Configuration Manual

MSc Research Practicum
MSc in Artificial Intelligence

Surendra Adabala
Student ID: 23346418

School of Computing
National College of Ireland

Supervisor: Abdul Razzq

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: Surendra Adabala

Student ID: 23346418

Programme: MSc in Artificial Intelligence **Year:** 2024-2025

Module: MSc Research Practicum

Lecturer: Abdul Razzaq

Submission Due Date: 15/09/2025

Project Title: Multi-Modal Urban Mobility Forecasting: A Graph Neural Network-Based Approach with Spatiotemporal Hypergraph Attention

Word Count: 851 **Page Count:** 3

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary

Signature: Surendra Adabala

Date: 14/09/2025

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Surendra Adabala
23346418

1 Introduction

This configuration manual provides a concise and guided walkthrough of the setup process for the research project titled DSTHGCN: A Novel Deep Learning Framework for Urban Mobility Forecasting. The project explores spatiotemporal patterns of urban mobility using deep learning, with a specific focus on integrating dynamic hypergraph learning and temporal modeling techniques. This guide is designed for students, researchers, and evaluators who wish to replicate the results, test the model, or expand upon the existing framework.

DSTHGCN model trained on real world NYC yellow taxi and CitiBike service data, for a multi-modal time series forecasting pipeline. This manual explains the prerequisites such as tools, software environments and provides a walkthrough to install the project dependencies and run the model. This ensures reproducibility and makes sense to the reader either working on Google Colab or a local Jupyter setup. The broad goal of this guide is to make it easier for anyone to be able to use one of the most powerful deep learning models in existence.

2 System Requirements & Dependencies

Model training and evaluation are done in the cloud using Google Colab, so on the user sides this project only needs a minimum system specifications. This way everyone, including users with limited local resources, can reproduce results and interact with the model.

Every libraries required sho

Recommended System Configuration:

Browser: Google Chrome, Firefox, or Safari (latest version)

Internet: Stable connection for Colab usage and file uploads

Software Stack:

Environment: Google Colab (GPU-enabled)

Python: Version 3.10+ (handled automatically by Colab)

Dependencies: Every libraries required should already be installed or can be install in notebook cells using pip. Which includes:

PyTorch, NumPy, Pandas

NetworkX, Scikit-learn

Matplotlib, Seaborn, tqdm

Core Libraries Used:

PyTorch: Deep learning framework used for model construction and training

NetworkX: For hypergraph creation and manipulation

Scikit-learn: For preprocessing and metric computation

Matplotlib/Seaborn: Visualizing outputs and efficacy

3 Project Execution Workflow

DSTHGCN model was implemented in the Google Colab environment with GPU acceleration, which can improve the training and testing performance without requiring hardware. The execution is initiated by uploading the datasets — bike_flow_tensor npy, taxi_flow_tensor. npy, and zone_lookup. json – normalised spatiotemporal inflow-outflow data for the different NY modes (Taxi, green Taxi and Citi Bike) The files are loaded and concatenated in a 3D tensor which captures the temporal patterns inside zones across modalities.

After loading of the data, we formed a combined tensor is split into overlapping sequences using a sliding window methodology (12 timesteps each) and along with one timestep ahead target. These resultant sequences are divided into training, validation and test in the ratio of 70-15-15 respectively. We package each sequence and its corresponding label into custom PyTorch Dataset object and load these DataLoaders for training in mini batches.

Model initialization is done directly in the script and architecture parameters, like N of hidden units, attention dimensions and dropout are clearly declared. There is no requirement of an external configuration file but we specify hyperparameters within the notebook cells. We have used Adam as an optimizer, and Mean Squared Error (MSE) for our loss function. The training goes up to a certain number of epochs (e.g., 60) with options for early stopping or checkpointing by monitoring the validation loss. Finally, we evaluate the performance of our model on multiple metrics such as MSE, RMSE, MAE, R^2 Score, as well as an accuracy estimation using a derived MAE. We display the results directly after evaluation, so they can be easily interpreted.

4 Model Architecture Explanation

The DSTHGCN model is designed to learn complex spatio-temporal dependencies in urban mobility data by coupling the dynamic hypergraph convolution, hierarchical attention and temporal modeling via GRU. Fundamentally, the model bootstraps by creating a time-evolving hypergraph for each timestep: zones are nodes and higher-order interactions are hyperedges that can be inferred from many-to-many OD (Origin-Destination) flows. While classical static graphs are capable to encode only pairwise interactions, this formulation of hypergraph is much more flexible as it supports explicit group-wise spatial dependencies occurring in real-world urban mobility.

Hypergraph Convolution: After getting the spacial embeddings, they pass through a hierarchial attention. This module provides learnable weights at the node and attribute level, which instructs the model to attend to critical zones and features in each context hyperedge. Such an approach makes the estimation easier to interpret while enabling adaptive weighting based on mobility patterns. The attention-weighted spatial features are passed through a stack of GRU (Gated Recurrent Unit) layers which model temporal dependencies by holding an internal state across time. We use GRUs instead of LSTM as they are computationally efficient and perform well for moderately long sequences (12-steps lookback window for this task).

This combines the dynamic hypergraphs representation with hierarchical attention and GRU to gain an understanding of spatial irregularities, adapt to local context, as well as effectively track temporal fluctuations. This architectural synthesis provides an effective solution to the limited capacity of existing GCN or LSTM-only methods, by integrating higher-order spatial encoding with temporal dynamics within a compact yet expressive stanza. The last output is then taken through a fully-connected regression layer to predict mobility inflows for all zones and timesteps.

References

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, 32(1), 4–24.

Zhao, L., Song, Y., Zhang, C., Liu, Y., Wang, J., Lin, T., & Deng, M. (2020). T-GCN: A Temporal Graph Convolutional Network for Traffic Prediction. IEEE Transactions on Intelligent Transportation Systems, 21(9), 3848–3858.

Wei, L., Gu, Y., Yang, H., He, W., Zheng, H., & Zhang, D. (2023). SDSC: Self-supervised Dynamic Spatiotemporal Graph Learning for Traffic Prediction. AAAI Conference on Artificial Intelligence (AAAI).

Kipf, T. N., & Welling, M. (2017). Semi-Supervised Classification with Graph Convolutional Networks. International Conference on Learning Representations (ICLR).