance of BPINN is further compared against classic NN and PINN models, with the results summarized in Table 5.1. While the mean absolute error (MAE) and root mean square error (RMSE) metrics for BPINN and NN are relatively close for lower time instances, BPINN outperforms the classic NN at higher time instances, with a percentage reduction in MAE by 69% and RMSE by 63%. This demonstrates the improved generalizability of our proposed framework compared to classic NN approaches. Additionally, BPINN outperforms the classic PINN model. This improvement is attributed to BPINN’s ability to update and re-estimate the uncertain physics parameters, which the classic PINN framework does not address. Table 5.2 shows the uncertain parameter estimates obtained using the BPINN approach for the specified training interval, compared to the true values. It can be seen that the estimated values closely align with the true values. By incorporating these updated parameter values, BPINN achieves enhanced predictive performance, particularly in scenarios where parameter uncertainty plays a significant role.

Numerical experiments on MRI data from ex-vivo tissues

The applicability of proposed framework was tested on experimental data using MRI images taken during microwave ablation experiments on ex-vivo tissues. Training was conducted using the first 3 minutes of heating data. As in the previous case, symmetry was considered to simplify the analysis, and only half of the MRI data along the spatial coordinates was used for a single time shot. To reduce noise and smooth the MRI data as a preprocessing step, we applied a 4x4 averaging filter. The decay constant for the exponential model of Q was set to $a = 0.23$. Performance evaluation was carried out similarly to the simulated dataset. The heatmap of the actual temperature values 294.4 seconds after heating started is given in Figure 5.5, the heatmap of the temperature predictions made by BPINN and the error heatmap, depicting the difference between true temperature values and predicted values, are also presented. Similarly the predictions and error heatmaps for 300.8 seconds and 307.2 seconds after the start of heating are shown in Figures 5.6 and 5.7 respectively. It can be observed that our proposed framework successfully captures the underlying dynamics during thermal ablation in tissues, tracking the temperature changes reasonably well. This is evident from the error heatmap, where the maximum error is less than 2°C . It should be noted that predicting the peak temperature in tissue (or predicting temperatures in tissue that are within $\sim 5\text{mm}$ of the ablation applicator) is not of greatest significance, as these regions will almost surely be ablated (i.e. raised to absolute temperatures exceeding 60°C). Rather, predicting temperature of tissue in regions at the periphery of the ablation zone (where absolute temperature is in the range $\sim 45 - 55^{\circ}\text{C}$), is of considerably greater significance as it represents the edge of the treatment zone, and would identify regions of targeted tissue that remain undertreated. As seen in Figures 5.5-5.7, the prediction error is considerably lower at regions of tissue further away from the applicator.

The predictive performance of BPINN on MRI data, compared to the baseline approaches, is summarized in Table 5.3. A similar trend to that observed in the simulated data experiments can be seen. While BPINN consistently outperforms other methods, the error is notably higher for the NN when compared to BPINN, especially for predictions at later

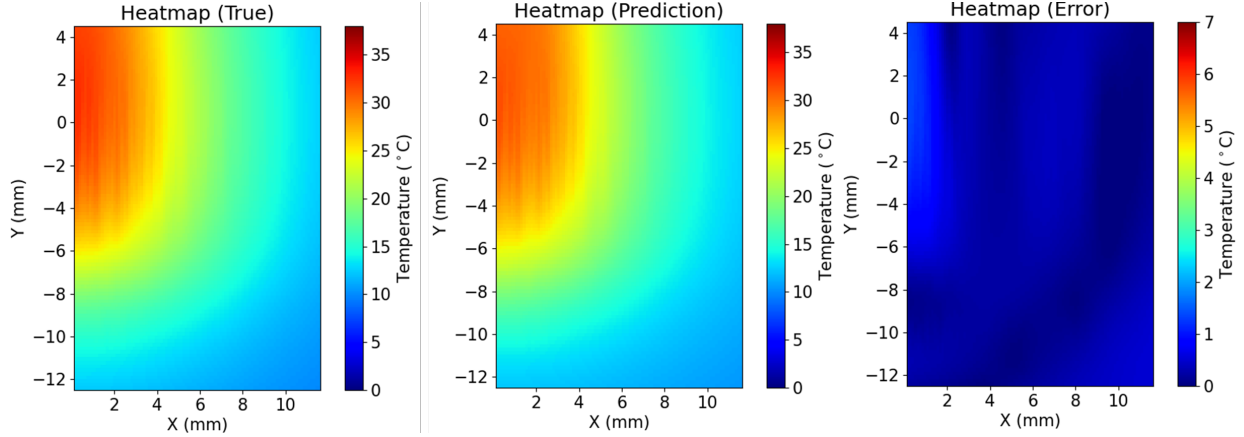


Figure 5.2: Heat map for true, prediction of temperature profile (T in $^{\circ}C$) at the colocation points (x, y in mm), and the error 10 sec outside of training window for simulation data

Metric	Time=320sec			Time=326.4sec			Time=332.8sec		
	PINN	NN	BPINN	PINN	NN	BPINN	PINN	NN	BPINN
RMSE	1.47	2.02	1.25	2.14	2.94	1.83	3.21	3.93	2.76
MAE	0.84	0.99	0.71	1.24	1.57	1.05	1.98	2.23	1.59

Table 5.3: Prediction error at different time points using various frameworks on MRI data, evaluated at 320.0 s, 326.4 s, and 332.8 s after the start of heating

time points, highlighting the superior generalizability of our model. The uncertainty quantified on the BPINN predictions using ADF approach is depicted in the Figure 5.9. The 1D predictions along the x-axis for a fixed y-value are shown, with uncertainty represented as the standard deviation. It can be noted that the uncertainty is higher at the initial points (closer to the microwave applicator), likely due to a mismatch between the actual variation in absorbed electromagnetic energy Q and the assumptions made in our model. The uncertainty quantified using the ADF approach assumes a Gaussian distribution and is utilized as likelihood information within the Bayesian framework. This aids in estimating the posterior distribution, which corresponds to the values of the uncertain parameters. Table 5.4 reports the estimated uncertain parameters for MRI data using the BPINN framework, alongside their true values. The 1D variation of the estimated and actual electromagnetic energy absorbed from the microwave applicator along the x-axis for $y = 0\text{mm}$ is shown in Figure 5.8. The exponential model is truncated to zero after $x = 7\text{mm}$ to match the variation of the electromagnetic energy in our framework.

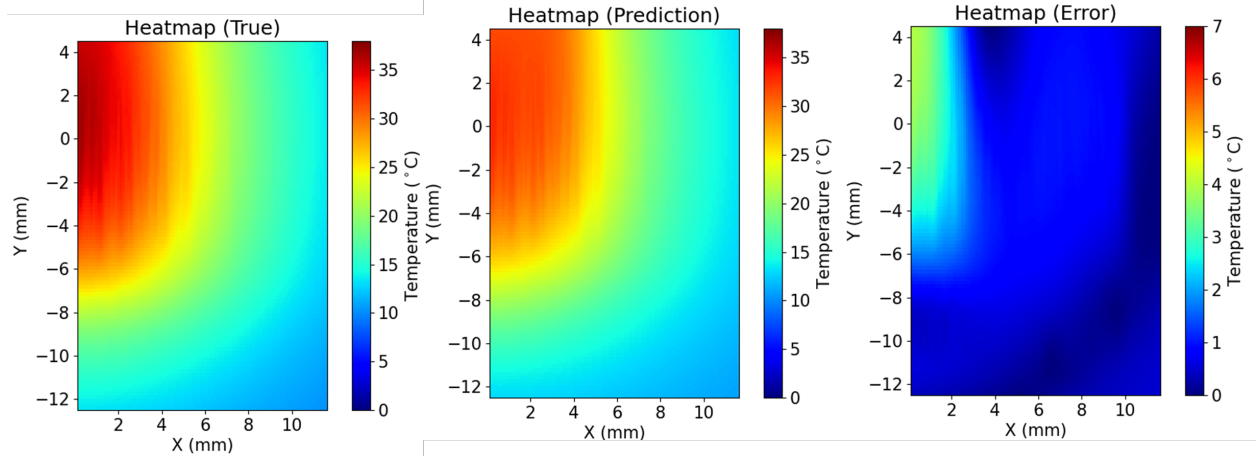


Figure 5.3: Heat map for true, prediction of temperature profile (T in $^{\circ}C$) at the colocation points (x, y in mm), and the error 20 sec outside of training window for simulation data

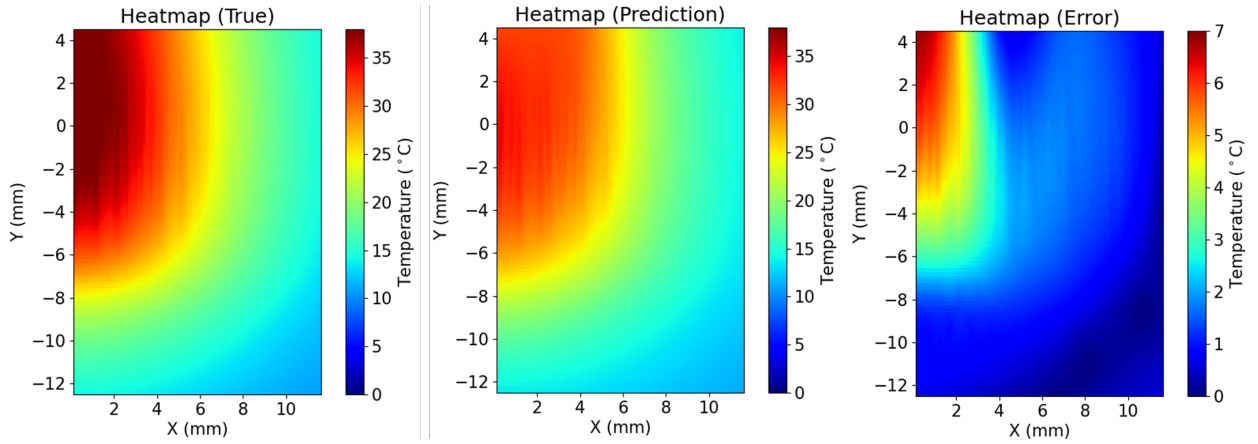


Figure 5.4: Heat map for true, prediction of temperature profile (T in $^{\circ}C$) at the colocation points (x, y in mm), and the error 30 sec outside of training window for simulation data

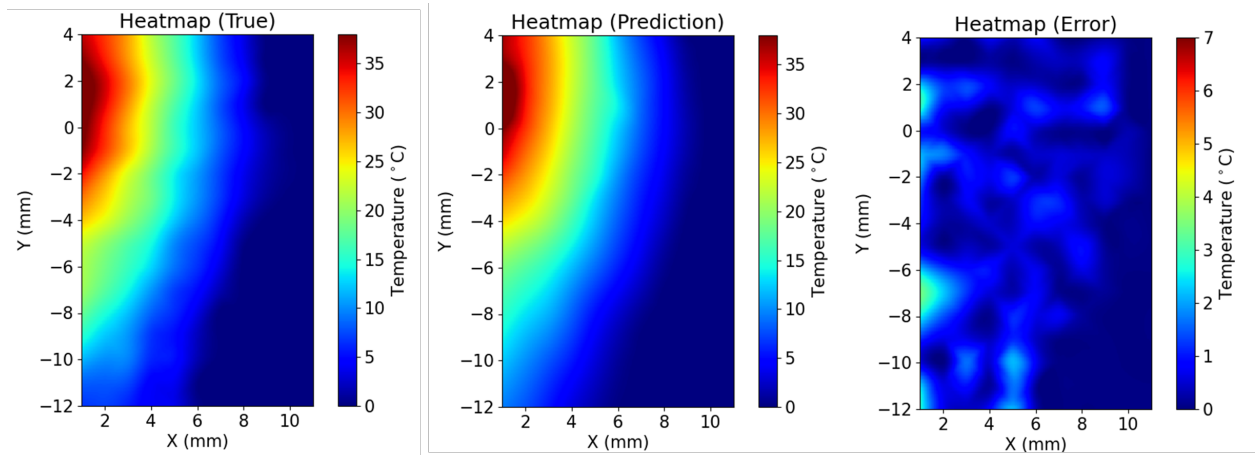


Figure 5.5: Heat map for true and prediction of temperature profile (T in $^{\circ}C$) at the colocation points (x, y in mm), 294.4 seconds after start of heating for MRI data

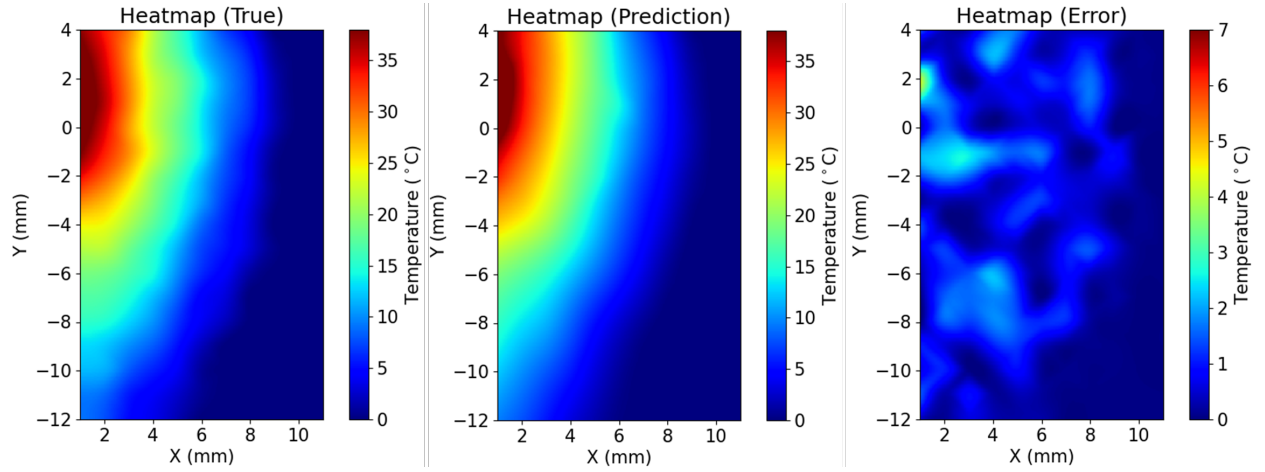


Figure 5.6: Heat map for true and prediction of temperature profile (T in $^{\circ}C$) at the colocation points (x, y in mm), 300.8 seconds after start of heating for MRI data

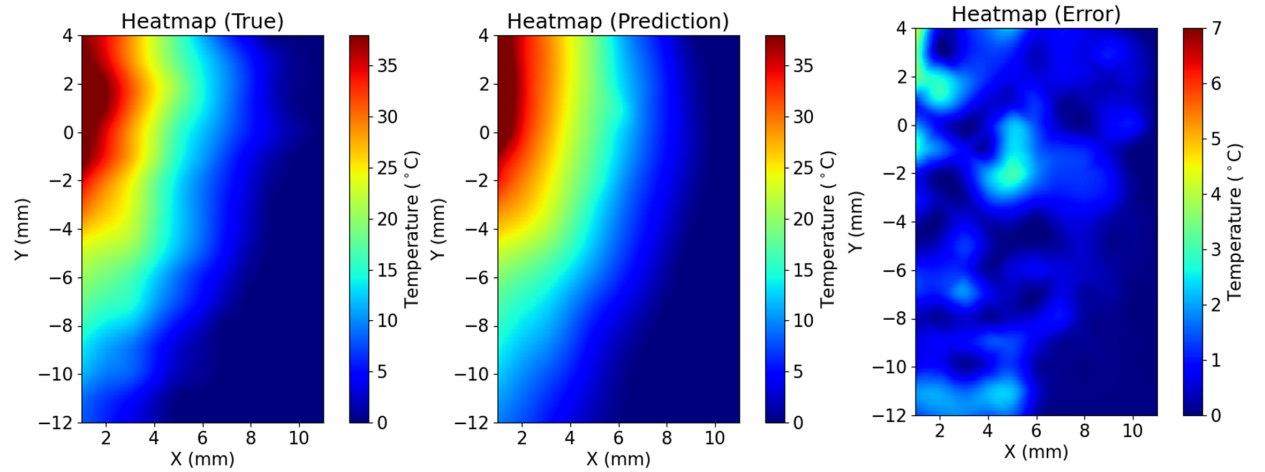


Figure 5.7: Heat map for true and prediction of temperature profile (T in $^{\circ}C$) at the colocation points (x, y in mm), 307.2 seconds after start of heating for MRI data

Parameter	True	Mean	Variance
$Q_0(W/m^3)$	16.43×10^7	16.80×10^7	0.00011
$k(W/m.K)$	0.46-0.57	0.53	4.26×10^{-5}

Table 5.4: Values of the uncertain parameter obtained using BPINN for MRI data

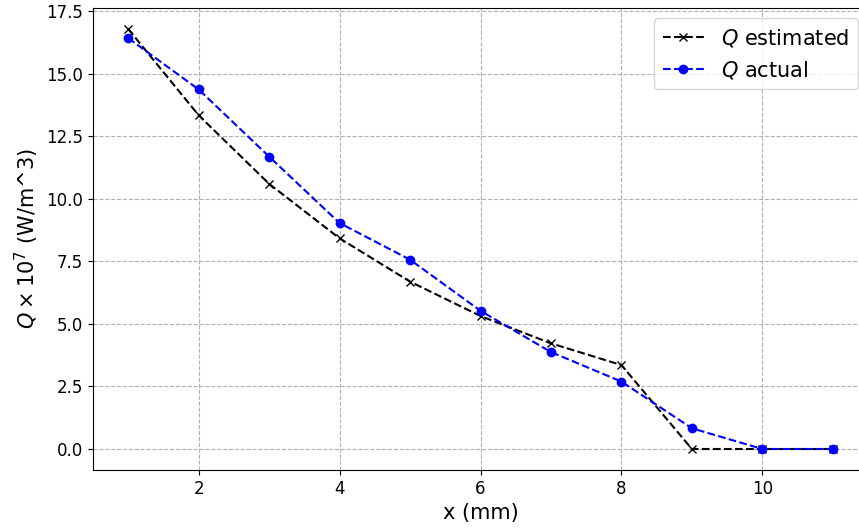


Figure 5.8: Variation of the estimated and actual electromagnetic energy absorbed from the microwave applicator along the x-axis for MRI data

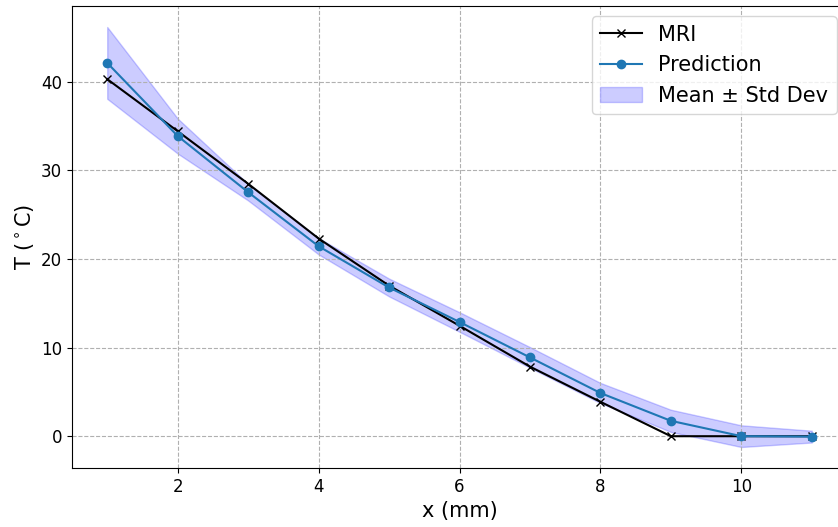


Figure 5.9: The 1D plot along x direction for the BPINN predictions on MRI data along with one std deviation obtained using assumed density filtering

5.1.7 Discussion

Predicting temperature changes accurately during thermal ablation has the potential to assist physicians in delivering technically successful procedures for cancer treatment, where accurate estimation of the spatial temperature profile at a specific time point may significantly impact treatment efficacy. Although computational models are used to design equipment and guide treatment decisions, they are often computationally expensive and rely on physical parameters that yield high uncertainty due to limited experimental data. In this work, we proposed a Bayesian physics-informed machine learning framework that not only improves the prediction of temperature dynamics, i.e., the 2D distribution of temperature at different time points during ablation, but also provides estimates of uncertain model parameters, offering enhanced generalizability compared to traditional approaches.

While the proposed approach demonstrates strong predictive performance, particularly in the context of temperature evolution during ablation, it is important to analyze the findings within the broader ablation modeling literature. Traditional computational models are often limited by computational cost, sensitivity to uncertain tissue-specific properties, and difficulties in modeling nonlinearities over time. BPINNs offer a promising data-driven approach by incorporating prior physics knowledge within the learning framework. However, the current implementation addresses a static inverse problem, where uncertain parameters are estimated and assumed constant over time. This limits the model’s ability to capture nonlinear temporal dynamics, such as the evolution of tissue properties with heating, which can in turn affect the ablation response in real-world scenarios.

A possible next step to overcome this limitation involves extending the Bayesian framework to dynamic inverse problems. Furthermore, spatial heterogeneity in tissue properties was not explicitly modeled, $\mathbf{Q}$ was inferred using 1D radial data and assumed to generalize across the 2D domain. This simplification may contribute to performance degradation as predictions extend beyond the training period, particularly in later phases of heating. Incorporating spatially and temporally adaptive parameter estimation would enhance the clinical relevance and robustness of BPINNs for thermal ablation.

5.2 Case Study 2: Graph-Based Machine Learning for Biomarker Identification

5.2.1 Related Work

When an external force damages the head - the result is head trauma, and if there's temporary or permanent brain dysfunction, we call it a traumatic brain injury, or TBI. This external force could be a blunt impact, like a baseball bat, a penetrating injury, like a gunshot wound, a blast wave, like an explosion, or an accelerating-decelerating force, like in a motor vehicle crash. TBI is becoming common and various methods are employed to identify the severity of TBI. Traditional methods to identify TBI rely heavily on imaging data, which, while useful, often lack biologically significant information compared to omics-based data cite. Several studies have utilized supervised learning techniques that require large amounts of labeled data [126], posing a challenge for practical deployment. Graph-based learning methods have recently emerged as a promising alternative, enabling the integration of biological pathway knowledge with proteomics data [127]. While graph neural networks (GNNs) have been applied in proteomics, their application specifically to TBI biomarker discovery remains largely unexplored. Prior work has focused on general proteomic network analysis but has not incorporated temporal changes in protein interactions over time, a crucial aspect of understanding TBI progression.

5.2.2 Contributions

- We propose a GNN-based inductive learning framework to identify potential biomarkers for Traumatic Brain Injury (TBI) using proteomics data.
- Topological information is incorporated through Protein-Protein Interaction (PPI) networks, which serve as prior biological knowledge.
- Temporal aspects are captured as node features, enabling the identification of potential biomarkers at various time points post-injury.

- We also quantify the epistemic uncertainty associated with the model predictions

5.2.3 Background

Traumatic Brain Injury (TBI) is a form of head trauma that results in temporary or permanent brain dysfunction. TBI is a significant global cause of death and disability. In the United States alone, the Center for Disease Control and Prevention reported over 220,000 TBI-related hospitalizations and 60,000 TBI-related deaths in 2021 [128]. TBI is a complex neurological disorder with primary (mechanical injury) and secondary mechanisms, such as neuroinflammation, excitotoxicity, oxidative stress, cellular apoptosis, and mitochondrial dysfunction, that contribute to long-term motor and cognitive deficits¹⁻⁴. Current TBI diagnosis relies on neurological exams (Glasgow Coma Scale) [129] and imaging (CT, MRI) [130], which are inadequate for early and rapid detection, assessment, and monitoring of severity and specific brain injury components (cell death, neuroinflammation, or blood-brain barrier, BBB, breakdown). Proteomic biomarkers offer a more precise and objective means of diagnosis by analyzing protein interactions and pathway activations [127]. However, challenges such as high variability in data, sample heterogeneity, and detection limitations hinder effective biomarker identification. Machine learning approaches, particularly graph-based models, provide a structured way to extract meaningful biological information from complex proteomic datasets [131].

5.2.4 Experimental Set up

The following section outlines the procedure used for data collection.

Mouse Model of Traumatic Brain Injury: All animal procedures were conducted in accordance with the guidelines of the Institutional Animal Care and Use Committee at the University of Texas Medical Branch and adhered to the US National Institutes of Health guidelines. C57BL6J male mice (The Jackson Laboratory, #000664), aged 10-14 weeks, were housed in a controlled environment with a 12-hour light/12-hour dark cycle at a temperature of 21-23°C, with free access to water and a standard chow diet.

To induce TBI, a non-penetrating, closed-skull weight-drop model was employed on unrestricted mice, adapted from previously published methods. Briefly, mice underwent general health assessments and handling for 10 minutes the day prior to the injury to minimize stress related to handling. On the day of the injury, mice were anesthetized with 3-5% isoflurane until the righting reflex was lost, and they were immediately placed in a prone position on top of a tin foil with slits, with the cranium directly underneath a plunger with a brass disc at the end. A 150 g weight was dropped from a height of 1.5 meters. The impact from the plunger caused the mice to break through the aluminum foil barrier and undergo a 180-degree flip while falling 10 cm onto a foam cushion. After injury, mice were placed on a warm pad until they regained consciousness and attained a prone position. Sham animals underwent anesthesia without injury. At various time points post-injury, mice were euthanized, and blood samples were collected for further analysis.

Blood Plasma Harvest: Mice were anesthetized with 3-5% isoflurane until the righting reflex was lost and then placed in a supine position with their head in a nose cone, with a continuous flow of isoflurane administered throughout the procedure. An incision was made in the skin from the bottom of the rib cage to expose the chest cavity, ensuring that the intrathoracic pressure remained intact. The heart was located, and a 1 mL syringe with a 25G x 5/8 in. needle (BD, #309626) was carefully inserted at a 45-degree angle to puncture the apex of the left ventricle. Blood plasma was then slowly extracted, and the animals were humanely euthanized by cervical dislocation followed by decapitation. Proteins in the plasma were extracted and quantified at various time points. Label-Free Quantification (LFQ) intensity of proteins from blood plasma was obtained using mass spectrometry (MS)-based proteomics, which served as input for the proposed GNN-based framework for potential biomarker identification.

5.2.5 Proposed approach

Proteomics data was subjected to Graph Neural Network (GNN) analysis, a computational machine learning process, to identify potential biomarkers for TBI by incorporating addi-

tional features such as neurological associations, predicted protein-protein interactions, and time-dependent LFQ intensity values of each protein. By including these parameters, GNN provides a more comprehensive analysis that better captures the biological context of each identified protein to increase the likelihood of discovering a potential biomarker. We constructed a Protein-Protein Interaction (PPI) network using available databases from the STRING [132]. The identification of potential biomarkers was formulated as a node classification problem, with node labels representing ranks from 1 to 6. Rank 1 indicated the highest probability and rank 6 the lowest probability of a protein being a potential biomarker.

5.2.6 Results

The training labels, constituting approximately 5%–8% of the overall dataset, were obtained based on domain-expert knowledge. The GraphSAGE model was employed for learning node representations, capturing both structural and feature-based relationships within the dataset. The model architecture consists of three node-embedding layers and one feed-forward layer, with ReLU activations applied at intermediate layers and a Softmax function at the output for classification. A learning rate of 0.001 and Adam optimizer was used to ensure convergence of the training loss function.

We performed GNN-based learning techniques to predict probabilities of individual proteins being potential biomarkers for all nodes in the network. A time point specific network with ranks for potential biomarkers for 3hpi is shown in Figure 5.10. We detected apolipoproteins and inflammatory response associated proteins that were matched by biostatistical analysis. Additionally, the uncertainty quantification framework proposed specifically for graph neural networks in Chapter 4 was used to quantify the uncertainty in model predictions, as shown in Table 5.5. It can be observed that the uncertainty quantified by the proposed GNSDE-UQ method yields lower ENCE scores compared to the widely used MC Dropout, demonstrating its reliability as a superior uncertainty quantification approach.

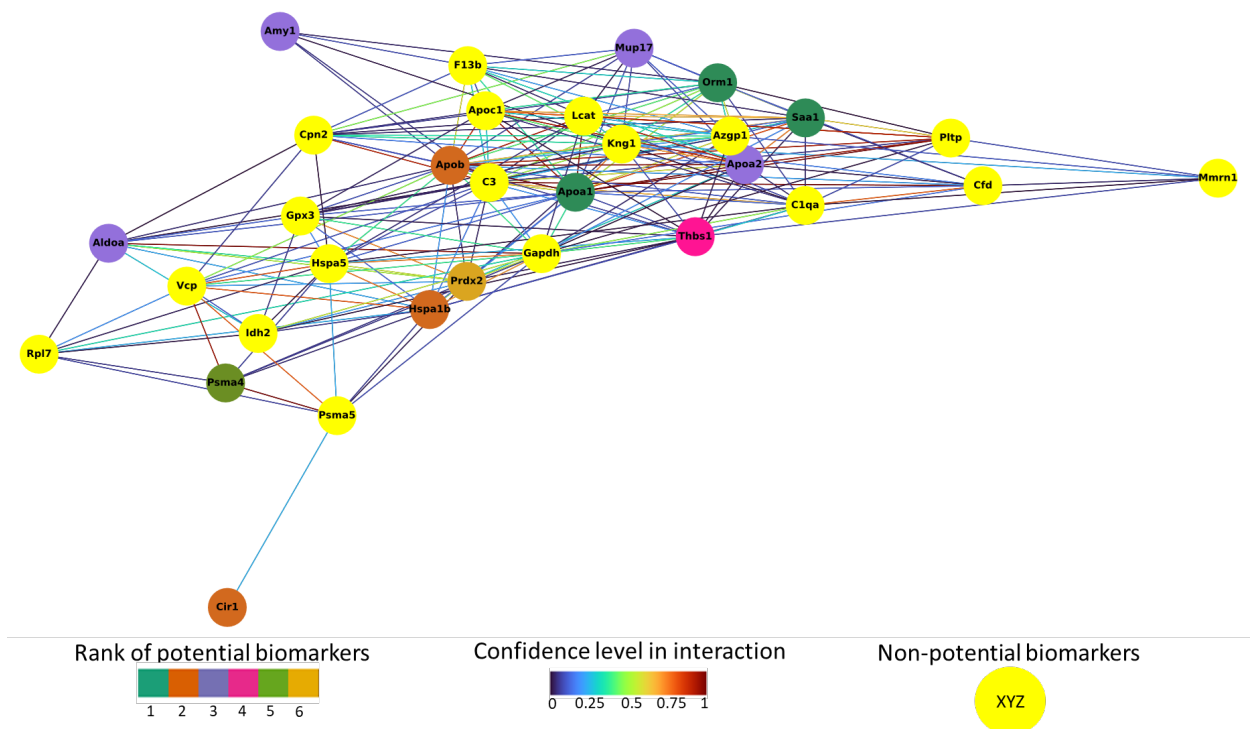


Figure 5.10: Potential Biomarker identified for Traumatic Brain Injury at 3 hours post injury

P	GNSDE-UQ		MC dropout	
	ENCE	Training Loss	ENCE	Training Loss
0.1	0.6095	0.004	9.96	0.0366
0.3	0.3836	0.0075	12.76	0.373
0.5	0.3844	0.0026	1.996	0.0367
0.7	0.2831	0.0075	2.795	0.0404
0.9	0.8151	0.0029	1.528	0.0614

Table 5.5: UQ for the biomarker identification using GNSDE-UQ and MC dropout framework

5.3 Summary

In this chapter, we proposed two frameworks that incorporate prior knowledge as learning bias and observational bias in limited-data scenarios. Additionally, the learning framework is equipped with uncertainty quantification on the output predictions. We explored two case studies to demonstrate the effectiveness of the proposed frameworks.

The first case study focused on predicting temperature changes during thermal ablation. Accurate estimation of the spatial temperature profile at specific time points can significantly impact treatment efficacy. While computational models are commonly used to design equipment and guide treatment decisions, they are often computationally expensive and rely on uncertain physical parameters. To address this, we proposed a Bayesian physics-informed machine learning framework that not only improves the prediction of temperature dynamics, i.e. the 2D distribution of the temperature at different time points during ablation but also provides estimates of uncertain model parameters, offering enhanced generalizability compared to traditional approaches. Therefore, this framework serves as an initial step toward developing personalized treatment planning for patients.

In the second case study, we propose a graph neural network-based machine learning algorithm for identifying potential biomarkers in traumatic brain injury (TBI). The algorithm utilizes protein-protein interactions as edge connections in the input data. The proposed framework incorporates prior domain knowledge as observational bias by embedding physics-based information from an existing database. Specifically, the underlying data is modified to reflect the known interactions between proteins, allowing for a more biologically meaningful identification of novel TBI biomarkers in proteomics data.

Chapter 6

Uncertainty Guided Active Sampling of Spatio-Temporal Data

Data sparsity in spatiotemporal datasets, specifically in biological applications often arises due to the high cost and laborious nature of sensing modalities as we discussed in Chapter 1. These constraints limit both the number of sensors that can be deployed and the volume of data that can be collected. A promising approach to address this challenge is to deploy sensors strategically, ensuring that the limited data acquired is as informative as possible. In this chapter, we introduce a framework for active sampling of spatio-temporal data that leverages uncertainty quantification to guide these strategic decisions. This approach is designed to maximize the informational value of each measurement, thereby enhancing the overall effectiveness of downstream analysis performed on such data.

6.1 Related Work

In modern agricultural practices, there is an environmental obligation to ensure efficient usage of water. Frequent monitoring of soil moisture and temperature can help farmers to optimize irrigation, reduce water usage and ensure that plants receive the exact amount of water required [133]. Accurately monitoring soil moisture and temperature can also be a

valuable tool to detect changes in the soil environment that may indicate the presence of plant disease [134]. Thus, frequent monitoring of soil moisture is critical for many applications and efficient agriculture management. Soil dynamics vary across both space and time, necessitating regular measurements for various applications. For instance, in regional planning, it is essential to have soil moisture data at different spatial and temporal scales, which require measurements at various depths and time intervals [135]. To effectively represent these variations, data can be represented in the form of a spatio-temporal graph, aiding in the quantitative monitoring of soil health.

Over the past decade, the use of machine learning (ML) algorithms such as deep learning [136], support vector machines [137], and random forests [138] has significantly increased in soil health research. This increase is attributed to their non-parametric nature and their ability to capture complex, non-linear relationships. Recently, advanced ML methodologies like graph neural networks [139] and recurrent neural networks [140] have also been explored in order to efficiently capture spatial and temporal variations among soil measurements. Despite the importance of soil health data, its availability from agricultural lands is often hindered by several factors, including sensor reliability, data collection challenges, maintenance issues, and inadequate network connectivity. Moreover, deploying a large number of sensors in the field can be prohibitively expensive, resulting in spatially sparse data [141].

Missing data creates substantial issues for ML-based analysis. Omitting missing data can lead to biased or incorrect conclusions, complicating the identification of temporal or spatial patterns [142]. Therefore, developing effective approaches to handle incomplete data is crucial when designing machine learning-based decision support models. Given the constraints of sparse measurements, we propose a novel uncertainty-guided framework to strategically choose sensor locations rather than relying on random placement. This can be achieved using semi-supervised approaches, such as Active Learning (AL), which enable the use of a limited number of in situ sensors while maintaining high-quality predictive modeling of soil health [143].

Spatio-temporal graphs incorporate both spatial and temporal dimensions into their structure. They represent connections or relationships between entities not only in space

but also over time and are useful in modeling complex systems such as transportation systems [144], climate patterns [145] and precision agriculture [146]. Owing to their applications in a wide variety of complex systems, learning in spatio-temporal graphs has attracted significant research attention recently. Among these studies, there is a proposal to explicitly model the structure information into the convolutional operation and combine it with pseudo three-dimensional networks to capture temporal dependencies in the traffic data [147]. Another work proposes to learn temporal features using an attention-based Long Short-Term Memory (LSTM) network and the spatial features using Multi-Task Learning output layers [52]. Deep Spatio-Temporal Graph Convolutional Network exploits a combination of graph convolution operations and standard convolutions to capture the dynamic variations in both spatial and temporal perspective [148]. LSTM unit and convolutional neural network are used to encode the spatial features of the target points, which are then fed to temporal attention mechanism to complete the decoding process and predict grain temperature [149]. A graph-based Recurrent Neural Network (RNN) is proposed that incorporates both geographical and temporal knowledge in the model for crop yield prediction [150]. More recently, Dynamic Traffic Correlation-based Spatio-Temporal Graph Convolutional Network is proposed to extract a dynamic adjacency matrix from different traffic characters to describe dynamic spatio-temporal correlations. Further, an attention and dynamic adjacency matrix-based GCNs framework is used to capture dynamic spatial features, while a LSTM is used to capture urban traffic temporal features, respectively [151]. A multi-output and spatiotemporal model combining GCN and transformer is used to capture spatial correlations and temporal dependencies of the sensor network in a granary [152]. Hierarchical Spatio-temporal Graph Neural Network incorporates an adaptive graph learning module that constructs a self-learning hierarchical graph and leverages convolution and gated temporal convolution with a dilated inception as the backbone to capture hidden spatial dependencies and diverse long-term meteorological trends [153]. Most of the approaches proposed for learning in spatio-temporal graphs require huge amounts of training data, which might be difficult to obtain in practical scenarios like precision agriculture. Despite the advances in sensor technology, several challenges remain in collecting consistent and comprehensive

soil data from the agricultural fields. Sensor reliability, data transmission issues, maintenance, and field operations often lead to sparse and uneven data distribution. The high cost and logistical challenges of widespread sensor deployment further worsen these issues [154]. Developing machine learning-based predictive frameworks to model the dynamics of these sparse soil measurements is challenging. Sparse measurements can introduce bias and are not always representative of the space that is being learned [155]. There are existing works that capture spatial and temporal dynamics in sparse data setups by imputing the data at missing locations [23]. However, these methods may bias the results as the missingness itself might be informative [156]. Semi-supervised learning methods such as Active Learning (AL), have addressed the sparsity in data in a machine learning setting. AL can significantly reduce the need for extensive labeled data by strategically selecting the most informative samples for labeling, resulting in higher prediction accuracy with limited data [157], [158]. The integration of AL with Graph Neural Networks (GNNs) has emerged as a potent approach for spatial and temporal data prediction. The application of AL and GNNs in environmental and agricultural data prediction has been explored in recent studies. [159] demonstrated the efficacy of GNNs in handling spatial dependencies in various applications. Similarly, [46] have illustrated the advantages of semi-supervised learning on graph structures, facilitating effective learning from sparsely labeled data. Though GNNs excel in capturing complex dependencies and relationships in data structured as graphs, they fail to capture temporal dependencies. This work, for the first time, presents a novel uncertainty-guided AL framework for applications involving spatio-temporal processes and graphs.

6.1.1 Contributions

This work addresses the problem of active learning in spatio-temporal graphs. For the first time, we propose a novel uncertainty-guided learning scheme involving a combination of Dilated Causal Convolutional Neural Network (DC-CNN) and Graph Neural Network (GNN). We systematically propagate the uncertainties associated with graph node features (repre-

senting the noise in sensor measurements) through all the layers of DC-CNN and GNN. This helps us estimate the uncertainty associated with the model predictions, which is then leveraged to guide the informative sample selection via querying in the context of AL. We evaluate the effectiveness of the proposed methodology by conducting experiments over a field-scale sensor network data set for monitoring and modeling the spatial and temporal variation of soil moisture and temperature in a dryland agricultural field. The experimental evaluation reveals that it is possible to predict soil moisture/temperature over a large spatial area with over 90% accuracy even with a limited fraction of sensors and strategically selected measurement locations. The proposed methodology is generic and can be applied to the problem of monitoring of any soil parameter (such as nitrogen, phosphorus, potassium etc.) within any spatio-temporal graph setup, irrespective of the learning scheme and model architecture. Such an approach of utilizing uncertainty information to support AL in spatio-temporal graphs has not been reported in the literature so far and is a novel contribution of this work. The remainder of this article is organized as follows: background on spatio-temporal graphs, DC-CNN, GNN and AL is provided in Section 2. We discuss the specific problem formulation targeted in this work and the proposed approach to implement uncertainty-guided AL on spatio-temporal graphs in Section 3. Experimental evaluation is outlined in Section 4 and the article ends with concluding remarks in Section 5.

6.2 Background

6.2.1 Spatio-Temporal Graphs

Spatiotemporal graphs represent data that is indexed in both spatial and temporal dimensions. These types of graphs are useful for modeling complex systems with features and relationships that change over space and time, such as transportation networks, soil parameter dynamics in agriculture, climate patterns, and biological processes [160]. In the context of spatiotemporal graphs, nodes/vertices represent spatial entities such as traffic intersections or location of field sensors. Edges/links represent the relationships between the

nodes, such as roads between the intersection, communication links between sensing units or parametric dependencies across soil sensor locations. As a result, the spatial aspect of the spatiotemporal graph captures the geographical or logical locations of the nodes, as well as the spatial relationships represented by distance metrics, adjacency matrices, or spatial clustering techniques. On the other hand, temporal aspect captures the dynamic nature of the graph, such as the change in features or attributes over time and the appearance or disappearance of nodes or edges, thereby helping to model the evolution of the system it represents. Advances in machine learning have led to the development of novel models such as Graph Attention Networks (GAT) [161], a combination of Graph Neural Networks (GNNs) and Recurrent Neural Networks [162], and Temporal Convolution Neural Networks [147] which can effectively learn from spatiotemporal data. Leveraging recent technological advancements, spatiotemporal graphs offer a powerful framework for modeling and analyzing complex systems where spatial and temporal dimensions interplay.

6.2.2 Dilated Causal Convolutional Neural Networks

Dilated Causal Convolutional Neural Networks (DC-CNN) provide an efficient mechanism for capturing temporal information in sequential data and for temporal prediction tasks [163]. The major component of the dilated causal convolution is the convolutional neural network. In the context of sequential data, a 1D convolutional layer is applied to the data. Causal convolution ensures that the output at time step t only depends on inputs from time steps t and earlier. This is crucial for sequence prediction tasks where future data points should not be used to predict past or present values. In a causal convolution, the convolution operation is modified so that each filter only accesses past and present input values. Traditional convolutional layers have a limited receptive field, meaning they can only consider a small window of the input at a time. Dilated convolution introduces a spacing (dilation) between the filter elements, allowing the network to have a larger receptive field without increasing the number of parameters or computational cost excessively [164]. The dilation factor decides the number of input elements to skip which is decided by the spacing between the filter

elements. For instance, with a dilation factor of 2, the filter skips every other input element. Compared to other frameworks used for capturing temporal relationships, such as RNN, DC-CNN maintain a manageable number of parameters while being able to look back at a larger context in the input sequence [165]. The hierarchical nature of the dilation factor allows the network to capture both short-term and long-term dependencies. Lower layers with smaller dilation factor capture finer details, while higher layers with larger dilation rates capture broader patterns [164].

6.2.3 Graph Neural Networks

The field of graph neural networks (GNNs) has experienced remarkable and swift advancements in recent years. GNNs are a class of deep learning models designed to perform inferencing on graph structured data [166]. GNNs can capture complex relations and interactions in various domains such as sensor networks, transportation systems, social networks and so on [167]. Unlike traditional neural networks that operate on regular grids like images or sequences, GNNs are tailored to capture the underlying structure and properties of graph-based data, which consists of nodes representing entities or objects and edges representing relationships or interactions between these nodes [142]. The nodes representing entities are called node embeddings of a graph. These vectors capture the structural information and properties of the nodes, making them suitable for various downstream tasks like node classification, link prediction, and clustering. GNNs operate by passing messages between nodes, where each node updates its representation by aggregating messages from its neighbors and combining them with its current state. Various GNN variants exist based on different aggregation and combination operations [168], [169]. In this work, we use GraphSAGE, an inductive framework that can generalize to unseen nodes and subgraphs, unlike transductive models that require re-training when new nodes are added [49]. GraphSAGE enhances scalability and adaptability to dynamic graphs by sampling a fixed-size set of neighbors for each node and aggregating their features to form the updated representation for the node of interest. This approach significantly reduces the computational complexity associated with

large graphs. The equation used to update node features of a node u at i^{th} node embedding layer is as follows.

$$\begin{aligned} h_u^{(i)} &= f^{(i)} \left(h_u^{(i-1)}, h_{N(u)}^{(i-1)} \right) \\ &= g \left(\theta_c^{(i)} h_u^{(i-1)}, \theta_A^{(i)} \tilde{A}(h_{N(u)}^{(i-1)}) \right) \end{aligned}$$

6.2.4 Active Learning

AL is a semi-supervised machine learning approach that selectively queries the most informative data points for labeling, aiming to build accurate models with fewer labeled instances [158]. It allows the learning algorithm to leverage information from external sources (e.g., human experts) and train an efficient decision model using much fewer labeled instances as compared to traditional supervised ML methodologies. This approach is particularly useful in situations where labeling is expensive or time-consuming, such as in natural language processing, computer vision, medical imaging, and speech recognition. The learning algorithm in AL settings is referred to as Active Learner and the external information source is termed as the Oracle. The Active Learner is permitted to select the data it wants to learn from. Thus, it poses queries to the Oracle in the form of unlabeled data instances, and the Oracle is expected to supply corresponding labels. By employing various query strategies such as uncertainty sampling, query by committee and sampling diversity, AL helps in selecting the candidate query instance that is the most informative, enhancing model performance while operating under a labeling budget [157]. In this work, we employ an AL strategy combined with Graph Neural Networks (GNN) and Dilated Convolutional Neural Networks (DC-CNN) to predict soil health in agricultural fields.

6.3 Problem Formulation

This work presents a novel framework to perform Active Learning on spatio-temporal graphs. The proposed architecture involves two modules: (i) Dilated Causal Convolutional Neural Network for learning in the temporal domain, and (ii) Graph Neural Network for the spatial counterpart. The proposed framework supports systematic propagation of uncertainty in node features through the DC-CNN and GNN layers using Assumed Density Filtering (ADF), followed by moment matching. This allows us to derive the uncertainty associated with model predictions, which is further leveraged to select the most informative nodes for AL in spatio-temporal graphs.

We consider a spatio-temporal graph $G = (A, E, X_t)$, where $A \in R^{N \times N}$ denotes the spatial adjacency matrix; N denotes the number of nodes and the element $a_{i,j} \in A$ is 1 if the nodes i and j are connected via an edge. E represents the spatial relationship between the nodes, and $X_t \in R^{N \times D}$ is the set of D dimensional dynamic features corresponding to each node at time t . The uncertainty in node features is assumed to be Gaussian with zero mean and known variance. For a node $u \in G$, the uncertainty in the feature vector can be represented as $h_u \sim \mathcal{N}(h_u^*, \Sigma)$ where h and h^* are the measured and true values of the feature vector, respectively, and Σ is a diagonal matrix comprising of known variances of noise in individual features. It is also assumed that the noise in features of different nodes is uncorrelated. The uncertainties in ML frameworks can be either aleatoric or epistemic. The intrinsic randomness of data caused due to noisy or faulty measurements is referred to as aleatoric uncertainty. In a graph-structured data, it can be captured via uncertainty associated with node features and probabilistic links [170]. The uncertainty in node features takes care of the measurement noise and probabilistic links handle the imprecise information about underlying graph structure. The inability of a model to comprehend the underlying process leads to epistemic uncertainty. In a graph-structured data, it can be captured via uncertainty in parameters of GNN and the associated activation functions. In this work, we consider the propagation of aleatoric uncertainty through the layers of DC-CNN and GNN, and propagation of epistemic uncertainty shall be considered as a part of the future work. The proposed

methodology shall be evaluated over a field-scale sensor network data set for monitoring and modeling the spatial and temporal variation of soil health in a dryland agricultural field [171]. Therefore, in the context of measurements obtained from agricultural fields, every node in G corresponds to a location with two sensors, soil temperature and volumetric water content, each with a hourly time-series measurement for 1000 time instances. The node features comprise of the truncated Singular Value Decomposition (SVD) of the weighted adjacency matrix [52]. This helps to improve the learning scheme by incorporating higher order neighbor embeddings, while keeping self-embedding separate from the neighbor embeddings in aggregation function of GNN [52]. In this work, we systematically quantify the uncertainties through different layers of DC-CNN and GNN in order to finally obtain the uncertainty in model predictions because of noise in the node features. Incorporating prior information about the data, such as sensor or measurement noise, allows for more reliable predictions by quantifying uncertainties in data through confidence intervals around the predictions. For instance, it is critical to account for measurement noise in the readings obtained from various sensors like temperature and soil moisture sensors installed in agricultural fields. If these uncertainties are overlooked and the features are assumed to be deterministic, it can lead to unreliable predictions. It is well known that the widespread installation of sensors can be excessively costly and can place a substantial burden on communication networks. This work, for the first time, presents an uncertainty guided AL framework for predicting measurements across the entire field by using sparsely available data. The uncertainty associated with the model prediction of individual nodes is used as the heuristic for iterative selection of nodes in the context of AL. This allows to learn a robust decision model by utilizing significantly lesser number of labeled nodes for training, as compared to conventional supervised learning approaches. Such a framework is highly beneficial for developing predictive engines using information from the agricultural fields, as the data is frequently impeded by absence of information owing to multiple factors like sensor dependability, data collection, communication disruptions, maintenance problems, and field operations.

6.4 Proposed Framework

This work proposes a novel uncertainty-guided framework to implement AL on spatio-temporal graphs. The learning scheme utilized in the proposed methodology involves a combination of DC-CNN and GNN. While DC-CNN addresses the temporal relations associated with individual nodes, GNN captures the spatial dependencies between different nodes of the given spatio-temporal graph G . The proposed framework is illustrated in Figure 6.1. A small fraction of nodes (10%) from G is chosen for initial training of the decision model. During this process, we systematically propagate the uncertainties associated with node features (assumed as sensor or measurement noise) through different layers of DC-CNN and GNN in order to finally obtain the uncertainty in model predictions. This is achieved by making use of ADF, followed by moment matching. We consider propagation and matching the means and variances of the output probability density functions of all DC-CNN and GNN layers. The uncertainty in model predictions is then leveraged to iteratively select the most informative nodes in the context of AL. This process is repeated until the query budget is exhausted.

This work proposes a novel uncertainty-guided framework to implement AL on spatio-temporal graphs. The learning scheme utilized in the proposed methodology involves a combination of DC-CNN and GNN. While DC-CNN addresses the temporal relations associated with individual nodes, GNN captures the spatial dependencies between different nodes of the given spatio-temporal graph G . The proposed framework is illustrated in Figure 6.1. A small fraction of nodes (10%) from G is chosen for initial training of the decision model. During this process, we systematically propagate the uncertainties associated with node features (assumed as sensor or measurement noise) through different layers of DC-CNN and GNN in order to finally obtain the uncertainty in model predictions. This is achieved by making use of ADF, followed by moment matching. We consider propagation and matching the means and variances of the output probability density functions of all DC-CNN and GNN layers. The uncertainty in model predictions is then leveraged to iteratively select the most informative nodes in the context of AL. This process is repeated until the query budget

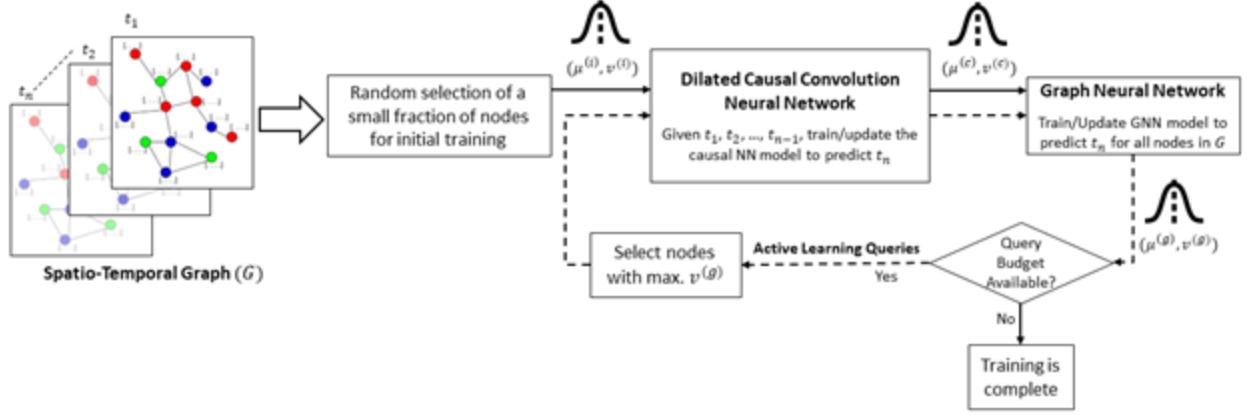


Figure 6.1: Proposed framework for uncertainty-guided AL on spatio-temporal graphs. The following result formalizes the propagation of uncertainty across DC-CNN and GNN layers.

is exhausted.

Theorem 6. *The expected value $(\mu_{u,t})$ and variance $(v_{u,t})$ of node embedding for node u at time t , accounting for aleatoric uncertainty with mean aggregation and linear activation functions are:*

$$\mu_{u,t}^{(i)} = \theta_{C,t}^{(i)} \mu_{u,t}^{(i-1)} + \theta_{A,t}^{(i)} \frac{n}{k |N(u)| D(u)} \sum_{s=0, \nu \in N(u)}^{k-1} p_{\nu u, t} \mu_{\nu, t}^{(i-1)} (t - \eta s) \quad (6.1)$$

$$v_{u,t}^{(i)} = \theta_{C,t}^{(i)2} v_{u,t}^{(i-1)} + \theta_{A,t}^{(i)2} \frac{n}{k |N(u)| D(u)} \sum_{s=0, \nu \in N(u)}^{k-1} p_{\nu u, t}^2 v_{\nu, t}^{(i-1)} (t - \eta s) \quad (6.2)$$

where, (i) corresponds to quantities of i^{th} layer, θ_C denotes the combination parameters, θ_A denotes the aggregation parameters, $N(u)$ is the neighborhood of the node u , $D(u)$ is the degree of node u and η is the dilation factor. $p_{\nu u, t}$ represents the probability of the link between nodes u and ν at time t .

Proof. For a node u , let the noisy feature vector be: $h_u \sim \mathcal{N}(h_u^*, \Sigma)$. The aggregation and the combination operations at the i^{th} node embedding layer is written as:

$$\begin{aligned}
h_u^{(i)} &= f\left(h_u^{(i-1)}, h_{N(u)}^{(i-1)}\right) \\
&= g(\theta_C^{(i)} h_u^{(i-1)}, \theta_A^{(i)} \tilde{A}(h_{N(u)}^{(i-1)}))
\end{aligned} \tag{6.3}$$

where, $g[\bullet]$ is the activation function and $\tilde{A}$ is the aggregation operator. At time t , for node u , the temporal representation learned by the causal convolution can be written as:

$$\begin{aligned}
h_{u,t}^{(i)} &= h_u^{(i-1)} * k^{(i)}(t) \\
&= \sum_{s=0}^{k-1} k^{(i)}(s) \odot h_u^{(i-1)}(t - \eta s)
\end{aligned} \tag{6.4}$$

where k is the kernel and $\odot$ is the Hadamard product. Using ADF, the joint density of all embeddings for a node u , i.e., $p(h_{u,(t-\eta s):t}^{(o:l)})$ can be approximated [124], [170]. This can be done by minimizing KL divergence and moment matching (first two moments) [170]:

$$\mu_{u,t}^{(i)} = \mathbb{E}_q(h_{u,t}^{(i-1)})[f^{(i)}(h_{u,t}^{(i-1)}, h_{N(u)}^{(i-1)}(t - \eta s))] \tag{6.5}$$

$$v_{u,t}^{(i)} = \text{var}_q(h_{u,t}^{(i-1)})[f^{(i)}(h_{u,t}^{(i-1)}, h_{N(u)}^{(i-1)}(t - \eta s))] \tag{6.6}$$

where, $\mathbb{E}$ is the expectation operator and var is the variance operator. The following assumptions were made: $\tilde{A}$ is a mean operator and $g[\bullet]$ is linear activation. As a result of these assumption, (i) the variance of mean is normalized form of mean of variances; and (ii) the mean of expectations is same as expectation of means.

By substituting equation 6.3 and 6.4 in equation 6.5 and 6.6 the mean and variance can be obtained as:

$$\mu_{u,t}^{(i)} = \theta_{C,t}^{(i)} \mu_{u,t}^{(i-1)} + \theta_{A,t}^{(i)} \frac{n}{k |N(u)| D(u)} \sum_{s=0, \nu \in N(u)}^{k-1} p_{\nu,t} \mu_{\nu,t}^{(i-1)}(t - \eta s) \tag{6.7}$$

$$v_{u,t}^{(i)} = \theta_{C,t}^{(i)^2} v_{u,t}^{(i-1)} + \theta_{A,t}^{(i)^2} \frac{n}{k |N(u)| D(u)} \sum_{s=0, \nu \in N(u)}^{k-1} p_{u\nu,t}^2 v_{\nu,t}^{(i-1)} (t - \eta s) \quad (6.8)$$

This proves Theorem 6. □

6.5 Results

We evaluate the proposed methodology over a field-scale sensor network data set for monitoring and modeling the spatial and temporal variation of soil health in a dryland agricultural field [171]. We consider hourly measurements of volumetric water content and soil temperature collected at 42 monitoring locations and 5 depths (30, 60, 90, 120, and 150 cm) across Cook Agronomy Farm [171]. A window of 1000 measurements is chosen for each measurement location at every depth. The next step is to create a connected graph G with 210 nodes (i.e., 42 locations at 5 depth). The Euclidean distances d_{ij} between all node pairs i and j were calculated, and an edge was considered between the ones where d_{ij} was found to be less than a pre-defined threshold distance d_T . The selection criteria for d_T was to ensure that the resulting graph is connected. Therefore, every node in G corresponds to a location with two sensors, soil temperature and volumetric water content, each with a hourly time-series measurement for 1000 time instances. The node features comprise of the truncated Singular Value Decomposition (SVD) of the weighted adjacency matrix [52]. This helps to improve the learning scheme by incorporating higher order neighbor embeddings, while keeping self-embedding separate from the neighbor embeddings in aggregation function of GNN [52]. The detailed architecture used in our experiments is as follows: (i) DC-CNN: number of convolution layers: 1; number of LSTM layers: 1; number of linear layers: 3; hidden dimensions: 48, 24; kernel size: 3; window size: 100 (ii) GNN: depth (i.e., number of node embedding modules): 2; number of neurons in 2 layers: 48, 36; number of multi-layer perceptron (MLP) layers: 2; activation function: linear (except last layer with softmax); aggregation function: mean. The training of GNN is carried out in a mini-batch manner. An initial labeled dataset was curated by randomly selecting around 10% of nodes in G . It

is assumed that both soil temperature and volumetric water content data are available for all these nodes for the first 100-time instances. Subsequently, this data is used to train an initial model as per the aforementioned architecture. During this process, the uncertainties arising from sensor or measurement noise are also propagated systematically through different layers of DC-CNN and GNN, as outlined in Theorem 6. After training an initial model, the querying procedure is commenced, and the decision model is updated after each query step. We use the uncertainty in model predictions obtained during the previous step as a heuristic to iteratively select the most informative nodes at each successive query step. Around 30% of the nodes in G are used for AL queries and 40% are held out separately for testing.

We have designed four tasks in order to demonstrate the utility of our proposed framework. The first two tasks are intended to predict soil temperature (TR) and volumetric water content (VWR) for the test nodes during all time instances by using truncated SVD values as node features. The third task is designed to predict TR by using both truncated SVD values and VWR measurements as node features. The last task is to predict VWR by using a combination of truncated SVD values and TR measurements as node features. We use Mean Squared Error (MSE) and Predictive Performance (PP) as metrics to evaluate the performance of the proposed methodology over different tasks. PP is calculated using eq. (7). Here, A_i and P_i are actual and predicted values, respectively, and n is the total number of observations.

$$\text{Predictive performance (PP)} = 100 - \frac{100}{n} \sum_{i=1}^n \left| \frac{A_i - P_i}{A_i} \right| \quad (6.9)$$

Firstly, we establish the benefits of utilizing AL for strategic selection of nodes for model training during each query step. The random selection of nodes from the dataset is treated as the baseline. Table 6.1 illustrates the percentage of node labels required to achieve 90% predictive performance. It can be observed that the proposed framework achieves a predictive performance of 90% using 30% or lesser node labels for all the tasks. On the other hand, 65%–70% of node labels are required to achieve the same level of performance using baseline,

Sr. No.	Task	Baseline	Proposed Framework
1	Prediction of TR	71.42%	30.95%
2	Prediction of VWR	70.47%	29.52%
3	Prediction of TR based on VWR	64.28%	28.57%
4	Prediction of VWR based on TR	65.23%	27.61%

Table 6.1: Percentage of node labels required to achieve 90% predictive performance.

Sr. No.	Task	Performance Metrics after number of queries					
		20		40		60	
		MSE	PP	MSE	PP	MSE	PP
1	Prediction of TR	0.1123	52.23%	0.0911	77.90%	0.0537	92.07%
2	Prediction of VWR	0.1345	51.54%	0.1032	75.15%	0.0625	92.85%
3	Prediction of TR based on VWR	0.1056	55.87%	0.0744	81.33%	0.0498	94.67%
4	Prediction of VWR based on TR	0.0988	56.12%	0.0779	80.47%	0.0386	96.58%

Table 6.2: Performance Evaluation at different stages of Active Learning.

i.e., if the node labels are selected at random for model training. This is because the proposed framework strategically selects the nodes with highest uncertainty at each query step, thereby improving the label efficiency. Such an approach is highly beneficial for developing predictive engines using information from the agricultural fields, where data availability is sparse. It can also be seen that the percentage of node labels required to achieve 90% PP is lesser for tasks 3 and 4, as compared to those for tasks 1 and 2. This is intuitive because a larger amount of information is utilized as node features for model training in these tasks, namely, VWR for task 3 and TR for task 4, respectively, which is missing in tasks 1 and 2.

Secondly, we demonstrate the efficacy of AL approach by evaluating performance for all the tasks after specific query steps. The values of MSE and PP after regular intervals of query steps are provided in Table 6.2. It is clearly observed that the values of MSE reduce and PP improves as more queries are made. This indicates the incremental learning of the decision model, as it is updated after each query step. Once again, it can be observed that the performance of decision model is better for tasks 3 and 4 as compared to tasks 1 and 2. This is because additional information is incorporated as node features during model training for these tasks. We also capture the uncertainty associated with model predictions for all the tasks after specific query steps, as outlined in Table 6.3. These uncertainty values were

Sr. No.	Task	Uncertainty after number of queries		
		20	40	60
1	Prediction of TR	0.0823	0.0621	0.0408
2	Prediction of VWR	0.0965	0.0612	0.0536
3	Prediction of TR based on VWR	0.0809	0.0577	0.0245
4	Prediction of VWR based on TR	0.0767	0.0448	0.0287

Table 6.3: Uncertainty Evaluation at different stages of Active Learning (considering 0.10 at input).

Sr. No.	Node Instance	Performance Metrics after number of queries					
		20		40		60	
		PP	Uncertainty	PP	Uncertainty	PP	Uncertainty
1	13	55.41%	0.0798	82.08%	0.0542	95.31%	0.0265
2	46	56.22%	0.0877	81.53%	0.0603	94.87%	0.0247
3	88	54.85%	0.0843	83.16%	0.0528	94.63%	0.0316
4	162	55.91%	0.0812	81.74%	0.0537	93.96%	0.0255

Table 6.4: Longitudinal Studies for specific nodes: Prediction of TR based on VWR (uncertainty measured by considering 0.10 at input).

calculated by considering that the sensor measurements are corrupted with 10% Gaussian noise, i.e., an input uncertainty of 0.10. It was observed that the values of uncertainties reduce as more queries are made. This is intuitive because learning of the decision model improves with each query step, as the information about strategically selected uncertain nodes is incorporated in the learning scheme. Table 6.2. Performance Evaluation at different stages of Active Learning.

Finally, the effectiveness of the proposed framework is demonstrated by carrying out longitudinal studies corresponding to specific nodes in G for task of predicting TR based on VWR. A few nodes were selected at random and the variations in PP and uncertainty values are examined. Table 6.4 presents a summary of the results for four randomly selected nodes from G. It can be seen that the predictive performance improves, and uncertainty reduces as more queries are made. These observations illustrate the systematic propagation of uncertainties across all layers of DC-CNN and GNN, as well as incremental learning of the decision model after subsequent query steps in the context of AL.

6.6 Summary

In this chapter, we propose a novel framework to perform AL on spatio-temporal graphs. Specifically, we developed an uncertainty-guided learning scheme that integrates a DC-CNN to capture temporal relationships at individual nodes with a GNN to model spatial dependencies among nodes. The framework systematically propagates uncertainty in node features through the DC-CNN and GNN layers using an approximate density filtering (ADF) approach, followed by moment matching to obtain robust uncertainty estimates in the model predictions. This allows for more reliable predictions by quantifying uncertainties in data through confidence intervals around the predictions. The uncertainty in model predictions is then leveraged to iteratively select the most informative nodes in the context of AL. It is worthwhile to note that the proposed methodology is generic and hence, agnostic to learning architecture, AL setup and query strategy. Consequently, this framework is particularly well-suited for data-limited scenarios where strategic selection of data points is critical. The proposed methodology is evaluated by conducting experiments over a field-scale sensor network data set for monitoring and modeling the spatial and temporal variation of soil health in a dryland agricultural field. Experimental results demonstrate the applicability of the proposed methodology to train a decision model for soil moisture and temperature monitoring using a limited number of sensor measurements which are strategically selected to minimize uncertainty in model predictions.

A similar approach can be applied to biological applications, such as monitoring the tumor microenvironment to estimate physiological states and optimize treatment strategies. The framework can assist in determining when and where to sample measurements or acquire imaging data, ultimately enhancing decision-making in clinical settings.

Chapter 7

Conclusions and Future Work

7.1 Conclusions

This dissertation addresses the challenges associated with inferencing from sparse spatio-temporal data, specifically focusing on biological systems. While existing statistical parametric models and machine learning frameworks have addressed certain aspects of inferencing in sparse spatio-temporal data, they both have their limitations. This thesis is an attempt to combine the strengths of both statistical and machine learning approaches while trying to mitigate their disadvantages.

The key contributions of this thesis can be summarized as follows: To address the imputation dilemma, in Chapter 2 we proposed an Informative Missing Indicator Method (IMIM), a theoretical framework that determines when imputation does not introduce bias or information loss, particularly in deep learning models. Additionally, we developed a spatio-temporal imputation framework in Chapter 3 that leverages graph neural networks (GNNs) and recurrent neural networks (RNNs) to systematically capture underlying spatial and temporal dependencies in data, thereby enhancing predictive accuracy. Together these approaches combined can help impute gaps in sparse spatio-temporal data without loss of information and bias and improve predictive accuracy on downstream tasks.

Since safety is a critical aspect of healthcare applications, where biological data is used, it

is equally important to quantify the confidence associated with model predictions. To achieve this, we introduced a Neural Stochastic Differential Equation (Neural SDE) and Graph Neural SDE framework in Chapter 4, which provides a generalized, task-agnostic approach to quantifying epistemic uncertainty in deep neural networks and graph neural networks, respectively. This framework enables better-calibrated probability estimates, ensuring that machine learning models provide not only accurate predictions but also reliable uncertainty estimates that are crucial for safety-critical applications.

We further demonstrate how machine learning frameworks can be used to make meaningful predictions under limited-data conditions in Chapter 5. To improve generalizability and ensure that predictions conform to scientific principles, we integrated domain knowledge into machine learning models in the form of observational and learning biases. In the case of predicting thermal ablation in biological tissue, we proposed a Bayesian physics-informed machine learning framework, with improved prediction accuracy and analytically estimated uncertainty quantified on the predictions. We also show how the quantified uncertainty can be further used to estimate uncertain quantities that cannot be directly measured. In the context of biomarker identification for traumatic brain injury (TBI), we introduced a graph-based machine learning approach that incorporated prior biological knowledge as observational bias. Specifically, we leveraged known protein-protein interactions from an existing database, resulting in more biologically meaningful predictions while also quantifying uncertainty.

Finally, we proposed an uncertainty-guided framework in Chapter 6 to efficiently sample the most informative data points while taking into account the spatio-temporal variations. We introduced an uncertainty-guided active learning framework for spatio-temporal graphs, which systematically propagates measurement uncertainty and utilizes it to guide data acquisition. This framework optimizes the selection of the most informative samples, reducing overall model uncertainty and improving predictive performance, particularly in data limited setups such as healthcare.

7.2 Future work

- In Chapter 2, we explored how the imputation dilemma can be addressed to preserve the informativeness of data. Handling Missing Not at Random (MNAR) data is particularly challenging, as it depends on unobserved values, whereas Missing at Random (MAR) is comparatively easier to manage. Most imputation frameworks are designed for MCAR or MAR scenarios, while MNAR requires modeling the missingness mechanism itself. In such cases, incorporating physics constraints could help us better understand the type of missingness and enable more meaningful imputations. Since graph-structured data is common in spatio-temporal contexts, extending the informative missing indicator method to graph neural networks would allow us to account for graph structure when assessing informative missingness
- We discussed various types of uncertainty in Chapter 4. In many healthcare applications, uncertainty in ground truth labels is common, especially in scenarios involving subjective clinical judgment or multiple expert opinions. These uncertainties are often irreducible. While consensus-based approaches can help determine a probable ground truth, they can also be time-consuming and labor-intensive. Quantifying label-related uncertainty could be an interesting direction for future research
- In Chapter 5, we demonstrated how incorporating prior knowledge into machine learning models can enhance generalizability. A promising future direction involves embedding causal inference techniques into physics-informed models, enabling deeper insights into biological processes and improving model interpretability.
- In Chapter 5, we introduced a Bayesian physics-informed machine learning framework for solving inverse problems, i.e., estimating unknown parameters from observed data. Currently, this framework is designed for static setups. However, in many real-world applications, as data evolves over time, these uncertain parameters may also change. Developing a dynamic approach to solve such time-varying inverse problems would be a valuable extension

- The active learning framework proposed in Chapter 6 is currently tailored for spatio-temporal graphs with homogeneous sensor data. Expanding this methodology to accommodate multi-modal data, such as imaging, clinical records, and genomics, could significantly enhance its applicability in real-world biomedical applications.

Bibliography

- [1] Reza Asadi and Amelia Regan. A convolution recurrent autoencoder for spatio-temporal missing data imputation. *arXiv preprint arXiv:1904.12413*, 2019.
- [2] Christopher K Wikle, Andrew Zammit-Mangion, and Noel Cressie. *Spatio-temporal statistics with R*. Chapman and Hall/CRC, 2019.
- [3] José-María Montero, Gema Fernández-Avilés, and Jorge Mateu. *Spatial and spatio-temporal geostatistical modeling and kriging*. John Wiley & Sons, 2015.
- [4] Andrew Zammit-Mangion and Noel Cressie. Fixed rank kriging: the r package. *URL <https://cran.r-project.org/web/packages/FRK/index.html>*, 2021.
- [5] Sergios Theodoridis and Konstantinos Koutroumbas. *Pattern recognition*. Elsevier, 2006.
- [6] Pengcheng Xu, Xiaobo Ji, Minjie Li, and Wencong Lu. Small data machine learning in materials science. *npj Computational Materials*, 9(1):42, 2023.
- [7] Torgyn Shaikhina, Dave Lowe, Sunil Daga, David Briggs, Robert Higgins, and Natasha Khovanova. Machine learning for predictive modelling based on small data in biomedical engineering. *IFAC-PapersOnLine*, 48(20):469–474, 2015.
- [8] Fabian Guignard. *On Spatio-Temporal Data Modelling and Uncertainty Quantification Using Machine Learning and Information Theory*. Springer Nature, 2022.
- [9] Leo Breiman. Statistical modeling: The two cultures (with comments and a rejoinder by the author). *Statistical science*, 16(3):199–231, 2001.

- [10] Konstantinos Poulinakis, Dimitris Drikakis, Ioannis W Kokkinakis, and Stephen Michael Spottswood. Machine-learning methods on noisy and sparse data. *Mathematics*, 11(1):236, 2023.
- [11] Haoxing Chen, Huaxiong Li, Yaohui Li, and Chunlin Chen. Sparse spatial transformers for few-shot learning. *Science China Information Sciences*, 66(11):210102, 2023.
- [12] Charu C Aggarwal, Xiangnan Kong, Quanquan Gu, Jiawei Han, and Philip S Yu. Active learning: A survey. In *Data classification*, pages 599–634. Chapman and Hall/CRC, 2014.
- [13] Rayan Krishnan, Pranav Rajpurkar, and Eric J Topol. Self-supervised learning in medicine and healthcare. *Nature Biomedical Engineering*, 6(12):1346–1352, 2022.
- [14] Cátia M Salgado, Carlos Azevedo, Hugo Proença, and Susana M Vieira. Missing data. *Secondary analysis of electronic health records*, pages 143–162, 2016.
- [15] Noel Cressie and Christopher K Wikle. *Statistics for spatio-temporal data*. John Wiley & Sons, 2011.
- [16] Thomas R Sullivan, Ian R White, Amy B Salter, Philip Ryan, and Katherine J Lee. Should multiple imputation be the method of choice for handling missing data in randomized trials? *Statistical methods in medical research*, 27(9):2610–2626, 2018.
- [17] Hyun Kang. The prevention and handling of the missing data. *Korean journal of anesthesiology*, 64(5):402, 2013.
- [18] Tlamelo Emmanuel, Thabiso Maupong, Dimane Mpoeleng, Thabo Semong, Banyatsang Mphago, and Oteng Tabona. A survey on missing data in machine learning. *Journal of Big data*, 8:1–37, 2021.
- [19] Rolf HH Groenwold, Ian R White, A Rogier T Donders, James R Carpenter, Douglas G Altman, and Karel GM Moons. Missing covariate data in clinical research: when and

- when not to use the missing-indicator method for analysis. *Cmaj*, 184(11):1265–1269, 2012.
- [20] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- [21] Mike Van Ness, Tomas M Bosschieter, Roberto Halpin-Gregorio, and Madeleine Udell. The missing indicator method: From low to high dimensions. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 5004–5015, 2023.
- [22] Thomas F Johnson, Nick JB Isaac, Agustin Paviolo, and Manuela González-Suárez. Handling missing values in trait data. *Global Ecology and Biogeography*, 30(1):51–62, 2021.
- [23] Aabila Tharzeen, Sai Munikoti, Punit Prakash, Jungkwun Kim, and Balasubramaniam Natarajan. A general spatiotemporal imputation framework for missing sensor data. In *2023 IEEE Conference on Artificial Intelligence (CAI)*, pages 55–58. IEEE, 2023.
- [24] Shahidul Islam Khan and Abu Sayed Md Latiful Hoque. Sice: an improved missing data imputation technique. *Journal of big Data*, 7(1):37, 2020.
- [25] Mehmet S Aktaş, Sinan Kaplan, Hasan Abacı, Oya Kalipsiz, Utku Ketenci, and Umut O Turgut. Data imputation methods for missing values in the context of clustering. In *Big data and knowledge sharing in virtual organizations*, pages 240–274. IGI Global, 2019.
- [26] Utomo Pujianto, Aji Prasetya Wibawa, Muhammad Iqbal Akbar, et al. K-nearest neighbor (k-nn) based missing data imputation. In *2019 5th International Conference on Science in Information Technology (ICSITech)*, pages 83–88. IEEE, 2019.

- [27] Adrienne D Woods, Pamela Davis-Kean, Max Andrew Halvorson, Kevin King, Jessica Logan, Menglin Xu, Sierra Bainter, Denver Brown, James M Clay, Rick Anthony Cruz, et al. Missing data and multiple imputation decision tree. 2021.
- [28] Leo Breiman. Random forests. *Machine learning*, 45:5–32, 2001.
- [29] Buliao Huang, Yunhui Zhu, Muhammad Usman, Xiren Zhou, and Huanhuan Chen. Graph neural networks for missing value classification in a task-driven metric space. *IEEE Transactions on Knowledge and Data Engineering*, 35(8):8073–8084, 2022.
- [30] Yimei Zhang, Xiangjie Kong, Wenfeng Zhou, Jin Liu, Yanjie Fu, and Guojiang Shen. A comprehensive survey on traffic missing data imputation. *IEEE Transactions on Intelligent Transportation Systems*, 2024.
- [31] Xiaofeng Yuan, Jiale Zhang, Kai Wang, Yalin Wang, Chunhua Yang, Weihua Gui, Feifan Shen, and Lingjian Ye. Missing data imputation for industrial time series with adaptive median iteration based on generative adversarial networks. *IEEE Sensors Journal*, 2024.
- [32] Rolf HH Groenwold. Informative missingness in electronic health record systems: the curse of knowing. *Diagnostic and prognostic research*, 4(1):8, 2020.
- [33] Shengwen Xie and Juan Ren. Linearization of recurrent-neural-network-based models for predictive control of nano-positioning systems using data-driven koopman operators. *IEEE Access*, 8:147077–147088, 2020.
- [34] Téo Sanchez, Baptiste Caramiaux, Pierre Thiel, and Wendy E Mackay. Deep learning uncertainty in machine teaching. In *27th International Conference on Intelligent User Interfaces*, pages 173–190, 2022.
- [35] Arash Amini, Guangyi Liu, and Nader Motee. Robust learning of recurrent neural networks in presence of exogenous noise. In *2021 60th IEEE Conference on Decision and Control (CDC)*, pages 783–788. IEEE, 2021.

- [36] Matthias Feurer, Jan N Van Rijn, Arlind Kadra, Pieter Gijsbers, Neeratyoy Mallik, Sahithya Ravi, Andreas Müller, Joaquin Vanschoren, and Frank Hutter. Openml-python: an extensible python api for openml. *Journal of Machine Learning Research*, 22(100):1–5, 2021.
- [37] Bumjoon Bae, Hyun Kim, Hyeonsup Lim, Yuandong Liu, Lee D Han, and Phillip B Freeze. Missing data imputation for traffic flow speed using spatio-temporal cokriging. *Transportation Research Part C: Emerging Technologies*, 88:124–139, 2018.
- [38] Hongtai Yang, Jianjiang Yang, Lee D Han, Xiaohan Liu, Li Pu, Shih-miao Chin, and Ho-ling Hwang. A kriging based spatiotemporal approach for traffic volume data imputation. *PloS one*, 13(4):e0195957, 2018.
- [39] Andrea Cini, Ivan Marisca, and Cesare Alippi. Filling the gaps: Multivariate time series imputation by graph neural networks. *arXiv preprint arXiv:2108.00298*, 2021.
- [40] Xinyu Chen, Mengying Lei, Nicolas Saunier, and Lijun Sun. Low-rank autoregressive tensor completion for spatiotemporal traffic data imputation. *IEEE Transactions on Intelligent Transportation Systems*, 2021.
- [41] Hanpeng Liu, Yaguang Li, Michael Tsang, and Yan Liu. Costco: A neural tensor completion model for sparse tensors. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 324–334, 2019.
- [42] Jendrik Poloczek, Nils André Treiber, and Oliver Kramer. Knn regression as geo-imputation method for spatio-temporal wind data. In *International Joint Conference SOCO’14-CISIS’14-ICEUTE’14*, pages 185–193. Springer, 2014.
- [43] Ningrinla Marchang and Rakesh Tripathi. Knn-st: Exploiting spatio-temporal correlation for missing data inference in environmental crowd sensing. *IEEE Sensors Journal*, 21(3):3429–3436, 2020.
- [44] Yanjie Duan, Yisheng Lv, Yu-Liang Liu, and Fei-Yue Wang. An efficient realization

- of deep learning for traffic data imputation. *Transportation research part C: emerging technologies*, 72:168–181, 2016.
- [45] Yongchao Ye, Shiyao Zhang, and James JQ Yu. Spatial-temporal traffic data imputation via graph attention convolutional network. In *International Conference on Artificial Neural Networks*, pages 241–252. Springer, 2021.
- [46] Yuankai Wu, Dingyi Zhuang, Aurelie Labbe, and Lijun Sun. Inductive graph neural networks for spatiotemporal kriging. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 4478–4485, 2021.
- [47] Sanmukh R Kuppannagari, Yao Fu, Chung Ming Chueng, and Viktor K Prasanna. Spatio-temporal missing data imputation for smart power grids. In *Proceedings of the Twelfth ACM International Conference on Future Energy Systems*, pages 458–465, 2021.
- [48] Abhilasha J Saroj, Angshuman Guin, and Michael Hunter. Deep lstm recurrent neural networks for arterial traffic volume data imputation. *Journal of Big Data Analytics in Transportation*, 3(2):95–108, 2021.
- [49] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017.
- [50] Yong Yu, Xiaosheng Si, Changhua Hu, and Jianxun Zhang. A review of recurrent neural networks: Lstm cells and network architectures. *Neural computation*, 31(7):1235–1270, 2019.
- [51] Bing Yu, Haoteng Yin, and Zhanxing Zhu. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. *arXiv preprint arXiv:1709.04875*, 2017.
- [52] Vijay Lingam, Rahul Ragesh, Arun Iyer, and Sundararajan Sellamanickam. Simple truncated svd based model for node classification on heterophilic graphs. *arXiv preprint arXiv:2106.12807*, 2021.

- [53] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [54] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural network. In *International conference on machine learning*, pages 1613–1622. PMLR, 2015.
- [55] Alex Graves. Practical variational inference for neural networks. *Advances in neural information processing systems*, 24, 2011.
- [56] Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.
- [57] Matthew A Kupinski, Darrin C Edwards, Maryellen L Giger, and Charles E Metz. Ideal observer approximation using bayesian classification neural networks. *IEEE transactions on medical imaging*, 20(9):886–899, 2001.
- [58] Y Gal and Z Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning [eb/ol]. *arXiv preprint arxiv:1506.02142*, 2015.
- [59] B Lakshminarayanan, A Pritzel, and C Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. arxiv. *arXiv preprint arXiv:1612.01474*, 2016.
- [60] Lingkai Kong, Jimeng Sun, and Chao Zhang. Sde-net: Equipping deep neural networks with uncertainty estimates. *arXiv preprint arXiv:2008.10546*, 2020.
- [61] Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.
- [62] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. corr abs/1512.03385 (2015), 2015.

- [63] Xuanqing Liu, Tesi Xiao, Si Si, Qin Cao, Sanjiv Kumar, and Cho-Jui Hsieh. Neural sde: Stabilizing neural ode networks with stochastic noise. *arXiv preprint arXiv:1906.02355*, 2019.
- [64] Xuanqing Liu, Tesi Xiao, Si Si, Qin Cao, Sanjiv Kumar, and Cho-Jui Hsieh. How does noise help robustness? explanation and exploration under the neural sde framework. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 282–290, 2020.
- [65] Simo Särkkä and Arno Solin. *Applied stochastic differential equations*, volume 10. Cambridge University Press, 2019.
- [66] R Kelley Pace and Ronald Barry. Sparse spatial autoregressions. *Statistics Probability Letters*, 33(3):291–297, 1997.
- [67] Wing S. Hutton L. Sigillito, V. and K. Baker. Ionosphere. UCI Machine Learning Repository, 1989. DOI: <https://doi.org/10.24432/C5W01B>.
- [68] Dan Levi, Liran Gispan, Niv Giladi, and Ethan Fetaya. Evaluating and calibrating uncertainty prediction in regression tasks. *Sensors*, 22(15):5540, 2022.
- [69] Alexander Pauls and J Yoder. Determining optimum drop-out rate for neural networks. In *Midwest Instructional Computing Symposium (MICS)*, 2018.
- [70] J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun. Graph neural networks: A review of methods and applications. *AI Open*, 1:57–81, 2020. doi: 10.1016/j.aiopen.2021.01.001.
- [71] Fangxin Wang, Yuqing Liu, Kay Liu, Yibo Wang, Sourav Medya, and Philip S Yu. Uncertainty in graph neural networks: A survey. *arXiv preprint arXiv:2403.07185*, 2024.

- [72] Puja Trivedi, Mark Heimann, Rushil Anirudh, Danai Koutra, and Jayaraman J Thiragarajan. Accurate and scalable estimation of epistemic uncertainty for graph neural networks. *arXiv preprint arXiv:2401.03350*, 2024.
- [73] Soroor Motie and Bijan Raahemi. Financial fraud detection using graph neural networks: A systematic review. *Expert Systems with Applications*, 240:122156, 2024.
- [74] Heloísa Oss Boll, Ali Amirahmadi, Mirfarid Musavian Ghazani, Wagner Ourique de Moraes, Edison Pignaton de Freitas, Amira Soliman, Farzaneh Etminani, Stefan Byttner, and Mariana Recamonde-Mendoza. Graph neural networks for clinical risk prediction based on electronic health records: A survey. *J. Biomed. Informatics*, 151: 104616, 2024.
- [75] Chuhan Wu, Fangzhao Wu, Lingjuan Lyu, Tao Qi, Yongfeng Huang, and Xing Xie. A federated graph neural network framework for privacy-preserving personalization. *Nature Communications*, 13(1):3091, 2022.
- [76] Shankar Sankararaman and Sankaran Mahadevan. Likelihood-based representation of epistemic uncertainty due to sparse point data and/or interval data. *Reliability Engineering & System Safety*, 96(7):814–824, 2011.
- [77] Moloud Abdar, Farhad Pourpanah, Sadiq Hussain, Dana Rezazadegan, Li Liu, Mohammad Ghavamzadeh, Paul Fieguth, Xiaochun Cao, Abbas Khosravi, U Rajendra Acharya, et al. A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Information Fusion*, 76:243–297, 2021.
- [78] Maximilian Stadler, Bertrand Charpentier, Simon Geisler, Daniel Zugner, and Stephan Gunnemann. Graph posterior network: Bayesian predictive uncertainty for node classification. *ArXiv*, abs/2110.14012, 2021. URL <https://api.semanticscholar.org/CorpusID:239998097>.
- [79] Rohitash Chandra, Ayush Bhagat, Manavendra Maharana, and Pavel N Krivitsky.

- Bayesian graph convolutional neural networks via tempered mcmc. *IEEE Access*, 9: 130353–130365, 2021.
- [80] Sai Munikoti, Laya Das, and Balasubramaniam Natarajan. Bayesian graph neural network for fast identification of critical nodes in uncertain complex networks. In *2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 3245–3251. IEEE, 2021.
- [81] Arman Hasanzadeh, Ehsan Hajiramezanali, Shahin Boluki, Mingyuan Zhou, Nick G. Duffield, Krishna R. Narayanan, and Xiaoning Qian. Bayesian graph neural networks with adaptive connection sampling. In *International Conference on Machine Learning*, 2020. URL <https://api.semanticscholar.org/CorpusID:219530761>.
- [82] Md Farhadul Islam, Sarah Zabeen, Fardin Bin Rahman, Md Azharul Islam, Fahmid Bin Kibria, Meem Arafat Manab, Dewan Ziaul Karim, and Annajiat Alim Rasel. Exploring node classification uncertainty in graph neural networks. In *Proceedings of the 2023 ACM Southeast Conference*, pages 186–190, 2023.
- [83] Jian Kang, Qinghai Zhou, and Hanghang Tong. Jurygn: quantifying jackknife uncertainty on graph convolutional networks. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 742–752, 2022.
- [84] Seohyeon Cha, Honggu Kang, and Joonhyuk Kang. On the temperature of bayesian graph neural networks for conformal prediction. *arXiv preprint arXiv:2310.11479*, 2023.
- [85] Richard Bergna, Felix Opolka, Pietro Liò, and Jose Miguel Hernandez-Lobato. Graph neural stochastic differential equations. *arXiv preprint arXiv:2308.12316*, 2023.
- [86] Yaofeng Desmond Zhong, Biswadip Dey, and Amit Chakraborty. Symplectic ode-net: Learning hamiltonian dynamics with control. *arXiv preprint arXiv:1909.12077*, 2019.
- [87] Shweta Dahale, Sai Munikoti, and Balasubramaniam Natarajan. A general framework

- for uncertainty quantification via neural sde-rnn. *arXiv preprint arXiv:2306.01189*, 2023.
- [88] Richard Bergna, Sergio Calvo-Ordóñez, Felix L Opolka, Pietro Liò, and Jose Miguel Hernandez-Lobato. Uncertainty modeling in graph neural networks via stochastic differential equations. *arXiv preprint arXiv:2408.16115*, 2024.
- [89] Matthias Fey and Jan Eric Lenssen. Fast graph representation learning with pytorch geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019.
- [90] Wolfgang Schramm, Deshan Yang, and Dieter Haemmerich. Contribution of direct heating, thermal conduction and perfusion during radiofrequency and microwave ablation. In *2006 International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 5013–5016. IEEE, 2006.
- [91] Kerri A Simo, Victor B Tsirlin, David Sindram, Matthew T McMillan, Kyle J Thompson, Ryan Z Swan, Iain H McKillop, John B Martinie, and David A Iannitti. Microwave ablation using 915-mhz and 2.45-ghz systems: what are the differences? *Hpb*, 15(12):991–996, 2013.
- [92] Chris Brace. Thermal tumor ablation in clinical use. *IEEE pulse*, 2(5):28–38, 2011.
- [93] Katrina F Chu and Damian E Dupuy. Thermal ablation of tumours: biological mechanisms and advances in therapy. *Nature Reviews Cancer*, 14(3):199–208, 2014.
- [94] Caroline J Simon, Damian E Dupuy, and William W Mayo-Smith. Microwave ablation: principles and applications. *Radiographics*, 25(suppl_1):S69–S83, 2005.
- [95] Pegah Faridi, Stefan H Bossmann, and Punit Prakash. Simulation-based design and characterization of a microwave applicator for mr-guided hyperthermia experimental studies in small animals. *Biomedical physics & engineering express*, 6(1):015001, 2019.

- [96] Jun Ying Tan, Malea Williams, Santosh Kumar Mandal, Anna Bottiglieri, Aabila Tharzeen, Rahul Sheth, Balasubramaniam Natarajan, Punit Prakash, et al. Development of a multi-modal electrochemical sensing (mes) device for real-time monitoring of tumor microenvironment parameters in cancer immunotherapy. In *2024 IEEE 37th International Conference on Micro Electro Mechanical Systems (MEMS)*, pages 441–444. IEEE, 2024.
- [97] Xinyi Wang, Shiqing Zhao, and Aili Zhang. Image-based monitoring of thermal ablation. *Bioengineering*, 12(1):78, 2025.
- [98] Pegah Faridi, Stefan H Bossmann, and Punit Prakash. Image-guided cancer thermal therapies. *Magnetic Nanomaterials: Applications in Catalysis and Life Sciences*, ed. S. H. Bossmann and H. Wang, The Royal Society of Chemistry,, 2017.
- [99] Punit Prakash. Microwave ablation: Physical principles and technology. In *Principles and Technologies for Electromagnetic Energy Based Therapies*, pages 139–167. Elsevier, 2022.
- [100] Christopher L Brace. Microwave tissue ablation: biophysics, technology, and applications. *Critical ReviewsTM in Biomedical Engineering*, 38(1), 2010.
- [101] Amirreza Heshmat, Caleb S O’Connor, Jessica Albuquerque Marques Silva, Iwan Paolucci, Aaron Kyle Jones, Bruno C Odisio, and Kristy K Brock. Using patient-specific 3d modeling and simulations to optimize microwave ablation therapy for liver cancer. *Cancers*, 16(11):2095, 2024.
- [102] Xiaoqian Song, Maokun Li, Fan Yang, Shenheng Xu, and Aria Abubakar. Three-dimensional joint inversion of em and acoustic data based on contrast source inversion. *IEEE Journal on Multiscale and Multiphysics Computational Techniques*, 5: 28–36, 2020.
- [103] Ulrich Katscher and Cornelius AT van den Berg. Electric properties tomography:

- Biochemical, physical and technical background, evaluation and clinical applications. *NMR in Biomedicine*, 30(8):e3729, 2017.
- [104] Viola Rieke and Kim Butts Pauly. Mr thermometry. *Journal of Magnetic Resonance Imaging: An Official Journal of the International Society for Magnetic Resonance in Medicine*, 27(2):376–390, 2008.
 - [105] Shengze Cai, Zhiping Mao, Zhicheng Wang, Minglang Yin, and George Em Karniadakis. Physics-informed neural networks (pinns) for fluid mechanics: A review. *Acta Mechanica Sinica*, 37(12):1727–1738, 2021.
 - [106] Minwoo Shin, Minjee Seo, Seonaeng Cho, Juil Park, Joon Ho Kwon, Deukhee Lee, and Kyungho Yoon. Physrfanet: Physics-guided neural network for real-time prediction of thermal effect during radiofrequency ablation treatment. *Engineering Applications of Artificial Intelligence*, 138:109349, 2024.
 - [107] Victor Rosario Núñez, Alfonso Hernández, Iván Rodríguez, Ignacio Fernández-Pacheco Ruiz, and Luis Acevedo. Artificial intelligence applied to microwave heating systems: Prediction of temperature profile through convolutional neural networks. *Thermo*, 4(3), 2024.
 - [108] Zhongkai Hao, Songming Liu, Yichi Zhang, Chengyang Ying, Yao Feng, Hang Su, and Jun Zhu. Physics-informed machine learning: A survey on problems, methods and applications. *arXiv preprint arXiv:2211.08064*, 2022.
 - [109] Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics*, 378: 686–707, 2019.
 - [110] Suchuan Dong and Yiran Wang. A method for computing inverse parametric pde problems with random-weight neural networks. *Journal of Computational Physics*, 489:112263, 2023.

- [111] Yongchao Li, Yanyan Wang, and Liang Yan. Surrogate modeling for bayesian inverse problems based on physics-informed neural networks. *Journal of Computational Physics*, 475:111841, 2023.
- [112] Harbir Antil, Howard C Elman, Akwum Onwunta, and Deepanshu Verma. Novel deep neural networks for solving bayesian statistical inverse. *arXiv preprint arXiv:2102.03974*, 2021.
- [113] Marija Radmilović-Radjenović, Nikola Bošković, and Branislav Radjenović. Computational modeling of microwave tumor ablation. *Bioengineering*, 9(11):656, 2022.
- [114] George Em Karniadakis, Ioannis G Kevrekidis, Lu Lu, Paris Perdikaris, Sifan Wang, and Liu Yang. Physics-informed machine learning. *Nature Reviews Physics*, 3(6):422–440, 2021.
- [115] Martin Genzel, Jan Macdonald, and Maximilian März. Solving inverse problems with deep neural networks—robustness included? *IEEE transactions on pattern analysis and machine intelligence*, 45(1):1119–1134, 2022.
- [116] Simon Arridge, Peter Maass, Ozan Öktem, and Carola-Bibiane Schönlieb. Solving inverse problems using data-driven models. *Acta Numerica*, 28:1–174, 2019.
- [117] Enrui Zhang, Ming Dao, George Em Karniadakis, and Subra Suresh. Analyses of internal structures and defects in materials using physics-informed neural networks. *Science advances*, 8(7):eabk0644, 2022.
- [118] Han Gao, Matthew J Zahr, and Jian-Xun Wang. Physics-informed graph neural galerkin networks: A unified framework for solving pde-governed forward and inverse problems. *Computer Methods in Applied Mechanics and Engineering*, 390:114502, 2022.
- [119] Liu Yang, Xuhui Meng, and George Em Karniadakis. B-pinns: Bayesian physics-informed neural networks for forward and inverse pde problems with noisy data. *Journal of Computational Physics*, 425:109913, 2021.

- [120] Dhruv V Patel, Deep Ray, and Assad A Oberai. Solution of physics-based bayesian inverse problems with deep generative priors. *Computer Methods in Applied Mechanics and Engineering*, 400:115428, 2022.
- [121] Tetsuro Morimura, Takayuki Osogami, and Tsuyoshi Idé. Solving inverse problem of markov chain with partial observations. *Advances in neural information processing systems*, 26, 2013.
- [122] Pegah Faridi, Paul Keselman, Hojjatollah Fallahi, and Punit Prakash. Experimental assessment of microwave ablation computational modeling with mr thermometry. *Medical physics*, 47(9):3777–3788, 2020.
- [123] Pegah Faridi and Punit Prakash. Experimental validation of computational models of microwave tissue heating with magnetic resonance thermometry. In *2018 IEEE/MTT-S International Microwave Symposium-IMS*, pages 1282–1284. IEEE, 2018.
- [124] Jochen Gast and Stefan Roth. Lightweight probabilistic deep networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.
- [125] Garron Deshazer, Punit Prakash, Derek Merck, and Dieter Haemmerich. Experimental measurement of microwave ablation heating pattern and comparison to computer simulations. *International Journal of Hyperthermia*, 33(1):74–82, 2017.
- [126] Jhimli Mitra, Kai-kai Shen, Soumya Ghose, Pierrick Bourgeat, Jurgen Fripp, Olivier Salvado, Kerstin Pannek, D Jamie Taylor, Jane L Mathias, and Stephen Rose. Statistical machine learning to identify traumatic brain injury (tbi) from structural disconnections of white matter networks. *Neuroimage*, 129:247–259, 2016.
- [127] Xiaogang Wu, Mohammad Al Hasan, and Jake Yue Chen. Pathway and network analysis in proteomics. *Journal of theoretical biology*, 362:44–52, 2014.
- [128] Tony Z Tang, Yingxin Zhao, Deepesh Agarwal, Aabila Tharzeen, Igor Patrikeev, Yuanyi Zhang, Jana DeJesus, Stefan H Bossmann, Balasubramaniam Natarajan, Mas-

- soud Motamedi, et al. Serum amyloid a and mitochondrial dna in extracellular vesicles are novel markers for detecting traumatic brain injury in a mouse model. *Isience*, 27(2), 2024.
- [129] Brooke E Magnus, Steve Balsis, Joseph T Giacino, Michael A McCrea, Nancy R Temkin, John Whyte, Geoffrey T Manley, Lindsay D Nelson, and TRACK-TBI Investigators. Improving the precision of the glasgow outcome scale-extended using item response theory: a track-tbi study. *Journal of Neurotrauma*, 39(11-12):870–878, 2022.
- [130] Ling Hu, Siyu Yang, Bo Jin, and Chao Wang. Advanced neuroimaging role in traumatic brain injury: a narrative review. *Frontiers in Neuroscience*, 16:872609, 2022.
- [131] JA Cruz. Ds wishart applications of machine learning in cancer prediction and prognosis., 2007, 2. DOI: <https://doi.org/10.1177/117693510600200030>, pages 59–77, 2007.
- [132] Damian Szklarczyk, Rebecca Kirsch, Mikaela Koutrouli, Katerina Nastou, Farrokh Mehryary, Radja Hachilif, Annika L Gable, Tao Fang, Nadezhda T Doncheva, Sampo Pyysalo, et al. The string database in 2023: protein–protein association networks and functional enrichment analyses for any sequenced genome of interest. *Nucleic acids research*, 51(D1):D638–D646, 2023.
- [133] L. Gałezewski, I. Jaskulska, D. Jaskulski, A. Lewandowski, A. Szyplowska, A. Wilczek, and M. Szczepańczyk. Analysis of the need for soil moisture, salinity and temperature sensing in agriculture: A case study in poland. *Scientific Reports*, 11(1):16660, 2021. doi: 10.1038/s41598-021-96182-1.
- [134] D. R. Montgomery, P. Rabinowitz, Y. Sipos, and E. E. Wheat. Soil health: A common focus for one health and planetary health interventions. *One Health*, 18:100673, 2024. doi: 10.1016/j.onehlt.2023.100673.
- [135] C. Joshi and B. P. Mohanty. Physical controls of near-surface soil moisture across varying spatial scales in an agricultural landscape during smex02. *Water Resources Research*, 46(12), 2010. doi: 10.1029/2010WR009152.

- [136] M. H. Saleem, J. Potgieter, and K. M. Arif. Automation in agriculture by machine and deep learning techniques: A review of recent developments. *Precision Agriculture*, 22(6):2053–2091, 2021. doi: 10.1007/s11119-021-09806-x.
- [137] G. W. Pereira, D. S. M. Valente, D. M. de Queiroz, N. T. Santos, and E. I. Fernandes-Filho. Soil mapping for precision agriculture using support vector machines combined with inverse distance weighting. *Precision Agriculture*, 23(4):1189–1204, 2022. doi: 10.1007/s11119-022-09880-9.
- [138] S. M. Basha, D. S. Rajput, J. J, R. S. Somula, and S. Ram. Principles and practices of making agriculture sustainable: Crop yield prediction using random forest. *Scalable Computing: Practice and Experience*, 21(4):Article 4, 2020. doi: 10.12694/scpe.v21i4.1714.
- [139] Q. Zhang, B. Li, Y. Zhang, and S. Wang. Suitability evaluation of crop variety via graph neural network. *Computational Intelligence and Neuroscience*, 2022:5614974, 2022. doi: 10.1155/2022/5614974.
- [140] K. Mythili and E. Al. A swarm based bi-directional lstm-enhanced elman recurrent neural network algorithm for better crop yield in precision agriculture. *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, 12(10):Article 10, 2021. doi: 10.17762/turcomat.v12i10.5656.
- [141] R. P. Sishodia, R. L. Ray, and S. K. Singh. Applications of remote sensing in precision agriculture: A review. *Remote Sensing*, 12(19):Article 19, 2020. doi: 10.3390/rs12193136.
- [142] Y. Zhang and P. J. Thorburn. A dual-head attention model for time series data imputation. *Computers and Electronics in Agriculture*, 189:106377, 2021. doi: 10.1016/j.compag.2021.106377.
- [143] H. Chen and J. Wang. Active learning for efficient soil monitoring in large terrain with heterogeneous sensor network. *Sensors*, 23(5):2365, 2023. doi: 10.3390/s23052365.

- [144] M. Lablack and Y. Shen. Spatio-temporal graph mixformer for traffic forecasting. *Expert Systems with Applications*, 228:120281, 2023. doi: 10.1016/j.eswa.2023.120281.
- [145] M. P. Laurini. A spatio-temporal approach to estimate patterns of climate change. *Environmetrics*, 30(1):e2542, 2019. doi: 10.1002/env.2542.
- [146] M. San Emeterio de la Parte, J.-F. Martínez-Ortega, V. Hernández Díaz, and N. L. Martínez. Big data and precision agriculture: A novel spatio-temporal semantic iot data management framework for improved interoperability. *Journal of Big Data*, 10(1):52, 2023. doi: 10.1186/s40537-023-00729-0.
- [147] L. Zhao, Y. Song, C. Zhang, Y. Liu, P. Wang, T. Lin, M. Deng, and H. Li. T-gcn: A temporal graph convolutional network for traffic prediction. *IEEE Transactions on Intelligent Transportation Systems*, 21(9):3848–3858, 2020. doi: 10.1109/TITS.2019.2935152.
- [148] L. Yu, B. Du, X. Hu, L. Sun, L. Han, and W. Lv. Deep spatio-temporal graph convolutional network for traffic accident prediction. *Neurocomputing*, 423:135–147, 2021. doi: 10.1016/j.neucom.2020.09.043.
- [149] S. Duan, W. Yang, X. Wang, S. Mao, and Y. Zhang. Temperature forecasting for stored grain: A deep spatiotemporal attention approach. *IEEE Internet of Things Journal*, 8(23):17147–17160, 2021. doi: 10.1109/JIOT.2021.3078332.
- [150] J. Fan, J. Bai, Z. Li, A. Ortiz-Bobea, and C. P. Gomes. A gnn-rnn approach for harnessing geospatial and temporal information: Application to crop yield prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 11873–11881, 2022. doi: 10.1609/aaai.v36i11.21444.
- [151] Y. Xu, X. Cai, E. Wang, W. Liu, Y. Yang, and F. Yang. Dynamic traffic correlations based spatio-temporal graph convolutional network for urban traffic prediction. *Information Sciences*, 621:580–595, 2023. doi: 10.1016/j.ins.2022.11.086.

- [152] Z. Qu, Y. Zhang, C. Hong, C. Zhang, Z. Dai, Y. Zhao, X. Wu, Y. gao, X. Jiang, J. Qian, and Z. Gu. Temperature forecasting of grain in storage: A multi-output and spatiotemporal approach based on deep learning. *Computers and Electronics in Agriculture*, 208:107785, 2023. doi: 10.1016/j.compag.2023.107785.
- [153] M. Ma, P. Xie, F. Teng, B. Wang, S. Ji, J. Zhang, and T. Li. Histgmn: Hierarchical spatio-temporal graph neural network for weather forecasting. *Information Sciences*, 648:119580, 2023. doi: 10.1016/j.ins.2023.119580.
- [154] J. Lloret, S. Sendra, L. Garcia, and J. M. Jimenez. A wireless sensor network deployment for soil moisture monitoring in precision agriculture. *Sensors*, 21(21):Article 21, 2021. doi: 10.3390/s21217243.
- [155] S. R. Santana, L. A. Ortega, D. Hernández-Lobato, and B. Zaldívar. Correcting model bias with sparse implicit processes, 2022. ArXiv preprint.
- [156] M. Van Ness, T. M. Bosschieter, R. Halpin-Gregorio, and M. Udell. The missing indicator method: From low to high dimensions. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 5004–5015, 2023. doi: 10.1145/3580305.3599911.
- [157] D. Agarwal, P. Srivastava, S. Martin-del Campo, B. Natarajan, and B. Srinivasan. Addressing uncertainties within active learning for industrial iot. In *2021 IEEE 7th World Forum on Internet of Things (WF-IoT)*, pages 557–562, 2021. doi: 10.1109/WF-IoT51360.2021.9595397.
- [158] D. Agarwal and B. Natarajan. Active learning for node classification using a convex optimization approach. In *2022 IEEE Eighth International Conference on Big Data Computing Service and Applications (BigDataService)*, pages 96–102, 2022. doi: 10.1109/BigDataService55688.2022.00022.
- [159] Y. Wu, Y. Xu, A. Singh, Y. Yang, and A. Dubrawski. Active learning for graph neural networks via node feature propagation, 2019. ArXiv preprint.

- [160] G. Del Mondo, P. Peng, J. Gensel, C. Claramunt, and F. Lu. Leveraging spatio-temporal graphs and knowledge graphs: Perspectives in the field of maritime transportation. *ISPRS International Journal of Geo-Information*, 10(8):Article 8, 2021. doi: 10.3390/ijgi10080541.
- [161] Y. Wang, C. Jing, S. Xu, and T. Guo. Attention based spatiotemporal graph attention networks for traffic flow forecasting. *Information Sciences*, 607:869–883, 2022. doi: 10.1016/j.ins.2022.05.127.
- [162] A. Jain, A. R. Zamir, S. Savarese, and A. Saxena. Structural-rnn: Deep learning on spatio-temporal graphs. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5308–5317, 2016. doi: 10.1109/CVPR.2016.573.
- [163] J. Tully, R. Haight, B. Hutchinson, S. Huang, J.-Y. Lee, and S. Katipamula. Dilated causal convolutional neural networks for forecasting zone airflow to estimate short-term energy consumption. *Energy and Buildings*, 286:112890, 2023. doi: 10.1016/j.enbuild.2023.112890.
- [164] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu. Wavenet: A generative model for raw audio. <https://www.semanticscholar.org/paper/WaveNet%3A-A-Generative-Model-for-Raw-Audio-Oord-Dieleman/df0402517a7338ae28bc54acaac400de6b456a46>, 2016.
- [165] S. Chang, Y. Zhang, W. Han, M. Yu, X. Guo, W. Tan, X. Cui, M. Witbrock, M. A. Hasegawa-Johnson, and T. S. Huang. Dilated recurrent neural networks. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/hash/32bb90e8976aab5298d5da10fe66f21d-Abstract.html.
- [166] F. Scarselli, M. Gori, Ah Chung Tsoi, M. Hagenbuchner, and G. Monfardini. The

- graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2009. doi: 10.1109/TNN.2008.2005605.
- [167] A. Gupta, P. Matta, and B. Pant. Graph neural network: Current state of art, challenges and applications. *Materials Today: Proceedings*, 46:10927–10932, 2021. doi: 10.1016/j.matpr.2021.01.950.
- [168] S. Zhang, H. Tong, J. Xu, and R. Maciejewski. Graph convolutional networks: A comprehensive review. *Computational Social Networks*, 6(1):11, 2019. doi: 10.1186/s40649-019-0069-y.
- [169] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio. Graph attention networks, 2018. URL <http://arxiv.org/abs/1710.10903>.
- [170] S. Munikoti, D. Agarwal, L. Das, and B. Natarajan. A general framework for quantifying aleatoric and epistemic uncertainty in graph neural networks. *Neurocomputing*, 521:1–10, 2023. doi: 10.1016/j.neucom.2022.11.049.
- [171] C. K. Gasch, D. J. Brown, C. S. Campbell, D. R. Cobos, E. S. Brooks, M. Chahal, and M. Poggio. A field-scale sensor network data set for monitoring and modeling the spatial and temporal variation of soil water content in a dryland agricultural field. *Water Resources Research*, 53(12):10878–10887, 2017. doi: 10.1002/2017WR021307.