Machine Learning-Driven Solutions for Hydrometric and Traffic Prediction

Naghmeh Shafiee Roudbari

A Thesis

in

The Department

of

Computer Science and Software Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy (Computer Science) at

Concordia University

Montréal, Québec, Canada

December 2024

© Naghmeh Shafiee Roudbari, 2024

CONCORDIA UNIVERSITY

School of Graduate Studies

This is to certify that the thesis prepared

By: **Naghmeh Shafiee Roudbari**

Entitled: **Machine Learning-Driven Solutions for Hydrometric and Traffic Prediction**

and submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy (Computer Science)

complies with the regulations of this University and meets the accepted standards with respect to originality and quality. Signed by the Final Examining Committee:

_____ Chair
Dr. Ahmed Soliman

_____ External Examiner
Dr. Ralph Evins

_____ External to Program
Dr. Nizar Bouguila

_____ Examiner
Dr. Adam Krzyzak

_____ Examiner
Dr. Yann-Gaël Guéhéneuc

_____ Supervisor
Dr. Charalambos Poullis, Dr. Zachary Patterson, Dr. Ursula Eicker

Approved by _____
Dr. Joey Paquet, Chair
Department of Computer Science and Software Engineering

_____ 2025

_____
Dr. Mourad Debbabi, Dean
Faculty of Engineering and Computer Science

Abstract

Machine Learning-Driven Solutions for Hydrometric and Traffic Prediction

Naghmeh Shafiee Roudbari, Ph.D.

Concordia University, 2024

This thesis explores advanced spatiotemporal machine learning techniques for traffic and hydrometric prediction using graph-based neural network models, RNN family, attention mechanism, and transformer architecture. First, a multilevel GNN-RNN architecture is proposed for traffic forecasting, effectively capturing complex spatial and temporal dependencies across urban road networks. This model significantly reduces computation time and improves prediction accuracy compared to existing methods. In the domain of hydrometric forecasting, a spatiotemporal model with an attention-augmented Graph Convolution Recurrent Neural Network (GCRN) is introduced. This model learns the connectivity between water stations adaptively through a graph learning module, addressing the dynamic nature of water systems. Additionally, a flood prediction model, LocalFloodNet, combines GNNs with a digital twin simulation tool, enabling interactive flood scenario analysis and prevention strategies. The model was applied to a case study for the city of Terrebonne. Finally, a hybrid model integrating Vision Transformers (ViTs) and LiDAR terrain data is developed for long-term hydrometric prediction, utilizing both static terrain features and dynamic temporal relationships. These models collectively enhance forecasting capabilities across multiple domains, providing more accurate and efficient solutions for traffic and hydrometric challenges.

Acknowledgments

First and foremost, I would like to sincerely thank my supervisors, Dr. Poullis, Dr. Patterson, and Dr. Eicker, for their invaluable guidance, patience, and expert advice throughout my PhD journey. Their mentorship has been instrumental in shaping my research. I also extend my thanks to the committee members, Dr. Krzyzak, Dr. Guéhéneuc, and Dr. Bouguila, whose insights and feedback have significantly enriched my work.

My heartfelt thanks go to my family for their constant support throughout this journey. Their encouragement has kept me going, especially during challenging times. I truly appreciate their strength and the sacrifices they've made to help me reach my goals.

Finally, I reflect on this journey with gratitude to everyone who has been a part of it.

Contribution of Authors

This thesis is manuscript-based, and the following is a list of publications, outlining my role in advancing the research presented.

(1) Naghmeh Shafiee Roudbari, Zachary Patterson, Ursula Eicker, and Charalambos Poullis. "Simpler is Better: Multilevel Abstraction with Graph Convolutional Recurrent Neural Network Cells for Traffic Prediction." 2022 IEEE Symposium Series on Computational Intelligence (SSCI).
Contribution: conceptual modeling and writing.
My role: lead author.
The significance of this work: within the context of my Ph.D. thesis.
Chapter: Chapter 3

(2) Naghmeh Shafiee Roudbari, Charalambos Poullis, Zachary Patterson, and Ursula Eicker. "TransGlow: Attention-augmented Transduction Model Based on Graph Neural Networks for Water Flow Forecasting." 2023 International Conference on Machine Learning and Applications (ICMLA).
Contribution: conceptual modeling and writing.
My role: lead author.
The significance of this work: within the context of my Ph.D. thesis.
Chapter: Chapter 4

(3) Naghmeh Shafiee Roudbari, Shubham Rajeev Punekar, Zachary Patterson, Ursula Eicker, and Charalambos Poullis. "From Data to Action in Flood Forecasting Leveraging Graph Neural Networks and Digital Twin Visualization." Scientific Reports (2024).

Contribution: conceptual modeling and writing (I was responsible for computational modeling, while Shubham Rajeev Punekar contributed to visualization and digital twin aspects).

My role: co-lead author.

The significance of this work: within the context of my Ph.D. thesis.

Chapter: Chapter 5

(4) Naghmeh Shafiee Roudbari, Ursula Eicker, Charalambos Poullis, and Zachary Patterson. "HydroVision: LiDAR-Guided Hydrometric Prediction with Vision Transformers and Hybrid Graph Learning." 19th International Symposium on Visual Computing (ISVC) 2024.

Contribution: conceptual modeling and writing.

My role: lead author.

The significance of this work: within the context of my Ph.D. thesis.

Chapter: Chapter 6

Contents

List of Figures xi

List of Tables xiii

1 Introduction 1

1.1 Forecasting Across Time and Space 1

1.2 Research Motivation 3

1.3 Research Objective 4

1.4 Methodology Overview 4

1.4.1 Traffic Prediction Using Multilevel Encoder Architecture 4

1.4.2 Hydrometric Prediction with Attention-Augmented Transduction 5

1.4.3 Flood Simulation and Prediction Using Digital Twins 5

1.4.4 Enhancing Hydrometric Prediction with LiDAR Data 6

1.5 Thesis Structure 7

2 Literature Review 8

2.1 Introduction 8

2.2 Predictive models 9

2.2.1 Recurrent Neural Networks (RNNs) 10

2.2.2 Graph Neural Networks (GNNs) 11

2.3 Attention Mechanisms and Transformers 12

2.4 Application Domains 16

2.4.1 Transportation . . . 17
2.4.2 Meteorological Prediction . . . 17
2.4.3 Disaster Situation Prediction . . . 17
2.4.4 Other Application Domains . . . 18
2.5 Challenges in Spatiotemporal Forecasting . . . 18

3 Traffic Prediction Using Multilevel Encoder Architecture 20
3.1 Abstract . . . 20
3.2 Introduction . . . 21
3.3 Related Work . . . 24
3.4 Proposed Model . . . 27
3.4.1 Graph Notation . . . 27
3.4.2 Problem statement . . . 28
3.4.3 Cell Architecture . . . 29
3.4.4 Multilevel Sequence-to-Sequence Architecture (MLS2S) . . . 30
3.5 Methodology . . . 32
3.6 Experiments . . . 33
3.6.1 Dataset Description . . . 33
3.6.2 Experimental Settings . . . 38
3.6.3 Baseline Methods . . . 38
3.6.4 Performance Comparison . . . 39
3.6.5 Computation Time . . . 40
3.7 Conclusion . . . 40

4 Hydrometric Prediction with Attention-Augmented Transduction 42
4.1 Abstract . . . 42
4.2 Introduction . . . 43
4.3 Related Work . . . 46
4.4 Methodology . . . 47
4.4.1 Problem Statement . . . 47

4.4.2 Graph Learning Module . . . 49
4.4.3 Graph Convolution Recurrent Block . . . 49
4.4.4 Encoder-Decoder Model . . . 50
4.4.5 Efficient Attention Layer . . . 51
4.5 Experiments . . . 52
4.5.1 Dataset Description . . . 52
4.5.2 Experimental Settings . . . 52
4.5.3 Baselines . . . 54
4.5.4 Performance Comparison . . . 55
4.5.5 Complexity . . . 56
4.6 Conclusion . . . 57

5 Flood Forecasting Using Digital Twins **59**

5.1 Abstract . . . 59
5.2 Introduction . . . 60
5.3 Related Work . . . 63
5.3.1 Digital Twins . . . 63
5.3.2 Forecasting . . . 65
5.4 Problem Formulation . . . 67
5.5 Methodology . . . 68
5.5.1 Architectural Enhancement: Attention-Augmented Encoder-Decoder . . . 68
5.6 Experiments . . . 70
5.7 Simulation and Visualization . . . 73
5.7.1 Digital Twin Modeling . . . 74
5.7.2 Viewing and Interaction . . . 80
5.7.3 Simulation experiments . . . 81
5.8 Conclusion . . . 84
5.9 Data availability . . . 85
5.10 Acknowledgement . . . 85

5.11 Additional Results 85

6 Enhancing Hydrometric Prediction with LiDAR Data **97**

6.1 Abstract 97

6.2 Introduction 98

6.3 Related Work 100

6.4 Dataset 102

6.4.1 LiDAR Data 102

6.4.2 Timeseries data 104

6.5 Objective 104

6.6 Methodology 105

6.6.1 Foundational Approach 105

6.6.2 Vision Transformer 107

6.6.3 Hybrid Graph Learning 108

6.7 Experiments 109

6.7.1 Settings 109

6.7.2 Comparative Performance Evaluation 109

6.7.3 Ablation Study 111

6.8 Conclusion 112

7 Conclusion and Future Work **113**

7.1 Key Findings 114

7.2 Limitations 115

7.3 Error Metrics Discussion 116

7.4 Future Work 117

Bibliography **119**

List of Figures

Figure 2.1 The message-passing process in Graph Neural Networks (GNNs): Each node updates its representation based on its own features and the features of its neighbors, progressively incorporating higher-order relationships within the graph structure. This iterative process allows information to flow along the graph edges, enabling nodes to exchange context-specific information. 12

Figure 2.2 Illustration of the attention mechanism showing the interaction between Query (Q), Key (K), and Value (V) components. 13

Figure 2.3 Transformer encoder and Vision Transformer (ViT) architecture. The diagram shows the flow from input patches to patch embeddings through multiple transformer encoder blocks 15

Figure 3.1 Traffic data variability 23

Figure 3.2 Multilevel Sequence-to-Sequence (MLS2S) Architecture 28

Figure 3.3 Montreal Street Level Traffic Dataset preparation process 33

Figure 3.4 Traffic dataset coverage- urban area and highway 34

Figure 3.5 Comparison of smoothing effect on data variation 35

Figure 4.1 TransGlow main Architecture 48

Figure 4.2 Overview of CWFDD-186 dataset 53

Figure 5.1 LocalFloodnet network architecture 68

Figure 5.2 Station map and annual sensor data variability 71

Figure 5.3 Adding obstructions and inland sources of water 81

Figure 5.4 Simulation of water level = 200 cm over time 82

Figure 5.5 Simulation with real-world data for spring floods of 2017 84
Figure 5.6 Geospatial data sourced from open repositories and visualized in QGIS . . . 90
Figure 5.7 Digital twin modeled in Unreal Engine using geospatial data 91
Figure 5.8 Aligning the building meshes with the landscape 92
Figure 5.9 Heightfield fluid simulation 92
Figure 5.10 Modeling rivers in the simulation 93
Figure 5.11 Comparative visualization of the digital twins 94
Figure 5.12 Viewing perspectives and overview map 94
Figure 5.13 Simulations at varying initial water levels 95
Figure 5.14 Simulation of water level with and without obstruction 96
Figure 6.1 Stations, water level trends, and DTM near Sainte-Agathe-des-Monts 103
Figure 6.2 Architecture of the HydroVision Framework 105

List of Tables

Table 3.1 Traffic experimental results 37
Table 3.2 Efficiency comparison of two top performing methods 40
Table 4.1 Experimental results on water flow data 56
Table 4.2 Efficiency comparison of baseline methods 57
Table 5.1 Experimental results for flood forecasting 72
Table 6.1 Comparative Performance of Various Models for Water Level Forecasting . . 110
Table 6.2 Ablation Study Results for Hybrid Graph Learning 111

Chapter 1

Introduction

1.1 Forecasting Across Time and Space

Time series forecasting is a well-established field in data science and machine learning, with applications spanning various domains such as finance, healthcare, environmental science, and engineering. It focuses on predicting future values of a variable or set of variables based on their historical values over time. Time series models analyze temporal dependencies, enabling predictions about future trends or events by capturing patterns such as seasonality, trends, and short-term fluctuations. Classic models such as Autoregressive Integrated Moving Average (ARIMA), Exponential Smoothing, and more recently, Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks, have been widely used to forecast univariate and multivariate time series data.

However, many real-world systems, particularly in urban infrastructure and environmental monitoring, involve data that evolves not only over time but also across different spatial locations. These systems are inherently spatiotemporal, where both temporal dynamics and spatial relationships play a crucial role in understanding the underlying patterns. This brings us to **spatiotemporal forecasting**, an extension of traditional time series forecasting that incorporates spatial dependencies alongside temporal information. Spatiotemporal forecasting is essential in domains where the interaction between spatially distributed entities (such as road segments in a city or water stations in a river system) has a direct influence on the temporal patterns observed at each location.

Spatiotemporal forecasting has found applications in various fields, including **traffic forecasting** and **hydrometric forecasting**. In traffic management, the goal is to predict vehicle flow, speed, or congestion levels across a network of roads. Traffic data is inherently spatiotemporal, as vehicle flow at one location is often influenced by conditions in nearby locations. The interaction between road segments, combined with temporal patterns (such as rush hour or seasonal changes), makes traffic forecasting a challenging yet vital task for optimizing transportation systems and reducing congestion.

On the other hand, hydrometric forecasting focuses on predicting water-related metrics, such as water levels, flow rates, and flood risks, across interconnected water systems like rivers, lakes, and reservoirs. Similar to traffic forecasting, hydrometric forecasting involves both spatial and temporal dependencies. Water systems are influenced by spatial factors like terrain and interconnected networks of water bodies, while temporal factors include seasonal fluctuations and precipitation events. Accurate hydrometric forecasting is crucial for effective water resource management, flood prevention, and environmental conservation.

Both traffic and hydrometric forecasting share the challenge of dealing with complex spatiotemporal dependencies, where the dynamic interactions between spatial entities evolve over time. Traditional time series forecasting methods are often inadequate for these tasks, as they fail to capture the spatial relationships between the entities. This calls for the development of advanced machine learning models that can effectively learn both spatial and temporal dependencies in an integrated manner.

In this thesis, we explore the development and application of such models, focusing on **Graph Neural Networks (GNNs)** and **Recurrent Neural Networks (RNNs)**, which are well-suited for capturing spatiotemporal patterns. These models are applied to real-world datasets in traffic and hydrometric forecasting, aiming to improve prediction accuracy and computational efficiency in both domains.

1.2 Research Motivation

The motivation for this research lies in the critical role that underlying structures play in the accuracy of spatiotemporal predictions. In traffic prediction, the road network structure has a significant impact on vehicle flow and speed. Urban road networks, characterized by dense connections and high variability in speed, present different challenges compared to highways, which have sparse connections and relatively stable speed patterns. Existing models often struggle to efficiently adapt to both types of networks, leading to inconsistencies in predictions. Therefore, there is a clear need for a robust traffic prediction model that can work effectively in both sparse and dense road networks, handling the variability of traffic conditions while balancing accuracy and computational complexity.

In hydrometric forecasting, the challenge is further compounded by the implicit connectivity between water stations. Unlike road networks, where the connections between nodes (road segments) are explicit and well-defined, the connectivity between water stations is often implicit and dynamic. This interconnectivity is influenced by environmental factors such as terrain elevation, precipitation patterns, and seasonal fluctuations. Accurately capturing this hidden connectivity is crucial for improving the prediction of water levels and streamflow. Additionally, incorporating satellite data, such as LiDAR, provides another layer of spatial awareness, which can significantly enhance the predictive capability of these models by offering precise information about terrain and environmental conditions.

Another crucial aspect of the research motivation is the integration of digital twin technology. By developing digital twins of physical environments, it is possible to simulate different scenarios, such as traffic congestion patterns or flood risks, and evaluate the effectiveness of predictive models in real-world contexts. This approach allows for the testing of various strategies in a simulated environment, providing actionable insights for improving decision-making processes in environmental resource planning.

In summary, this research is driven by the need to develop advanced predictive models that not only enhance prediction accuracy but also maintain computational efficiency. The aim is to ensure that these models can be practically applied across a wide range of spatiotemporal forecasting tasks,

from urban planning to environmental sustainability.

1.3 Research Objective

The objective of this thesis is to address the critical challenge of improving the accuracy and efficiency of predictive models in both environmental science and urban planning using advanced machine learning techniques. Specifically, the research aims to develop models capable of effectively integrating diverse data sources, including satellite imagery and sensor data, while adapting to dynamic conditions in real-time.

A key focus of the research is to enhance sequence-to-sequence models, ensuring they are capable of maintaining ordering information over long sequences while selectively focusing on the most relevant parts of the input data. The goal is to improve predictive performance in systems with inherent variability and complexity, such as urban traffic networks and interconnected water systems.

Ultimately, this research seeks to create models that provide more accurate, efficient, and actionable predictions, helping to improve decision-making in complex and dynamic systems.

1.4 Methodology Overview

The following subsections summarize the contributions of each paper, explain the rationale behind the approaches, and outline the specific innovations introduced in each study. Each contribution was published in peer-reviewed venues, indicating that it was evaluated by experts in the field.

1.4.1 Traffic Prediction Using Multilevel Encoder Architecture

Paper: Simpler is Better: Multilevel Abstraction with Graph Convolutional Recurrent Neural Network Cells for Traffic Prediction. *Published at 2022 IEEE Symposium Series on Computational Intelligence (SSCI) by Naghmeh Shafiee Roudbari, Zachary Patterson, Ursula Eicker, and Charalambos Poullis.*

This paper addresses the challenge of capturing spatial and temporal features in traffic data by leveraging a combination of Graph Neural Networks (GNNs) and Recurrent Neural Networks

(RNNs). While previous GNN-RNN methods handle these dependencies, they often rely on complex graph convolution operations, which increase computational complexity. We introduced a novel multilevel encoder architecture that abstracts and combines dependencies at multiple complexity levels. This architecture is specifically designed to handle both sparse and dense road networks, which is crucial for forecasting in urban environments with high variability. Additionally, our model employs a sparse architecture to reduce computational costs while maintaining accuracy, particularly on benchmark datasets such as METR-LA and our own street-level dataset, MSLTD. This paper demonstrates that the proposed model improves accuracy and significantly reduces training time.

1.4.2 Hydrometric Prediction with Attention-Augmented Transduction

Paper: TransGlow: Attention-augmented Transduction Model Based on Graph Neural Networks for Water Flow Forecasting. *Published at 2023 International Conference on Machine Learning and Applications (ICMLA) by Naghmeh Shafiee Roudbari, Charalambos Poullis, Zachary Patterson, and Ursula Eicker.*

In hydrometric forecasting, the challenge lies in capturing the implicit connectivity between water stations. This paper proposes a novel spatiotemporal forecasting model, integrating an efficient attention mechanism to augment the hidden state in a sequence-to-sequence architecture. Our method focuses on learning the actual correlations between drainage basins using a graph-based approach, which is critical for accurate water flow prediction. We validated the model across 186 drainage basins in Canada using data from Environment and Natural Resources of Canada. The results show that our approach significantly outperforms existing state-of-the-art methods in terms of prediction accuracy across all horizons, further establishing the utility of the attention mechanism in spatiotemporal forecasting tasks.

1.4.3 Flood Simulation and Prediction Using Digital Twins

Paper: From Data to Action in Flood Forecasting Leveraging Graph Neural Networks and Digital Twin Visualization. *Published at Scientific Reports(2024) by Naghmeh Shafiee Roudbari, Shubham Rajeev Punekar, Zachary Patterson, Ursula Eicker, and Charalambos Poullis.*

Both first and second authors contributed equally to this work. I was responsible for the computational modeling, while Shubham Rajeev Punekar contributed to the visualization and digital twin aspects.

This paper introduces an innovative flood simulation and prediction model that combines GNN-based forecasting with digital twin visualization. By creating a detailed digital twin of the city of Terrebonne, Quebec, we simulate flood scenarios, enabling more accurate flood risk assessment and the design of mitigation strategies. Our approach extends traditional GCRN-based methods by allowing for local-scale flood predictions. Extensive experimentation demonstrates the accuracy of our model in predicting flood dynamics using digital twin techniques. This simulation tool offers a generalizable and reproducible framework for flood prevention and disaster response strategies, contributing significantly to local-scale flood prediction.

1.4.4 Enhancing Hydrometric Prediction with LiDAR Data

Paper: HydroVision: LiDAR-Guided Hydrometric Prediction with Vision Transformers and Hybrid Graph Learning. *Accepted for publication at 19th International Symposium on Visual Computing (ISVC) 2024 by Naghmeh Shafiee Roudbari, Ursula Eicker, Charalambos Poullis, and Zachary Patterson.*

In this paper, we integrate terrain elevation data derived from LiDAR into a hydrometric forecasting model, enabling more accurate predictions of water flow and connectivity between water stations. The core contribution lies in the introduction of a hybrid graph learning structure that combines static graphs, based on terrain data, with dynamic graphs that adapt to temporal changes in water systems. By using Vision Transformers (ViTs) to encode LiDAR data, the model captures critical terrain features, while the graph learning structure improves the overall spatial and temporal representation. Experimental results on water stations in Quebec, using data from the Environment and Natural Resources of Canada, show that this model outperforms state-of-the-art methods across all prediction horizons, underscoring the importance of incorporating spatial data into hydrometric models.

1.5 Thesis Structure

This thesis follows a manuscript-based format. The next chapter provides a literature review, setting the foundation for the research presented. Each subsequent chapter corresponds to a manuscript that has been published, detailing the specific methodologies and contributions of the work. Finally, the thesis concludes with a chapter dedicated to summarizing the key findings, drawing overarching conclusions, and outlining potential directions for future research.

Chapter 2

Literature Review

2.1 Introduction

With the rapid development of sensing technologies and data stream processing, large amounts of data from urban systems are now being efficiently collected and stored. This progress has paved the way for urban computing, which seeks to analyze urban patterns and dynamics across various domains like transportation, environment, and climate. Predictive learning, a supervised learning technique, uses historical data to forecast future trends. According to urban computing theories Y. Zheng, Capra, Wolfson, and Yang (2014), predictive learning based on extensive urban data plays a crucial role in supporting intelligent decision-making, scheduling, and management in smart cities. Furthermore, the ability to predict from urban big data opens the door to innovations like digital twin cities and the metaverse X. Wang, Li, Yuan, Ye, and Wang (2016).

Most urban data is spatio-temporal, meaning it relates to both spatial locations and changes over time. In urban systems, this spatio-temporal data is commonly characterized by two key properties: correlation and heterogeneity S. Wang, Cao, and Philip (2020). Correlation refers to the data being correlated over time and across different spatial locations. Heterogeneity, on the other hand, indicates that data patterns vary across different temporal or spatial scales. Traditional methods used in time series forecasting, such as Support Vector Regression (SVR) Drucker, Burges, Kaufman, Smola, and Vapnik (1996), Random Forest (RF) Breiman (2001), and Gradient Boosting Decision Tree (GBDT) Natekin and Knoll (2013), are not effective in generating accurate predictions. In the

past decade, the rapid growth of deep learning has led to the rise of hybrid neural networks combining Convolutional Neural Networks (CNNs) Gu et al. (2018) and Recurrent Neural Networks (RNNs) Y. Yu, Si, Hu, and Zhang (2019). Hybrid models, such as ConvLSTM Shi et al. (2015) and PredRNN Y. Wang, Long, Wang, Gao, and Yu (2017), have been applied to predictive learning in urban spatio-temporal data, demonstrating significant improvements. However, a major limitation of these models is their inability to directly process non-Euclidean data, which is prevalent in urban systems, such as traffic flow on road networks, and vehicle movement across routes. Recently, breakthroughs in representation learning for non-Euclidean data have been achieved through deep learning techniques, particularly with Graph Neural Networks (GNNs) Kipf and Welling (2016). These advances have paved the way for predictive learning models capable of handling diverse and complex urban data.

The purpose of this literature review is to provide an analysis of the existing research in machine learning (ML) as applied to environmental forecasting. This review will evaluate the key technological advancements, models, and methodologies that have been developed in recent years. By synthesizing this body of work, the review will lay the theoretical foundation for the research presented in this thesis.

2.2 Predictive models

To effectively model environmental changes and manage urban systems, ML models such as Recurrent Neural Networks (RNNs), Convolutional Neural Networks (CNNs), Graph Neural Networks (GNNs), and Transformer architectures have been employed in predictive learning tasks. These models are capable of capturing complex spatial and temporal dependencies within large datasets, making them ideal for applications that involve dynamic systems like weather forecasting, traffic management, and disaster prediction. Each architecture has its strengths: RNNs excel at handling sequential data, CNNs are efficient in extracting spatial features, GNNs can model non-Euclidean data like road networks and environmental systems, while Transformers leverage attention mechanisms to model long-range dependencies in both spatial and temporal data. By integrating these technologies, predictive models can more accurately forecast environmental changes,

providing critical insights for decision-making in smart cities.

2.2.1 Recurrent Neural Networks (RNNs)

Recurrent Neural Networks (RNNs) are a class of deep neural networks designed for sequential learning tasks, making them particularly well-suited for time series modeling and other temporal data analysis. RNNs process sequences by maintaining a hidden state that recursively updates as new input is received at each time step. This recursive computation allows RNNs to capture temporal dependencies in the data. However, the vanilla RNN architecture is known to suffer from a significant drawback known as the gradient vanishing or explosion problem during the training process, which can hinder the network's ability to learn long-term dependencies Fadziso (2020). To address this issue, two prominent variants of RNNs have been introduced: Long Short-Term Memory (LSTM) Hochreiter and Schmidhuber (1997) and Gated Recurrent Units (GRU) Dey and Salem (2017). These variants are designed to mitigate the vanishing gradient problem and improve the network's ability to model long-range dependencies. Among these, GRU has gained widespread adoption due to its balance of performance and computational efficiency.

GRU employs two gated mechanisms: the update gate (u_t) and the reset gate (r_t). The update gate controls how much of the new information at the current time step should be combined with the memory from the previous time step. Meanwhile, the reset gate determines how much of the previous time step's information should be retained in the current memory. This streamlined structure makes GRUs less computationally complex than LSTMs, while still providing competitive performance across various time series and sequential learning tasks. The reduction in learnable parameters in GRU compared to LSTM has been shown to improve training and inference efficiency. Despite its simpler architecture, GRU achieves similar, if not better, performance in several applications while also improving computational efficiency, making it a popular choice for time series forecasting and other sequence-based modeling tasks.

RNNs, particularly LSTM and GRU variants, have been widely adopted in forecasting applications. In traffic forecasting, RNNs are leveraged to model the temporal dynamics of traffic flow Y. Tian and Pan (2015), congestion patterns J. Guo, Liu, Yang, Wang, and Fang (2021), and travel time prediction Duan, Yisheng, and Wang (2016). By capturing the sequential nature of traffic data,

RNN-based models such as LSTM networks have demonstrated significant improvements over traditional statistical methods Siami-Namini, Tavakoli, and Namin (2018). Furthermore, GRUs have been utilized for real-time traffic prediction, where their computational efficiency allows for quicker processing of large-scale, high-frequency data Z. Zhao et al. (2023). In environmental forecasting, RNNs are used to predict weather patterns, air quality indices, and hydrological phenomena by learning from historical time series data. The capability of RNNs to handle multivariate input makes them suitable for complex environmental systems where various interdependent factors (e.g., temperature, humidity, wind speed) influence the predictions Y. Chen, Cheng, Cheng, Yang, and Yu (2018). Additionally, in urban planning, RNNs have been employed for smart city applications such as energy consumption prediction, urban mobility analysis, and public transportation scheduling Kong et al. (2019). Their ability to model sequential data efficiently has made RNNs an essential tool in building adaptive and intelligent forecasting systems for urban environments.

2.2.2 Graph Neural Networks (GNNs)

GNNs are a class of neural networks specifically designed to operate on graph-structured data, making them well-suited for tasks involving complex relational structures and interconnections. Unlike traditional neural networks that operate on grid-like data (e.g., images or sequences), GNNs leverage the graph's structure to capture relationships between nodes and edges. This capability is particularly advantageous for problems such as social network analysis, molecular chemistry, and traffic forecasting, where data is naturally represented as a graph.

The core idea of GNNs is to aggregate and transform information from a node's neighbors to update its representation iteratively. This process involves passing messages between nodes through their edges, allowing each node to incorporate information from its local neighborhood, as illustrated in Figure 2.1. One of the foundational architectures in GNNs is the Graph Convolutional Network (GCN) Kipf and Welling (2016), which performs convolution operations on graph data by aggregating information from a node's neighbors and applying learned weights to update node embeddings. This approach allows GCNs to effectively capture local structural information and node features.

However, GCNs have limitations in capturing long-range dependencies and handling graphs

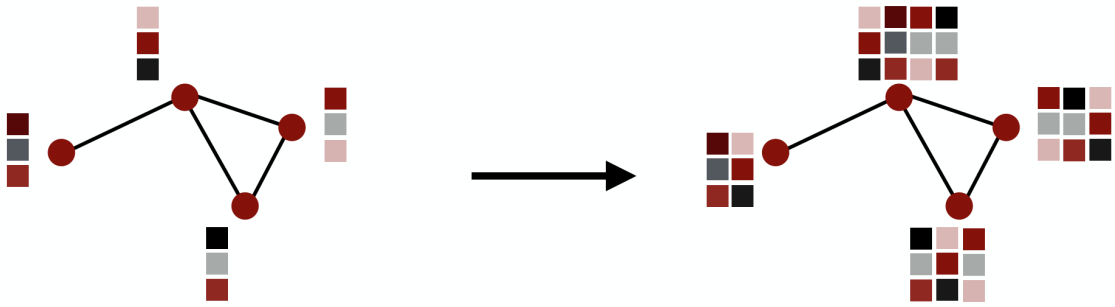


Figure 2.1: The message-passing process in Graph Neural Networks (GNNs): Each node updates its representation based on its own features and the features of its neighbors, progressively incorporating higher-order relationships within the graph structure. This iterative process allows information to flow along the graph edges, enabling nodes to exchange context-specific information.

with varying node degrees. To address these challenges, several advanced variants of GNNs have been proposed. For example, the Graph Attention Network (GAT) Veličković et al. (2017) introduces attention mechanisms to weigh the importance of different neighbors, allowing the model to focus on more relevant connections. This attention-based approach helps GATs handle graphs with varying node degrees and improves the model's ability to capture long-range dependencies.

2.3 Attention Mechanisms and Transformers

Attention mechanisms have revolutionized deep learning by enabling models to focus selectively on different parts of the input data, improving performance across a range of tasks. Initially introduced in the context of sequence-to-sequence models, attention mechanisms allow neural networks to weigh the importance of different elements in a sequence, enhancing the model's ability to capture relevant information and manage long-range dependencies.

The concept of attention was first popularized by the paper *Neural Machine Translation by Jointly Learning to Align and Translate* Bahdanau, Cho, and Bengio (2014), where Bahdanau et al. demonstrated its effectiveness in machine translation tasks. This attention mechanism computes a context vector by weighing the importance of each input element based on a learned alignment model, allowing the model to focus on different parts of the input sequence when generating each part of the output sequence.

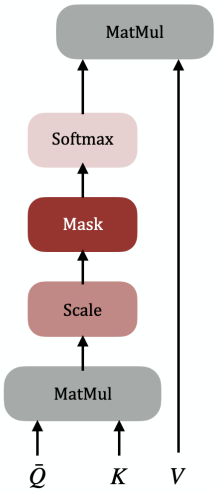


Figure 2.2: Illustration of the attention mechanism showing the interaction between Query (Q), Key (K), and Value (V) components.

The attention function begins by mapping inputs into three distinct components:

- **Query (Q):** Represents the element seeking relevant information.
- **Key (K):** Represents the element containing information to be compared against the query.
- **Value (V):** Contains the actual information used to compute the output.

The attention scores, which capture the alignment between queries and keys, are computed using the scaled dot product attention formula:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{1}$$

where d_k is the dimensionality of the keys. This scaling factor, $\sqrt{d_k}$, ensures numerical stability by preventing excessively large gradients when working with high-dimensional data. A **mask** can optionally be applied to the scores before the softmax operation. This mask is used to either:

- Ignore irrelevant or padded tokens.
- Prevent the model from attending to future positions in autoregressive tasks (causal masking).

Once the scores are calculated, they are passed through a softmax function, normalizing them into a probability distribution. These probabilities act as weights, determining the importance of each

value V in forming the final representation:

$$\text{Output} = \sum_i \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)_i V_i \tag{2}$$

The attention operation is illustrated in Figure 2.2, which shows how the Query, Key, and Value components interact and contribute to the final output.

Building on this foundation, the Transformer architecture Vaswani et al. (2017) introduced a novel approach that relies entirely on attention mechanisms, eliminating the need for recurrent layers. The Transformer model is based on self-attention, which allows each position in the input sequence to attend to every other position, capturing dependencies regardless of their distance. The self-attention mechanism computes attention scores for each pair of positions in the sequence, creating weighted representations that incorporate context from the entire sequence. The Transformer architecture consists of two main components: the encoder and the decoder. The encoder processes the input sequence into a set of embeddings using multiple layers of self-attention and feed-forward networks. The decoder then generates the output sequence by attending to the encoder's output and previous tokens in the sequence. Key innovations in the Transformer include multi-head attention, which allows the model to capture different types of relationships in parallel, and positional encoding, which provides information about the position of each token in the sequence.

Transformers have demonstrated remarkable performance across various tasks, including natural language processing, computer vision, and time series analysis. Models like BERT Devlin, Chang, Lee, and Toutanova (2018) and GPT Radford, Narasimhan, Salimans, Sutskever, et al. (2018) leverage the Transformer architecture to achieve state-of-the-art results in language understanding and generation tasks. BERT (Bidirectional Encoder Representations from Transformers) focuses on understanding context by considering both preceding and following tokens, while GPT (Generative Pre-trained Transformer) emphasizes autoregressive generation, predicting the next token based on previous tokens. Despite their success, Transformers can be computationally intensive and require large amounts of data for training. Techniques such as sparse attention Child, Gray, Radford, and Sutskever (2019) and efficient Transformer variants Choromanski et al. (2020) aim to address these challenges by reducing the computational complexity and memory requirements,

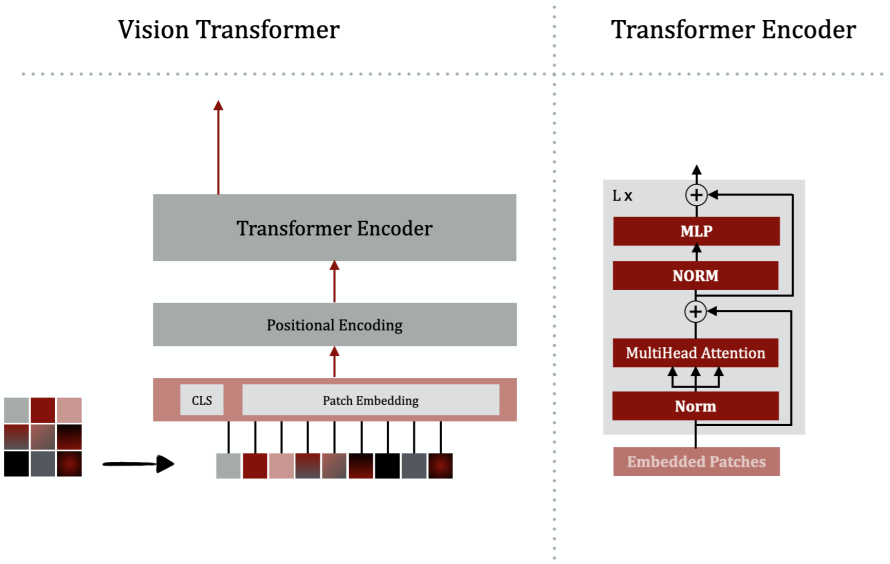


Figure 2.3: Transformer encoder and Vision Transformer (ViT) architecture. The diagram shows the flow from input patches to patch embeddings through multiple transformer encoder blocks

making Transformers more accessible and scalable.

Vision Transformers (ViTs) Dosovitskiy et al. (2020) is a powerful architecture for image-based tasks. By applying transformer models to images, ViTs capture global relationships between patches, making them highly effective for tasks such as image classification and segmentation. One key advantage of ViTs over traditional convolutional neural networks (CNNs) is their ability to model long-range dependencies across the entire input without relying on convolutions, leading to better scalability and performance in many scenarios.

The process in a Vision Transformer begins with the following functions related to the transformer encoder:

Patch Embedding: The input image is divided into fixed-size patches, such as 16×16. Each patch is flattened into a 1D vector and linearly projected into a fixed-dimensional embedding space. Positional encodings are added to these patch embeddings to retain spatial information, as the transformer architecture does not inherently capture positional context.

The sequence of embedded patches is then passed through multiple transformer encoder layers, each consisting of:

Multi-Head Self-Attention (MHSA): This mechanism computes pairwise attention scores between patches, allowing the model to learn relationships across the entire input sequence, thus capturing both local and global information.

Feed-Forward Network (FFN): This applies two linear transformations with a non-linearity in between, enabling complex feature transformation and enhancing the representational capacity of the model.

Layer Normalization and Residual Connections: These components help ensure stable training and efficient gradient flow, preventing vanishing gradients during the learning process.

Classification Token (Optional): A special learnable token, the [CLS] token, is prepended to the sequence of patches. This token aggregates global information from all patches, and the final representation can be used for classification tasks.

ViT Blocks: A Vision Transformer consists of stacked transformer encoder blocks, each containing the multi-head self-attention and feed-forward components. These blocks allow the model to process increasingly abstract representations of the input image at each layer, ultimately producing a global feature representation that can be used for tasks such as classification or segmentation.

The architecture of the transformer encoder and Vision Transformer is depicted in Figure 2.3. This diagram illustrates how the input image is transformed into patch embeddings, processed through multiple transformer blocks, and how the global representation is obtained through the [CLS] token.

2.4 Application Domains

Spatiotemporal forecasting has been widely applied in various domains critical to modern urban systems. These domains include transportation, environmental monitoring, and disaster prediction. Below are the main applications of spatiotemporal forecasting and how they leverage Graph Neural Networks.

2.4.1 Transportation

Modern urban systems are equipped with numerous sensors distributed across road networks and critical areas to monitor dynamic traffic states, such as flow and speed. The aim of traffic state prediction is to forecast future traffic conditions based on historical data within a specific spatial region. Traffic state prediction can be categorized into:

- **Network-based prediction**: The objective here is to predict traffic flow or speed across a given road network. Many studies rely on graph structures that can be directly derived from road networks. A combination of GNN and RNN models have been widely used for this purpose due to their ability to capture spatial-temporal dependencies S. Guo, Lin, Wan, Li, and Cong (2021); B. Yu, Yin, and Zhu (2017).

- **Region-based prediction**: This approach forecasts regional traffic patterns, such as crowd flow in urban areas. The city is typically divided into regular or irregular regions, and spatiotemporal graphs are constructed based on distances, connectivity, and semantic correlations between these regions Y. Wang et al. (2021); X. Zhang, Huang, Xu, and Xia (2020).

2.4.2 Meteorological Prediction

Meteorological forecasting is another vital domain closely linked to both environmental sustainability and human activity. Similar to air quality monitoring, meteorological data is gathered from distributed monitoring stations. However, the correlations between stations can be influenced by a more significant number of factors, making it a complex forecasting task. GNN-based methods have demonstrated strong performance in several meteorological applications, including flood forecasting Kazadi, Doss-Gollin, Sebastian, and Silva (2022), temperature prediction Jia et al. (2021), frost prediction Lira, Martí, and Sanchez-Pi (2021), and wind prediction Khodayar and Wang (2018).

2.4.3 Disaster Situation Prediction

Natural disasters, such as earthquakes, floods, and fires, present substantial challenges to human safety. Accurate disaster prediction enables governments and organizations to take proactive

measures, such as distributing resources, organizing evacuations, and preparing for relief operations. GNNs have shown considerable success in modeling the correlations and heterogeneity of data across geographical locations. These models have been applied to a variety of disaster prediction scenarios, such as fire prediction Jin et al. (2020), typhoon forecasting Farahmand, Xu, and Mostafavi (2023), and earthquake prediction McBrearty and Beroza (2022).

2.4.4 Other Application Domains

In addition to the primary domains mentioned above, spatiotemporal forecasting models have been extended to other fields, such as energy and economics. For instance:

- **Energy**: GNNs have been employed for forecasting wind power generation Z. Li et al. (2022) and photovoltaic power prediction Simeunović, Schubnel, Alet, and Carrillo (2021). These models assist in balancing energy production and consumption in modern grids.
- **Economy and Finance**: In the economic domain, GNNs have been used for regional economy prediction, where researchers explore correlations between different economic regions and indicators F. Xu, Li, and Xu (2020).

These application domains demonstrate the broad utility of spatiotemporal forecasting techniques, particularly when using GNNs, to predict complex phenomena across multiple sectors.

2.5 Challenges in Spatiotemporal Forecasting

While significant progress has been made in spatiotemporal forecasting using models like RNNs, GNNs, and CNNs, several challenges remain, particularly when dealing with dynamic and complex environmental systems. One major challenge lies in the fact that many forecasting applications, such as hydrological modeling and traffic prediction, involve underlying graphs that are either constantly changing or not explicitly defined. Traditional GNNs typically rely on fixed, predefined graph structures, which may not capture the evolving relationships within these systems. This limitation calls for adaptive graph learning techniques that can dynamically adjust the graph representation based on changing environmental factors, such as water flow patterns.

Another challenge is achieving a balance between model accuracy and computational efficiency. Complex hybrid models that combine RNNs, GNNs, and attention mechanisms can yield high accuracy in forecasting tasks but often suffer from increased computational costs, making real-time prediction difficult. Therefore, there is a need for models that can maintain a high level of accuracy while optimizing efficiency, especially in scenarios where large-scale and high-frequency data are involved.

Furthermore, the integration of additional contextual information, such as terrain data obtained through LiDAR, remains underexplored. Most existing models either ignore such static environmental features. Terrain elevation and other topographical features can significantly influence environmental dynamics, particularly in applications like flood prediction and hydrological forecasting. Incorporating these features into the forecasting model can enhance its spatial representation, leading to more accurate and actionable predictions.

Finally, the transition from data to actionable insights presents a crucial but often overlooked aspect of forecasting models. While many models provide predictions, they fall short in evaluating the influence of these predictions on the city's infrastructure and environment. There is a need for models that predict environmental changes and test different scenarios based on these predictions to assess their impact on urban systems. This involves exploring various outcomes, such as the effect of predicted floods on infrastructure, to enable informed decision-making and proactive city management.

These challenges form the basis for this thesis, which introduces adaptive graph learning techniques for dynamically adjusting to changing environments, optimizes model efficiency without compromising accuracy, integrates LiDAR data to enrich the spatial context in forecasting models, and extends from data to action by simulating different scenarios to evaluate the influence of predicted changes on urban environments. By addressing these key challenges, the proposed methods aim to advance the state of spatiotemporal forecasting, providing valuable tools for environmental and urban applications.

Chapter 3

Traffic Prediction Using Multilevel Encoder Architecture

This chapter is a copy of the manuscript titled *Simpler is Better: Multilevel Abstraction with Graph Convolutional Recurrent Neural Network Cells for Traffic Prediction*, published in the 2022 IEEE Symposium Series on Computational Intelligence (SSCI). The manuscript was co-authored by Naghmeh Shafiee Roudbari, Zachary Patterson, Ursula Eicker, and Charalambos Poullis.

3.1 Abstract

In recent years, graph neural networks (GNNs) combined with variants of recurrent neural networks (RNNs) have reached state-of-the-art performance in spatiotemporal forecasting tasks. This is particularly the case for traffic forecasting, where GNN models use the graph structure of road networks to account for spatial correlation between links and nodes. Recent solutions are either based on complex graph operations or avoiding predefined graphs. This chapter proposes a new sequence-to-sequence architecture to extract the spatiotemporal correlation at multiple levels of abstraction using GNN-RNN cells with sparse architecture to decrease training time compared to more complex designs. Encoding the same input sequence through multiple encoders, with an incremental increase in encoder layers, enables the network to learn general and detailed information through multilevel abstraction. We further present a new benchmark dataset of street-level segment traffic

data from Montreal, Canada. Unlike highways, urban road segments are cyclic and characterized by complicated spatial dependencies. Experimental results on the METR-LA benchmark highway and our MSLTD street-level segment datasets demonstrate that our model improves performance by more than 7% for one-hour prediction compared to the baseline methods while reducing computing resource requirements by more than half compared to other competing methods.

3.2 Introduction

Spatiotemporal forecasting is an essential tool for understanding variations in space-time data and ultimately helps inform resource allocation, risk management, and policy-making decisions. It has long been a popular field of research in machine learning. Even though many influential machine learning (ML) approaches have been proposed in this field, there is still a considerable gap between the state-of-the-art and accurate predictions, particularly when it comes to traffic forecasting.

Traffic behaviour on more granular urban street networks is fundamentally different from highways due to differences in spatial complexity and temporal variability:

Spatial complexity differences Connectivity is the density of connections in a road network *Online TDM Encyclopedia* (n.d.). Urban streets include numerous short links and intersections, so the density of connections in the urban road network is way higher than those in the highway network and they are characterized by more complex spatial dependencies than highways.

Temporal variability differences Because of the impact of traffic signals and traffic volume, the speed variability on an urban road segment is significantly higher compared to a highway Karr, Graves, Mockus, and Schuster (2002). Fig. 3.1 explores the difference in speed variability between the two different contexts of traffic. Fig. 3.1c shows the Gaussian distributions for speed in an urban road (3.1a) and highway (3.1b). In urban road segments, traffic speed is lower than highway speed. Besides, for urban road segment data, the Gaussian graph is short and wide, proving the standard deviation is significant compared to highway speed data with a narrow and tall Gaussian distribution.

In this chapter, we present experiments on urban streets and highway benchmark datasets that

demonstrate the robustness of our proposed model against substantial temporal variability and the ability to learn more complex spatial correlations than baseline methods.

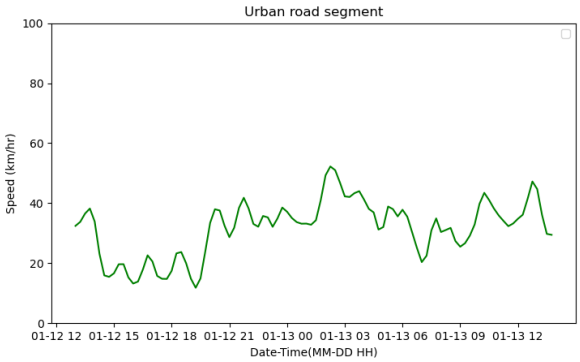
Previous research has studied different approaches such as statistical models, machine learning techniques, and deep learning methods to explore the spatial and temporal features in traffic data. RNN family Chung, Gulcehre, Cho, and Bengio (2014); Hochreiter and Schmidhuber (1997) combined with GNNs Z. Wu, Pan, Chen, et al. (2020) have shown great potential to learn complexities in traffic data effectively. Even though the GNN-RNN based methods introduced so far address both the spatial and temporal dependencies, these methods either do not fully exploit the spatial information or have an over-complex structure, particularly in graph convolution operation, that affects the training time. This chapter addresses the problem of cyclic graph networks with complex spatiotemporal inter-dependencies. The main contributions of this chapter include:

(1) Multilevel encoder architecture: We formulate the problem as a sequence-to-sequence modelling task and introduce a novel multilevel encoder architecture that abstracts and combines dependencies at multiple complexity levels. Uniquely, the proposed technique handles sparse and dense road networks and enhances the accuracy of the predictions.

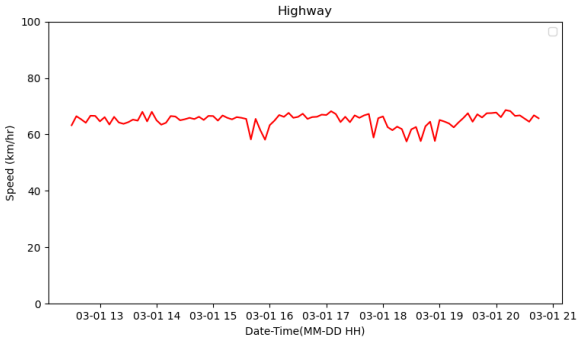
(2) Sparse architecture: Furthermore, when compared with state-of-the-art, our method employs a sparse architecture that improves time efficiency by reducing the computational complexity of the training.

(3) We present our experiments on the application of traffic forecasting and report the results on benchmark datasets of highway-level data METR-LA and the newly released street-level MSLTD dataset presented in this chapter. Detailed descriptions of the datasets are provided in Section 3.6.1.

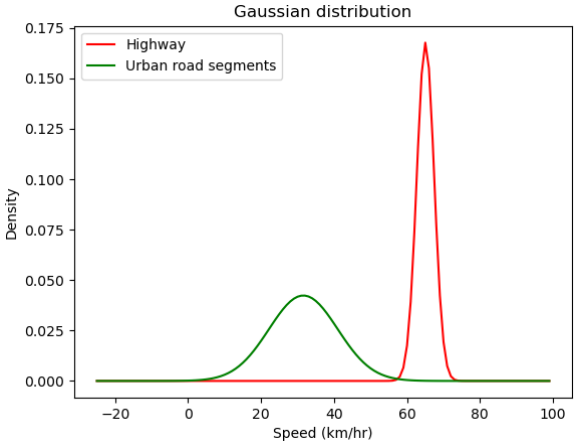
The rest of the chapter is organized as follows. Section 3.3 reviews the current state of the literature in traffic flow forecasting. Section 3.4 formally introduces the model, and Section 3.5 addresses the traffic prediction problem using the proposed model. Finally, Section 3.6 shows numerical experiments on real-world datasets and compares the proposed model's results with other state-of-the-art models reported in the literature.



(a)



(b)



(c)

Figure 3.1: Differences in data variability in two different contexts of traffic. Traffic speed changes over 24 hours. (a) An urban road segment from our new MSLTD street-level dataset, (b) a highway sensor from the METR-LA dataset, and (c) Gaussian distribution of the speed in (a) (green) and (b) (red).

3.3 Related Work

A wide range of data-driven methods regarding traffic prediction has been proposed in the past. Traditional approaches for predicting traffic state include Kalman-filter based methods J. Guo, Huang, and Williams (2014); Xie, Zhang, and Ye (2007), Vector Auto Regressive models Chandra and Al-Deek (2009), Autoregressive Integrated Moving Average (ARIMA) Van Der Voort, Dougherty, and Watson (1996) and variants such as seasonal ARIMA Kumar and Vanajakshi (2015) and Space-Time ARIMA Ding, Wang, Zhang, and Sun (2011). These statistical methods assume that traffic data has a fixed variability pattern. In reality, however, traffic data does not obey stationary assumptions, so traditional approaches mostly fail to achieve a good performance. To overcome this issue, machine learning methods with promising performance compared to linear methods became popular in time-series forecasting. Since then, many machine learning models, including K-nearest neighbour approaches Cai et al. (2016), Support Vector Machine models Cong, Wang, and Li (2016); YAO, SHAO, and GAO (2006), and Bayesian Network approaches Sun, Zhang, and Yu (2006) have been applied to traffic forecasting problem.

Before the emergence of deep learning, various shallow Neural network (NN) models were proposed to address traffic prediction, such as Multi-Layer Perceptron networks (MLP) Innamaa (2000) to predict traffic parameters such as speed and flow and study the effect of MLP parameters on prediction. Similarly, radial basis function (RBF) neural networks have been used for freeway traffic flow prediction with fuzzy c-means clustering to find the center position of the hidden layer Xiao and Wang (2004). Although shallow NN models improved traffic prediction compared to traditional methods, they are still limited in their ability to capture spatial and temporal dependencies in complex traffic data. This inspired researchers to apply deep learning methods. In Y. Lv, Duan, Kang, Li, and Wang (2014), the authors used Stacked Auto Encoders (SAE) with a standard logistic regression model to train the network in a supervised manner for traffic flow prediction. Yang, Dillon, and Chen (2016) proposed another class of autoencoders named stacked autoencoder Levenberg-Marquardt model to improve forecasting accuracy by learning features through multiple layers in a "greedy" way. W. Huang, Song, Hong, and Xie (2014) proposed a deep architecture comprising two main modules, a deep belief network (DBN) to learn the features automatically and

a multitask regression layer.

A Recurrent Neural Network (RNN) is a type of neural network containing at least one loop within the structure of network connections, enabling RNNs to have internal memory to remember a summarization of previous states and understand sequential information. This feature makes RNN models an excellent building block for a time-series forecasting network. LSTM and GRU are gated RNNs designed to overcome the vanishing gradient problems. Gated RNNs can also successfully learn temporal correlation in traffic data by capturing long-term information. Many RNN-based models such as bidirectional LSTM Cui, Ke, Pu, and Wang (2018), Mixture Deep LSTM R. Yu, Li, Shahabi, Demiryurek, and Liu (2017), shared hidden LSTM Song, Kanasugi, and Shibasaki (2016) models and GRU models Agarap (2018) have been applied to traffic forecasting. Y. Tian and Pan (2015) proposed an LSTM RNN model and demonstrated how it outperforms previously proposed models, including SVM, single layer feed-forward neural network (FFNN), and stacked autoencoder (SAE). Liu, Wang, Yang, and Zhang (2017) uses an LSTM model for travel time prediction. Ji and Hou (2017) has applied GRU models to predict bus trip demand and shown how GRU and NN models outperform ARIMA models for different prediction horizons. Although RNN-based methods consider the dynamic variations of traffic conditions, the future traffic state is affected by spatial dependencies besides the temporal changes. Since these models fail to explore spatial complexities, they cannot generate accurate predictions.

Recent studies Y. Wu and Tan (2016), Ma et al. (2017) have attempted to model spatial dependencies using the Convolutional Neural Networks (CNN) to address this issue. CNN models are a class of neural networks used primarily for extracting features from images Oquab, Bottou, Laptev, and Sivic (2014). To formulate the traffic prediction problem using CNNs, Ma et al. (2017) constructed a time-space matrix, where each element in the matrix contains information about the time interval and the street section associated with that element; H. Yu, Wu, Wang, Wang, and Ma (2017) also proposed CNN models to investigate the spatial correlation in traffic data; Z. Lv et al. (2018) proposed a combination of RNN and CNN models to take advantage of both by learning time-series dependencies and capturing the features related to the road network; and S. Guo, Lin, Li, Chen, and Wan (2019) introduced 3D convolution to capture spatial correlations and temporal correlations

from both long-term patterns and local patterns. However, CNN models can only extract spatial dependencies from a rigid grid structure representing the information, and abstract features extracted by CNN represent the relationship in Euclidean space.

There are numerous applications where Euclidean space cannot perfectly describe spatial dependence such as chemical synthesis, social network analysis, 3D vision and transportation networks. A graph is a powerful data structure that can express complex relationships in unstructured data Z. Wu, Pan, Chen, et al. (2020). Graph Neural Networks (GNNs) are deep learning-based methods that have been widely applied to graph domains in recent years. Neural networks based on the graph theory Butler and Chung (2006), Bruna, Zaremba, Szlam, and LeCun (2013) process the graph-structured data. In order to capture spatial dependence in a graph context, different types of Graph Convolutional Network (GCN) models have been introduced. GCNs are categorized into two main domains, spectral and spatial. The spectral graph convolution is based on graph Laplacian to localize a graph signal on the spectral domain, and the spatial graph convolution is the aggregation of information within a node's neighbourhood S. Zhang, Tong, Xu, and Maciejewski (2019). Most recent studies Y. Li, Yu, Shahabi, and Liu (2017), Cui, Henrickson, Ke, and Wang (2019), L. Bai, Yao, Li, Wang, and Wang (2020) on traffic forecasting focus on using a hybrid architecture of GCN to extract the spatial dependencies and RNN to explore the temporal changes. Y. Li et al. (2017) maps traffic state to a diffusion process on a directed graph and uses diffusion convolution integrated with gated recurrent units to capture both spatial and temporal dependencies. A sequence-to-sequence model is used in this study as a learning structure, in which they employ a scheduled sampling approach for training. L. Zhao et al. (2019) proposes a temporal graph convolution for traffic prediction and uses GRU to understand temporal dynamics. Cui et al. (2019) presents a traffic graph convolution operator and a convolutional LSTM to predict traffic speed. Z. Wu, Pan, Long, et al. (2020) avoid using a pre-defined graph in their proposed model, which includes a graph learning layer, a graph convolution component, and a temporal convolution component. L. Bai et al. (2020) also argues against pre-defined graphs since domain knowledge is necessary to generate the graph, as such information expressed by the graph might not be directly related to the prediction task, and may even introduce bias.

Sequence-to-sequence architecture is an end-to-end approach to processing sequence data. It includes a wide range of applications like speech modelling Sutskever, Vinyals, and Le (2014), natural language processing Ma and Hovy (2016), and recently time-series forecasting to enhance capturing long-term dependencies and enabling multistep prediction. Y. Li et al. (2017) used sequence-to-sequence modelling for traffic speed prediction by applying an encoder-decoder architecture, where the encoder and decoder blocks consisted of their proposed diffusion convolution recurrent neural network. C. Zheng, Fan, Wang, and Qi (2020) proposed a sequence-to-sequence architecture consisting of attention blocks to predict traffic. Z. Wang, Su, and Ding (2020) proposed an encoder-decoder design with LSTM blocks in addition to a control layer on top of it to cumulatively learn the data.

The literature indicates the strong potential of RNN models to learn the temporal information in traffic data. GCN has also achieved satisfactory results in unlocking the spatial relation in the traffic network. Although these approaches have improved accuracy considerably compared to previous methods, they either employ a complex network design that increases the number of parameters and training time or cannot achieve comparable accuracy to other deep learning tasks. This chapter proposes a novel multilevel sequence-to-sequence architecture to extract discriminative features effectively. The proposed model is based on a sparse architecture graph convolutional GRU enabling the network to reduce computational costs.

3.4 Proposed Model

3.4.1 Graph Notation

A graph is a data structure with a solid mathematical basis used to represent complex interactions between a set of objects. A graph denoted by $\mathcal{G}(\mathcal{V}, \mathcal{E})$ includes a set of $n \in \mathbb{Z}$ nodes represented as $\mathcal{V} = \{v_1, v_2, \ldots, v_n\}$ and a set of edges $\mathcal{E} = \{e_{i,j} | i, j \in \{1, 2, .., n\}\}$, where each edge e_{ij} indicates an interaction between nodes i and j. An adjacency matrix $\mathbf{A}_{n \times n}$ is one way to represent a graph. $\mathbf{A}$ is a square matrix, where a non-zero element at $\mathbf{A}_{i,j}$ indicates the existence of an edge connecting nodes i and j. Every node of a graph can have a set of features/attributes associated with it. To this end, a matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$ denotes an attribute representation of graph G, where

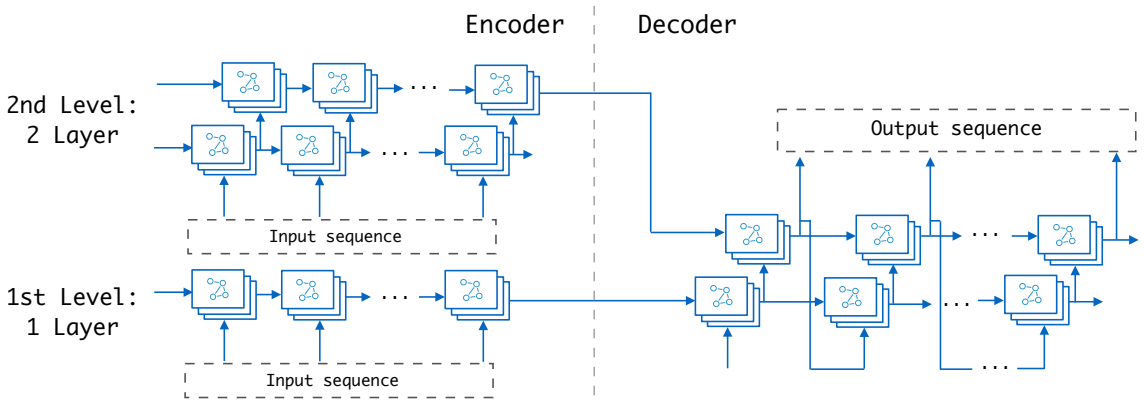


Figure 3.2: Multilevel Sequence-to-Sequence (MLS2S) Architecture for a 2-level encoder. The output from the last layer of each encoder is used to initialize the decoder. The number of encoder levels equals the number of decoder layers.

each row of the matrix is a vector that corresponds to a specific node of the graph representing its d features Hamilton (2020).

In a spatiotemporal graph, the objects that are the resource of temporal changes are considered nodes. The feature matrix denoted as $\mathbf{X}^t \in \mathbb{R}^{n \times d}$ dynamically changes over time t Z. Wu, Pan, Chen, et al. (2020). To represent a spatiotemporal forecasting problem as a graph, one must first translate the temporal data sequence to a graph signal. Each node v_i is assigned a feature vector $X_i^t \in \mathbb{R}^{1 \times d}$ where d is the length of the historical time-series of node i at time t needed to predict the future sequence.

3.4.2 Problem statement

Spatiotemporal forecasting aims to predict the future sequence of a parameter given its historical time-series. More formally, this is equivalent to finding a function $f : X^t, A \rightarrow Y^{t+h}$ which given an adjacency matrix $\mathbf{A}$ representing the graph structure and a sequence of observed data X^t representing d previous historical states of n nodes,

$$X^t = \begin{bmatrix} x_0^{t-d+1} & \dots & x_0^{t-1} & x_0^t \\ x_1^{t-d+1} & \dots & x_1^{t-1} & x_1^t \\ \dots & \dots & \dots & \dots \\ x_{n-1}^{t-d+1} & \dots & x_{n-1}^{t-1} & x_{n-1}^t \end{bmatrix}_{n*d}$$

it gives a sequence of values Y at a future time $t+h$ of up to h steps. Thus, Y^{t+h} is given by,

$$Y^{t+h} = \begin{bmatrix} x_0^{t+1} & x_0^{t+2} & \dots & x_0^{t+h} \\ x_1^{t+1} & x_1^{t+2} & \dots & x_1^{t+h} \\ \dots & \dots & \dots & \dots \\ x_{n-1}^{t+1} & x_{n-1}^{t+2} & \dots & x_{n-1}^{t+h} \end{bmatrix}_{n*h}$$

In this context, the graph nodes $\mathcal{V}$ represent a set of n objects generating time-series data, and the existence of an edge e_{ij} denotes a dependency between objects i and j. Since the rows of the matrices above correspond to the graph's nodes, the prediction is network-wide.

3.4.3 Cell Architecture

Cells are the basic building blocks of the prediction network. A cell consists of two main components. First, the spatial module captures the spatial features given the road network graph. The second component combines the spatial module operation with RNNs to learn temporal dependencies. As explained in the following sections, our work focuses on finding a practical sequence-to-sequence model solution, which at the same time maintains a simple cell design. As a result, the computational complexity for training decreases by more than a factor of 2 compared to state-of-the-art employing more complex architectures.

Spatial Learning Module Graph Convolutional Network (GCN) models aggregate node features and graph structure information to capture spatial correlation in data. For the spatial module, we leverage the spectral graph convolution operation based on normalized Laplacian by utilizing the simplified operation using Chebyshev polynomial approximation Kipf and Welling (2016):

$$H^{(k)} = \sigma(\hat{D}^{-1/2}\hat{A}\hat{D}^{-1/2}H^{(k-1)}W^{(k)}) + b^k \tag{3}$$

$$\hat{A} = A + I, \hat{D} = D + I$$

where A is the adjacency matrix of graph $\mathcal{G}$, D is the degree matrix, a diagonal matrix $n * n$ where D_{ii} represents the number of edges connected to node i. D_{ii} element value can be acquired by

summing up the row or column elements associated with node i in an undirected graph's adjacency matrix. W^k and b^k are the trainable weight and bias matrices for layer k, and σ is the activation function. H^{k-1} and H^k are the input signal and output of GCN operation on layer k, respectively.

Temporal Learning Module RNNs have achieved promising performance for natural language processing and sequence modelling applications. RNN gated models like Gated Recurrent Units (GRUs) can account for long-range dependencies based on their built-in architecture. We use GRU layers in our temporal module design to extract the temporal dynamics.

We integrate the spatial and temporal learning modules by replacing all the fully-connected layers of the GRU with the GCN layer introduced in the spatial module leading to,

$$r^t = \sigma(\mathcal{G}_{conv}(A, [X^t, h^{t-1}]) + b_r \tag{4}$$

$$u^t = \sigma(\mathcal{G}_{conv}(A, [X^t, h^{t-1}]) + b_u \tag{5}$$

$$c^t = tanh(\mathcal{G}_{conv}(A, [X^t, r^t * h^{t-1}]) + b_c \tag{6}$$

$$h^t = (u^t * h^{t-1})(1.0 - u^t) * c \tag{7}$$

where $\mathcal{G}_{conv}$ is the graph convolution operation from Equation 3, r^t and u^t are the reset gate and the update gate, c^t is the cell state, X^t is the input signal, $\sigma(.)$ and $tanh(.)$ are activation functions, and h is the hidden state. We aim at an efficient solution with reduced training time; hence we avoid adding complexity to the cell design by keeping the architecture simple.

3.4.4 Multilevel Sequence-to-Sequence Architecture (MLS2S)

Generally, Encoder-Decoder architectures include a pair of networks trained in tandem on target sequences given their input sequences. The encoder transforms the input series with variable size to a hidden state with a fixed size, and the decoder converts back the hidden state representation to a variable size prediction series Cho et al. (2014). It is not necessary to keep the input and output sequences fixed.

In this chapter, we propose a novel architecture that modifies the original encoder-decoder structure and uses multilevel encoders. The multilevel encoder integrates simple abstraction from the first level with the more detailed abstraction of higher levels and constructs a stack of hidden states with expressive information to initialize the decoder layers. The number of encoder levels matches that of the decoder layers since the final output state of every level of encoder is used as the input of the decoder layer at the same level. An example two-level MLS2S is shown in Fig. 3.2. The input sequence is a fixed-length time-series of historical data that indicates a sequence of inputs. The same input sequence feeds into the encoders at all levels.

The decoder uses its previous hidden state as input and the encoder output as its initial state to generate the prediction. For each encoder level, only the final layer hidden state is used as the initial state of the corresponding decoder layer.

Multilevel Encoder The encoders are the cell blocks introduced in Section 3.4.3. Given each element of the input sequence, the encoder layer l updates its state as follows:

$$h_l^t = f_l(h_{l-1}^t, h_l^{t-1}, A) \tag{8}$$

where h_0^t equals the input sequence X^T. After reaching the end of the sequence, the final hidden state of the encoder is acquired. The Multilevel Encoder is a network of multiple encoders reading the same input sequence. The first level is a one-layer encoder, and the subsequent number of layers increases incrementally for higher levels. Ultimately, the final hidden state of all levels are concatenated and become the initial hidden state of the decoder. For an L-level encoder, the final stacked hidden state is given by,

$$h = Concat[h_1^t, h_2^t, ..., h_L^t] \tag{9}$$

Decoder The decoders also follow the cell block architecture; they are trained to produce the predicted series given the hidden state and the previous output. For multilayer decoder, the hidden

state of the decoder on layer l at time t is computed by:

$$Y_l^t = f_l(y_{l-1}^{t-1}, h_l^{t-1}, A) \tag{10}$$

where h_l^0 equals the corresponding element of the final hidden state from encoder, $h[l]$. The final predicted sequence, for an L-layer decoder equals Y_L.

3.5 Methodology

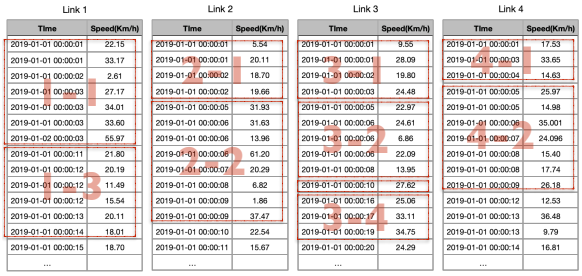
We investigate the use of the MLS2S model on traffic speed forecasting. We predict traffic speed parameters given past traffic speed observations and the road network. We represent the road network's graph $\mathcal{G}$ as an unweighted adjacency matrix A. Graph nodes $\mathcal{V}$ represent a set of n road segments or highway sensors, and a non-zero value for e_{ij} denotes that the two road segments i and j are connected. Every element of the historical and future sequence provides traffic speed at the specific time range t for all the nodes, $\{v_1, v_2, \ldots, v_n\}$ and the final prediction covers the entire road network.

At timestamp t, we simultaneously look ahead at the next η steps $\{t+1, t+2, \ldots, t+\eta\}$. We train the network using a Mean Absolute Error (MAE) loss between the predicted sequence and the ground truth given by,

$$MAE = \sum_{i=1}^{h} |Y^{t+i} - X^{t+i}| \tag{11}$$

where X^t is a vector providing ground truth speed values for the entire network at time t and Y^t is a vector providing the network-wide predicted speed values at time t.

We use backpropagation with Adam optimizer for the training with a learning rate decay ratio to decrease the learning rate by a constant factor as a function of time. The general intuition behind this strategy is to avoid being stuck in local minima, speed up the training process during the initial steps, and reduce the learning rate to prevent oscillation in the final epochs. We follow this approach because starting with a larger learning rate prevents the network from learning noisy data while decaying the learning rate over time enables the network to learn complex patterns You, Long, Wang, and Jordan (2019).



(a)

	2019-01-01 00:00:00	2019-01-01 00:00:05	2019-01-01 00:00:10	2019-01-01 00:00:15	...
Link 1	1 - 1	NAN	1 - 3		
Link 2	2 - 1	2 - 2			
Link 3	3 - 1	3 - 2	3 - 3	3 - 4	
Link 4	4 - 1	4 - 2			
...					

(b)

Figure 3.3: MSLTD dataset preparation process (a) Travel time records provided for links oredered by time(b) Speed matrix table covering the whole time range in 5 minute intervals for each link (the elements labeled in red, point to the group of travel time records in corresponding link table. The blank elements are to be filled based on the rest of information from links table)

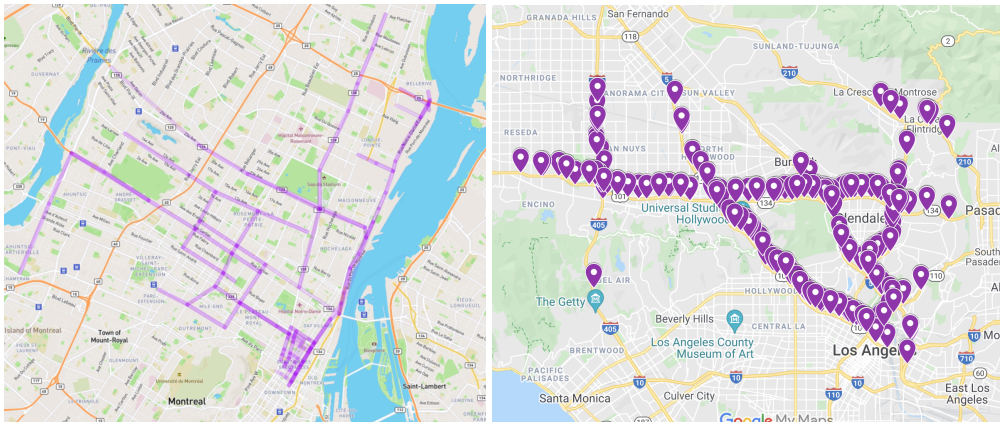
3.6 Experiments

3.6.1 Dataset Description

We present experiments on a highway-scale dataset and our newly introduced street-level dataset to demonstrate the efficacy of the proposed approach. We quantitatively evaluate the MLS2S model on two real-world traffic datasets: METR-LA containing the highway data of Los Angeles, USA, and the proposed MSLTD street-level segment traffic data from Montreal, QC, Canada.

Highway benchmark dataset METR-LA

This benchmark dataset contains the average traffic speed at 5-minute intervals for 207 locations. The data is collected from sensors located on highways in Los Angeles, USA, over four months from March 2012 to June 2012 Jagadish et al. (2014). The coverage of sensor points is visualized on a map in Fig. 3.4b.



(a) (b)

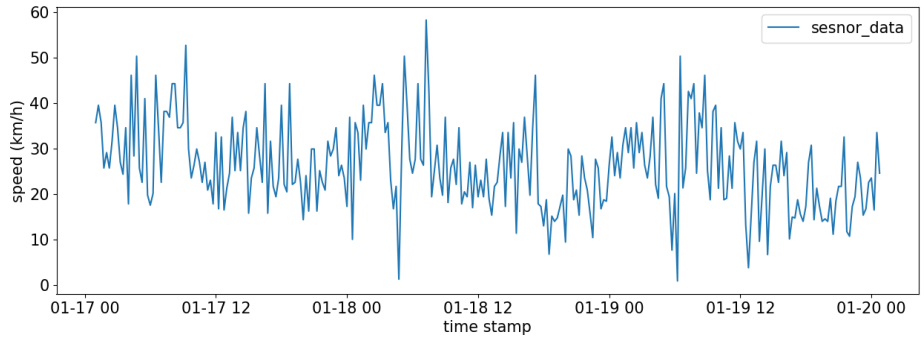
Figure 3.4: Datasets coverage (a) Montreal Street-level Traffic Dataset covering certain strategic road segments of Montreal. (b) METR-LA dataset sensor distribution.

Montreal Street-level Traffic Dataset

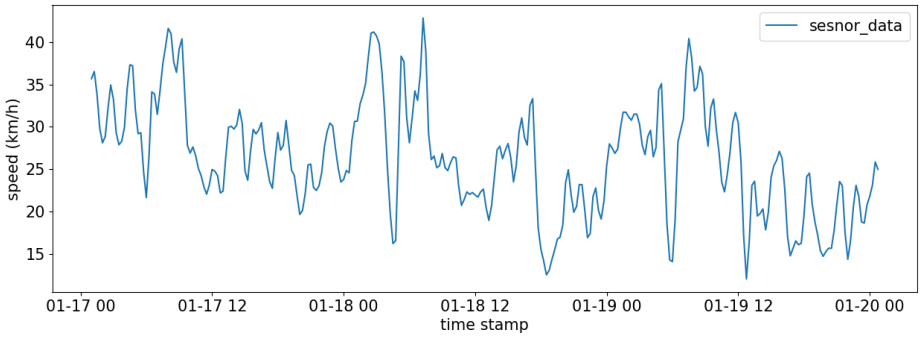
We present the Montreal street-level traffic dataset (MSLTD), which provides traffic speed over 15-minute intervals [1]. The data was collected from street segments in Montreal, QC, Canada, during the six months from January 2019 to June 2019. The raw data is the historical travel time on road segments provided by the City of Montreal. The City of Montreal has deployed a network of sensors using Bluetooth technology on specific strategic road segments to present speed, travel time, and origin-destination information of journeys *Travel time on road segments (historical)* (n.d.-a) together with the information on the road segments where travel times are collected *Travel time on road segments (historical)* (n.d.-b). For the first six months of 2019, the data contains more than 14 million travel time records on 252 road segments.

The adjacency matrix corresponding to the pre-defined graph for GNN methods is calculated based on the road segment information. The variable features are mapped to the nodes, and the static states are considered as the edges to have a fixed graph structure. The goal is to capture network structural flow correlation since traffic speeds at one node should be correlated to traffic speeds at other nodes with which they share links. Each road segment is represented as a node and contains information including its origin and destination detector IDs. If two roads share the same start or end points, then a connection is formed in the graph.

[1] Publicly available at https://github.com/naghm3h/MSLTD



(a)



(b)

Figure 3.5: Comparison of smoothing effect on data variation (a) before smoothing and (b) after smoothing operation using Gaussian filter.

$$\forall v_i, v_j | i \neq j, if OriginId_i == OriginId_j or\ OriginId_i == DestinationId_j \Rightarrow e_{ij} \in \mathcal{E}$$

To convert historical travel time data to traffic flow data, first, the travel time records whose corresponding segment information does not exist are removed. Since the dataset provides the average speed during a journey, for journeys taking more than 5 minutes, we split them into shorter time intervals.

Algorithm 1 Breaking up of long trips

Require: $trips = [\{link_id, start_time, end_time\}]$

1: **while** $end_time - start_time > interval$ **do**
2: $\quad trips.append(\{trip_id, start_time, start_time + interval\})$
3: $\quad start_time += interval$
4: **end while**
5: **return** $trips$

Next, a temporal list is created for each link covering January 2019 to June 2019 at 5-minute intervals as shown in Fig. 3.3b. We fill in the list elements by the aggregated speed information from

travel time records corresponding to each time interval as shown in Fig. 3.3a, the filled elements of the table point to the multiple travel time record matching the time interval. The traffic state data (speed) is set to NAN for the elements whose time interval does not match any travel time record. Links containing more than τ consecutive NAN elements ($\tau = 1000$) are removed because of excessive missing data.

After removing rejected links, the adjacency matrix is revised to exclude those links. Ultimately, 105 major road segments with the highest number of trips are selected. Link coverage is shown in Fig. 3.4a. At this step, the speed matrix is generated in 5-minute intervals for the selected segments and determines the speed through those specific segments. Since we are aggregating trips' data to fill in time intervals, for time intervals with multiple trips, we have multiple speed aggregation options as the representative speed of all trips in that particular timestamp, such as maximum, average, and minimum speeds of all trips that have occurred in that specific time range. In this work, for demonstrating MSLTD, we have selected the maximum speed since it better reflects the road network characteristics compared to other options that might be affected by drivers' behaviour, for example, the first element of the speed matrix in Fig. 3.3b is defined as:

$$1 - 1 : max(22.15, 33.17, 2.61, 27.17, 34.01, 33.60, 55.97)$$

for the element $1 - 2$ since there is no travel time record in its corresponding time interval, it is set to NAN. Element $3 - 3$ includes only one travel time record that would be the representative of this time interval itself.

Furthermore, for any remaining links with missing values, we aggregate the trip information in 15-minute time intervals to overcome the sparsity in the data. The rest of the missing temporal data for X-minute windows are replaced by the Historical Average method, using the weighted average of previous weeks.

The aggregation of the trip information may lead to noisy data for time intervals containing fewer trips, as shown in Fig. 3.5a. To address this, we apply a Gaussian filter to smooth the data, as shown in Fig. 3.5b.

Finally, the adjacency matrix represents road segment connectivity information, and the temporal matrix encodes the speed changes for all segments.

Table 3.1: Experimental results of various baseline methods conducted on METR-LA and Montreal datasets for different values of time steps ahead

Dataset		METR-LA			MSLTD		
		Horizon			Horizon		
Method	Metric	30 min	45 min	60 min	30 min	45 min	60 min
HA	RMSE	7.77	7.77	7.77	7.54	7.54	7.54
	MAE	4.15	4.15	4.15	5.58	5.58	5.58
	MAPE	12.9%	12.9%	12.9%	28.20%	28.20%	28.20%
VAR	RMSE	9.37	10.01	10.68	5.65	6.93	7.28
	MAE	5.40	6.07	6.50	4.27	5.25	5.51
	MAPE	12.75%	14.5%	15.84%	19.85%	25.46%	27.09%
GC-LSTM	RMSE	7.41	8.96	10.19	5.20	6.39	7.13
	MAE	4.16	5.32	6.32	3.99	4.77	5.22
	MAPE	11.38%	13.75%	15.68%	19.62%	23.82%	26.55%
DCRNN	RMSE	6.45	7.21	7.59	5.13	5.93	6.59
	MAE	3.15	3.42	3.60	3.63	4.59	4.93
	MAPE	8.8%	9.91%	10.5%	17.28%	21.42%	23.05%
MTGNN	RMSE	6.17	6.79	7.25	5.19	6.53	6.89
	MAE	3.05	3.29	3.50	3.86	4.84	5.10
	MAPE	8.21%	8.94%	9.88%	17.46%	23.37%	24.75%
AGCRN	RMSE	9.96	11.52	12.72	5.27	6.66	7.11
	MAE	4.31	5.06	5.74	3.93	4.97	5.30
	MAPE	10.87%	12.90%	14.75%	18.25%	24.18%	26.18%
MLS2S	RMSE	**5.62**	**6.12**	**6.51**	**4.92**	**5.86**	**6.28**
	MAE	**2.80**	**2.96**	**3.10**	**3.54**	**4.27**	**4.57**
	MAPE	**7.33%**	**7.96%**	**8.48%**	**16.31%**	**20.38%**	**22.00%**

3.6.2 Experimental Settings

For both datasets, we use 70% of the data for training, 10% for validation, and 20% for testing. We use 12 historical time steps information to predict the next horizon steps. For 30-minutes to 1-hour prediction horizon varies between 6 and 12 on METR-LA data since it is provided in 5-minute time intervals. It varies between 2 to 4 for Montreal Street-level Traffic Dataset that is presented in 15 minutes intervals. We train the proposed model using the Adam optimizer with a base learning rate of 0.01 and decay ratio of 0.1. The network is implemented using PyTorch v1.7.1. All experiments are conducted on NVIDIA GeForce RTX 2080 Ti with 11GB memory.

3.6.3 Baseline Methods

We compare MLS2S with the commonly used baseline methods and the state-of-the-art approaches: 1) Historical Average model (HA), which uses a weighted average of historical periods to predict future values; 2) Vector Auto-Regression (VAR), a statistical model of multivariate forecasting; 3) Diffusion Convolutional Recurrent Neural Network (DCRNN) Y. Li et al. (2017), a graph-based method that applies random walk on graph and uses common encoder-decoder structure, the spatial information in this method is provided to the network given a graph based on distances between sensors; 4) An improved version of Traffic Graph Convolutional Recurrent Neural Network (TGC-LSTM) Cui et al. (2019), which defines the graph based on road network topology. In addition to the spatial information and temporal changes provided to the network, their model has an extra dependency on road network characteristics information, which is provided to the network as a free-flow reachability matrix. Following their network design, we achieved similar performance while removing this dependency, see experiment results in Appendix. Thus, in the experiments conducted, we use our modified version (GC-LSTM) of their approach by only providing spatial and temporal information to the network. 5) Adaptive Graph Convolutional Recurrent Network (AGCRN) Z. Zhou and Li (2017), which considers temporal information as intra-dependencies and spatial information as inter-dependencies, then introduces two adaptive modules based on RNNs and GNNs, finally combines them to capture both dependencies; 6) Connecting the dots (MTGNN) Z. Wu, Pan, Long, et al. (2020), introduces a graph learning module to avoid using a predefined

graph for multivariate time-series forecasting.

3.6.4 Performance Comparison

Table 3.1 shows a comparison of the performance for the above-mentioned methods. We employ three widely used measures in traffic forecasting literature to measure the performance, including (1) Mean Absolute Error (MAE), (2) Mean Absolute Percentage Error (MAPE), and (3) Root Mean Squared Error (RMSE).

$$RMSE = \frac{1}{n}\sum_{i=1}^{h} |\frac{Y^{t+i} - X^{t+i}}{Y^{t+i}}| \tag{12}$$

$$MAPE = \sqrt{\frac{1}{n}\sum_{i=1}^{h}(Y^{t+i} - X^{t+i})^2} \tag{13}$$

The metrics are reported for three different prediction horizons, from 30 minutes to 1 hour. A comparison of the results performed on METR-LA dataset shows that the Historical Average method has a constant performance for different horizons since it relies on long-range information. Moreover, both baseline methods HA and VAR do not perform well compared to the deep neural network approaches. TGC-LSTM and AGCRN have a comparable performance which can be attributed to the fact that they both use GRU and GCN as their main network components. DCRNN achieves a better performance than the previously investigated methods, which shows the effectiveness of diffusion convolution operation in the encoder-decoder architecture. DCRNN and MTGNN performance results are almost in the same range for highway data. The proposed MLS2S approach outperforms all other methods on all reported metrics. In particular, the multilevel encoder design improves the MAE by more than 11% for one-hour prediction.

DCRNN and MTGNN employ different mechanisms to learn spatial and temporal dependencies. While the MTGNN network avoids using predefined graphs, DCRNN initially needs spatial information to be provided to the network in a graph format. To learn temporal dependencies, DCRNN makes use of RNN family while MTGNN utilizes dilated convolution. Finally, comparing their performance on Montreal's urban road segments dataset clearly shows that DCRNN outperforms MTGNN specifically for greater horizon values. In MLS2S network design, we also use

Table 3.2: Efficiency comparison of two top performing methods

	Training time per epoch(s)	
Method	MSLTD	METR-LA
DCRNN	105	382
MLS2S	**26.4**	**106**

RNN for addressing temporal complexities and predefined graphs to provide spatial information to the network. Given the two top-performing methods on the urban road segments dataset (MLS2S and DCRNN), we can conclude that using a predefined graph is more effective for urban road connections with a higher density adjacency matrix.

3.6.5 Computation Time

Table 3.2 shows training time per epoch for DCRNN and MLS2S, which are the two top-performing methods based on forecasting performance that use a predefined graph in their structure. MSLTD generally runs faster than METR-LA since their dimensions differ. The number of nodes for the MSLTD dataset is smaller, and although it covers six months, it is provided in 15 minutes intervals. MLS2S method reduces time complexity by more than half on both MSLTD and METR-LA datasets. The DCRNN graph operation design adds to the number of graph operations by applying diffusion process and dual random walks, while in our sparse architecture design, we avoid these complexities, which significantly reduces the training time per epoch.

3.7 Conclusion

In this chapter, we improve the efficiency and accuracy of graph-based deep-learning spatiotemporal forecasting with a novel multilevel sequence-to-sequence architecture for understanding correlations at different levels of abstraction. The proposed architecture is based on a sparse cell block that uses a traditional graph convolution operation combined with gated recurrent units to address the spatial and temporal dependencies. The cell block is designed in its simplest form to avoid complexities that result in additional computational time during the training of the network. To show the effectiveness of the proposed model on two different scales of traffic, we further introduced an

urban street-level segment dataset collected from Montreal, QC, Canada over a 6-month time period. Experiments on two real-world datasets show the advantage of the proposed model in terms of computational time and performance improvement.

Our work can inform decision-making for traffic management by providing improved predictions. On the other hand, the impact of computational time becomes more evident while running the model on a larger area with a considerable number of parameters in operational stages like predicting traffic covering all over the city.

As a spatiotemporal forecasting approach, this work has the potential to be applied to different research areas like meteorological forecasting, infectious disease surveillance, energy consumption, and economic analysis.

The road segment connections carry out more information than a pairwise relationship since all the links belong to the road network. For future work, we explore the higher-order relationship of data instead of the pairwise relationship represented by adjacent nodes information in graph-based neural network models.

Chapter 4

Hydrometric Prediction with Attention-Augmented Transduction

This chapter is a copy of the manuscript titled *TransGlow: Attention-augmented Transduction Model Based on Graph Neural Networks for Water Flow Forecasting*, published in the 2023 International Conference on Machine Learning and Applications (ICMLA). The manuscript was co-authored by Naghmeh Shafiee Roudbari, Charalambos Poullis, Zachary Patterson, and Ursula Eicker.

4.1 Abstract

The hydrometric prediction of water quantity is useful for a variety of applications, including water management, flood forecasting, and flood control. However, the task is difficult due to the dynamic nature and limited data of water systems. Highly interconnected water systems can significantly affect hydrometric forecasting. Consequently, it is crucial to develop models that represent the relationships between other system components. In recent years, numerous hydrological applications have been studied, including streamflow prediction, flood forecasting, and water quality prediction. Existing methods are unable to model the influence of adjacent regions between pairs of variables. In this chapter, we propose a spatiotemporal forecasting model that augments the hidden state in Graph Convolution Recurrent Neural Network (GCRN) encoder-decoder using an efficient

version of the attention mechanism. The attention layer allows the decoder to access different parts of the input sequence selectively. Since water systems are interconnected and the connectivity information between the stations is implicit, the proposed model leverages a graph learning module to extract a sparse graph adjacency matrix adaptively based on the data. Spatiotemporal forecasting relies on historical data. In some regions, however, historical data may be limited or incomplete, making it difficult to accurately predict future water conditions. Further, we present a new benchmark dataset of water flow from a network of Canadian stations on rivers, streams, and lakes. Experimental results demonstrate that our proposed model TransGlow significantly outperforms baseline methods by a wide margin.

4.2 Introduction

Accurate water flow prediction plays a crucial role in flood forecasting and mitigation. By understanding and predicting the dynamics of water flow, authorities can issue timely warnings and implement proactive measures to minimize the impact of floods, protecting human lives and reducing property damage. This proactive approach allows for better emergency response planning and the implementation of effective flood control strategies. Furthermore, water flow prediction is essential for optimal water resource management, fair distribution of water, ensuring sustainable use, and minimizing waste.

Water systems are interconnected with interdependencies, which can significantly impact hydrometric prediction. Water levels, flow, and quality changes in one part of the system can have cascading effects on the other parts. For example, changes in precipitation in one part of a river basin can affect water levels and flows downstream. These dependencies are challenging to understand, as different components can interact in complex ways that rely on various factors. Hence, it is essential to develop models that can capture the relationships between other system components. Spatiotemporal forecasting in water flow prediction involves capturing the complex relationships and patterns of water flow in a given geographical area over time. It takes into account the interconnections of hydrological processes across different locations and time intervals. The concept of spatiotemporal forecasting recognizes that water flow is not only influenced by local conditions

but also by the spatial context and interactions within the hydrological system. It considers how changes in one area can propagate and affect water flow patterns in neighboring or downstream locations. Additionally, it takes into account the temporal dynamics, such as seasonality, trends, and short-term variations, that influence water flow.

Previous studies in the field of hydrometric prediction can be categorized into distinct research areas. These categories include streamflow forecasting Kişi (2008), drought prediction Hao, AghaKouchak, Nakhjiri, and Farahmand (2014); Hao, Singh, and Xia (2018); Hao, Yuan, Xia, Hao, and Singh (2017), flood forecasting S. K. Jain et al. (2018); Karyotis et al. (2019); Nayak, Sudheer, Rangan, and Ramasastri (2005), and water quality prediction Ahmed et al. (2019); Haghiabi, Nasrolahi, and Parsaie (2018); Mohr, Drainas, and Geist (2021). Previous studies in hydrometric prediction have significantly contributed to the field; however, they are not without limitations. These studies have often relied on limited and fragmented datasets, which can result in uncertainties and reduced accuracy. Additionally, oversimplified assumptions about hydrological processes and the disregard of spatial and temporal variability can reduce the accuracy of the predictions. Furthermore, limited focus on realtime applications poses challenges for the field.

To address the mentioned challenges, we propose TransGlow, a spatiotemporal forecasting solution based on a transductive model with an augmented decoder hidden state using an efficient attention mechanism. The attention ability to focus on relevant parts of the input allows the model to reduce the risk of losing context information from the beginning of the sequence. In time series modeling, it is necessary to preserve the ordering information. However, the permutation-invariant self-attention mechanism results in temporal information lossZeng, Chen, Zhang, and Xu (2023). Recurrent neural network (RNN) family Dey and Salem (2017); Graves and Graves (2012); Medsker and Jain (2001) have been well known as state of the art approaches in sequence modeling and transduction problems such as language modeling and machine translation Datta, David, Mittal, and Jain (2020); Mikolov and Zweig (2012). We use attention in parallel with a vanilla RNN-based encoder to maintain ordering information and capture more relevant contextual information simultaneously. With this approach, the information can be spread throughout the RNN encoder and the attention layer, then selectively retrieved by the decoder allowing the model to process sequential data effectively. The main weakness of attention mechanism is the high computational complexity and

memory usage while computing the dot product. To avoid this, we use ProbSparse self-attention, an efficient attention mechanism proposed by H. Zhou et al. (2021).

The purpose of Graph Neural Networks (GNNs) Hamilton (2020) is to analyze data represented as graphs. They can operate on nodes and edges, capturing the graph's structural information and relationships. We can leverage the power of GNNs by incorporating Graph Convolution operations Kipf and Welling (2016) into the encoder-decoder design; therefore, we employ Graph Convolution Recurrent Neural Networks Y. Li et al. (2017) blocks to capture spatial dependencies and extract high-level representations for each water station. The self-learning graph structure has numerous benefits. It captures the changing relationships among variables over time, allowing the model to adapt to shifting patterns and dependencies. This dynamic graph construction is especially useful when the relationships between variables are ambiguous or change over time intervals. The model can learn to establish relationships between relevant variables based on their temporal dependencies and the current forecasting task. This adaptability enables TransGlow to capture both local and global dependencies between variables, resulting in more accurate forecasting. Our principal contributions are as follows:

(1) An augmented transduction model with an efficient attention mechanism for spatiotemporal forecasting.

(2) To the best of our knowledge, this is the first study of water flow forecasting from a graph-based perspective to learn the actual correlation between drainage basins.

(3) We present our experiments on 186 drainage basins across Canada. The raw data is provided by the Environment and natural resources of Canada of Canada (2018). Experimental results show that our method outperforms the state-of-the-art methods on all prediction horizons and performance metrics. Detailed descriptions of the datasets are provided in Section 4.5.

The rest of the chapter is organized as follows. Section 4.3 reviews the current state of the literature in spatiotemporal forecasting. Section 4.4 formally states the problem and the proposed methodology. Finally, Section 4.5 includes dataset description, experimental settings, and experiment results on a real-world dataset to show the effectiveness of our proposed approach.

4.3 Related Work

A plethora of data-driven methods have been proposed for Time series Forecasting (TSF) in the past. The prevailing focus in the literature is on statistical approaches, with notable methods like Vector Auto Regressive models Zivot and Wang (2006), Autoregressive Integrated Moving Average (ARIMA)Lai and Dzombak (2020), and its variants Narasimha Murthy, Saravana, and Vijaya Kumar (2018). While statistical models have gained popularity due to their simplicity and interpretability, they heavily rely on assumptions related to stationary processes, which may not always hold true in real-world scenarios, especially for multivariate time series data. On the other hand, machine learning methods demonstrated a solid ability to learn nonlinearity in TSF. Kişi (2008) developed a wavelet model for stream flow prediction. For water quality prediction, Ahmed et al. (2019) proposed a Support Vector Machine (SVM) approach, and Haghiabi et al. (2018) proposed a Multi-Layer Perepcetron (MLP), the results showed that the model has good predictive performance compared to other baselines. All these studies demonstrate strong capabilities of machine learning for complex nonlinear feature extraction and improve prediction accuracy. However, limitations in capturing spatial and temporal dependencies for more complex data prompted the exploration of deep learning methods.

Recurrent Neural Networks (RNNs) Medsker and Jain (2001) with internal memory became an excellent choice for time-series forecasting, particularly Long Short-Term Memory (LSTM) Graves and Graves (2012) and Gated Recurrent Unit (GRU) Dey and Salem (2017) models with the ability to address vanishing gradient problems and effectively learning longterm temporal dependency. Numerous RNN-based models, such as bidirectional LSTM Siami-Namini, Tavakoli, and Namin (2019), Mixture Deep LSTM R. Yu et al. (2017), and GRU models H. Lin et al. (2022), were applied to TSF problems with impressive results. By introducing attention mechanisms Vaswani et al. (2017), different Transformer based methods have also been applied to TSF applications such as traffic M. Xu et al. (2020), air quality Liang et al. (2023), and energy H. Zhou et al. (2021) forecasting. However, a recent study Zeng et al. (2023) claims that Transformers are not effective for time series forecasting by comparing the Transformer-based models against simple one-layer linear

models. To validate this claim, we also conduct experiments to explore the impacts of transformer-based models in TSF; detailed descriptions are provided in Section 4.5.

To capture spatial dependencies, Convolutional Neural Networks (CNNs) were introduced, treating TSF data as a time-space matrix Ma et al. (2017). However, CNN models are limited to grid-like structures and Euclidean space representation. Then, Graph Neural Networks (GNNs) came into the spotlight, offering a powerful way to express complex relationships in unstructured data using graph-based data structure. Graph-based methods have been widely used in different spatiotemporal applications such as solar energy, traffic, and electricity Z. Wu, Pan, Long, et al. (2020). Still, their application in predicting hydrological-related parameters and water resources is relatively limited. Recent studies combined GNNs with RNNs to explore spatial and temporal changes L. Bai et al. (2020), achieving promising results. Encoder-Decoder architectures have shown great potential for processing sequence data Makin, Moses, and Chang (2020). Researchers applied this architecture to TSF, using GCRN Y. Li et al. (2017), attention C. Zheng et al. (2020), and transformer-based architectures M. Xu et al. (2020). Here we propose a novel transduction architecture using the attention mechanism to augment the hidden state and enable better information flow and context preservation. To utilize the strong potential of RNN models in capturing temporal information and GNN in uncovering spatial relationships, we use GCRN as the building blocks of our core Encoder-Decoder model. For the graph convolution operation in GCRN, we employ a self-learning graph module to adaptively understand implicit spatial dependencies.

4.4 Methodology

4.4.1 Problem Statement

The spatiotemporal input dataset containing water flow measurements at different monitoring stations over time is given as a two-dimensional tensor $X \in \mathbb{R}^{n \times d}$, where n refers to the total number of water stations generating their own time series, and d refers to the total number of time intervals covered by the dataset. $X^t \in \mathbb{R}^{n \times T}$ denotes a sequence of historical data from time $t - T + 1$ to time t over all the n resources.

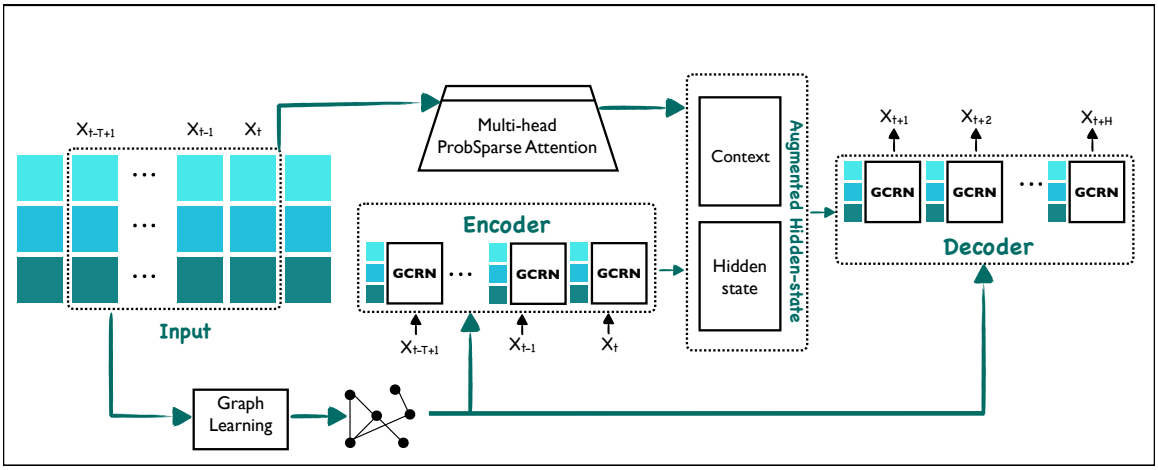


Figure 4.1: TransGlow main Architecture: graph learning module learns the graph to be used in GCRN blocks. The final hidden state of the Encoder is augmented by the attention context vector to pass over to the Decoder

$$X^t = \begin{bmatrix} x_0^{t-T+1} & \dots & x_0^{t-1} & x_0^t \\ x_1^{t-T+1} & \dots & x_1^{t-1} & x_1^t \\ \dots & \dots & \dots & \dots \\ x_{n-1}^{t-T+1} & \dots & x_{n-1}^{t-1} & x_{n-1}^t \end{bmatrix}_{n*T}$$

Here, the goal is to predict future tensor values up to H steps ahead (X^{t+H}) given historical data from the last T steps (X^t). The methodology involves improving accuracy by taking spatial dependency between all water stations (n nodes) into account through graph convolution operation. Since the graph structure is unknown, the graph adjacency matrix $A_{n\times n}$ needs to also be extracted. So, the objective is to find a function f that learns the graph as A from input sequence X^t:

$$f : X^t \rightarrow A$$

then find a function g that maps the input X^t and A to the future values:

$$g : X^t, A \rightarrow X^{t+H}$$

4.4.2 Graph Learning Module

The underlying graph structure for graph convolution operation is either defined based on distance and similarity functions or needs to be constructed. Having a predefined graph requires specific prior knowledge of the problem. The spatial relationship between the objects is often implicit, so even prior information might be biased and misleading for the prediction task. Several existing studies Z. Wu, Pan, Long, et al. (2020),Jiang et al. (2023),L. Bai et al. (2020) serve this purpose mostly as a function of the node embedding's product. Here we adopt the well-established adaptive graph generation defined in L. Bai et al. (2020):

$$\hat{A} = softmax(relu(E1.E2^T)) \tag{14}$$

where E1 and E2 represent randomly initialized node embeddings, which are subject to learning during the training process.

4.4.3 Graph Convolution Recurrent Block

The blocks of the encoder-decoder architecture consist of two primary modules: a graph convolution operation to capture spatial dependencies and an RNN-based unit to exploit temporal variability. GCRN has been firmly established as state of the art approach in spatiotemporal forecasting problems. Building upon the widely used approach of combining graph convolution within an RNN layer in the literature L. Bai et al. (2020); Jiang et al. (2023); Y. Li et al. (2017), we adopt the original GCRN formulation initially proposed by Y. Li et al. (2017), by eliminating the dual random walk. This modification is aimed at avoiding computational complexities. The graph convolution operation is a simplified version of normalized Laplacian formulated as:

$$H^{(l)} = \sigma(\hat{A}H^{(l-1)}W^{(l)}) + b^l \tag{15}$$

where $\hat{A}$ denotes the learned graph, W^l and b^l are the trainable weight and bias matrices for layer l, and σ is the activation function. H^{l-1} and H^l are the input signal and output of graph convolution on layer l, respectively. Finally, the MLP layers of the Gated Recurrent Unit are replaced by the

introduced graph convolution operation above to form the block:

$$r^t = \sigma(\mathcal{G}_{conv}(A, [X^t, h^{t-1}]) + b_r \tag{16}$$

$$u^t = \sigma(\mathcal{G}_{conv}(A, [X^t, h^{t-1}]) + b_u \tag{17}$$

$$c^t = tanh(\mathcal{G}_{conv}(A, [X^t, r^t * h^{t-1}]) + b_c \tag{18}$$

$$h^t = (u^t * h^{t-1})(1.0 - u^t) * c \tag{19}$$

where $\mathcal{G}_{conv}$ is the graph convolution operation from Equation 15, r^t and u^t are the reset gate and the update gate, c^t is the cell state, X^t is the input signal, $\sigma(.)$ and $tanh(.)$ are activation functions, and h is the hidden state.

4.4.4 Encoder-Decoder Model

The flexibility of the encoder-decoder design and its effectiveness in handling sequence-to-sequence tasks have made it a popular choice across various domains, such as machine translation Datta et al. (2020) and time series forecasting H. Zhou et al. (2021), enabling Neural Networks architectures to process and generate human-like data sequences. The vanilla encoder-decoder model works in a sequence-to-sequence manner. The encoder processes the input sequence to create a fixed-size representation that carries the relevant information. This context vector is then passed to the decoder, which uses it to generate the output sequence one step at a time. The model building blocks can be implemented using RNNs, LSTMs, or transformers.

The main problem of the vanilla encoder-decoder architecture is that it may suffer from the issue of information compression and loss. Since the encoder produces a fixed-size representation (context vector) to summarize the entire input sequence, it needs to capture all the relevant information within this fixed-size vector. However, this process can lead to lossly information compression, where important details from the input sequence may get lost or diluted in the context vector. Additionally, the encoder-decoder architecture may face difficulty in handling long sequences. When processing long input sequences, the encoder's fixed-size context vector may not be sufficient to retain all the essential information, resulting in inadequate generation of the output sequence by

the decoder. To address these problems, we propose a modification and improvement to the vanilla encoder-decoder architecture by using attention mechanisms to mitigate information loss, handle longer sequences more effectively, and improve the model's overall performance in spatiotemporal forecasting. The added layer between the input data and the decoder is a way of attending to the input sequence by an attention distribution mechanism that calculates a weighted sum of the inputs at all time steps. The final augmented hidden state H passed over to the decoder is:

$$H = Concat[h^t, C] \tag{20}$$

where h^t is the final hidden state of the encoder, and C is the context vector from the attention layer. The attention vector is then incorporated into the decoder's decision-making process, allowing it to focus on relevant information from the source data during decoding. Figure 4.1 visually illustrates an encoder-decoder model with an augmented attention layer.

4.4.5 Efficient Attention Layer

The bottleneck of the original attention mechanism lies in its quadratic computational complexity with respect to the sequence length. H. Zhou et al. (2021) have been able to alleviate the bottleneck of canonical attention and make Transformer-based models more scalable for sequence-to-sequence tasks by proposing ProbSparse Self-attention. Instead of having a full query matrix with all the queries for every token in the sequence, ProbSparse attention selects only a subset of k queries based on a certain measure or probability distribution:

$$\hat{Q} = M(Q, K) \tag{21}$$

$$Attention(Q, K, V) = Softmax(\frac{\hat{Q}K^T}{\sqrt{d}})V \tag{22}$$

where Q,K,V denote query, key, and value, respectively, and d is the input dimension. The measure or probability distribution M determines the importance or relevance of each token in the sequence with respect to the current token. Tokens with higher importance or relevance are more likely to

be included in the sparse query matrix, while tokens with lower importance have a lower probability of being included. Therefore, we utilize the ProbSparse attention mechanism for the attention augmented layer in the encoder-decoder design.

4.5 Experiments

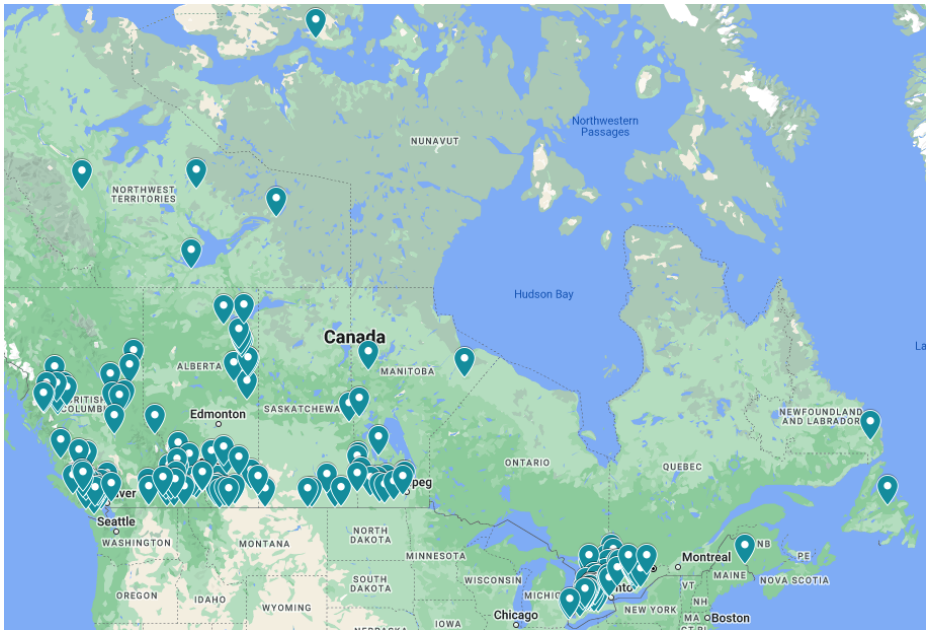
4.5.1 Dataset Description

In this work, we present a water flow daily discharge dataset called CWFDD-186, which provides daily discharge data from 186 stations on rivers, streams, and lakes across Canada. The raw data is provided by Environment and Climate Change Canada Environment and Canada (2023). The presented dataset, publicly available on the project repository, covers 40 years from 1981 to 2021. Figure 4.2a shows the station's coverage on the map, and 4.2b is the daily discharge of one of the stations over a year period.

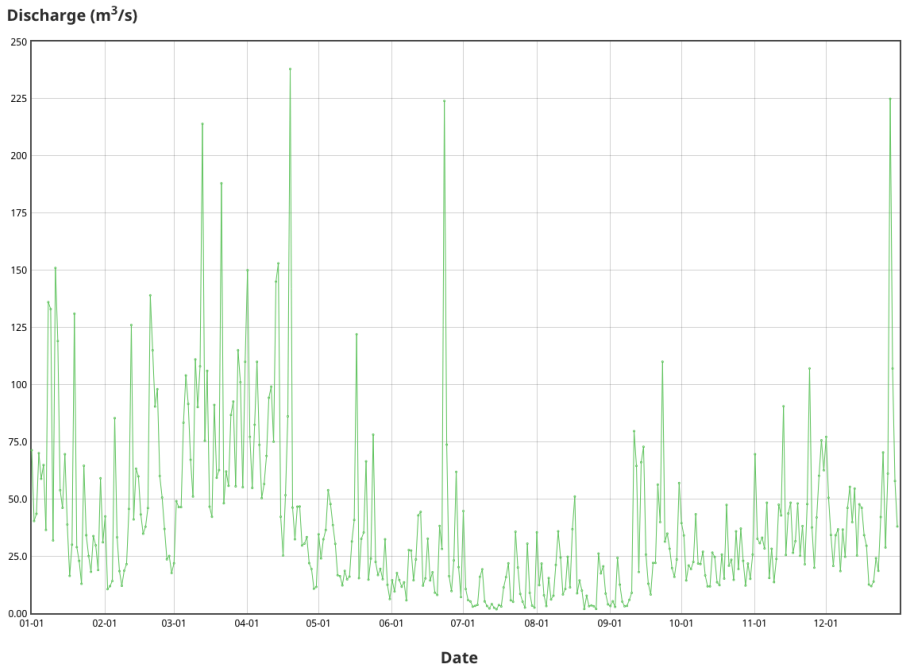
The missing values are replaced using the Historical Average method, with the weighted average of previous/next years. Occasionally spikes or dips can occur in the data due to sensor malfunctions, data transmission errors, or calibration issues with the measurement instruments. Environmental factors such as debris, ice, or sudden changes in water flow conditions can also lead to temporary fluctuations in the data. We have applied a Gaussian smoothing filter on data to address this without losing the general pattern and trends of the data. Smoothed data helps the model catch actual patterns by removing the noise.

4.5.2 Experimental Settings

We use 70% of the data for training, 10% for validation, and 20% for testing with batch size 64. The historical sequence length and the prediction horizon are both set to 12. The maximum number of epochs is set to 200, while the training may stop earlier if validation converges for 20 consecutive epochs. We use the Adam optimizer for training with Mean Absolute Error (MAE) as the loss function and curriculum learning for better generalization. The base learning rate is 0.01, the decay ratio is 0.1, and the number of heads for attention is 8. The network is implemented using PyTorch v1.7.1. All experiments are conducted on NVIDIA GeForce RTX 2080 Ti with 11GB memory.



(a)



(b)

Figure 4.2: CWFDD-186 dataset (a) 186 station's distribution on the map. (b) Data variability pattern for a specific sensor during a year.

4.5.3 Baselines

We compare our proposed TransGlow method with various baseline models, including graph neural networks, transformers, attention-based, and statistical methods. The baseline models include the following:

(1) The Historical Average model (HA) is forecasting method that predicts the future value of a time series based on the weighted average of historical data points.

(2) Vector Auto-Regression (VAR) is a statistical model used for multivariate time series forecasting. It is an extension of the univariate Auto-Regression (AR) model to handle multiple related time series variables simultaneously. VAR is widely used in econometrics, finance, and various other fields for analyzing and predicting the interactions between multiple variables over time.

(3) The Adaptive Graph Convolutional Recurrent Network (AGCRN) Z. Zhou and Li (2017) is designed to capture both temporal and spatial dependencies within graph-structured data effectively. AGCRN takes into account temporal information as intra-dependencies and spatial information as inter-dependencies. To achieve this, it integrates two crucial adaptive modules based on Recurrent Neural Networks (RNNs) and Graph Neural Networks (GNNs).

(4) The Informer model H. Zhou et al. (2021) is introduced as an enhanced version of the Transformer architecture specifically designed for efficient and accurate forecasting of long time-series sequences. The proposed modifications, including temporal attention and ProbSparse self-attention mechanisms, enable Informer to achieve state-of-the-art performance while efficiently handling long-range dependencies in time-series data.

(5) Spatial-Temporal Transformer Networks for Traffic Flow Forecasting (STTransformer) M. Xu et al. (2020) introduces a novel application of the Transformer architecture by proposing spatial and temporal transformers to capture spatial and temporal dependencies in traffic flow data. The model achieves state-of-the-art performance in traffic flow forecasting tasks by treating the traffic network as a graph and utilizing multi-head attention mechanisms.

(6) MTGNN Z. Wu, Pan, Long, et al. (2020) uncovers the relationships between variables using a graph learning module. The model also incorporates a unique mix-hop propagation layer and a dilated inception layer. All three modules are jointly learned within an end-to-end framework, ensuring optimal integration of the learning process.

(7) Discrete Graph Structure (GTS) Shang, Chen, and Bi (2021) is a scalable spatiotemporal forecasting method where the number of parameters does not grow quadratically with the number of time-series. It was originally proposed for traffic forecasting and uses the GCRN block structure and a graph learning approach they present in the paper.

4.5.4 Performance Comparison

Table 4.1 shows a comparison of the performance for the above-mentioned methods. We report performance metrics for three different horizons: 3, 6, and 12 days. Widely used measures in TSF literature are employed to report the performance, including (1) Mean Absolute Error (MAE), (2) Root Mean Squared Error (RMSE).

$$RMSE = \frac{1}{n}\sum_{i=1}^{h} |\frac{Y^{t+i} - X^{t+i}}{Y^{t+i}}| \tag{23}$$

The method HA has a constant performance for all prediction horizons since it is based on log range information. Both HA and VAR fail to represent a good performance. They do not consider any trends, seasonality, or other factors that may influence the time series, making it less accurate for long-term forecasting or when dealing with volatile data patterns. Nonetheless, HA provides a simple and fast solution for simple forecasting tasks. AGCRN improves MAE performance compared to the previous two baselines but performs poorly on RMSE. The next two methods are transformer-based approaches. Informer lacks considering spatial dependencies since it only has attention and no graph convolution operations in its architecture. Whereas STTransformer employs both attention and graph convolution, so it performs better than Informer. Although transformers perform well on most sequential data, they are not the best choice for TSF, as it was also shown in the findings from Zeng et al. (2023). In TSF the primary objective is to capture the temporal relationships between data points within an ordered sequence. Although positional encoding and token embeddings are

Table 4.1: Experimental results of various baseline methods conducted on water flow data for different values of time steps ahead

	Dataset	CWFDD-186		
		Horizon		
Method	Metric	3 Days	6 Days	12 Days
HA	MAE	44.82	44.82	44.82
	RMSE	61.39	61.39	61.39
VAR	MAE	23.29	30.19	36.39
	RMSE	94.17	131.76	173.55
AGCRN	MAE	11.27	13.64	16.40
	RMSE	44.57	46.30	49.64
Informer	MAE	17.68	17.93	18.76
	RMSE	44.71	45.51	48.19
STTransformer	MAE	9.44	11.03	14.33
	RMSE	27.20	30.85	39.27
MTGNN	MAE	7.98	9.47	58.72
	RMSE	24.04	27.59	107.34
GTS	MAE	7.33	9.93	12.53
	RMSE	22.56	28.97	34.52
TransGlow	MAE	**6.83**	**9.46**	**12.19**
	RMSE	**21.48**	**27.71**	**33.48**

utilized in Transformers to retain some level of ordering information for sub-series, the inherent nature of the permutation-invariant self-attention mechanism still leads to a loss of temporal information.

MTGNN performs well on smaller prediction horizons. Its poor performance for 12 days prediction can be attributed to the fact that it employs Convolutional Neural Networks to understand temporal dependencies, while RNNs have proved to be a better choice for this purpose. GTS is the second-best-performing method after our proposed TransGlow, which shows the effectiveness of our Augmented encode-decoder architecture. Finally, our proposed TransGlow outperforms the other methods over all prediction horizons.

4.5.5 Complexity

Table 4.2 shows the total number of parameters for each method. While larger models may have the potential to achieve better performance on complex tasks, it is essential to balance model complexity with efficiency considerations to find the optimal trade-off between model performance

and resource utilization. The total number of parameters can impact the efficiency of model training. During the inference phase, models with fewer parameters usually require less memory and computational power. Smaller models are often faster and more efficient in real-time applications. Of the two top-performing methods, GTS has the largest number of parameters, while our proposed TransGlow with the best MAE has third place in the parameters table.

Table 4.2: Efficiency comparison of baseline methods

Method	Number of Parameters
STTransformer	292,281
MTGNN	375,548
TransGlow	405,593
AGCRN	747,600
Informer	12,251,850
GTS	16,578,387

4.6 Conclusion

Water flow prediction is crucial for assessing the current water levels and identifying any sudden changes that could indicate an impending flood. TransGlow is designed to continuously predict water flow over various monitoring stations along rivers, streams, and other water bodies. It can be integrated into flood warning systems to trigger alerts when the predicted water flow exceeds certain critical levels. Our proposed model can also help create flood hazard maps that identify areas at high risk of flooding based on predicted water flow patterns.

Water flow in a hydrological system is highly dependent on the interactions between different monitoring stations, such as rivers, streams, and lakes. GNNs can naturally capture these spatial dependencies by considering the graph structure of the network, which allows the model to incorporate the influence of nearby regions. GNNs can also effectively model temporal dynamics using recurrent units, such as the Graph Convolution Recurrent Neural Network (GCRN). Understanding the graph structure in the context of water systems can be challenging. It may not be static and can change over time due to various factors, such as weather conditions or human interventions. Graph learning methods can adaptively learn the connectivity between stations based on the available data.

The encoder-decoder design is commonly used in prediction tasks. The benefit of the augmented attention layer proposed for the encoder-decoder architecture lies in its ability to focus selectively on relevant parts of the input sequence. The attention mechanism allows the decoder to access different parts of the encoded input sequence based on their importance for generating the output. Experiments on a real-world dataset show the advantage of our proposed model in terms of complexity and performance improvement.

Future works can explore integrating additional external factors influencing water flow, such as rainfall data, temperature, or land-use patterns. These factors can provide valuable context and improve the accuracy of the predictions. Investigating multi-task learning techniques to jointly predict other hydrological variables, such as water quality parameters or groundwater levels, alongside water flow can also lead to a more comprehensive understanding of the water system's behavior.

Chapter 5

Flood Forecasting Using Digital Twins

This chapter is a copy of the manuscript titled *From Data to Action in Flood Forecasting Leveraging Graph Neural Networks and Digital Twin Visualization*, published in Scientific Reports, 2024. The manuscript was co-authored by Naghmeh Shafiee Roudbari, Shubham Rajeev Punekar, Zachary Patterson, Ursula Eicker, and Charalambos Poullis. Both first and second authors contributed equally to this work. I was responsible for the computational modeling, while Shubham Rajeev Punekar contributed to the visualization and digital twin aspects.

5.1 Abstract

Forecasting floods encompasses significant complexity due to the nonlinear nature of hydrological systems, which involve intricate interactions among precipitation, landscapes, river systems, and hydrological networks. Recent efforts in hydrology have aimed at predicting water flow, floods, and quality, yet most methodologies overlook the influence of adjacent areas and lack advanced visualization for water level assessment. Our contribution is twofold: firstly, we introduce a graph neural network model equipped with a graph learning module to capture the interconnections of water systems and the connectivity between stations to predict future water levels. Secondly, we develop a simulation prototype offering visual insights for decision-making in disaster prevention and policy-making. This prototype visualizes predicted water levels and facilitates data analysis using decades of historical information. Focusing on the Greater Montreal Area (GMA), particularly Terrebonne,

Quebec, Canada, we apply our model and prototype to demonstrate a comprehensive method for assessing flood impacts. By utilizing a digital twin of Terrebonne, our simulation tool allows users to interactively modify the landscape and simulate various flood scenarios, thereby providing valuable insights into preventive strategies. This research aims to enhance water level prediction and evaluation of preventive measures, setting a benchmark for similar applications across different geographic areas.

5.2 Introduction

Since 2021, North America has been consistently experiencing devastating flood disasters that leave a trail of destruction in their wake. In a recent event, heavy rainfall and climate change-induced conditions triggered catastrophic flooding in British Columbia, Canada. It resulted in resident displacement, critical infrastructure damage, and a significant economic toll *B.C. floods caused at least $450M in damage* (n.d.). Communities were inundated, and the cleanup and recovery process were arduous and costly. Similarly, in 2021, the Mississippi River experienced severe flooding *There were flash floods, strong winds and at least two deaths in Mississippi* (n.d.), impacting several states. This inundation led to the destruction of homes, businesses, and agricultural lands. It disrupted transportation and caused widespread economic losses. The catastrophic floods underscore the importance of flood forecasting and simulation. Communities require timely and accurate warnings to prepare, mitigate, and respond effectively to impending flood events. By simulating potential flood scenarios, we can assess the extent of damage and plan for effective flood control measures. Simulation of barriers along the river sides where inundation is frequent during flooding helps in decision-making for the best distribution of preventative resources in a timely manner. This pivotal task not only bolsters public safety but also underpins resilient urban planning, climate change adaptation, and scientific research, making it an essential component of modern-day disaster preparedness and response efforts.

One of the fundamental difficulties in flood forecasting arises from the intricate network of interconnected hydrological systems. Rivers, streams, tributaries, and drainage networks collectively form a dynamic, interdependent web. Predicting how water will traverse this network, especially

in response to extreme weather events, requires sophisticated modelling. The cascading effect of changes in one part of the system can have far-reaching consequences downstream, adding layers of complexity to flood forecasting. Consequently, flood forecasting goes beyond time series forecasting, evolving into a spatiotemporal problem. Graph Neural Networks (GNN) are designed to model complex relationships and spatial dependencies. For flood prediction, using a GNN-based model is necessary because it can accurately represent how hydrological systems are connected and changing over time. This gives us a better understanding of how water moves through water system networks. Their adaptability to changing conditions, reduced need for manual feature engineering, and capacity to integrate diverse data sources make them indispensable for improving prediction accuracy.

Another challenge is the spatial variability of flood events. Floods vary significantly across different regions and are not uniform. The local topography, land use, and geographical features significantly influence flood dynamics. Therefore, forecasting must account for these site-specific conditions, demanding meticulous and often site-specific modelling. This spatial variability can complicate predictions, as what holds true for one area may not apply to another, even within the same watershed. Our digital twin simulation addresses this challenge by creating a dynamic and detailed virtual replica of the real-world environment. Simulating flood scenarios within the digital twin makes it possible to account for these spatial variations and understand how they influence flood dynamics. Different scenarios can reveal the sensitivity of the system to specific variables. For instance, we assess how the placement of concrete barriers or the density of tree trunks influences water flow and how it impacts floods. This sensitivity analysis aids in identifying critical factors and their associated variability. Moreover, visual representations of potential flood scenarios are critical in engaging and educating local communities about flood risks and the importance of being prepared. This visual communication is instrumental in raising awareness and supporting flood mitigation efforts.

New research in hydrological prediction covers a wide range of topics. It helps us understand and predict various water-related events, like streamflow patterns Gore and Banning (2017), rainfall-runoff modeling Jaiswal, Ali, and Bharti (2020), drought onset prediction L. Xu, Chen, Zhang, and Chen (2018), and flood forecasting Dtissibe, Ari, Titouna, Thiare, and Gueroui (2020). However,

many of these works often need to adequately consider spatial complexity, which is the cascading influence of water system components on each other. The omission of spatial variables, like localized rainfall variations, terrain differences, and land use changes, can limit the precision and reliability of predictions. Future research should bridge this gap by ensuring a more holistic understanding of hydrological processes to improve predictive capabilities.

In addressing the current challenges in flood forecasting, we have employed a cutting-edge spatiotemporal forecasting approach using graph neural networks. This technique, recently introduced in Roudbari, Poullis, Patterson, and Eicker (2023), outperforms traditional methods in predicting water flow. The TransGlow model uses Graph Convolution Recurrent Neural Network (GCRN) blocks to build an encoder-decoder structure. These blocks preserve sequence information and unravel spatial correlations. The transductive model structure is further enhanced with an augmented encoder layer employing attention mechanisms to focus on important elements of the input data. We adapt the TransGlow model to the specific context of Quebec, focusing on water level predictions in Terrebonne. We further compliment the flood forecasting framework with a flood-simulation prototype comprising Terrebonne's digital twin. Hence, our approach not only offers accurate flood forecasting but also improves our understanding of flood variability across different city zones and provides important insights for effective mitigation strategies. Our key contributions are as follows:

(1) We present an extendable flood simulation and visualization technique, creating a detailed digital model of the city of Terrebonne, QC for flood simulation. [1] This is a significant advancement, providing a valuable tool for understanding and predicting flood dynamics specific to this area. Notably, the proposed technique is designed to be reproducible and, more importantly, generalizable to other geographic locations, leveraging open-source data and standard practices to ensure its applicability beyond this initial area of focus. Our approach utilizes historical data and predictions from a graph neural network model to simulate changes in the water levels of nearby water bodies. Furthermore, our method supports interaction, including the simulation of concrete barriers for the purposeful redirection of river flow during flood events. By simulating and visualizing the controlled redirection of floodwaters, our approach facilitates the design of

[1] To ensure that the images shown in the chapter are consistent, we focus on a 1.25x1.25km^2 area encompassing Terrebonne, with a central focus on Ile St. Jean, adjacent to the Mille-Iles (Riviere Des) Au Barrage De Terrebonne weather station located at 45°41'35"N 73°38'58"W.

effective mitigation strategies tailored to diverse scenarios in a timely fashion.

(2) We introduce a state-of-the-art GCRN-based method, originally developed for predicting hydrological parameters and now extended to local-scale flood prediction in Terrebonne, QC. Extensive experimentation has shown that this adaptation leads to better performance compared to current methods, emphasizing the effectiveness of the approach and its potential to improve flood prediction accuracy in specific local contexts.

In summary, our work contributes to (i) more precise flood forecasting using state-of-the-art deep learning frameworks, and (ii) evaluating mitigation strategies using a digital twin for simulation and visualization. By understanding localized flood patterns, stakeholders can make informed decisions to protect the community and infrastructure, ultimately reducing the impact of floods on the city. Furthermore, the research lays the foundation for similar applications in various geographical areas, extending the potential benefits beyond the city of Terrebonne, QC.

5.3 Related Work

In this section, we offer a concise overview of related work in the fields of digital twin creation and forecasting methodologies.

5.3.1 Digital Twins

Recent years have seen significant improvements in the use of geospatial data generated from LiDAR scans for landscape modelingGamba and Houshmand (2000). Along with high-resolution topography data, other geospatial data sources, such as roads, building footprints, have been effectively used to create detailed and accurate representations of urban landscapes. The integration of deep learning techniques with photogrammetry and satellite imagery has revolutionized the field of 3D reconstruction. However, these works focus primarily on the visual acuity of the reconstructed models, rather than their physical aspects. For instance, 3D point clouds reconstructed with methods in photogrammetry such as structure-from-motion and multi-view stereo algorithms Massimiliano Pepe and Crocetto (2022); Schönberger and Frahm (2016); Schönberger, Zheng, Pollefeys, and Frahm (2016) or implicit 3D representations such as neural radiance fields Mildenhall et al. (2020); Xiangli et al. (2022), while highly useful in visualizing the city do not produce static meshes that

can test for physics-based collision with fluids, as required by the problem statement at hand. To model urban features digitally which render with lower visual acuity but provide accurate physical interactions, techniques such as procedural extrusion of the building footprintsLedoux and Meijers (2011) or deep learning based reconstructionD. Yu, Ji, Liu, and Wei (2021) have been proposed to produce models with low level-of-detail.

The simulation of fluids, particularly for applications like flood prediction, has greatly benefited from advances in physics-based computational models. Research in this area has focused on the development of scalable and accurate simulation algorithms capable of handling the complex dynamics of water movement. Various physics-based fluid simulation techniques have been explored in literature, such as Eulerian methodsChentanez and Müller (2011), which track the fluid properties of fixed points in space, and Lagrangian methodsMüller, Charypar, and Gross (2003), which track individual particles moving in space and time. Novel methods combining Eulerian and Lagrangian approachesMacklin and Müller (2013) use particles to carry information about fluid motion and a Eulerian grid to solve for pressures and velocities. However, these methods are computationally expensive to be used on consumer hardware, which precludes them from being used in implementations to be accessed by stakeholders at ground zero during flooding, to offer insight for the decision making process. With this objective in mind, we refer to the work of Chentanez and MüllerChentanez and Müller (2010) which represents the fluid surface as a two dimensional heightfield grid, where each cell in the grid stores a height value indicating the height of the fluid surface at that point. The simulation works by iteratively updating the height values of the grid cells based on physical principles like conservation of mass and momentum, as well as external forces like gravity and wind. It offers a good balance between computational efficiency and visual fidelity, making it suitable for real-time applications as well as high-quality rendering. This approach is particularly suited for large-scale simulations like rivers where vertical detail -such as formation of waves and splashing of water- is less important than the overall motion and surface appearance.

Our area of research involves the integration of a detailed 3D city model with appropriate fluid simulation technique to enhance flood prediction and management. In our proposed work, we implement the simulation and visualization application using the Unreal Engine as the foundational framework, augmented with supporting libraries for the procedural generation of urban features

such as roads and buildings, and the fluid height field simulation. This interdisciplinary approach allows for more accurate simulations of how water interacts with urban infrastructure, aiding in the development of more effective flood defense and urban planning strategies.

5.3.2 Forecasting

Time series forecasting is a fundamental task in various domains, from economics to meteorology, and it has witnessed a continuous evolution in methodology over the years. Traditional time series forecasting methods, particularly those rooted in statistical techniques, have historically assumed linearity within the data. These methods include Vector Auto Regressive models Zivot and Wang (2006), Autoregressive Integrated Moving Average (ARIMA) Ho, Xie, and Goh (2002), and exponential smoothing Taylor (2003). The assumption of linearity simplifies the forecasting process but is often at odds with the intricate, nonlinear patterns frequently present in real-world time series data. However, machine learning has shown remarkable aptitude in uncovering and modeling non-linearity within time series data, significantly enhancing forecasting accuracy across various domains. In their work, Makwana and Tiwari (2014) introduced a wavelet-based model for stream flow prediction. Tan, Yan, Gao, and Yang (2012) explored a Support Vector Machine (SVM) approach for water quality prediction, while Niroobakhsh, Musavi-Jahromi, Manshouri, and Sedghi (2012) utilized a Multilayer Perceptron (MLP) and Radial Basis Function (RBF) model for the same task. Their findings demonstrated that these models exhibited strong predictive performance compared to conventional baseline methods. While machine learning methods show proficiency in learning non-linearity in time series data, the ever-increasing complexity of modern datasets necessitates exploring deep learning approaches.

Recurrent Neural Networks (RNNs) with internal memory Medsker and Jain (2001) have emerged as a compelling choice for time-series forecasting. Specifically, models like Long Short-Term Memory (LSTM) Graves and Graves (2012) and Gated Recurrent Unit (GRU) models Dey and Salem (2017) have proven proficiency at mitigating issues such as vanishing gradients while effectively capturing long-term temporal dependencies. The integration of attention mechanisms Vaswani et al. (2017) has extended the horizons of time series forecasting. Transformer-based methods have

made notable progress in Time Series Forecasting (TSF) applications, spanning domains like hydrometric forecasting Wei, Wang, Schmalz, Hagan, and Duan (2023), air quality prediction Méndez, Montero, and Núñez (2022), and energy forecasting C. Wang et al. (2022). These models have showcased their effectiveness in modeling complex temporal relationships.

In the context of spatiotemporal forecasting, the landscape of TSF has witnessed an evolution driven by innovative approaches and their intersection with the spatial domain. Initially, to capture spatial dependencies within time series data, Convolutional Neural Networks (CNNs) emerged as a solution, treating TSF data as a time-space matrix Ma et al. (2017). However, CNN models exhibit limitations in their applicability, particularly when dealing with non-grid-like structures and the representation of data in Euclidean space.

Subsequently, the spotlight turned to GNNs, which offered a robust framework for expressing intricate relationships within unstructured data via a graph-based data structure. While graph-based methods found extensive use in various spatiotemporal applications, including renewable energies Khodayar and Wang (2018), traffic C. Zheng et al. (2020), and electricity forecasting Z. Wu, Pan, Long, et al. (2020), their utilization in predicting hydrological-related parameters and water resources remained relatively restricted.

Recent studies have bridged this gap by combining GNNs with Recurrent Neural Networks (RNNs), exploring the interplay of spatial and temporal changes Geng, He, Xu, and Yu (2022) and achieving promising results. The advent of Encoder-Decoder architectures, renowned for their efficacy in processing sequential data Makin et al. (2020), has further propelled innovations in TSF. Researchers have adapted this architecture to TSF, employing models such as GCRN Y. Huang, Weng, Yu, and Chen (2019), attention-based mechanisms H. Zhou et al. (2019), and Transformer-based architectures S. Wu et al. (2020).

In this context, we leverage a novel transduction architecture Roudbari et al. (2023) that uses the attention mechanism to enhance the hidden state, facilitating improved information flow and context preservation. This method harnesses the strengths of RNN models in capturing temporal information and GNNs in unveiling spatial relationships. To support the Graph Convolutional operation, the prediction model uses a self-learning graph module that can autonomously discover the implicit connections within the data. This is invaluable since sometimes the actual network structure

between different water stations is not known in advance or evolves over time.

5.4 Problem Formulation

The input to the prediction network can be represented as a two-dimensional matrix, denoted as X. This matrix serves as the fundamental input data structure to capture the spatiotemporal dynamics of water levels, where each row in the matrix represents a different water station, and each column corresponds to a specific timestamp covering T time steps (from $t-T+1$ to t). Thus, each element of the matrix, represented as $X[i][j]$, signifies the water level at a certain water station (indexed by i) for a particular time stamp (indexed by j), allowing the prediction network to learn and model the complex relationships and patterns associated with water levels at different stations over time.

$$X = \begin{bmatrix} x_1^{t-T+1} & \dots & x_1^{t-1} & x_1^t \\ x_2^{t-T+1} & \dots & x_2^{t-1} & x_2^t \\ \dots & \dots & \dots & \dots \\ x_n^{t-T+1} & \dots & x_n^{t-1} & x_n^t \end{bmatrix}$$

The objective is to find a function f that captures the connections between water systems. The function f maps the spatiotemporal matrix X to a connectivity matrix A, representing the relationships between different water stations, $A = f(X)$. Given the connectivity matrix A, the current spatiotemporal matrix X, and a specified number of future time steps h, the goal is to find a function g that predicts the spatiotemporal matrix X^h for the future h time steps. Hence, function g takes the current connectivity matrix, the current spatiotemporal matrix, and the number of future time steps as inputs and produces the predicted spatiotemporal matrix for the future, i.e. $X^h = g(C, X, h)$.

The overall objective is to find functions f and g that enable the network to understand the connections between water systems from the current spatiotemporal data and use this understanding to forecast water levels at multiple stations for future time steps.

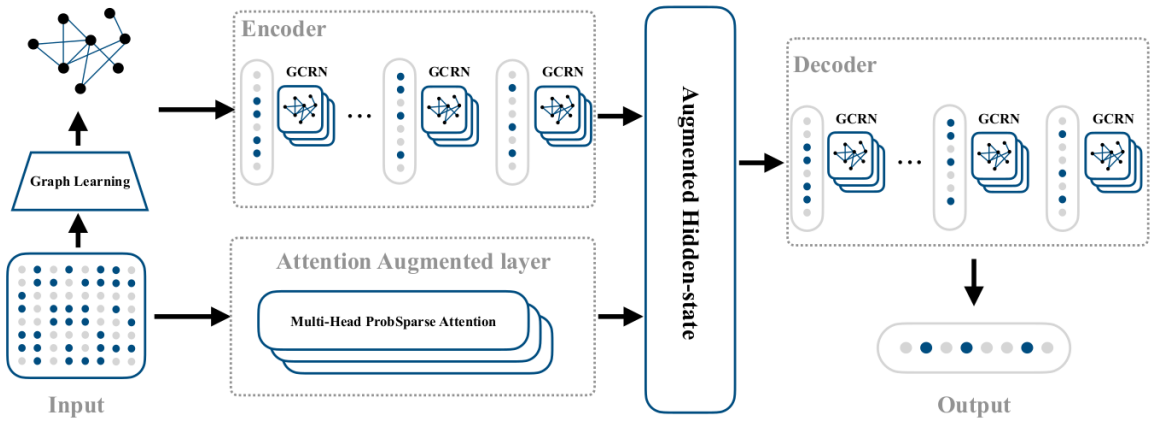


Figure 5.1: figure

Network architecture. The model input is a spatiotemporal matrix, the model learns the graph and makes prediction in a GCRN based encoder-decoder design, an attention layer is added to augment the decoder input

5.5 Methodology

This section details the theoretical approach utilized in this work for predicting the water level, and is illustrated in Figure 5.1. An extended explanation can be found in the Supplementary Material, Section "Graph Neural Networks".

5.5.1 Architectural Enhancement: Attention-Augmented Encoder-Decoder

The encoder-decoder design, inspired by RNNs and their variations, addresses the challenges of capturing and processing sequential information in a wide array of applications, from natural language processing, where it excels in language translation tasks Sutskever et al. (2014), to time series forecasting, where it adeptly handles the intricacies of temporal patterns He, Chow, and Zhang (2018).

At its core, the encoder-decoder structure is a two-step process strategically tailored to handle sequences of variable lengths. The encoder, functioning as the initial stage, systematically processes the input sequence, distilling its salient features into a fixed-size representation known as the context vector.

This condensed context becomes a comprehensive summary of the input, capturing its essential characteristics. The vanilla encoder-decoder architecture processes an input sequence $X =$

$(x_1, x_2, ..., x_T)$ with an encoder to create a context vector. The encoder processes each element of the input sequence sequentially, producing hidden states h_t for each time step t, i.e., $h_t = Encoder(x_t, h_{t-1}), C = h_T$ where C is the context vector and the function $Encoder$ can be any recurrent or non-recurrent layer, such as an LSTM or GRU. In this chapter, the foundational component is GCRN. It captures the information from the input sequence and creates a hidden state for each time step. Then, the decoder generates an output sequence using this context vector. For each time step in the output sequence, the decoder generates an output y_t and updates its hidden state s_t, i.e., $s_t = Decoder(y_{t-1}, s_{t-1}), s_0 = C$ where the function $Decoder$ is similar to the encoder's function but typically has a different set of parameters.

In order to tackle challenges related to information loss and compression, the model refines the traditional encoder-decoder architecture by introducing an augmented attention layer. This supplementary attention layer dynamically focuses on elements within the input sequence and calculates an attention vector that emphasizes the significance of different segments of the input sequence. The resultant attention vector $Attention$ is subsequently merged with the final hidden state obtained from the encoder, $C = Concat[h_T, Attention]$. This fusion produces an enriched context vector tailored for the decoder. By incorporating this augmented attention mechanism, the model adapts more flexibly to varying degrees of importance within the input sequence. This modification enhances the model's capacity to reduce information loss and improves its ability to generate contextually rich output sequences in sequence-to-sequence tasks.

The attention mechanism plays an important role in sequence-to-sequence tasks by allowing the model to focus on different parts of the input sequence when generating each element of the output sequence. However, the standard attention mechanism has a quadratic computational complexity in terms of sequence length. This is primarily because, for each element in the output sequence, the model computes an attention score for every component of the input sequence. Our used prediction model adopts ProbSparse attention, as proposed by H. Zhou et al. (2021). In this version of attention, a subset of K queries is selectively chosen based on specific measure M:

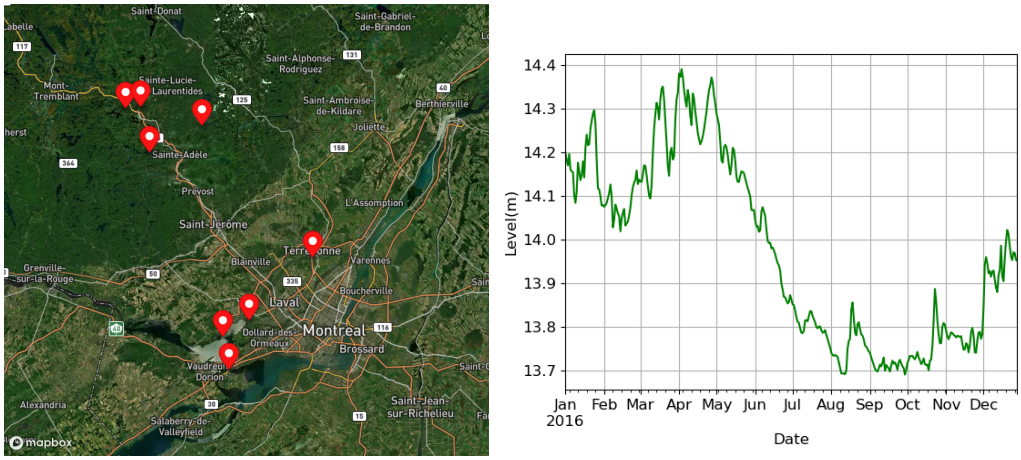
$$\hat{Q} = M(Q, K)$$
$$Attention(Q, K, V) = Softmax(\frac{\hat{Q}K^T}{\sqrt{d}})V$$

where Q,K, and V denote query, key, and value, respectively, and d is the input dimension. The probability distribution M decides the relevance of each token in the sequence relative to the current token. Tokens with higher importance have a higher chance of being incorporated into the sparse query matrix, while tokens with lower significance have a lower probability of inclusion. This approach ensures that the attention mechanism selectively focuses on relevant tokens, contributing to the overall efficiency of the model.

5.6 Experiments

The experiments in this work focus on Terrebonne, Quebec, Canada, situated in the North Shore region of the Greater Montreal Area. Terrebonne's climate features warm summers and cold winters with variable weather, including heavy rainfall and rapid snow melt periods. While the research is rooted in Terrebonne, its proposed approach transcends geographic constraints, offering generalization potential for application in other cities. We leverage water level data from 8 monitoring stations across rivers, streams, and lakes, enhancing the model understanding of interconnected water bodies. Sourced from Environment and Climate Change Canada Environment and Canada (2023) spanning 21 years (2000-2021), the spatial distribution of the monitoring stations and water level fluctuations is visualized in Figures 5.2a and Figure 5.2 demonstrates temporal variability of Terrebonne station for the year 2016. To address missing values, the Historical Average is employed, supplemented by Gaussian smoothing, to remove anomalies. This filtering technique mitigates sudden data fluctuations without compromising underlying patterns. By enhancing data reliability, the model is better equipped to derive meaningful insights from the water level data.

For a fair comparison, we use the same training, validation, and test sets for all the methods and allocate 70% of the data for training, 10% for model validation, and the remaining 20% for final testing. To verify the effectiveness of the prediction model, we ran the experiments on six baseline and state-of-the-art models explained below. Table 5.1 shows the experiment results for different examined periods over three evaluation metrics, including mean absolute error (MAE), mean absolute percentage error (MAPE) and root mean square error (RMSE). The smaller these errors, the higher the prediction accuracy, and vice versa.



(a) (b)

Figure 5.2: (a) Location of the stations (b)Water levels for 2016

We compare a number of baseline and state-of-the-art methods for spatiotemporal forecasting. Baseline methods, such as the Historical Average (HA) method, provide simplistic yet valuable benchmarks by predicting future values based on historical averages, serving as a foundational reference point for evaluating more complex techniques. Meanwhile, state-of-the-art methods represent the forefront of spatiotemporal forecasting research, incorporating advanced algorithms and architectures to capture intricate spatial and temporal dependencies within the data. These methods include the Informer model H. Zhou et al. (2021), which is designed for time series prediction, integrating a novel Transformer-based architecture with a multi-level feature fusion mechanism to efficiently capture long-term dependencies and spatial-temporal patterns, thereby improving forecasting accuracy in diverse time series datasets, GTS method Shang et al. (2021) which is highlighted for its scalability, making it well-suited for large datasets, especially in traffic forecasting, utilizing the GCRN block structure and a graph learning approach, Dynamic Causal Graph Convolutional Network (DCGCN) J. Lin et al. (2023) which uses graph convolutional networks based on generated stepwise dynamic causal graphs to make predictions and Spatial-Temporal Attention Wavenet (STAWnet) method C. Tian and Chan (2021) which uses temporal convolution to handle long-time sequences and self-attention networks to capture dynamic spatial dependencies between different nodes.

In analyzing the performance comparison based on the table of results, several trends emerge

Table 5.1: Experimental results of baseline methods conducted on water level data for different future time horizons e.g. 3, 6, 9 days. The smaller the errors, the higher the prediction accuracy.

		Horizon		
Method	Metric	3 Days	6 Days	9 Days
HA	MAE	0.28	0.28	0.28
	MAPE	9.56	9.56	9.56
	RMSE	0.42	0.42	0.42
Informer	MAE	0.35	0.35	0.35
	MAPE	0.10	0.10	0.10
	RMSE	0.53	0.53	0.53
GTS	MAE	0.34	0.33	0.33
	MAPE	0.12	0.13	0.14
	RMSE	0.50	0.49	0.49
DCGCN	MAE	0.16	0.25	0.51
	MAPE	0.06	0.11	0.83
	RMSE	0.22	0.32	1.81
STAWnet	MAE	0.08	0.12	0.15
	MAPE	0.03	0.04	0.04
	RMSE	0.12	0.18	0.23
TransGlow	MAE	**0.05**	**0.09**	**0.13**
	MAPE	**0.01**	**0.02**	**0.03**
	RMSE	**0.09**	**0.16**	**0.21**

across different spatiotemporal

forecasting models. The HA method does not take into account any trends, seasonality, or other patterns in the data, making it less accurate compared to the rest of the methods. Informer and GTS exhibit similar performance levels, while DCGCN surpasses both methods in shorter-term predictions. However, its performance diminishes when forecasting over longer periods, indicating the advantages of correlation-based spatial dependencies over causality-based ones in this context. STAWnet consistently demonstrates strong performance across all evaluated metrics, showcasing the effectiveness of its attention mechanism, particularly for longer forecasting horizons. Notably, TransGlow, which combines graph convolution with an RNN-based encoder-decoder and augmented attention layer, achieves the most promising results among the analyzed models. This underscores its capability to effectively capture complex spatiotemporal patterns and surpass other models in this comparative assessment.

5.7 Simulation and Visualization

We have developed an immersive, interactive application utilizing Unreal Engine to simulate and visualize fluvial floods (river floods) and pluvial or flash floods on a digital representation of the desired geographical region.

We selected Unreal EngineEpic Games (n.d.) as our simulation and visualization platform due to its robust physics engine, extensive support for asset creation, and comprehensive fluid simulation libraries. Additionally, it has the ability to execute complex simulations on consumer-grade hardware. The optimization capabilities of Unreal Engine, such as the conversion of dynamic elements into static meshes to minimize draw calls, significantly improve rendering performance. This feature is particularly beneficial for rendering large-scale, detailed urban environments. Moreover, Unreal Engine's flexibility facilitates the seamless integration of new data, ensuring that our digital twins accurately mirror the changing urban landscapes and support swift development cycles.

Another point that merits consideration is the purpose which is to be served by the reconstructed digital twin of an urban landscape. Only certain features of an urban city digital twin pertain to accurate flood simulation and visualization and the decision-making strategies for mitigating the adverse effects, namely the buildings, roads, and urban tree canopy. We distill these features and focus our efforts on reconstructing a representative digital twin of the city, which is qualified in scope for the purpose of flood simulation. We implement a procedural approach to reconstructing buildings, roads, and trees in the urban landscape, leveraging the procedural tools provided in Unreal Engine by utilizing curated geospatial data.

The application facilitates various forms of interaction based on the specific viewing perspective, aerial and ground. The rise in the level of the water bodies can be specified to simulate flooding to varying extents. Our application incorporates functionality to add or remove obstructions flexibly at custom scales and orientations, enabling users to manipulate the water flow and providing valuable insight for flood mitigation strategies. Furthermore, we introduce the capability to simulate inland flooding by enabling users to specify water sources within the urban terrain. Our simulation and visualization technique offers a comprehensive platform for understanding and addressing flood scenarios, encompassing both natural and human-made factors for effective decision-making in

flood risk management.

5.7.1 Digital Twin Modeling

In subsequent sections, we provide a detailed account of the methodology employed in the development of digital twins, beginning with the establishment of a digital twin for Terrebonne. This initial digital twin lays the groundwork for subsequent analysis, simulations, and decision-making processes. Our development process commences with the collection and integration of geospatial data, encompassing a diverse array of spatial information such as topography, environmental elements, and infrastructure. Section "Data-source Pre-visualizations" in the Supplementary Material depicts renders of the data used. Our primary objective is to formulate a method that is efficient and adaptable to different geographical contexts with minimal adjustments, thereby affirming the generalizability, scalability, and broader application of our methods. Furthermore, to ensure the robustness and generalizability of our approach, we give precedence to utilizing data from open repositories. These sources are publicly available and typically conform to established standards and guidelines, facilitating broader accessibility and compatibility.

After data preprocessing in QGIS, we import it into Unreal Engine through a landscaping plugin tool. This transition marked a pivotal phase in our methodology, allowing for the integration of geospatial data into a dynamic, interactive 3D modeling environment. The use of Unreal Engine enables the realistic and detailed reconstruction of urban landscapes, significantly enhancing our ability to simulate and analyze the potential impacts of changing water levels within these environments. Figure 5.7f shows renders of the various stages during the reconstruction of the digital twin in Unreal Engine.

Modeling the terrain

For the specific task of acquiring high-quality elevation data, we utilize a High-Resolution Digital Elevation Surface Model (HRDEM), which provides detailed topographical information. The HRDEM data utilized in this work is obtained from the Open Data Portal of Canada Canada (2019). The HRDEM dataset is a comprehensive source of terrain data that encompasses a variety of models, including both DSM (Digital Surface Model) and DTM (Digital Terrain Model) among other

terrain-related data types. The DSM is distinct in its approach to elevation data, as it captures elevations that account for both natural and artificial features on the earth's surface, such as buildings, vegetation, and infrastructure. Conversely, the DTM focuses exclusively on the ground surface, offering elevation data that considers only the natural undulations and contours of the terrain, absent of any buildings, vegetation, or man-made structures. This delineation between the DSM and DTM models within the HRDEM dataset is critical, as we require a terrain model that isolates the natural ground surface to compute the levels of flooding accurately. The HRDEM uses the Universal Transverse Mercator (UTM) coordinate system based on the North American Datum of 1983 (NAD83), specifically its Canadian Spatial Reference System (CSRS) variant. Each 1-meter resolution dataset encompasses a $10 \times 10km^2$ area, making it highly focused and detailed for localized studies. To ensure high-precision geo-referencing, all datasets are projected into the EPSG:2959 coordinate system. EPSG:2959 is a specific implementation of the UTM system optimized for the region, providing an additional layer of accuracy. By standardizing the coordinate system, it simplifies data integration and sharing among different GIS applications. Figure 5.6g shows a render of the Digital Terrain Model (DTM) from the HRDEM dataset for our test area.

Following the preprocessing of the Digital Terrain Model (DTM) to isolate the target area, we integrated the processed data into Unreal Engine to construct a landscape, utilizing the elevation data as a heightmap. To achieve a realistic terrain representation, we employed a sophisticated material strategy that leverages geospatial land use data for the informed distribution of various terrain materials. This approach enables the accurate depiction of different land covers: grass in open spaces and parks, mud in areas designated for buildings, and a vegetation template for tree-rich zones, including urban canopies and forests. Figure 5.7a illustrates the visual accuracy achieved by aligning land use data with the precise application of materials based on terrain characteristics derived from elevation data, establishing a foundation for subsequent fluid simulation tasks.

Modeling urban infrastructure

Following the completion of the surface terrain modeling, we focused on the modeling of urban infrastructure elements, namely buildings and roads.

Buildings. For the integration of building-specific information, we obtained multi-polygonal geospatial vector data from the municipal urban planning authority. This data, characterized by high spatial accuracy, delineated the geographical extents of all man-made structures within the jurisdiction of the city. Figure 5.6b shows a render of this multi-polygon vector data of building footprints in QGIS. Utilizing this vector data, static 3D representations of buildings were incorporated into the digital twin environment. To generate the buildings within the digital twin of the city, the data was imported into Unreal Engine via a landscaping plugin and a standard procedural modeling algorithm was employed by extending a wall mesh along the spline of the building footprint to construct the exterior facets of these buildings. This resulted in topologically closed, watertight 3D models that are representative of their actual physical counterparts in terms of water collision that will be tested for flood simulation. Given our primary focus on flood simulation scenarios, the vertical elevation or rooftop height of individual buildings was standardized, as it did not influence the hydrodynamic computations. This procedural generation algorithm ensured that the buildings adhered to collision boundaries essential for water simulation, allowing for a meaningful output that accurately reflected the bounds of the real structures.

After the reconstruction of the 3D models of the buildings themselves, a notable challenge we encountered involved the accurate placement of these structures on uneven terrain. This issue was critical, as buildings needed to be positioned in a manner that ensured water would correctly interact with their walls during simulations. Specifically, it was imperative to avoid scenarios where water could seep through gaps resulting from the terrain's irregularities beneath the buildings. To address this, we programmatically adjusted the buildings' placement to ensure that their walls extended sufficiently into the ground, thereby eliminating any potential spaces through which water could erroneously pass. This alignment was essential for achieving realistic simulation results, particularly in flood scenarios. Figure 5.8 shows the renders before and after aligning the building structures on the landscape, in perspective and orthographic views. The line traces shown in the images show the guiding points used to place the buildings. The line traces are red to indicate no intersection with the landscape underneath, and they are green after computing the points of intersection. It should be noted that the figure shows a sub-sampled region of the map for the purpose of illustrating the method of aligning procedurally generated buildings with the map. Figure 5.7b shows the landscape

with building structures after their correct placement from a high viewpoint.

Roads. For road infrastructure, multi-line string geospatial vector data was also acquired from the city's database. Preliminary steps involved the conversion of these multi-line strings to geospline entities. These were imported into Unreal Engine using the landscaping plugin as georeferenced splines. Subsequent adjustments were made to the splines' elevational attributes to ensure they were co-planar with the pre-modeled terrain. A procedural algorithm was then applied to transform these geosplines into three-dimensional road models with proper texture and material attributes, by procedurally applying a road section mesh constructed to the city proportions along these splines to reconstruct a representative digital model of the roads. Note that appropriate manual adjustments were made in cases of roads placed for bridges. Figure 5.6c shows a render of the vector geospatial data in QGIS and 5.7c shows the landscape with roads in the digital twin model.

Modeling environmental elements

We dedicated the final step to the modelling of environmental components, specifically the tree canopy and bodies of water.

Trees. The data utilized for canopy mapping was obtained from the Donnees Quebec database Québec (2023). In this data, the canopy is demarcated by the ground-level projection of arboreal crowns, encompassing leaves, branches, and trunks, and is observable via aerial imagery. All vegetation exceeding a height of 2 meters was included in the data. The canopy mapping was executed through advanced deep learning techniques, leveraging variables calculated at a 1-meter spatial resolution from raw airborne LiDAR data gathered over the period 2010-2020. Figure 5.6f shows a render of the vector data representing the tree canopy for our test area. Since the data for tree canopy is represented using multi-polygon geometry and we do not have the locations of the tree trunks themselves, a procedural foliage spawning method was used to create an digital modeling of the urban vegetation. The trees were placed at randomly sampled points to ensure they cover the regions of canopy retrieved from the canopy dataset with appropriate density. We ensured that in modeling the trees, the resistance offered by the trees to the flood water was accurate rather than achieving faithful physical placement of trees with respect to the their locations in the real world. Figure 5.7d shows a render of the placed foliage in the digital twin reconstructed in Unreal Engine.

Modeling water bodies. This flood simulation model uses geospatial data sourced from Geofabrik Geofabrik GmbH (n.d.). The dataset comprises two main types of vector geospatial information: water bounds represented by polygons and waterways represented by strings. The waterways form the basis for generating waterbodies in the simulation. The model integrates historical or predicted data to dynamically simulate water volume by specifying the height of water bodies. The simulation aims to accurately represent the behavior of water systems over time, providing a valuable tool for applications such as water resource management and flood prediction. Figure 5.6e shows a render of the two types of vector data representing the water for our test area.

To model the water bodies accurately in our project, we utilized waterways geospatial vector data represented as multiline strings. In our heightfield fluid simulation (Supplementary Material, Section "Heightfield Fluid Simulation"), we introduce a collection of water sources in the shape of customizable elliptical cylinders to represent a volume of water within each water body (river/stream). This cylinder's dimensions are adjustable along its major and minor axes, allowing for the precise modeling of water volumes, and the height is computed from the specified water level, thus closely mirroring the actual water volume present in the geospatially defined water bodies.

To ensure that the water bodies have the correct volume of water, we have implemented the following series of steps, which produce a close approximation of the actual water volume presented in the water bodies, given the water level.

(1) We first augmented the geospatial multiline string vector data for waterways by adding any unrepresented streams, identified visually by dashed lines in Figure 5.6d showing a QGIS render.

(2) We then imported the geospatial multiline string vectors into Unreal Engine with the landscaping plugin so that they are georeferenced in the same coordinate system as the landscape (and the other features as well), the corresponding objects in Unreal Engine are splines. Figure 5.10a shows the splines representing water bodies loaded up in Unreal Engine.

(3) Each point in a spline has location coordinates and a vector representing the normal to the direction along the spline at the point. For each spline point in the water body (rivers/streams)

splines, we computed the intersection of two line traces projected along the normals to the spline at the point in clock-wise and anticlockwise directions at a configurable height above the river bed, which ensures that our model approximates the physical dimensions of the actual water in the river/stream. Figure 5.10b shows the line traces in red, and the points of intersections of these traces with the landscape along the contours of the river, marked in red as well.

(4) For every point in the spline, we used the length of the spline segment from the current point to the next point halved and the distance between the line trace intersections to compute the scale of the elliptical surface of the water and placed it at the midpoint of the line trace intersections, orienting it correctly using the direction along the spline at the point. The height of each water cylinder is computed so that the actual water surface is aligned with the specified water level. Figures 5.10c and 5.10d show the spawning of the cylindrical water sources and the eventual surface of the water body itself.

For inland water flooding, we have implemented the ability for the user to specify cylindrical sources of water, with configurable height of the water cylinder to simulate flash floods due to rains.

Comparative Visualization

Figure 5.11 presents a comparative analysis of our Unreal Engine reconstructed digital twin of an urban environment against the representation provided by Google Earth. While the Google Earth digital twin does present a more aesthetically pleasing representation due to the use of satellite images for textures, as seen from figures 5.11a and 5.11b, it does reveal a limitation in utility of Google Earth's approach, notably its reliance on low-polygon mesh approximations for urban features. This simplification compromises the accuracy necessary for flood simulation scenarios, as exemplified by Figures 5.11d and 5.11f, which highlight Google Earth's inadequacy in measuring the depth of water bodies with the necessary precision for hydrological analysis.

In contrast, Figures 5.11c and 5.11e illustrate the enhanced detail and dynamic water body simulation of our digital twin model within Unreal Engine. This model includes accurate terrain modeling and comprehensive geospatial elements like roads, buildings, and canopies that influence

water flow, showcasing our model's superior fidelity and precision. The detailed urban features and accurately simulated water dynamics of our digital twin are essential for conducting dependable flood scenario simulations. This precision enables more accurate impact assessments and supports the formulation of effective mitigation strategies.

5.7.2 Viewing and Interaction

Viewing perspectives. There are two distinct viewing perspectives for observation: the aerial perspective, commonly referred to as the birds-eye view, and the terrestrial perspective, commonly referred to as the ground-view. Utilizing an aerial perspective, the camera is positioned at a significant altitude, affording a comprehensive vantage point capable of surveying up to an area of $1.25km^2$. The camera elevation can be flexibly controlled within the altitude limits of 35 meters up to 500 meters (with reference to the sea level). From this perspective, given the aim of comprehending the impact of the flood on a broader region, the digital twin encompasses all intricate geospatial entities such as vegetation, buildings, roads, and so forth, as described in the previous section.

In contrast, the ground-view perspective employs a third-person camera that tracks a user-controlled character positioned at the point of interest on the ground. From this perspective, the primary aim is to examine the impact of water flow at a lower geographical scale. This view allows the user to gauge the extent of flooding in comparison to the scale of urban landmarks, such as the roads and buildings in the vicinity. To provide real-world context, both viewing modes feature an onscreen overlay displaying the user location in WGS84 reference system coordinates and altitude in meters, thereby enabling users to accurately reference their position in the real world. The aerial and ground views are depicted in Figures 5.12a and 5.12b, respectively. Note that in Figure 5.12a, a green marker shows the location of the first person character that will be possessed on switching to ground view.

Interaction modes. Users can toggle between this navigation mode and an interaction mode, which allows them to navigate freely in either aerial or ground views or to add obstructions or water sources and specify the rise of water levels in various bodies. In this interaction mode, users have the ability to engage with the terrain by strategically positioning concrete water barriers in order to redirect the movement of water. Users also have the ability to add sources of water inland to simulate flash



(a) (b)

Figure 5.3: Interaction mode: Adding obstructions and inland sources of water

flood conditions. In Figure 5.3, 5.3a shows how barriers and inland sources of water can be set up, and 5.3b shows the simulation.

Overview map. We have implemented an orthographic projection of the map without the tree canopy, which allows real-time tracking of the water level as the simulation occurs. Figure 5.12c shows the overview map, which resembles a floodplain map.

5.7.3 Simulation experiments

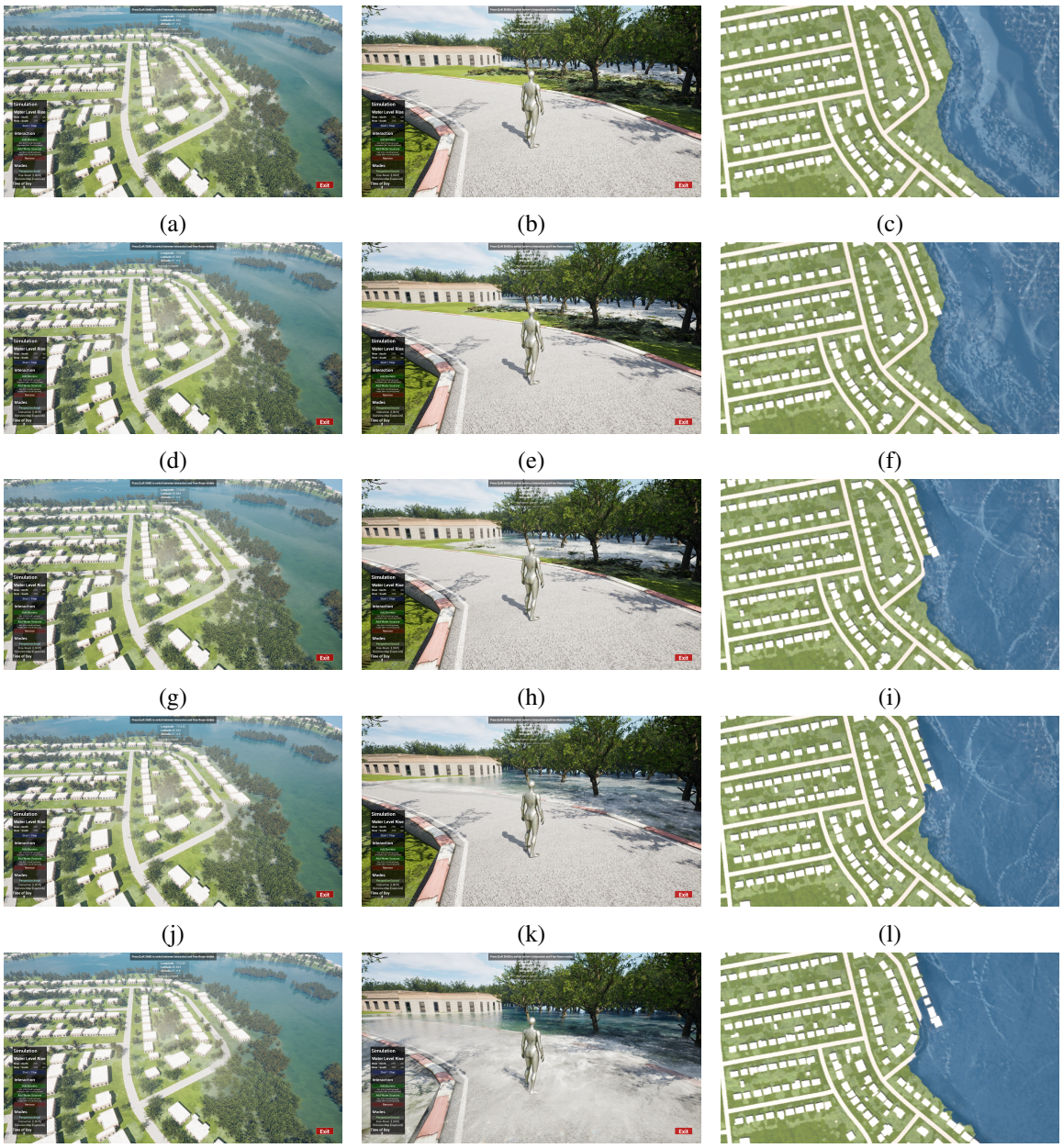
This section outlines the experiments conducted using our simulation and visualization application, showcasing its qualitative performance by thoroughly exploring a range of scenarios in a detailed, comprehensive manner.

Simulation of the rise in water level

In our simulation, we modeled a scenario where the water level was raised by 200cm from the riverbed's baseline. Figure 5.4 illustrates the evolution of the water simulation across three different perspectives (aerial, ground, and overview) captured at successive moments in time.

Simulations with varying rises in water levels

In this experiment, we systematically investigated the impact of different rises in water levels on our simulation's performance and outcomes. We initiated the simulation with a water level set at 100 cm above the riverbed's baseline and incrementally increased this level by 25 cm for



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

(m) (n) (o)

Figure 5.4: Simulation of water level = 200 cm over time

each subsequent experiment, continuing up to a maximum of 300 cm. This chosen range, from 100 cm to 300 cm, was designed to span a spectrum of scenarios: the lower threshold of 100 cm closely approximates conditions of regular river flow, where water bodies do not flood (refer to Figure 5.13a), while the upper limit of 300 cm is representative of extreme flooding situations (refer to Figure 5.13i). Through this approach, we aimed to comprehensively test the simulation's

behavior under varying degrees of water elevation, from normal to critical flooding conditions. The overview map was utilized to illustrate the floodplain regions for each scenario, providing a visual representation of the extent and impact of flooding across the different water level simulations. (refer to Figure 5.13)

Simulation with obstructions

We ran the simulation for a water level of 200 cm (above the base level in the river bed), and tested two scenarios, viewed from both the perspectives and monitored in the overview map. In the first scenario, we ran the simulation without creating any barriers obstructing the flooding water flow, resulting in puddling of water over the streets and close to the houses, as exhibited in figures 5.14a, 5.14c and 5.14e. In the second scenario, barriers were added to the map by carefully considering the expected flow of the water as observed from the previous scenario. Figures 5.14b, 5.14d and 5.14f show the migitated circumstances, thereby demonstrating the utility of this platform in providing insight into the mitigation strategy design for floods.

Simulation with real-world data

Historical & Predicted Data. There are two primary data sources for weather analysis: historical data and predictive data. Historical data refers to data that has been recorded over a period of time using sensors installed at the weather station. On the other hand, the predictive data is derived from utilizing the model outlined in Section "Methodology" to conduct weather forecasts for a specified time frame, such as the subsequent day, the following three days, and so forth. Based on the aforementioned factors, the application utilizes simulations to replicate the projected water levels, allowing users to analyze potential flood events and make informed decisions.

In this simulation experiment, using the historical dataset, we retrieved records and accessed the water level data for well-documented spring flooding events in 2017. Our model predicted the anticipated water levels by choosing the same time period. Figure 5.5a presents a demarcation map of the floodplain that occurred during the spring floods of 2017 and 2019 (provided by the government of Quebec*ArcGIS Web Application — cehq.gouv.qc.ca* (n.d.)). Figure 5.5b illustrates the simulated water levels based on observed data, while Figure 5.5c displays the simulated water

levels using predicted data. Both figures represent the same time period and geographical area, providing a comparative analysis between the observed and predicted water level scenarios.

Our flood simulation and visualization platform performs by considering terrain variations and incorporating all modeled urban features. This comprehensive approach ensures that simulations of water inundation generated from both historical 5.5b and predictive 5.5c data offer precise depictions of floodplain extents on par with the demarcated floodplain 5.5a. Moreover, the platform's capability to simulate scenarios with obstructions, in conjunction with predicted water level rises, provides a robust framework for testing and implementing effective flood mitigation strategies.



(a) QC Govt - Spring Floods 2017/2019

(b) Flood simulation based on historical data

(c) Flood simulation based on our prediction

Figure 5.5: Simulation with real-world data for spring floods of 2017

5.8 Conclusion

In conclusion, our research has demonstrated the substantial potential of advanced simulation tools in enhancing urban flood management and policy development. Through the innovative integration of digital twins, dynamic obstruction modeling and predictive analytics, we have unlocked new avenues for detailed and proactive disaster response planning. The capability to introduce obstructions dynamically within the simulation plays a pivotal role in the context of policy formulation, enabling users to explore various scenarios effectively. Given the constrained timeframe usually associated with flood warnings, the integration of a graph neural network facilitates precise forecasts up to nine days ahead. By considering available resources, including workforce, physical materials, and man hours, this simulation tool significantly enhances decision-making processes at a detailed

level. The added feature of navigating through the simulation environment offers a tangible sense of anticipated flood water levels and the potential impact of obstructions. This immersive experience aids in prioritizing and, if necessary, triaging responses to flood events, thereby optimizing resource allocation and response strategies.

5.9 Data availability

The datasets generated and analyzed during the current study are available from the corresponding author on reasonable request.

5.10 Acknowledgement

C.P. This work is financially supported by the Mathematics of Information Technology and Complex Systems' Accelerate programme under grant agreement IT29301 and the Natural Sciences and Engineering Research Council of Canada Grants RGPIN-2021-03479 (NSERC DG).

U.E. This research was undertaken, in part, thanks to funding from the Canada Excellence Research Chairs Program.

Z.P. This research was undertaken, in part, based on support from the Gina Cody School of Engineering of Concordia University FRS.

This work would not have been possible without the invaluable insights and support provided by Rémi Asselin, Directeur, Direction des technologies de l'information, Ville de Terrebonne; Philippe Hamel, Chef de section, sécurité organisationnelle et réseautique, Ville de Terrebonne; Anh Phuong Tran, Architecte de solutions, Ville de Terrebonne; and Sacha Leprêtre, CTO at Presagis Inc.

5.11 Additional Results

In the following, we include the pertinent background theory involved in the Graph Neural Network and renders of each step of the creation of the digital twin.

Graph Neural Networks

Graph Notation

A graph is a foundational data structure for representing intricate relationships between objects. It is denoted as $\mathcal{G}(\mathcal{V}, \mathcal{E})$ and consists of nodes and edges. Nodes are objects or entities represented as $\mathcal{V}$, they can be defined as $\mathcal{V} = \{v_1, v_2, \ldots, v_n\}$, where n is an integer representing number of nodes. Edges denoted as $\mathcal{E}$ establish connections between nodes. They can be expressed as $\mathcal{E} = \{e_{i,j} \,|\, i, j \in \{1, 2, .., n\}\}$, where e_{ij} indicates an interaction between nodes i and j. This interaction signifies the presence of a relationship between the connected nodes.

To represent a graph, one common approach is through an adjacency matrix, denoted as $\mathbf{A}_{n\times n}$. The adjacency matrix is a square matrix, where the entry $\mathbf{A}_{i,j}$ contains information about the existence of an edge connecting nodes i and j. A non-zero element at $\mathbf{A}_{i,j}$ signifies the presence of an edge, effectively indicating a relationship between the respective nodes. Conversely, a zero-element indicates no direct connection. Furthermore, each node within the graph can be associated with a set of features or attributes. These attributes can be encapsulated in a feature matrix $\mathbf{X} \in \mathbb{R}^{n\times d}$. Here, each row in the matrix corresponds to a specific node, and the columns represent its d features. This feature matrix provides a means to enrich the representation of nodes in the graph.

The graph structure enables the modelling of intricate relationships between objects. It is a robust framework for representing complex systems, networks, and data, making it a fundamental concept in various domains, including network analysis, social sciences, and data science. In this problem, we employ nodes to represent the hydrological stations. The graph edges represent the relationship between stations, which can be their spatial distribution at different geographic locations within a water system or the water flow from upstream to downstream stations.

Graph Learning Module

Using a graph learning module to understand the connections between hydrological stations is beneficial since the connections between hydrological stations in a complex water system are not always straightforward. Different factors, such as topography, land use, rainfall patterns, and human

activities, influence station relationships. Furthermore, the connections between hydrological stations can evolve over time due to various factors. For instance, seasonal variations, changes in land use, infrastructure development, and climate change can alter the relationships between stations. Several studies L. Bai et al. (2020); Jiang et al. (2023); Z. Wu, Pan, Long, et al. (2020) employ a graph learning module so the system can adaptively identify and represent these connections, even when they are not explicitly defined. It can uncover hidden connections, adapt to changes over time, and provide a dynamic representation of the network of hydrological stations.

The objective is to model the relationships between nodes, often represented as a function of the node embeddings, based on the widely recognized adaptive graph generation approach as outlined in L. Bai et al. (2020) using the following equation:

$$\hat{A} = softmax(relu(E_1.E_2^T))$$

Here, E_1 and E_2 represent randomly initialized node embeddings, and it is important to note that these embeddings are subject to learning and refinement during the training process. This equation computes an adaptive graph representation, denoted as A, which captures the dynamic relationships between nodes in the context of the specific problem being addressed.

Foundational Component: Graph Convolutional Recurrent Neural Network

Graph Convolutional Neural networks (GCN) are one of the multiple variants of GNNs used to extract and process information from graphs representing relationships between entities. In the context of spatiotemporal forecasting, this method becomes valuable as it allows the model to capture spatial dependencies inherent in the data. GCN involves applying convolutional operations to graph structures. Unlike traditional convolutions in grid-like data (e.g., images), GCN operates directly on the graph topology, utilizing node features and their connections to aggregate information. During convolution, information aggregation occurs by aggregating neighboring node features, considering their connectivity within the graph. This step allows the model to capture spatial dependencies by weighting the influence of neighboring nodes in the forecast. Our adapted model employs GCN operation based on normalized Laplacian using Chebyshev polynomial approximationKipf and Welling (2016), providing a computationally efficient solution:

$$H^{(k)} = \sigma(\hat{A}H^{(k-1)}W^{(k)}) + b^k \tag{24}$$

$$\hat{A} = A + I,$$

where A is the adjacency matrix of graph, W^k and b^k are the trainable weight and bias matrices for layer k, and σ is the activation function. H^{k-1} and H^k are the input signal and output of the graph convolution operation on layer k, respectively.

After establishing the foundation of the spatiotemporal forecasting model with the incorporation of graph convolution to capture spatial dependencies, the selected model broadens its architectural scope by incorporating a widely adopted approach in the literature L. Bai et al. (2020); Jiang et al. (2023); Y. Li et al. (2017), integrating temporal dynamics through the inclusion of gated recurrent units (GRUs). This approach enables the model to seamlessly navigate both spatial and temporal dimensions. The GRUs serve as a pivotal addition, enhancing the model's ability to capture intricate temporal patterns and dependencies within the data. The introduced graph convolution operation replaces the Multi-Layer Perceptron (MLP) layers of the Gated Recurrent Unit (GRU):

$$r^t = \sigma(\mathcal{G}_{conv}(A, [X^t, h^{t-1}]) + b_r$$

$$u^t = \sigma(\mathcal{G}_{conv}(A, [X^t, h^{t-1}]) + b_u$$

$$c^t = tanh(\mathcal{G}_{conv}(A, [X^t, r^t * h^{t-1}]) + b_c$$

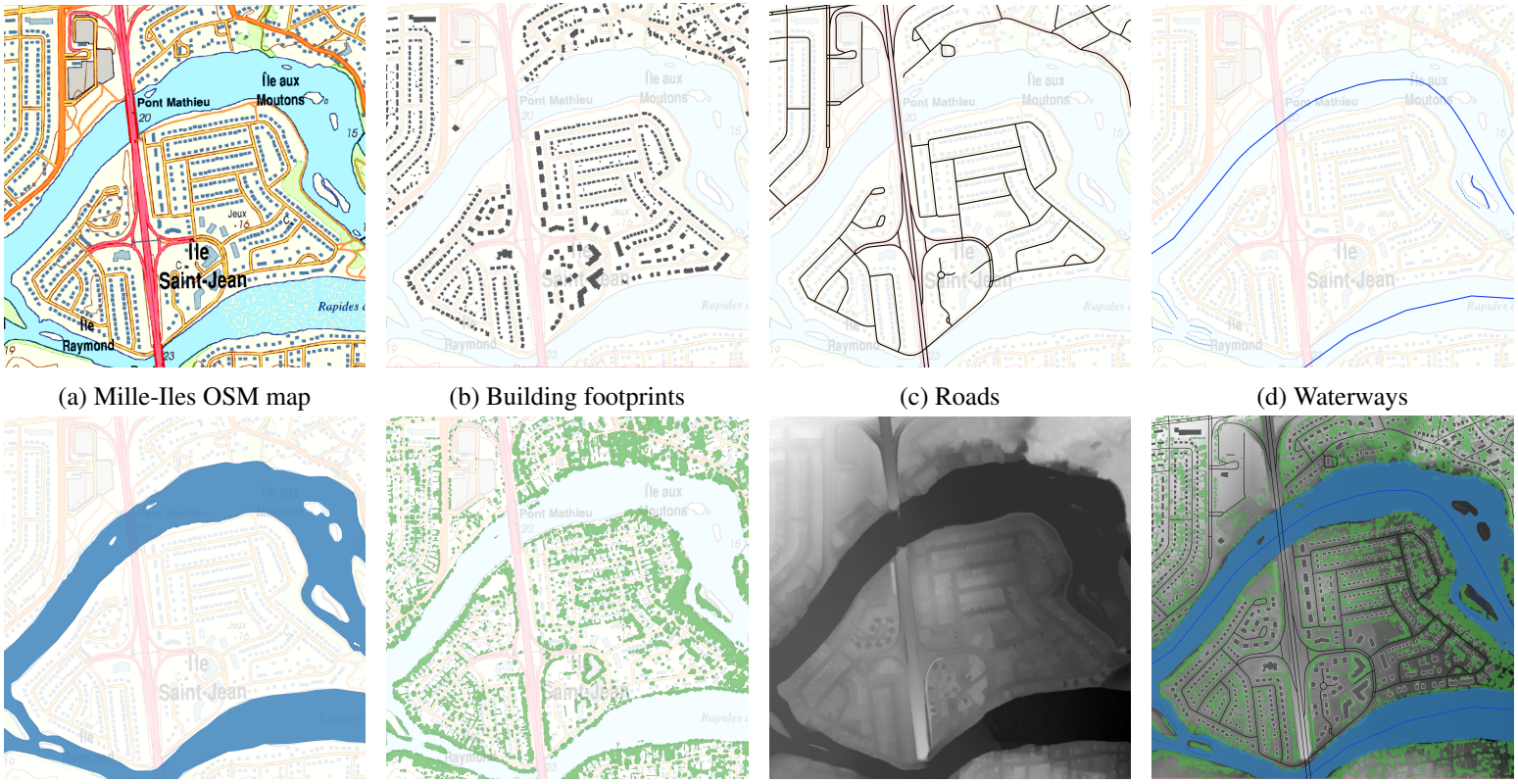
$$h^t = (u^t * h^{t-1})(1.0 - u^t) * c$$

where $\mathcal{G}_{conv}$ is the graph convolution operation from Equation 24, r^t and u^t are the reset gate and the update gate, c^t is the cell state, X^t is the input signal, $\sigma(.)$ and $tanh(.)$ are activation functions, and h is the hidden state.

Data-source Pre-visualizations

In our work, we utilized QGIS, an open-source Geographic Information System (GIS), for rendering and preliminary processing of geospatial data. This software enables detailed examination,

editing, and composition of spatial information, ensuring data readiness for subsequent project stages. Figure 5.6 shows renders of the geospatial data captured in QGIS.

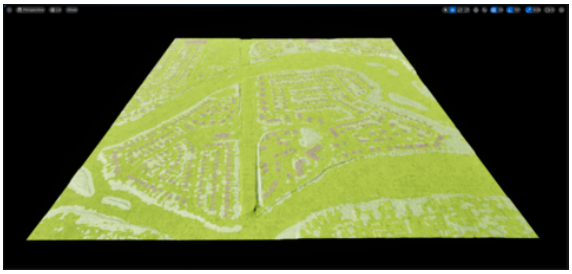


(a) Mille-Iles OSM map (b) Building footprints (c) Roads (d) Waterways

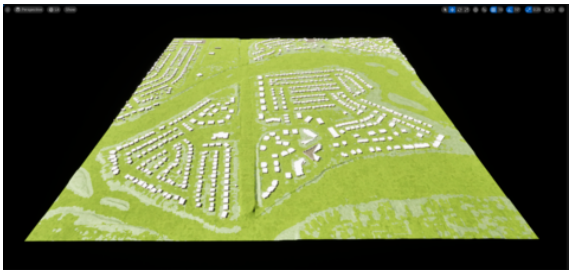
(e) Water bodies (f) Tree canopy (g) Digital Terrain Model (h) Geospatial features

Figure 5.6: Geospatial data sourced from open repositories and visualized in QGIS

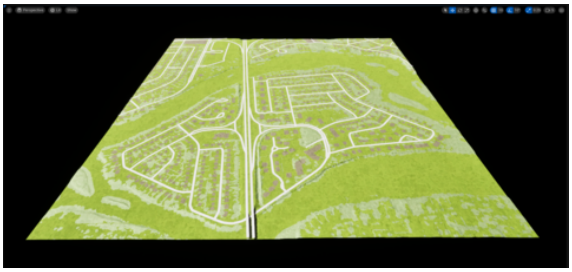
Digital Twin



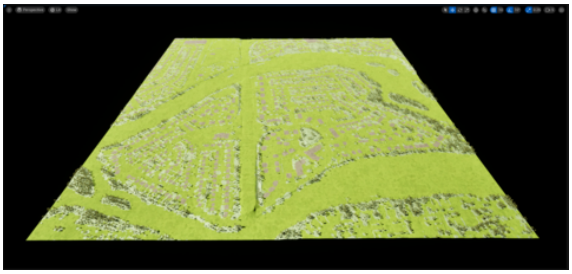
(a) Landscape generated DTM



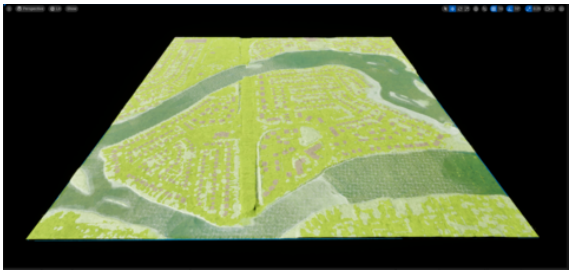
(b) Procedurally generated buildings



(c) Procedurally generated roads



(d) Procedural Foliage



(e) Water simulation

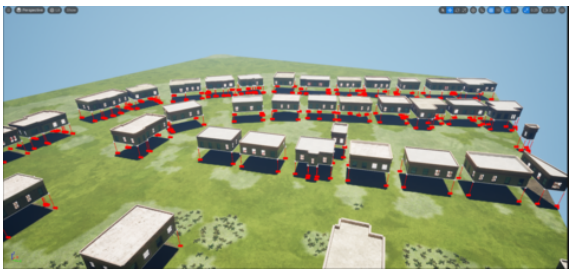


(f) Digital twin

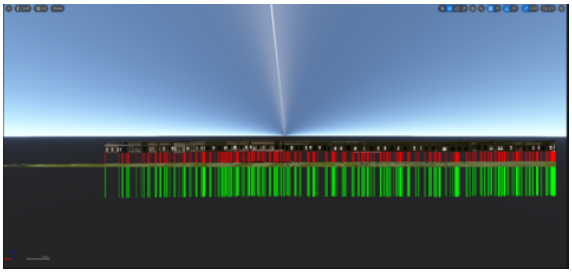
Figure 5.7: Digital twin modeled in Unreal Engine using geospatial data

Heightfield Fluid Simulation

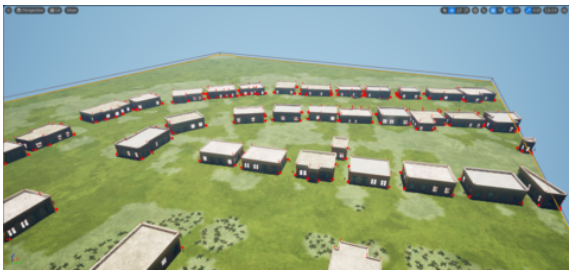
The heightfield fluid simulation technique we utilize employs a two-dimensional grid system, wherein a heightmap is generated through an overhead camera capture. This process records the height of each point within the grid by considering the objects present in each grid cell. In our implementation, the resolution of the heightfield mesh is set at 4096×4096 cells, with each cell measuring 30.5cm across. Consequently, the total area covered by the grid is approximately $1.25km^2$ as shown inFigure 5.9).



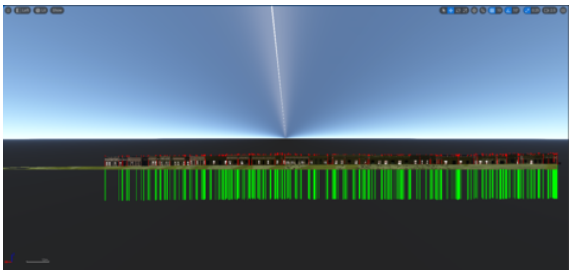
(a) Before (Perspective view)



(b) Before (Orthographic view)



(c) After (Perspective view)



(d) After (Orthographic view)

Figure 5.8: Aligning the building meshes generated procedurally from the geospatial building footprints with the landscape

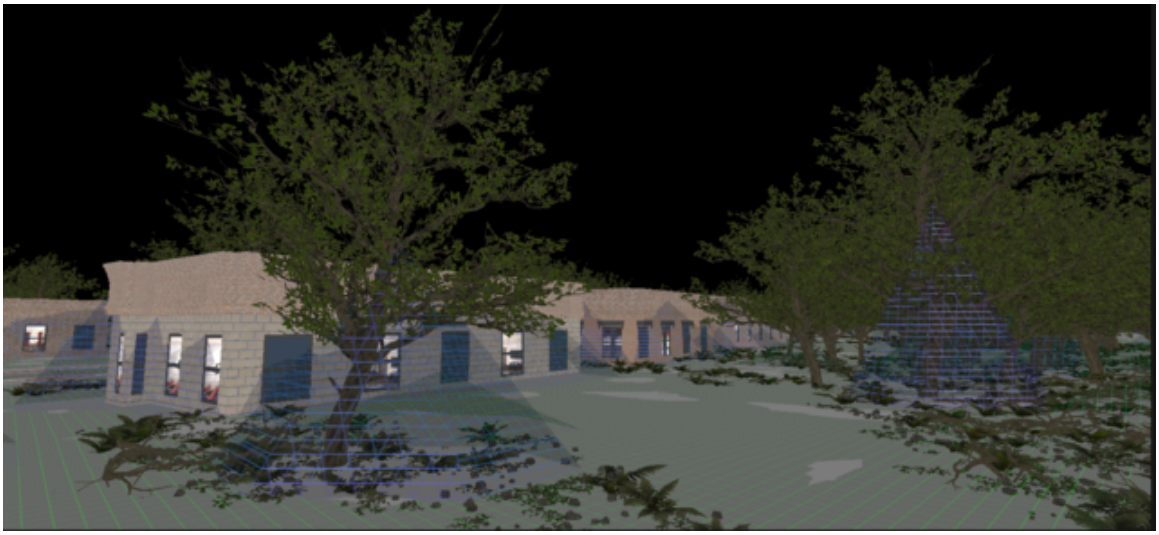


Figure 5.9: Heightfield fluid simulation: Grid resolution of 4096×4096 cells, Cell Size = $0.305m$, Area = $1.2498km^2$. The translucent mesh depicted within the simulation delineates the collision boundaries that are assessed for fluid interaction.



(a) Splines representing the geospatial waterways



(b) Determining water body extents using line traces



(c) Cylindrical water sources covering the span of the water body



(d) Simulated water level of the water body

Figure 5.10: Modeling rivers in the simulation by procedurally adding water sources along the spline points of the geospatial waterways



(a) Île Saint-Jean: Overhead view (ours)



(b) Île Saint-Jean: Overhead view (Google Earth)



(c) Autoroute 25S (ours)



(d) Autoroute 25S (Google Earth)



(e) Parc de Vérone (ours)



(f) Parc de Vérone (Google Earth)

Figure 5.11: Comparative visualization of the digital twins: Proposed work (ours) and Google Earth



(a) Aerial view perspective

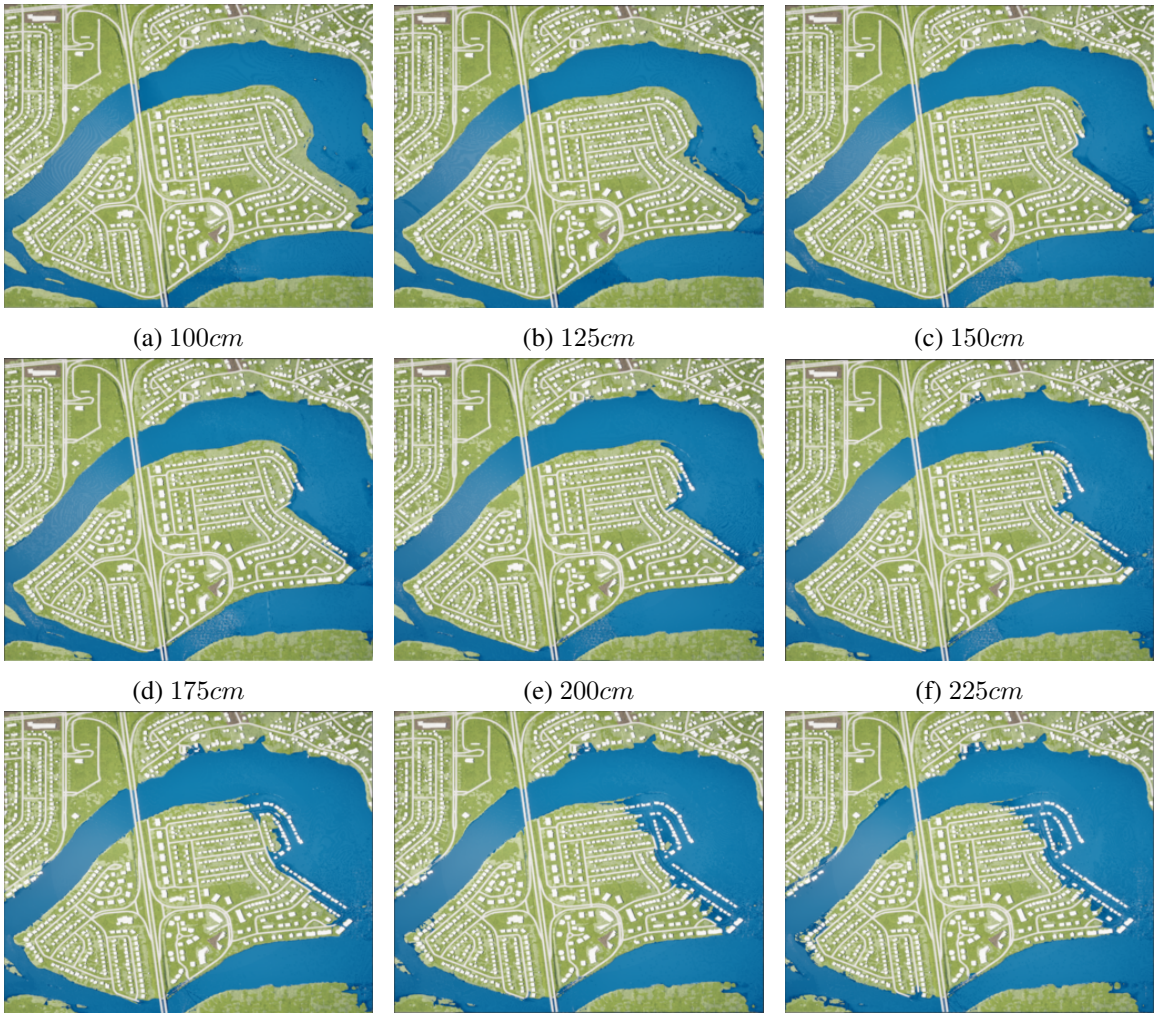


(b) Ground view perspective



(c) Overview map

Figure 5.12: Viewing perspectives and overview map



(a) 100cm (b) 125cm (c) 150cm

(d) 175cm (e) 200cm (f) 225cm

(g) 250cm (h) 275cm (i) 300cm

Figure 5.13: Simulations at varying initial water levels



(a) No obstructions (aerial view)



(b) Added obstructions (aerial view)



(c) No obstructions (ground view)



(d) Added obstructions (ground view)



(e) No obstructions (overview)



(f) Added obstructions (overview)

Figure 5.14: Simulation of water level = 200 cm with and without obstructions

Chapter 6

Enhancing Hydrometric Prediction with LiDAR Data

This chapter is a copy of the manuscript titled *HydroVision: LiDAR-Guided Hydrometric Prediction with Vision Transformers and Hybrid Graph Learning*, accepted for publication at the 19th International Symposium on Visual Computing (ISVC 2024). The manuscript was co-authored by Naghmeh Shafiee Roudbari, Ursula Eicker, Charalambos Poullis, and Zachary Patterson.

6.1 Abstract

Hydrometric forecasting is crucial for managing water resources, flood prediction, and environmental protection. Water stations are interconnected, and this connectivity influences the measurements at other stations. However, the dynamic and implicit nature of water flow paths makes it challenging to extract a priori knowledge of the connectivity structure. We hypothesize that terrain elevation significantly affects flow and connectivity. To incorporate this, we use LiDAR terrain elevation data encoded through a Vision Transformer (ViT). The ViT, which has demonstrated excellent performance in image classification by directly applying transformers to sequences of image patches, efficiently captures spatial features of terrain elevation. To account for both spatial and temporal features, we employ GRU blocks enhanced with graph convolution, a method widely used in the literature. We propose a hybrid graph learning structure that combines static and dynamic

graph learning. A static graph, derived from transformer-encoded LiDAR data, captures terrain elevation relationships, while a dynamic graph adapts to temporal changes, improving the overall graph representation. We apply graph convolution in two layers through these static and dynamic graphs. Our method makes daily predictions up to 12 days ahead. Empirical results from multiple water stations in Quebec demonstrate that our method significantly reduces prediction error by an average of 10% across all days, with greater improvements for longer forecasting horizons.

6.2 Introduction

Hydrometric forecasting is a critical component of water resource management, with significant implications for public safety, economic stability, and environmental conservation. Among the various aspects of hydrometric forecasting, water level prediction stands out due to its direct and widespread impact. Accurate forecasts provide early warnings for flood prevention, ensure the integrity of dams and bridges, and optimize water distribution for agricultural, industrial, and domestic use. Additionally, water level forecasting is essential for maintaining aquatic ecosystems and biodiversity.

Water systems exhibit temporal variability influenced by seasonal patterns, weather events, and long-term climate trends. These temporal dynamics shape the flow rates, water levels, and overall hydrological behavior over different time scales. Simultaneously, understanding spatial interactions is crucial for predicting how changes in one location propagate throughout the hydrological system. Therefore, forecasting water flow in hydrological systems inherently poses a spatiotemporal forecasting challenge, as it necessitates capturing both the temporal variability and spatial interconnections within the hydrological network to provide accurate predictions and insights into water system dynamics.

The spatial correlation of water systems is influenced by terrain elevation changes, which determine how water flows through a landscape. This parameter is crucial as it affects the speed and direction of runoff, with steeper slopes leading to faster runoff and potentially higher water levels in lower areas. It also affects the accumulation and distribution of water across different regions, contributing to the overall dynamics of the water system.

Most papers in this area focus on historical water level data and primarily consider only the temporal correlations. Various statistical and machine learning models, such as autoregressive integrated moving average (ARIMA) Bazrafshan, Salajegheh, Bazrafshan, Mahdavi, and Fatehi Maraj (2015), support vector machine (SVM) Asefa, Kemblowski, McKee, and Khalil (2006), and artificial neural network (ANN) Aichouri et al. (2015), have been widely used for this purpose. However, these methods often fall short in capturing the spatial interactions and complex dependencies within hydrological systems. Some recent work T. Bai and Tahmasebi (2023); Roudbari et al. (2023) have adopted graph neural network (GNN) based approaches to address these limitations. GNN methods excel in capturing the spatial relationships among multiple water stations, providing a more comprehensive understanding of hydrological dynamics. Despite their advantages, GNN methods still face challenges in accurately modeling the effects of terrain elevation on water flow patterns. In this study, we address the influence of terrain elevation on water level and connectivity in hydrometric forecasting. To incorporate this essential factor, we utilize LiDAR terrain elevation data encoded through a Vision Transformer (ViT). The use of Vision Transformer (ViT) stems from transformers' success in natural language processing tasks Kalyan, Rajasekharan, and Sangeetha (2021) and, more recently, in computer vision by directly applying transformers to image patches Dosovitskiy et al. (2020). This approach allows us to understand how variations in terrain elevation affect water flow patterns across different regions, providing a robust foundation for our forecasting model.

To model both temporal dependencies and spatial relationships among water stations, we employ Gated Recurrent Unit (GRU) blocks enhanced with graph convolution as successfully used in the recent literatureCui, Ke, Pu, Ma, and Wang (2020); Roudbari et al. (2023); J. Zhang et al. (2018). GRU blocks are well-suited for capturing sequential dependencies in time-series data, such as water flow measurements Gharehbaghi, Ghasemlounia, Ahmadi, and Albaji (2022). By integrating graph convolution, which models spatial dependencies through graph structures where nodes represent water stations and edges denote relationships, we extend our model's capability to capture complex interactions in hydrological systems. Furthermore, we propose a novel hybrid graph learning structure that combines static and dynamic graph learning. Static graphs, derived from transformer-encoded LiDAR data, capture terrain elevation relationships that remain consistent over time. In

contrast, dynamic graphs adapt to temporal changes in water flow patterns and connectivity between stations, thereby improving the overall graph representation and adaptability of the model. In this chapter, we present the following contributions:

- Incorporating Terrain Elevation for Water Level Forecasting: We integrate LiDAR-derived terrain elevation data into our hydrometric forecasting model, acknowledging its critical impact on water flow and connectivity.
- Proposing a Hybrid Graph Learning Structure: We introduce a novel hybrid graph learning structure that combines static and dynamic graph learning. Static graphs, derived from transformer-encoded LiDAR data, capture terrain elevation relationships, while dynamic graphs adapt to temporal changes, enhancing the overall graph representation.
- Demonstrating Superior Performance in Experiments: Through experiments conducted on water stations in Quebec, using the data from Environment and Natural Resources of Canada Environment and Canada (2024), our method outperforms state-of-the-art methods across all prediction horizons and performance metrics.

6.3 Related Work

Building a model that considers all the influencing parameters on the water cycle is complex due to the intricate nature of hydrological systems. Hydrometric parameters forecasting has evolved significantly over the years. Initially, statistical models such as ARIMA were the primary tools used for time series forecasting Irvine and Eberhardt (1992); Montanari, Rosso, and Taqqu (1997); Papamichail and Georgiou (2001). ARIMA models are relatively easy to implement and interpret, making them suitable for short-term forecasting. However, ARIMA models have notable limitations, particularly in handling non-linear relationships and complex temporal patterns, which are often present in hydrometric data. Machine learning models like Support Vector Machines (SVM) Sapankevych and Sankar (2009), Artificial Neural Networks (ANN) A. Jain and Kumar (2007), and Radial Basis Function (RBF) networks Moradkhani, Hsu, Gupta, and Sorooshian (2004) emerged as

alternatives to overcome the limitations of statistical models. SVMs have been used for their robustness in handling non-linear relationships. ANNs, particularly feedforward neural networks, have been widely applied due to their ability to approximate any continuous function, offering greater flexibility than ARIMA models. However, ANN models often require large amounts of training data and are prone to overfitting. RBF networks, a variant of ANNs, provide better generalization capabilities but still face challenges in capturing long-term dependencies in time series data. The advent of Recurrent Neural Networks (RNNs) J. Zhang and Man (1998) marked a significant advancement in time series forecasting, particularly for sequential data. RNNs and their variants like Long Short-Term Memory (LSTM) networks Hochreiter and Schmidhuber (1997) are specifically designed to capture long-term dependencies and temporal correlations in data, addressing the limitations of traditional ANN models. However, RNNs and LSTMs struggle with spatial dependencies, prompting the integration of Convolutional Neural Networks (CNNs) and Graph Neural Networks (GNNs). CNNs, known for their effectiveness in capturing spatial relationships, have been applied to hydrometric forecasting by modeling spatial dependencies among multiple stations Atashi, Kardan, Gorji, and Lim (2023). GNNs further enhance this capability by operating on graph-structured data, capturing complex spatial relationships and interactions within the hydrological system C. Chen et al. (2021).

To address the limitations of both spatial and temporal modeling, recent work Roudbari, Patterson, Eicker, and Poullis (2022); Shang et al. (2021); Z. Wu, Pan, Long, et al. (2020) have proposed hybrid models such as Graph Convolutional Recurrent Networks (GCRNs). GCRNs combine the strengths of GRU blocks for sequential data processing with graph convolution operations to capture spatial dependencies, providing a comprehensive approach to spatiotemporal forecasting. These hybrid models, inspired by the success of both RNNs and GNNs in their respective domains, offer a more robust framework for hydrometric forecasting. However, even these advanced models have shortcomings, particularly in incorporating complex domain information. Therefore, we propose integrating terrain elevation through Vision Transformers (ViTs) to create static graph structures to enhance prediction.

Transformers, initially introduced for machine translation tasks Vaswani et al. (2017), have revolutionized natural language processing (NLP) tasks. These models have been pretrained on vast

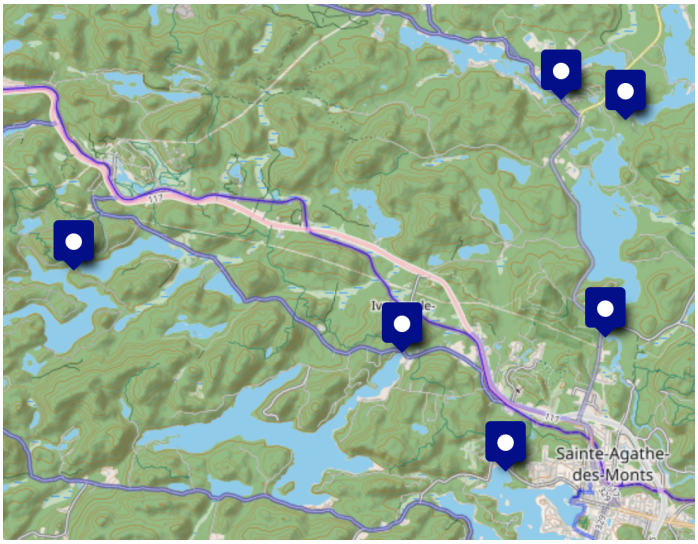
amounts of text and fine-tuned for various applications. Notable examples include BERT Devlin et al. (2018) and the GPT series Floridi and Chiriatti (2020), which uses language modeling for pretraining. Drawing inspiration from the success of Transformers in NLP, Vision Transformers (ViT) have emerged as a powerful tool in computer vision. The original ViT Dosovitskiy et al. (2020) divides images into patches, applies Multi-Head Attention (MHA) Vaswani et al. (2017) to these patches, and uses a learnable classification token to capture a global visual representation, enabling effective image classification.

6.4 Dataset

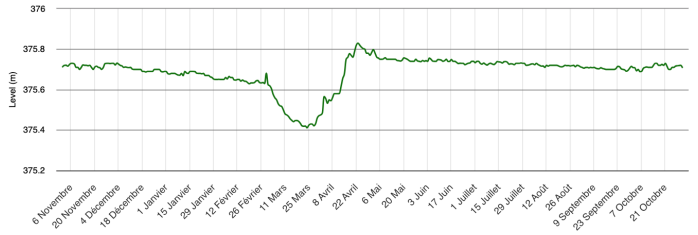
In this work, we utilize two types of data: LiDAR data and time series water level data, as outlined below. These datasets provide complementary information that enhances our water level forecasting model.

6.4.1 LiDAR Data

The LiDAR data, provided by the Ministère des Ressources naturelles et des Forêts (MRNF) Ministry of Natural Resources and Forests (2016) as part of the provincial LiDAR sensor data acquisition project, includes the Digital Terrain Model (DTM). This DTM is a raster file with a spatial resolution of 1 meter, providing precise numerical values representing altitudes in meters relative to mean sea level. Elevation values are derived through linear interpolation across an irregular triangle network created from ground points. The DTM images are produced by superimposing the Digital Elevation Model (DEM) with the shaded DEM to accentuate relief, using color gradients and transparency. With its high spatial resolution, it is also extensively used in creating hydrological models, planning road construction, managing flood risks, and conducting visual landscape analyses. Figure 6.1c visualizes the DTM of the study area.



(a)



(b)



(c)

Figure 6.1: (a) The geographic distribution of water level monitoring stations around the Sainte-Agathe-des-Monts. (b) Variation in water levels throughout the year at Lake Papineau in Sainte-Agathe-des-Monts. (c) Visualization of the Digital Terrain Model (DTM) near Sainte-Agathe-des-Monts.

6.4.2 Timeseries data

The time series data consists of daily water level measurements from six stations on bodies of water in a specific region in Quebec, spanning 40 years from 1981 to 2021. Provided by Environment and Climate Change Canada Environment and Canada (2024), this dataset is crucial for understanding water level variations over time. Missing values in the data are replaced by a weighted average of the previous and next year's data. Figures 6.1a and 6.1b illustrate the station coverage on the map and the variation of water levels at one station over a year, respectively.

To ensure that computational resources are efficiently utilized and data processing remains manageable, we have selected the closest geographically clustered monitoring stations from all available stations scattered across a wide region. This selection is due to the necessity of loading the LiDAR data that covers the entire study area.

6.5 Objective

Given the LiDAR data, our goal is to uncover the underlying spatial relationships. These spatial relationships will then be used as inputs for another function, combined with 2D time series data from n stations over m timestamps, to predict future values.

Let L represent the LiDAR data, and T represent the time series data, where $T \in \mathbb{R}^{n \times m}$.We aim to find a function f that captures the spatial relationships from L:

$$f : L \to S$$

where S represents the spatial relationships. Next, we define another function g that takes S and T as inputs to predict future values $\hat{T}$:

$$g : (S, T) \to \hat{T}$$

The overall objective is to find the optimal functions f and g such that the predicted future values $\hat{T}$ closely match the actual future values.

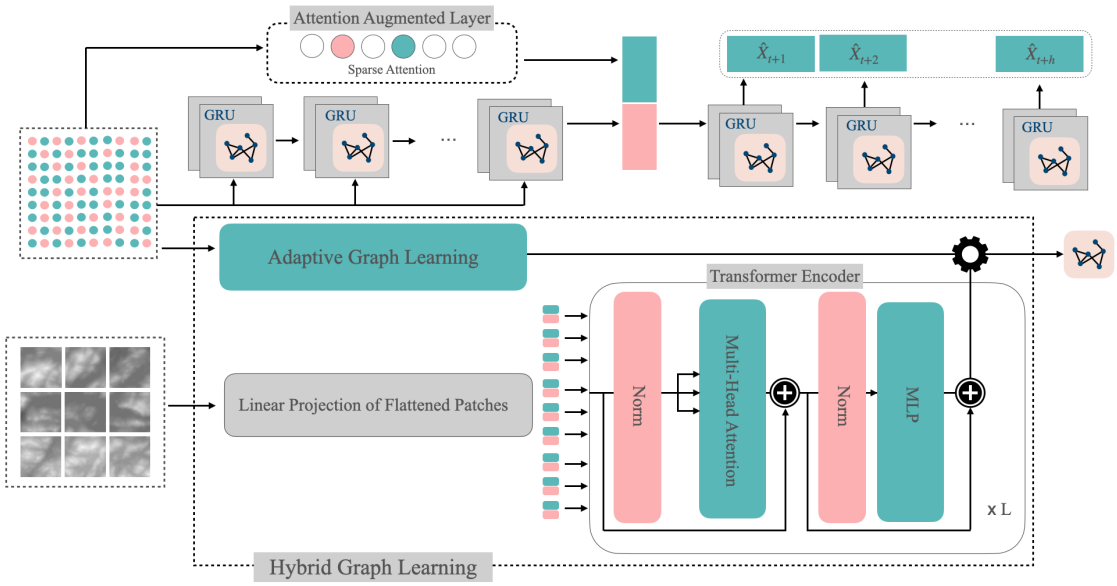


Figure 6.2: Main Architecture of the HydroVision Framework. The architecture integrates LiDAR data via a Vision Transformer and combines adaptive graph learning with GCRNs to capture both spatial and temporal dependencies for accurate water level forecasting.

6.6 Methodology

In this section, we introduce the HydroVision framework for water level forecasting, which integrates two essential components. First, we discuss the foundational approach, GCRN blocks. Second, we describe the hybrid graph learning layer incorporated in our study. Together, these elements constitute the HydroVision framework, as illustrated in Figure 6.2.

6.6.1 Foundational Approach

Graph Convolutional Recurrent Network

GCRN combines graph convolution operations and Gated Recurrent Units (GRUs) to tackle spatiotemporal forecasting challenges. The graph convolution captures spatial dependencies, while the GRU models temporal variability. Due to the success of previous research Y. Li et al. (2017); Shang et al. (2021) using this method for forecasting purposes, we adopt the GCRN formulation expressed as:

$$Z^{(l)} = \sigma(\tilde{A}Z^{(l-1)}W^{(l)} + b^l) \tag{25}$$

In this equation, $\tilde{A}$ is the learned adjacency matrix, W^l and b^l are the weight and bias matrices, and σ is the activation function. The GRU's traditional MLP layers are replaced by this graph convolution, resulting in the following equations for the reset gate (r^t), update gate (u^t), candidate cell state (c^t), and hidden state (h^t):

$$r^t = \sigma(\mathcal{F}(\tilde{A}, [X^t, h^{t-1}]) + c_r) \tag{26}$$

$$u^t = \sigma(\mathcal{F}(\tilde{A}, [X^t, h^{t-1}]) + c_u) \tag{27}$$

$$c^t = \tanh(\mathcal{F}(\tilde{A}, [X^t, r^t \odot h^{t-1}]) + c_c) \tag{28}$$

$$h^t = (u^t \odot h^{t-1}) + (1 - u^t) \odot c^t \tag{29}$$

Here, $\mathcal{F}$ represents the graph convolution operation, X^t is the input at time t, and $\odot$ denotes element-wise multiplication. The activation functions σ and $\tanh$ regulate the network's internal state transitions.

Encoder-Decoder with Augmented Attention

The encoder-decoder model is an effective approach for sequence-to-sequence tasks, widely used in fields like machine translation Cho et al. (2014) and time series forecasting Shang et al. (2021). The basic encoder-decoder model can struggle with information compression, especially for longer sequences. To address this, we use the architecture with an augmented attention layer. The attention mechanism computes a weighted sum of the input sequence elements, creating an augmented hidden state H:

$$H = \text{Concat}[h^t, C] \tag{30}$$

Here, h^t is the final hidden state of the encoder, and C is the context vector from the attention layer. This augmented state helps the decoder focus on relevant parts of the input data Roudbari et al. (2023). The original attention mechanism has a quadratic computational complexity with respect to the sequence length. To mitigate this, Zhou, Haoyi, et al. H. Zhou et al. (2021) proposed ProbSparse Self-attention, which selects a subset of k queries based on a probability distribution:

$$\hat{Q} = M(Q, K), \text{Attention}(Q, K, V) = \text{Softmax}\left(\frac{\hat{Q}K^T}{\sqrt{d}}\right)V \tag{31}$$

In these equations, Q, K, and V represent query, key, and value matrices, and d is the dimension. The probability distribution M determines the importance of each token in the sequence, including more relevant tokens in the sparse query matrix and excluding less relevant ones. This efficient attention mechanism improves the scalability of the encoder-decoder model.

6.6.2 Vision Transformer

In this study, we encode LiDAR elevation data using the Transformer model, as presented in the work by Dosovitskiy et al. Dosovitskiy et al. (2020). The Transformer model, originally designed for natural language processing, has been adapted to handle image data, proving highly effective in tasks requiring spatial understanding and feature extraction from images.

The core idea behind using Transformers for image recognition involves dividing the input image into a sequence of patches, which are then processed by the Transformer encoder. To encode the LiDAR elevation data, we first partition the data into non-overlapping patches of size 16×16. Each patch is then flattened into a vector and projected to a fixed-dimensional embedding.

The sequence of embedded patches, along with positional encodings $\mathbf{E}_{pos}$, is then fed into the Transformer encoder. The positional encodings are crucial for retaining spatial information, as the Transformer architecture does not inherently capture the order of the input sequence. The positional encoding can be defined as:

$$\mathbf{E}_{pos}(pos, 2i) = \sin\left(\frac{pos}{10000^{2i/d}}\right), \quad \mathbf{E}_{pos}(pos, 2i+1) = \cos\left(\frac{pos}{10000^{2i/d}}\right) \tag{32}$$

where pos is the position, i is the dimension, and d is the embedding size.

The embedded patches and positional encodings are combined and processed through the Transformer encoder layers, which consist of multi-head self-attention and feed-forward neural networks. The output from the Transformer encoder provides a rich, context-aware representation of the LiDAR elevation data, capturing both local and global spatial features.

The process can be summarized by the following equation for the Transformer encoder layer:

$$\mathbf{A_{elevation}} = \text{TransformerEncoder}(\mathbf{E} + \mathbf{E}_{pos}) \tag{33}$$

where $\mathbf{E}$ is the sequence of embedded patches, $\mathbf{E}_{pos}$ is the positional encoding, and $\mathbf{G1}$ is the output of the Transformer encoder. By leveraging the Transformer model for encoding LiDAR elevation data, we can effectively capture complex spatial relationships and provide a robust input representation for subsequent processing in our HydroVision framework.

6.6.3 Hybrid Graph Learning

To adaptively learn the spatial relationships between objects, we adopt the adaptive graph generation technique as defined in L. Bai et al. (2020):

$$A_{\text{adaptive}} = \text{softmax}(\text{ReLU}(E1 \cdot E2^T)) \tag{34}$$

In this equation, $E1$ and $E2$ represent node embeddings that are randomly initialized and subsequently learned during the training process. This method allows the model to dynamically adjust the spatial relationships between nodes based on the data. To enhance our model's performance, we integrate both the adaptively learned graph and the elevation encoded output into a combined graph representation. This can be expressed as:

$$\hat{A} = \alpha A_{\text{adaptive}} + (1 - \alpha) A_{\text{elevation}} \tag{35}$$

where α is a weighting parameter that balances the contribution of each graph. The combined graph $\hat{A}$ encapsulates the comprehensive underlying information used in our graph convolution operations.

6.7 Experiments

6.7.1 Settings

In our experiments, we allocate 70% of the dataset for training, 10% for validation, and the remaining 20% for testing. We use a batch size of 64. Both the length of the historical sequences and the prediction horizon are set to 12 time steps. The maximum number of training epochs is capped at 300, though early stopping is employed if the validation performance does not improve for 20 consecutive epochs.

Training is performed using the Adam optimizer with the Mean Absolute Error (MAE) as the loss function. To enhance generalization, curriculum learning is applied. The initial learning rate is set to 0.01, with a decay ratio of 0.1. The attention mechanism in the network utilizes 8 heads. The model is implemented using PyTorch version 1.7.1, and all experiments are conducted on an NVIDIA GeForce RTX 2080 Ti GPU with 11GB of memory.

6.7.2 Comparative Performance Evaluation

table 6.1 provides a comparative analysis of different models for water level forecasting. The performance metrics used are Mean Absolute Error (MAE) and Root Mean Square Error (RMSE).

AGCRN L. Bai et al. (2020), designed to capture both temporal and spatial dependencies within graph-structured data through RNNs and GNNs, demonstrates a pretty stable performance over different prediction horizons. This reflects its capability to handle both types of dependencies. The Informer H. Zhou et al. (2021) model is an enhanced version of the Transformer architecture, indicating some struggle with extended forecasts.

DCGCN J. Lin et al. (2023), employing the GCRN block structure and a graph learning approach, exhibits a significant increase in errors over time. This suggests that DCGCN faces considerable difficulty with longer-term forecasts. On the other hand, the STtransformer Network M. Xu et

Table 6.1: Comparative Performance of Various Models for Water Level Forecasting. The table displays the Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) for different forecasting models across various prediction horizons (3, 6, 9, and 12 days).

Method	Metric	Horizon			
		3 Days	6 Days	9 Days	12 Days
AGCRN	MAE	0.514	0.526	0.547	0.573
	RMSE	0.841	0.857	0.886	0.871
Informer	MAE	0.717	0.733	0.772	0.825
	RMSE	0.125	0.141	0.186	0.252
DCGCN	MAE	0.145	0.317	0.553	0.796
	RMSE	0.162	0.346	0.611	0.869
STtransformer	MAE	0.055	0.068	0.072	0.085
	RMSE	0.074	0.094	0.109	0.128
GTS	MAE	0.053	0.064	0.078	0.099
	RMSE	0.080	0.096	0.113	0.130
STAWnet	MAE	0.043	0.048	0.059	0.062
	RMSE	0.067	0.079	0.093	0.101
MTGNN	MAE	0.039	0.047	0.059	0.064
	RMSE	0.060	0.082	0.093	0.106
Hydrovision	MAE	**0.031**	**0.043**	**0.050**	**0.056**
	RMSE	**0.057**	**0.075**	**0.088**	**0.097**

al. (2020), which captures spatial and temporal dependencies using transformers, maintains lower errors compared to many other models, showcasing good performance across all prediction horizons.

The GTS method Shang et al. (2021), a scalable spatiotemporal forecasting approach, demonstrates strong performance with relatively low error increments over time, indicating its efficiency compared to AGCRN and Informer. Similarly, STAWnet C. Tian and Chan (2021), known for effectively capturing spatial and temporal information using advanced attention mechanisms, performs well, maintaining low errors across all horizons.

MTGNN Z. Wu, Pan, Long, et al. (2020), integrating a graph learning module with a mix-hop propagation layer and a dilated inception layer for optimal learning, shows very low errors, indicating robust performance across all prediction horizons. Finally, Hydrovision, the proposed model in this study, consistently outperforms the other models in both MAE and RMSE. Hydrovision's ability to maintain low error rates over all tested horizons underscores its superior effectiveness for water level forecasting.

It is important to note that given the domain of the dataset, which involves small variations within a meter, higher errors are particularly concerning. For instance, the errors exhibited by DCGCN, AGCRN, and Informer may not be considered good. Hydrovision, along with MTGNN and STAWnet, stands out for maintaining errors within an acceptable range, thereby proving more suitable for the precise nature of this domain.

6.7.3 Ablation Study

To evaluate the impact of incorporating the Vision Transformer (ViT) and LiDAR elevation data in our hybrid graph learning approach, we conducted an ablation study. We compared the performance of the model with the full hybrid graph learning approach, against a variant where the ViT and LiDAR elevation data were excluded from the main architecture. The results are summarized in Table 6.2.

Table 6.2: Ablation Study Results Comparing the Effectiveness of the Hybrid Graph Learning Approach

Metric	Hybrid Graph Leanring				Adaptive Graph Leanring			
	3 Days	6 Days	9 Days	12 Days	3 Days	6 Days	9 Days	12 Days
MAE	**0.031**	**0.043**	**0.050**	**0.056**	0.034	0.047	0.057	0.066
RMSE	**0.057**	**0.075**	**0.088**	**0.097**	0.061	0.079	0.096	0.108

These results clearly demonstrate that the Hybrid Graph Learning approach enhances forecasting accuracy more effectively than the Adaptive Graph Learning approach, particularly over longer prediction periods. The improvement in both MAE and RMSE indicates that integrating the hybrid approach results in a more robust and accurate forecasting model, making it a preferable choice for applications requiring extended time horizon predictions.

This analysis underscores the value of both the ViT and LiDAR data in enhancing the model's accuracy. The detailed terrain information provided by the LiDAR data, combined with the ViT's ability to capture complex spatial patterns, contributes to more precise water level forecasting, validating the effectiveness of our proposed approach.

6.8 Conclusion

In this chapter, we introduced a novel approach to water level forecasting by leveraging advanced graph learning techniques and LiDAR elevation data. Our approach integrates a Hybrid Graph Learning framework with a Vision Transformer (ViT) to enhance the accuracy of water level predictions across various time horizons.

Our study underscores the importance of high-resolution spatial data in enhancing the predictive performance of graph-based models. The LiDAR data provides detailed elevation information that enriches the spatial context of the predictions, leading to more accurate and reliable forecasts. This finding supports the value of integrating rich, domain-specific data into forecasting models to capture nuanced spatial dependencies.

Overall, the proposed Hybrid Graph Learning approach represents a significant advancement in the field of water level forecasting. It combines state-of-the-art graph convolution techniques with cutting-edge transformer models and spatially rich LiDAR data, setting a new benchmark for accuracy and reliability in this domain. Future work could explore further refinements to the model and assess its applicability to other environmental forecasting tasks, extending the benefits of advanced graph learning methods and high-resolution spatial data to a broader range of applications.

Chapter 7

Conclusion and Future Work

This thesis addresses several challenges inherent in modeling complex systems, including the dynamic nature of spatial and temporal dependencies, the integration of diverse data sources, and the need for efficient and scalable models that can support real-world applications.

Accurate forecasting in traffic and environmental systems is crucial for decision-making in urban planning, resource management, and disaster mitigation. Traditional models have struggled to handle the intricacies of these systems, especially when faced with dynamic graphs, multilevel dependencies, sparse data, and high computational demands. In response to these challenges, this thesis introduces methodologies that significantly enhance prediction accuracy, computational efficiency, and applicability across different domains. By incorporating innovative techniques such as multilevel encoder architectures, attention-augmented models, hybrid graph learning, and the integration of data like LiDAR-derived terrain information, this work pushes the boundaries of existing forecasting methods.

The proposed methods were rigorously evaluated across diverse datasets and real-world scenarios, including traffic networks, drainage basins, and flood simulations in specific urban environments. These experiments demonstrated the superiority of the developed models over current state-of-the-art methods and showcased their potential for practical implementation. As such, the findings contribute to both the theoretical understanding of spatiotemporal forecasting and the development of actionable solutions for complex urban and environmental problems.

In this chapter, we provide a summary of the main contributions made in this thesis. We also

acknowledge the limitations of the current research. Finally, we propose potential directions for future work, exploring ways to further improve forecasting accuracy, computational efficiency, and real-world applicability.

7.1 Key Findings

The primary contributions of this thesis are as follows:

1. Multilevel Encoder Architecture for Traffic Forecasting: This research introduces a novel multilevel encoder architecture that captures dependencies across multiple complexity levels in both sparse and dense road networks. This approach significantly enhances prediction accuracy while employing a sparse model to improve computational efficiency. The method was validated on benchmark datasets, including METR-LA and the newly introduced MSLTD, demonstrating superior performance over existing models.

2. Augmented Attention-Based Model for Water Flow Forecasting: A new transduction model with an efficient attention mechanism was developed to address water flow forecasting from a graph-based perspective. This is the first study of its kind to learn the actual correlations between drainage basins. Extensive experiments across 186 drainage basins in Canada showed that this model outperforms state-of-the-art methods across various prediction horizons and performance metrics.

3. Flood Simulation and Visualization Technique: This thesis presents an extendable flood simulation and visualization method by creating a detailed digital model of the city of Terrebonne, QC. Using historical data and predictions from a Graph Convolutional Recurrent Network (GCRN)-based model, the approach allows for interaction and scenario simulation, such as the controlled redirection of floodwaters. This tool supports stakeholders in evaluating mitigation strategies, providing actionable insights for community protection and infrastructure management.

4. Integration of LiDAR Data for Enhanced Hydrometric Forecasting: A novel hybrid graph learning structure was introduced, combining static and dynamic graph learning to capture terrain elevation relationships using LiDAR-derived data. This integration of terrain data into the forecasting model significantly improves the accuracy of water level predictions. The model's superior performance was validated through experiments on water station data from Quebec, outperforming

contemporary methods in all evaluation metrics.

Collectively, these contributions advance the field of spatiotemporal forecasting by addressing challenges related to dynamic graph learning, model efficiency, data integration, and actionable decision support in urban and environmental contexts.

7.2 Limitations

Despite the contributions presented in this thesis, several limitations remain that provide avenues for further research and model improvement.

One primary limitation of this work is related to data availability and quality. While the models presented in this thesis utilize a variety of datasets, including traffic data, water flow measurements, and LiDAR-derived terrain information, the results are still influenced by the quality, resolution, and completeness of the data. For example, the water flow forecasting model relies on historical data from specific drainage basins. Any gaps, inaccuracies, or noise within this data can affect the accuracy of the model's predictions. Therefore, the applicability of the proposed models is somewhat constrained by the quality of the input data, and the generalizability of the models to regions with different data characteristics may require further investigation.

Another limitation concerns the computational complexity of some of the proposed models. While efforts have been made to design more efficient architectures, such as the sparse multilevel encoder, certain components, especially those involving transformer and graph learning, still require substantial computational resources for training and inference. This can be a limiting factor for real-time applications and for deployment in resource-constrained environments. Future work could explore more lightweight models or techniques to optimize the training and inference processes without compromising the accuracy and robustness of the predictions.

The integration of dynamic and static data is a novel aspect of this research, yet it also introduces challenges. The hybrid graph learning structure that combines static terrain data with dynamic water flow variables represents a complex modeling task. While the approach has shown to improve prediction accuracy, it requires careful tuning and handling of various data types. Additionally, the current implementation assumes that static features, such as terrain elevation, remain

unchanged over time. However, environmental changes, such as urban development or natural landscape alterations, can affect these features, potentially limiting the model's long-term accuracy and adaptability. Future research should investigate methods for dynamically updating the static graph to reflect environmental changes over time.

Moreover, while the data-to-action simulation provides valuable insights into the potential impacts of forecasted events, the simulations are inherently based on the underlying model assumptions and data inputs. Therefore, they may not capture all real-world complexities, especially in scenarios where human intervention or unforeseen environmental factors play a significant role. For instance, the simulation of flood mitigation strategies, while useful, might not fully account for the interaction between different mitigation measures, infrastructure constraints, or changes in public behavior. As a result, these simulations should be interpreted as guidance tools rather than definitive predictions, emphasizing the need for further refinement and validation against real-world events.

In summary, this thesis presents several advancements in spatiotemporal forecasting, but its limitations highlight the complexity of modeling dynamic environmental systems. Addressing these challenges in future work could involve acquiring more diverse and high-quality datasets, optimizing model efficiency, dynamically updating static features, and enhancing the simulation process to incorporate a broader range of real-world variables and interactions.

7.3 Error Metrics Discussion

In this work, we have used Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and Root Mean Square Error (RMSE) as evaluation metrics, as they are widely used in the context of time series forecasting and regression. MAE measures the average absolute error, understanding deviations in the same unit as the target variable. MAPE expresses errors as a percentage, enabling scale-independent comparisons, while RMSE, by emphasizing larger errors due to squaring, is useful in scenarios where significant deviations need to be accounted.

While these metrics are well-established and effective for performance evaluation, they have certain limitations that are important to acknowledge. For instance, RMSE can be sensitive to

outliers, potentially exaggerating the impact of rare extreme values, while MAPE may become unstable when actual values are close to zero. Additionally, mean-based error metrics summarize performance as a single value, which may not fully capture the distribution of errors or highlight potential biases in predictions.

To complement these metrics, alternative approaches could be considered in future work. Methods such as the symmetric Mean Absolute Percentage Error (sMAPE) address some of the shortcomings of MAPE by adjusting for extreme values. Additionally, probabilistic measures like quantile-based loss functions can provide deeper insights into uncertainty and error distribution. While the chosen metrics remain a solid foundation for evaluation, incorporating such alternatives could further refine the assessment of model performance, particularly in applications where capturing uncertainty or specific error patterns is critical.

7.4 Future Work

Building on the advancements and limitations identified in this research, there are several directions for future work that can further enhance spatiotemporal forecasting, optimize computational efficiency, and explore the integration of new data sources and methodologies.

One promising avenue is to develop hybrid graph learning techniques that can dynamically update the model's statistical information in response to environmental changes over time. For example, in water flow forecasting, the current model assumes that static features, such as terrain elevation, remain fixed. Future research could explore methods to integrate temporal changes in these features, such as urban development or natural landscape alterations, into the graph representation. This would allow the model to adapt to evolving spatial configurations, improving long-term prediction accuracy. Additionally, incorporating more complex, real-time data, such as satellite imagery or drone-acquired terrain data, could further enhance the model's ability to reflect the current state critical directions potential improvement involves optimizing computational efficiency. While the models proposed in this thesis have been designed with efficiency in mind, there is still room to reduce computational complexity, particularly in the LiDARL-guided hybrid graph structures. Future work could explore the development of more lightweight neural network architectures. These

approaches can reduce the model's resource requirements, making it more suitable for real-time applications and deployment in resource-limited settings.

Data integration and enrichment is another critical direction for future work. Although this thesis has incorporated LiDAR data and hydrometric measurements, additional environmental and social data sources could be explored to provide a more holistic view of the forecasting system. Integrating real-time weather data, social media activity during disaster events, and urban infrastructure information could enhance the predictive capabilities of the models. Such data fusion could help refine the simulations in the data-to-action framework, leading to more accurate scenario analyses for urban and disaster management.

Furthermore, Using Large Language Models (LLMs) for Spatiotemporal Forecasting represents an innovative area of exploration. LLMs, such as GPT-4, have shown exceptional ability in learning patterns from vast amounts of data and generating complex, context-aware responses. Future research could investigate how LLMs can be adapted for spatiotemporal forecasting by training them to understand temporal sequences, spatial relationships, and domain-specific language descriptions of environmental changes. Additionally, LLMs could be used to analyze large volumes of unstructured text data, such as weather reports, emergency updates, or public sentiment, and compare their performance against traditional neural network models in forecasting tasks. Comparing LLM-based forecasting to the graph-based models presented in this thesis could yield valuable insights into their relative strengths, limitations, and potential areas of synergy.

Lastly, generalizability and transfer learning should be explored. The models developed in this thesis have been tested on specific datasets, such as those from Quebec's water stations and the city of Terrebonne. Future research could investigate the use of transfer learning techniques to adapt these models to new geographical regions and applications. By training on diverse datasets, the models could become more capable of generalizing their predictive power to a wide range of spatiotemporal forecasting scenarios.

These directions offer a pathway to more accurate, efficient, and actionable forecasting models that can better support urban planning, environmental management, and disaster mitigation efforts.

References

Agarap, A. F. M. (2018). A neural network architecture combining gated recurrent unit (gru) and support vector machine (svm) for intrusion detection in network traffic data. In *Proceedings of the 2018 10th international conference on machine learning and computing* (pp. 26–30).

Ahmed, A. N., Othman, F. B., Afan, H. A., Ibrahim, R. K., Fai, C. M., Hossain, M. S., ... Elshafie, A. (2019). Machine learning methods for better water quality prediction. *Journal of Hydrology*, *578*, 124084.

Aichouri, I., Hani, A., Bougherira, N., Djabri, L., Chaffai, H., & Lallahem, S. (2015). River flow model using artificial neural networks. *Energy Procedia*, *74*, 1007–1014.

ArcGIS Web Application — cehq.gouv.qc.ca. (n.d.). https://www.cehq.gouv.qc.ca/zones-inond/ZIS-20190715/index.html. ([Accessed 20-02-2024])

Asefa, T., Kemblowski, M., McKee, M., & Khalil, A. (2006). Multi-time scale stream flow predictions: The support vector machines approach. *Journal of hydrology*, *318*(1-4), 7–16.

Atashi, V., Kardan, R., Gorji, H. T., & Lim, Y. H. (2023). Comparative study of deep learning lstm and 1d-cnn models for real-time flood prediction in red river of the north, usa. In *2023 ieee international conference on electro information technology (eit)* (pp. 022–028).

Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Bai, L., Yao, L., Li, C., Wang, X., & Wang, C. (2020). Adaptive graph convolutional recurrent network for traffic forecasting. *arXiv preprint arXiv:2007.02842*.

Bai, T., & Tahmasebi, P. (2023). Graph neural network for groundwater level forecasting. *Journal of Hydrology*, *616*, 128792.

Bazrafshan, O., Salajegheh, A., Bazrafshan, J., Mahdavi, M., & Fatehi Maraj, A. (2015). Hydrological drought forecasting using arima models (case study: Karkheh basin). *Ecopersia*, *3*(3), 1099–1117.

B.c. floods caused at least $450m in damage. (n.d.). https://www.cbc.ca/news/canada/british-columbia/bc-flood-damage-1.6280393. (Accessed: 2023-10-30)

Breiman, L. (2001). Random forests. *Machine learning*, *45*, 5–32.

Bruna, J., Zaremba, W., Szlam, A., & LeCun, Y. (2013). Spectral networks and locally connected networks on graphs. *arXiv preprint arXiv:1312.6203*.

Butler, S., & Chung, F. (2006). Spectral graph theory. *Handbook of linear algebra*, 47.

Cai, P., Wang, Y., Lu, G., Chen, P., Ding, C., & Sun, J. (2016). A spatiotemporal correlative k-nearest neighbor model for short-term traffic multistep forecasting. *Transportation Research Part C: Emerging Technologies*, *62*, 21–34.

Canada, N. R. (2019). *High resolution digital elevation model (hrdem)–canelevation series–product specifications.* https://open.canada.ca/data/en/dataset/957782bf-847c-4644-a757-e383c0057995.

Chandra, S. R., & Al-Deek, H. (2009). Predictions of freeway traffic speeds and volumes using vector autoregressive models. *Journal of Intelligent Transportation Systems*, *13*(2), 53–72.

Chen, C., Luan, D., Zhao, S., Liao, Z., Zhou, Y., Jiang, J., & Pei, Q. (2021). Flood discharge prediction based on remote-sensed spatiotemporal features fusion and graph attention. *Remote Sensing*, *13*(24), 5023.

Chen, Y., Cheng, Q., Cheng, Y., Yang, H., & Yu, H. (2018). Applications of recurrent neural networks in environmental factor forecasting: a review. *Neural computation*, *30*(11), 2855–2881.

Chentanez, N., & Müller, M. (2010). Real-time simulation of large bodies of water with small scale details. In *Symposium on computer animation* (pp. 197–206).

Chentanez, N., & Müller, M. (2011). Real-time eulerian water simulation using a restricted tall cell grid. In *Acm siggraph 2011 papers.* New York, NY, USA: Association for Computing Machinery. Retrieved from https://doi.org/10.1145/1964921.1964977 doi: 10.1145/1964921.1964977

Child, R., Gray, S., Radford, A., & Sutskever, I. (2019). Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Choromanski, K., Likhosherstov, V., Dohan, D., Song, X., Gane, A., Sarlos, T., ... others (2020). Rethinking attention with performers. *arXiv preprint arXiv:2009.14794*.

Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

Cong, Y., Wang, J., & Li, X. (2016). Traffic flow forecasting by a least squares support vector machine with a fruit fly optimization algorithm. *Procedia Engineering*, *137*(1), 59–68.

Cui, Z., Henrickson, K., Ke, R., & Wang, Y. (2019). Traffic graph convolutional recurrent neural network: A deep learning framework for network-scale traffic learning and forecasting. *IEEE Transactions on Intelligent Transportation Systems*.

Cui, Z., Ke, R., Pu, Z., Ma, X., & Wang, Y. (2020). Learning traffic as a graph: A gated graph wavelet recurrent neural network for network-scale traffic prediction. *Transportation Research Part C: Emerging Technologies*, *115*, 102620.

Cui, Z., Ke, R., Pu, Z., & Wang, Y. (2018). Deep bidirectional and unidirectional lstm recurrent neural network for network-wide traffic speed prediction. *arXiv preprint arXiv:1801.02143*.

Datta, D., David, P. E., Mittal, D., & Jain, A. (2020). Neural machine translation using recurrent neural network. *International Journal of Engineering and Advanced Technology*, *9*(4), 1395–1400.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dey, R., & Salem, F. M. (2017). Gate-variants of gated recurrent unit (gru) neural networks. In *2017 ieee 60th international midwest symposium on circuits and systems (mwscas)* (pp. 1597–1600).

Ding, Q. Y., Wang, X. F., Zhang, X. Y., & Sun, Z. Q. (2011). Forecasting traffic volume with space-time arima model. In *Advanced materials research* (Vol. 156, pp. 979–983).

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., ... others (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Drucker, H., Burges, C. J., Kaufman, L., Smola, A., & Vapnik, V. (1996). Support vector regression machines. *Advances in neural information processing systems*, *9*.

Dtissibe, F. Y., Ari, A. A. A., Titouna, C., Thiare, O., & Gueroui, A. M. (2020). Flood forecasting based on an artificial neural network scheme. *Natural Hazards*, *104*, 1211–1237.

Duan, Y., Yisheng, L., & Wang, F.-Y. (2016). Travel time prediction with lstm neural network. In *2016 ieee 19th international conference on intelligent transportation systems (itsc)* (pp. 1053–1058).

Environment, & Canada, C. C. (2023). *Hydrometric statistics data.* Retrieved 2023-09-26, from https://wateroffice.ec.gc.ca/search/statistics_e.html

Environment, & Canada, C. C. (2024). *Hydrometric statistics data.* https://wateroffice.ec.gc.ca/search/statistics_e.html. (Accessed: 2024-02-20)

Epic Games. (n.d.). *Unreal engine.* Retrieved from https://www.unrealengine.com

Fadziso, T. (2020). Overcoming the vanishing gradient problem during learning recurrent neural nets (rnn). *Asian Journal of Applied Science and Engineering*, *9*(1), 197–208.

Farahmand, H., Xu, Y., & Mostafavi, A. (2023). A spatial–temporal graph deep learning model for urban flood nowcasting leveraging heterogeneous community features. *Scientific Reports*, *13*(1), 6768.

Floridi, L., & Chiriatti, M. (2020). Gpt-3: Its nature, scope, limits, and consequences. *Minds and Machines*, *30*, 681–694.

Gamba, P., & Houshmand, B. (2000, 08). Digital surface models and building extraction: A comparison of ifsar and lidar data. *Geoscience and Remote Sensing, IEEE Transactions on*, *38*, 1959 - 1968. doi: 10.1109/36.851777

Geng, X., He, X., Xu, L., & Yu, J. (2022). Graph correlated attention recurrent neural network for multivariate time series forecasting. *Information Sciences*, *606*, 126–142.

Geofabrik GmbH. (n.d.). *Openstreetmap data extracts for quebec.* https://download

.geofabrik.de/north-america/canada/quebec.html. (Accessed: [2023-11-01])

Gharehbaghi, A., Ghasemlounia, R., Ahmadi, F., & Albaji, M. (2022). Groundwater level prediction with meteorologically sensitive gated recurrent unit (gru) neural networks. *Journal of Hydrology*, *612*, 128262.

Gore, J. A., & Banning, J. (2017). Discharge measurements and streamflow analysis. In *Methods in stream ecology, volume 1* (pp. 49–70). Elsevier.

Graves, A., & Graves, A. (2012). Long short-term memory. *Supervised sequence labelling with recurrent neural networks*, 37–45.

Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., ... others (2018). Recent advances in convolutional neural networks. *Pattern recognition*, *77*, 354–377.

Guo, J., Huang, W., & Williams, B. M. (2014). Adaptive kalman filter approach for stochastic short-term traffic flow rate prediction and uncertainty quantification. *Transportation Research Part C: Emerging Technologies*, *43*, 50–64.

Guo, J., Liu, Y., Yang, Q., Wang, Y., & Fang, S. (2021). Gps-based citywide traffic congestion forecasting using cnn-rnn and c3d hybrid model. *Transportmetrica A: transport science*, *17*(2), 190–211.

Guo, S., Lin, Y., Li, S., Chen, Z., & Wan, H. (2019). Deep spatial–temporal 3d convolutional neural networks for traffic data forecasting. *IEEE Transactions on Intelligent Transportation Systems*, *20*(10), 3913–3926.

Guo, S., Lin, Y., Wan, H., Li, X., & Cong, G. (2021). Learning dynamics and heterogeneity of spatial-temporal graph data for traffic forecasting. *IEEE Transactions on Knowledge and Data Engineering*, *34*(11), 5415–5428.

Haghiabi, A. H., Nasrolahi, A. H., & Parsaie, A. (2018). Water quality prediction using machine learning methods. *Water Quality Research Journal*, *53*(1), 3–13.

Hamilton, W. L. (2020). Graph representation learning. *Synthesis Lectures on Artifical Intelligence and Machine Learning*, *14*(3), 1–159.

Hao, Z., AghaKouchak, A., Nakhjiri, N., & Farahmand, A. (2014). Global integrated drought monitoring and prediction system. *Scientific data*, *1*(1), 1–10.

Hao, Z., Singh, V. P., & Xia, Y. (2018). Seasonal drought prediction: advances, challenges, and future prospects. *Reviews of Geophysics*, *56*(1), 108–141.

Hao, Z., Yuan, X., Xia, Y., Hao, F., & Singh, V. P. (2017). An overview of drought monitoring and prediction systems at regional and global scales. *Bulletin of the American Meteorological Society*, *98*(9), 1879–1896.

He, Z., Chow, C.-Y., & Zhang, J.-D. (2018). Stann: A spatio–temporal attentive neural network for traffic prediction. *IEEE Access*, *7*, 4795–4806.

Ho, S.-L., Xie, M., & Goh, T. N. (2002). A comparative study of neural network and box-jenkins arima modeling in time series prediction. *Computers & Industrial Engineering*, *42*(2-4), 371–375.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735–1780.

Huang, W., Song, G., Hong, H., & Xie, K. (2014). Deep architecture for traffic flow prediction: deep belief networks with multitask learning. *IEEE Transactions on Intelligent Transportation Systems*, *15*(5), 2191–2201.

Huang, Y., Weng, Y., Yu, S., & Chen, X. (2019). Diffusion convolutional recurrent neural network with rank influence learning for traffic forecasting. In *2019 18th ieee international conference on trust, security and privacy in computing and communications/13th ieee international conference on big data science and engineering (trustcom/bigdatase)* (pp. 678–685).

Innamaa, S. (2000). Short-term prediction of traffic situation using mlp-neural networks. In *Proceedings of the 7th world congress on intelligent transport systems, turin, italy* (pp. 6–9).

Irvine, K., & Eberhardt, A. (1992). Multiplicative, seasonal arima models for lake erie and lake ontario water levels 1. *JAWRA Journal of the American Water Resources Association*, *28*(2), 385–396.

Jagadish, H. V., Gehrke, J., Labrinidis, A., Papakonstantinou, Y., Patel, J. M., Ramakrishnan, R., & Shahabi, C. (2014). Big data and its technical challenges. *Communications of the ACM*, *57*(7), 86–94.

Jain, A., & Kumar, A. M. (2007). Hybrid neural network models for hydrologic time series forecasting. *Applied Soft Computing*, *7*(2), 585–592.

Jain, S. K., Mani, P., Jain, S. K., Prakash, P., Singh, V. P., Tullos, D., ... Dimri, A. (2018). A brief review of flood forecasting techniques and their applications. *International Journal of River Basin Management*, *16*(3), 329–344.

Jaiswal, R., Ali, S., & Bharti, B. (2020). Comparative evaluation of conceptual and physical rainfall–runoff models. *Applied water science*, *10*, 1–14.

Ji, J., & Hou, J. (2017). Forecast on bus trip demand based on arima models and gated recurrent unit neural networks. In *2017 international conference on computer systems, electronics and control (iccsec)* (pp. 105–108).

Jia, X., Zwart, J., Sadler, J., Appling, A., Oliver, S., Markstrom, S., ... others (2021). Physics-guided recurrent graph model for predicting flow and temperature in river networks. In *Proceedings of the 2021 siam international conference on data mining (sdm)* (pp. 612–620).

Jiang, R., Wang, Z., Yong, J., Jeph, P., Chen, Q., Kobayashi, Y., ... Suzumura, T. (2023). Spatio-temporal meta-graph learning for traffic forecasting. In *Proceedings of the aaai conference on artificial intelligence* (Vol. 37, pp. 8078–8086).

Jin, G., Zhu, C., Chen, X., Sha, H., Hu, X., & Huang, J. (2020). Ufsp-net: a neural network with spatio-temporal information fusion for urban fire situation prediction. In *Iop conference series: Materials science and engineering* (Vol. 853, p. 012050).

Kalyan, K. S., Rajasekharan, A., & Sangeetha, S. (2021). Ammus: A survey of transformer-based pretrained models in natural language processing. *arXiv preprint arXiv:2108.05542*.

Karr, A. F., Graves, T., Mockus, A., & Schuster, P. (2002). Variability of travel times on arterial streets: effects of signals and volume. *Transportation Research Record C*, *10*, 000-000.

Karyotis, C., Maniak, T., Doctor, F., Iqbal, R., Palade, V., & Tang, R. (2019). Deep learning for flood forecasting and monitoring in urban environments. In *2019 18th ieee international conference on machine learning and applications (icmla)* (pp. 1392–1397).

Kazadi, A. N., Doss-Gollin, J., Sebastian, A., & Silva, A. (2022). Flood prediction with graph neural networks. *Climate Change AI. Climate Change AI*.

Khodayar, M., & Wang, J. (2018). Spatio-temporal graph deep neural network for short-term wind speed forecasting. *IEEE Transactions on Sustainable Energy*, *10*(2), 670–681.

Kipf, T. N., & Welling, M. (2016). Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Kişi, Ö. (2008). Stream flow forecasting using neuro-wavelet technique. *Hydrological Processes: An International Journal*, *22*(20), 4142–4152.

Kong, J., Huang, J., Yu, H., Deng, H., Gong, J., & Chen, H. (2019). Rnn-based default logic for route planning in urban environments. *Neurocomputing*, *338*, 307–320.

Kumar, S. V., & Vanajakshi, L. (2015). Short-term traffic flow prediction using seasonal arima model with limited input data. *European Transport Research Review*, *7*(3), 1–9.

Lai, Y., & Dzombak, D. A. (2020). Use of the autoregressive integrated moving average (arima) model to forecast near-term regional temperature and precipitation. *Weather and Forecasting*, *35*(3), 959–976.

Ledoux, H., & Meijers, M. (2011). Topologically consistent 3d city models obtained by extrusion. *International Journal of Geographical Information Science*, *25*(4), 557-574. Retrieved from https://doi.org/10.1080/13658811003623277 doi: 10.1080/13658811003623277

Li, Y., Yu, R., Shahabi, C., & Liu, Y. (2017). Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. *arXiv preprint arXiv:1707.01926*.

Li, Z., Ye, L., Zhao, Y., Pei, M., Lu, P., Li, Y., & Dai, B. (2022). A spatiotemporal directed graph convolution network for ultra-short-term wind power prediction. *IEEE Transactions on Sustainable Energy*, *14*(1), 39–54.

Liang, Y., Xia, Y., Ke, S., Wang, Y., Wen, Q., Zhang, J., ... Zimmermann, R. (2023). Airformer: Predicting nationwide air quality in china with transformers. In *Proceedings of the aaai conference on artificial intelligence* (Vol. 37, pp. 14329–14337).

Lin, H., Gharehbaghi, A., Zhang, Q., Band, S. S., Pai, H. T., Chau, K.-W., & Mosavi, A. (2022). Time series-based groundwater level forecasting using gated recurrent unit deep neural networks. *Engineering Applications of Computational Fluid Mechanics*, *16*(1), 1655–1672.

Lin, J., Li, Z., Li, Z., Bai, L., Zhao, R., & Zhang, C. (2023). Dynamic causal graph convolutional network for traffic prediction. *arXiv preprint arXiv:2306.07019*.

Lira, H., Martí, L., & Sanchez-Pi, N. (2021). Frost forecasting model using graph neural networks

with spatio-temporal attention. In *Ai: Modeling oceans and climate change workshop at iclr 2021*.

Liu, Y., Wang, Y., Yang, X., & Zhang, L. (2017). Short-term travel time prediction by deep learning: A comparison of different lstm-dnn models. In *2017 ieee 20th international conference on intelligent transportation systems (itsc)* (pp. 1–8).

Lv, Y., Duan, Y., Kang, W., Li, Z., & Wang, F.-Y. (2014). Traffic flow prediction with big data: a deep learning approach. *IEEE Transactions on Intelligent Transportation Systems*, *16*(2), 865–873.

Lv, Z., Xu, J., Zheng, K., Yin, H., Zhao, P., & Zhou, X. (2018). Lc-rnn: A deep learning model for traffic speed prediction. In *Ijcai* (pp. 3470–3476).

Ma, X., Dai, Z., He, Z., Ma, J., Wang, Y., & Wang, Y. (2017). Learning traffic as images: a deep convolutional neural network for large-scale transportation network speed prediction. *Sensors*, *17*(4), 818.

Ma, X., & Hovy, E. (2016). End-to-end sequence labeling via bi-directional lstm-cnns-crf. *arXiv preprint arXiv:1603.01354*.

Macklin, M., & Müller, M. (2013, jul). Position based fluids. *ACM Trans. Graph.*, *32*(4). Retrieved from https://doi.org/10.1145/2461912.2461984 doi: 10.1145/2461912.2461984

Makin, J. G., Moses, D. A., & Chang, E. F. (2020). Machine translation of cortical activity to text with an encoder–decoder framework. *Nature neuroscience*, *23*(4), 575–582.

Makwana, J. J., & Tiwari, M. K. (2014). Intermittent streamflow forecasting and extreme event modelling using wavelet-based artificial neural networks. *Water resources management*, *28*, 4857–4873.

Massimiliano Pepe, L. F., & Crocetto, N. (2022). Use of sfm-mvs approach to nadir and oblique images generated throught aerial cameras to build 2.5d map and 3d models in urban areas. *Geocarto International*, *37*(1), 120-141. Retrieved from https://doi.org/10.1080/10106049.2019.1700558 doi: 10.1080/10106049.2019.1700558

McBrearty, I. W., & Beroza, G. C. (2022). Earthquake location and magnitude estimation with graph neural networks. In *2022 ieee international conference on image processing (icip)* (pp.

3858–3862).

Medsker, L. R., & Jain, L. (2001). Recurrent neural networks. *Design and Applications*, *5*(64-67), 2.

Méndez, M., Montero, C., & Núñez, M. (2022). Using deep transformer based models to predict ozone levels. In *Asian conference on intelligent information and database systems* (pp. 169–182).

Mikolov, T., & Zweig, G. (2012). Context dependent recurrent neural network language model. In *2012 ieee spoken language technology workshop (slt)* (pp. 234–239).

Mildenhall, B., Srinivasan, P. P., Tancik, M., Barron, J. T., Ramamoorthi, R., & Ng, R. (2020). Nerf: Representing scenes as neural radiance fields for view synthesis. In *Eccv*.

Ministry of Natural Resources and Forests. (2016). *Lidar - digital models (terrain, canopy, slope)*. Dataset in Données Québec. Retrieved 2024-02-09, from https://www.donneesquebec.ca/recherche/dataset/produits-derives-de-base-du-lidar (Updated January 15, 2024)

Mohr, S., Drainas, K., & Geist, J. (2021). Assessment of neural networks for stream-water-temperature prediction. In *2021 20th ieee international conference on machine learning and applications (icmla)* (pp. 891–896).

Montanari, A., Rosso, R., & Taqqu, M. S. (1997). Fractionally differenced arima models applied to hydrologic time series: Identification, estimation, and simulation. *Water resources research*, *33*(5), 1035–1044.

Moradkhani, H., Hsu, K.-l., Gupta, H. V., & Sorooshian, S. (2004). Improved streamflow forecasting using self-organizing radial basis function artificial neural networks. *Journal of Hydrology*, *295*(1-4), 246–262.

Müller, M., Charypar, D., & Gross, M. (2003). Particle-based fluid simulation for interactive applications. In *Proceedings of the 2003 acm siggraph/eurographics symposium on computer animation* (pp. 154–159).

Narasimha Murthy, K., Saravana, R., & Vijaya Kumar, K. (2018). Modeling and forecasting rainfall patterns of southwest monsoons in north–east india as a sarima process. *Meteorology and Atmospheric Physics*, *130*, 99–106.

Natekin, A., & Knoll, A. (2013). Gradient boosting machines, a tutorial. *Frontiers in neurorobotics*, *7*, 21.

Nayak, P., Sudheer, K., Rangan, D., & Ramasastri, K. (2005). Short-term flood forecasting with a neurofuzzy model. *Water Resources Research*, *41*(4).

Niroobakhsh, M., Musavi-Jahromi, S., Manshouri, M., & Sedghi, H. (2012). Prediction of water quality parameter in jajrood river basin: application of multi-layer perceptron (mlp) perceptron and radial basis function networks of artificial neural networks (anns). *African Journal of Agricultural Research*, *7*(29), 4131–4139.

of Canada, G. (2018, Jan). *Environment and natural resources* [navigation page - theme landing page]. Retrieved from https://www.canada.ca/en/services/environment.html (Last Modified: 2023-07-11)

Online tdm encyclopedia. (n.d.). https://www.vtpi.org/tdm/tdm116.htm. (Accessed: 2021-10-21)

Oquab, M., Bottou, L., Laptev, I., & Sivic, J. (2014). Learning and transferring mid-level image representations using convolutional neural networks. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 1717–1724).

Papamichail, D. M., & Georgiou, P. E. (2001). Seasonal arima inflow models for reservoir sizing 1. *JAWRA Journal of the American Water Resources Association*, *37*(4), 877–885.

Québec, D. (2023). *Cartographie de la canopée de la rmr de montréal en format vectoriel.* https://www.donneesquebec.ca/recherche/dataset/canopee-des-six-rmr-du-quebec/resource/95bc3a15-bad9-46b4-b4f2-8000a391d770. (Accessed: 2023-11-01)

Radford, A., Narasimhan, K., Salimans, T., Sutskever, I., et al. (2018). Improving language understanding by generative pre-training.

Roudbari, N. S., Patterson, Z., Eicker, U., & Poullis, C. (2022). Simpler is better: Multilevel abstraction with graph convolutional recurrent neural network cells for traffic prediction. In *2022 ieee symposium series on computational intelligence (ssci)* (pp. 01–10).

Roudbari, N. S., Poullis, C., Patterson, Z., & Eicker, U. (2023). *Transglow: Attention-augmented transduction model based on graph neural networks for water flow forecasting.*

Sapankevych, N. I., & Sankar, R. (2009). Time series prediction using support vector machines: a survey. *IEEE computational intelligence magazine*, *4*(2), 24–38.

Schönberger, J. L., & Frahm, J.-M. (2016). Structure-from-Motion Revisited. In *Conference on computer vision and pattern recognition (cvpr).*

Schönberger, J. L., Zheng, E., Pollefeys, M., & Frahm, J.-M. (2016). Pixelwise View Selection for Unstructured Multi-View Stereo. In *European conference on computer vision (eccv).*

Shang, C., Chen, J., & Bi, J. (2021). Discrete graph structure learning for forecasting multiple time series. *arXiv preprint arXiv:2101.06861*.

Shi, X., Chen, Z., Wang, H., Yeung, D.-Y., Wong, W.-K., & Woo, W.-c. (2015). Convolutional lstm network: A machine learning approach for precipitation nowcasting. *Advances in neural information processing systems*, *28*.

Siami-Namini, S., Tavakoli, N., & Namin, A. S. (2018). A comparison of arima and lstm in forecasting time series. In *2018 17th ieee international conference on machine learning and applications (icmla)* (pp. 1394–1401).

Siami-Namini, S., Tavakoli, N., & Namin, A. S. (2019). The performance of lstm and bilstm in forecasting time series. In *2019 ieee international conference on big data (big data)* (pp. 3285–3292).

Simeunović, J., Schubnel, B., Alet, P.-J., & Carrillo, R. E. (2021). Spatio-temporal graph neural networks for multi-site pv power forecasting. *IEEE Transactions on Sustainable Energy*, *13*(2), 1210–1220.

Song, X., Kanasugi, H., & Shibasaki, R. (2016). Deeptransport: Prediction and simulation of human mobility and transportation mode at a citywide level. In *Proceedings of the twenty-fifth international joint conference on artificial intelligence* (pp. 2618–2624).

Sun, S., Zhang, C., & Yu, G. (2006). A bayesian network approach to traffic flow forecasting. *IEEE Transactions on intelligent transportation systems*, *7*(1), 124–132.

Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Advances in neural information processing systems* (pp. 3104–3112).

Tan, G., Yan, J., Gao, C., & Yang, S. (2012). Prediction of water quality time series data based on least squares support vector machine. *Procedia Engineering*, *31*, 1194–1199.

Taylor, J. W. (2003). Short-term electricity demand forecasting using double seasonal exponential smoothing. *Journal of the Operational Research Society*, *54*(8), 799–805.

There were flash floods, strong winds and at least two deaths in mississippi. (n.d.). https://www.nytimes.com/2021/08/31/us/hurricane-ida-mississippi.html. (Accessed: 2023-10-2)

Tian, C., & Chan, W. K. (2021). Spatial-temporal attention wavenet: A deep learning framework for traffic prediction considering spatial-temporal dependencies. *IET Intelligent Transport Systems*, *15*(4), 549–561.

Tian, Y., & Pan, L. (2015). Predicting short-term traffic flow by long short-term memory recurrent neural network. In *2015 ieee international conference on smart city/socialcom/sustaincom (smartcity)* (pp. 153–158).

Travel time on road segments (historical). (n.d.-a). http://www.donneesquebec.ca/recherche/dataset/vmtl-temps-de-parcours-sur-des-segments-routiers-historique.

Travel time on road segments (historical). (n.d.-b). Retrieved from https://www.donneesquebec.ca/recherche/dataset/vmtl-segments-routiers-de-collecte-des-temps-de-parcours

Van Der Voort, M., Dougherty, M., & Watson, S. (1996). Combining kohonen maps with arima time series models to forecast traffic flow. *Transportation Research Part C: Emerging Technologies*, *4*(5), 307–318.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, *30*.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., & Bengio, Y. (2017). Graph attention networks. *arXiv preprint arXiv:1710.10903*.

Wang, C., Wang, Y., Ding, Z., Zheng, T., Hu, J., & Zhang, K. (2022). A transformer-based method of multienergy load forecasting in integrated energy system. *IEEE Transactions on Smart Grid*, *13*(4), 2703–2714.

Wang, S., Cao, J., & Philip, S. Y. (2020). Deep learning for spatio-temporal data mining: A survey. *IEEE transactions on knowledge and data engineering*, *34*(8), 3681–3700.

Wang, X., Li, L., Yuan, Y., Ye, P., & Wang, F.-Y. (2016). Acp-based social computing and parallel intelligence: Societies 5.0 and beyond. *CAAI Transactions on Intelligence Technology*, *1*(4), 377–393.

Wang, Y., Long, M., Wang, J., Gao, Z., & Yu, P. S. (2017). Predrnn: Recurrent neural networks for predictive learning using spatiotemporal lstms. *Advances in neural information processing systems*, *30*.

Wang, Y., Yin, H., Chen, T., Liu, C., Wang, B., Wo, T., & Xu, J. (2021). Gallat: A spatiotemporal graph attention network for passenger demand prediction. In *2021 ieee 37th international conference on data engineering (icde)* (pp. 2129–2134).

Wang, Z., Su, X., & Ding, Z. (2020). Long-term traffic prediction based on lstm encoder-decoder architecture. *IEEE Transactions on Intelligent Transportation Systems*.

Wei, X., Wang, G., Schmalz, B., Hagan, D. F. T., & Duan, Z. (2023). Evaluate transformer model and self-attention mechanism in the yangtze river basin runoff prediction. *Journal of Hydrology: Regional Studies*, *47*, 101438.

Wu, S., Xiao, X., Ding, Q., Zhao, P., Wei, Y., & Huang, J. (2020). Adversarial sparse transformer for time series forecasting. *Advances in neural information processing systems*, *33*, 17105–17115.

Wu, Y., & Tan, H. (2016). Short-term traffic flow forecasting with spatial-temporal correlation in a hybrid deep learning framework. *arXiv preprint arXiv:1612.01022*.

Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., & Philip, S. Y. (2020). A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*.

Wu, Z., Pan, S., Long, G., Jiang, J., Chang, X., & Zhang, C. (2020). Connecting the dots: Multivariate time series forecasting with graph neural networks. In *Proceedings of the 26th acm sigkdd international conference on knowledge discovery & data mining* (pp. 753–763).

Xiangli, Y., Xu, L., Pan, X., Zhao, N., Rao, A., Theobalt, C., ... Lin, D. (2022). Bungeenerf: Progressive neural radiance field for extreme multi-scale scene rendering. In *European conference on computer vision* (pp. 106–122).

Xiao, J.-M., & Wang, X.-H. (2004). Study on traffic flow prediction using rbf neural network. In *Proceedings of 2004 international conference on machine learning and cybernetics (ieee cat.*

no. 04ex826) (Vol. 5, pp. 2672–2675).

Xie, Y., Zhang, Y., & Ye, Z. (2007). Short-term traffic volume forecasting using kalman filter with discrete wavelet decomposition. *Computer-Aided Civil and Infrastructure Engineering*, *22*(5), 326–334.

Xu, F., Li, Y., & Xu, S. (2020). Attentional multi-graph convolutional network for regional economy prediction with open migration data. In *Proceedings of the 26th acm sigkdd international conference on knowledge discovery & data mining* (pp. 2225–2233).

Xu, L., Chen, N., Zhang, X., & Chen, Z. (2018). An evaluation of statistical, nmme and hybrid models for drought prediction in china. *Journal of hydrology*, *566*, 235–249.

Xu, M., Dai, W., Liu, C., Gao, X., Lin, W., Qi, G.-J., & Xiong, H. (2020). Spatial-temporal transformer networks for traffic flow forecasting. *arXiv preprint arXiv:2001.02908*.

Yang, H.-F., Dillon, T. S., & Chen, Y.-P. P. (2016). Optimized structure of the traffic flow forecasting model with a deep learning approach. *IEEE transactions on neural networks and learning systems*, *28*(10), 2371–2381.

YAO, Z.-s., SHAO, C.-f., & GAO, Y.-l. (2006). Research on methods of short-term traffic forecasting based on support vector regression [j]. *Journal of Beijing Jiaotong University*, *30*(3), 19–22.

You, K., Long, M., Wang, J., & Jordan, M. I. (2019). How does learning rate decay help modern neural networks? *arXiv preprint arXiv:1908.01878*.

Yu, B., Yin, H., & Zhu, Z. (2017). Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. *arXiv preprint arXiv:1709.04875*.

Yu, D., Ji, S., Liu, J., & Wei, S. (2021). Automatic 3d building reconstruction from multi-view aerial images with deep learning. *ISPRS Journal of Photogrammetry and Remote Sensing*, *171*, 155-170. Retrieved from https://www.sciencedirect.com/science/article/pii/S092427162030318X doi: https://doi.org/10.1016/j.isprsjprs.2020.11.011

Yu, H., Wu, Z., Wang, S., Wang, Y., & Ma, X. (2017). Spatiotemporal recurrent convolutional networks for traffic prediction in transportation networks. *Sensors*, *17*(7), 1501.

Yu, R., Li, Y., Shahabi, C., Demiryurek, U., & Liu, Y. (2017). Deep learning: A generic approach

for extreme condition traffic forecasting. In *Proceedings of the 2017 siam international conference on data mining* (pp. 777–785).

Yu, Y., Si, X., Hu, C., & Zhang, J. (2019). A review of recurrent neural networks: Lstm cells and network architectures. *Neural computation*, *31*(7), 1235–1270.

Zeng, A., Chen, M., Zhang, L., & Xu, Q. (2023). Are transformers effective for time series forecasting? In *Proceedings of the aaai conference on artificial intelligence* (Vol. 37, pp. 11121–11128).

Zhang, J., & Man, K.-F. (1998). Time series prediction using rnn in multi-dimension embedding phase space. In *Smc'98 conference proceedings. 1998 ieee international conference on systems, man, and cybernetics (cat. no. 98ch36218)* (Vol. 2, pp. 1868–1873).

Zhang, J., Shi, X., Xie, J., Ma, H., King, I., & Yeung, D.-Y. (2018). Gaan: Gated attention networks for learning on large and spatiotemporal graphs. *arXiv preprint arXiv:1803.07294*.

Zhang, S., Tong, H., Xu, J., & Maciejewski, R. (2019). Graph convolutional networks: a comprehensive review. *Computational Social Networks*, *6*(1), 1–23.

Zhang, X., Huang, C., Xu, Y., & Xia, L. (2020). Spatial-temporal convolutional graph attention networks for citywide traffic flow forecasting. In *Proceedings of the 29th acm international conference on information & knowledge management* (pp. 1853–1862).

Zhao, L., Song, Y., Zhang, C., Liu, Y., Wang, P., Lin, T., ... Li, H. (2019). T-gcn: A temporal graph convolutional network for traffic prediction. *IEEE Transactions on Intelligent Transportation Systems*.

Zhao, Z., Yun, S., Jia, L., Guo, J., Meng, Y., He, N., ... Yang, L. (2023). Hybrid vmd-cnn-gru-based model for short-term forecasting of wind power considering spatio-temporal features. *Engineering Applications of Artificial Intelligence*, *121*, 105982.

Zheng, C., Fan, X., Wang, C., & Qi, J. (2020). Gman: A graph multi-attention network for traffic prediction. In *Proceedings of the aaai conference on artificial intelligence* (Vol. 34, pp. 1234–1241).

Zheng, Y., Capra, L., Wolfson, O., & Yang, H. (2014). Urban computing: concepts, methodologies, and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, *5*(3), 1–55.

Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., & Zhang, W. (2021). Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the aaai conference on artificial intelligence* (Vol. 35, pp. 11106–11115).

Zhou, H., Zhang, Y., Yang, L., Liu, Q., Yan, K., & Du, Y. (2019). Short-term photovoltaic power forecasting based on long short term memory neural network and attention mechanism. *Ieee Access*, *7*, 78063–78074.

Zhou, Z., & Li, X. (2017). Graph convolution: A high-order and adaptive approach. *arXiv preprint arXiv:1706.09916*.

Zivot, E., & Wang, J. (2006). Vector autoregressive models for multivariate time series. *Modeling financial time series with S-PLUS*, 385–429.