1 Overview

Understanding and predicting spatiotemporal dynamics in urban environments has become increasingly critical for effective city management and decision-making (Barr et al. 2020). The proliferation of Internet of Things (IoT) sensors across urban landscapes generates vast streams of spatiotemporally correlated data, presenting both opportunities and challenges for modelling urban mobility patterns (Karkouch et al. 2016). While these sensor networks provide unprecedented insight into urban dynamics, the complex, non-linear relationships between spatial and temporal dependencies, coupled with the high-dimensional nature of the data, necessitate sophisticated modelling approaches that can balance computational efficiency with predictive accuracy (Dong et al. 2023).

Deep Learning (DL) models have emerged as particularly effective tools for analysing spatiotemporal dependencies in sensor networks. These models can perform automatic feature extraction, learning intricate patterns from raw data that would be difficult or impossible to capture through manual feature engineering (Xiao et al. 2022). This capability is especially valuable for complex IoT systems like pedestrian or traffic networks, where traditional feature engineering approaches struggle with the context-dependent nature of the data and the intricate spatiotemporal interactions between variables (Yu et al. 2020, Dong et al. 2023).

The challenge of modelling spatiotemporal dependencies in sensor networks is further complicated by the high-dimensional nature of the data, which often involves multiple sensors, diverse modalities, and evolving time series. Deep learning architectures are capable of processing and extracting meaningful representations from such data (Dong et al. 2023, Kontolati et al. 2024). There are various architectures that effectively model spatial and temporal dependencies. Convolution Neural Networks (CNNs) leverage convolutional filters to extract spatial patterns and relationships (Wu et al. 2021, Xiao et al. 2022). Recurrent Neural Networks (RNNs) capture temporal dependencies through recurrent connections, maintaining memory of past information (Wu et al. 2021, Xiao et al. 2022, Hu et al. 2023). More recent architectures like Transformers have demonstrated remarkable capability in modelling both short-range and long-range dependencies across spatial and temporal dimensions (Yu et al. 2020).

However, these approaches face several key limitations that must be considered when assessing spatiotemporal dependencies. DL models typically require substantial training data and computational resources for training and inference (Scarselli et al. 2009, Qi & Majda 2020, Hu et al. 2023) - although, as discussed in the following section, time complexity (O(N)) is contingent on the specific model architecture, and the type of problem being solved. Lastly, interpretability of DL models can be challenging, especially for complex models such as transformers, due to their multiple layers of non-linear transformations and millions of interconnected parameters that make it difficult to trace how specific inputs lead to particular predictions (Dong et al. 2023).

The most powerful DL models for spatiotemporal data are transformers and spatiotemporal graph neural networks (Scarselli et al. 2009) (STGNNs). Transformers (Vaswani

et al. 2017) excel at capturing long-range dependencies in sequences, making them well-suited for tasks involving time series or spatial relationships (Yu et al. 2020, Zhou et al. 2021). Additionally, Zhou et al. (2021) highlights the self-attention mechanisms ability to reduce the maximum length of network signal travelling paths, offering potential for efficiency. Vaswani et al. (2017)'s seminal paper on transformers demonstrates how the transformers architecture allows for significantly faster training that architectures based on recurrent or convolutional layers. Zhou et al. (2021), Hu et al. (2023) both note that standard self attention mechanisms have quadratic time complexity and high memory usage, limiting scalability for long sequences. To mitigate this Hu et al. (2023) propose using downsampling techniques to reduce computational complexity, and Kim et al. (2024) propose using token embeddings to represent graph structures which can theoretically result in near linear time complexity.

As Wu et al. (2021) states, GNNs excel at capturing local dependencies and structural information within graphs, making them well suited for tasks where the relationships between nodes are crucial, such as node classification, link prediction, and graph classification.Dong et al. (2023) finds that integrating graph convolutional networks (GCNs) with RNNs allows for the extraction of topological features and the capture of temporal dependencies between graphs. However Dong et al. (2023) also notes that message-passing neural network-based STGNNs can suffer from scalability issues due to the requirement of processing messages from all involved agents. GCNs as proposed by Kipf & Welling (2017) required a fixed graph structure, making them unsuitable for dynamic problems, such as traffic forecasting where the graph structure changes over time. However, more recent advances such as the Graph Attention Network (GAT) proposed by Veličković et al. (2018) can adapt to structural changes in the graph, without the need for retraining. The benefit of combining self-attention into GNN architectures as highlighted by Kim et al. (2024) is a significant reduction in message-passing even when self-attention is restricted to local neighbourhoods.

Over the last few years GNNs and transformers have been applied to an increasing number of problems. In Károly et al. (2021) GNNs are employed in robotics for tasks like robot planning and control. For single robots, GNNs can model joints as nodes and connections as edges to predict behaviour. In multi-robot scenarios, GNNs help with decentralised control and communication, where robots communicate and process messages based on their interactions, represented as a graph (Wu et al. 2021, Kim et al. 2024, Kontolati et al. 2024). Both Zheng et al. (2019) and Keskes & Noumeir (2021) use GCNs in human activity recognition systems based on sensor data, where body joints are represented as nodes in a graph. Both GNNs and Transformers are applied extensively to smart cities applications. STD-Net a transformer-based network that extracts spatial-temporal features to forecast traffic volume is proposed by Hu et al. (2023). Zhang et al. (2021) designed a dynamic auto-structural GNN model to predict the origin-destination (OD) demand using a transportation graph (regions as nodes, OD pairs as edges). Jin et al. (2022) proposes a framework called STGNN-TTE (spatio-temporal graph neural network with travel time estimation) to predict the time it takes for a vehicle to traverse a particular route in a city,

considering the complex interplay of spatial and temporal factors affecting traffic flow which is capable of learning dynamic spatial correlations that go beyond the fixed topology of the road network and utilise transformer layers to extract individualised long-term temporal dynamics. Xia et al. (2021) addresses the challenge of predicting citywide crowd flow in irregular regions (forecasting the number of people moving in and out of different areas of a city) using a framework called 3DGCN (3-dimensional graph convolution network) which models crowd flow prediction as a dynamic spatio-temporal graph prediction problem, where nodes represent regions with time-varying flows, and edges represent the origin-destination (OD) flow between regions. Kim et al. (2024) finds that while transformers are increasingly being explored for graph-related tasks, ultimately, the choice between STGNN and Transformer-based architectures comes down to the specific application and data characteristics. As a result, factors such as graph size, the importance of temporal consistency, and computational constraints will be considered as the main factors in this decision for this piece of research Dong et al. (2023), Kim et al. (2024).

2 Research Objectives

Based on this review of current approaches and their limitations, this research aims to advance our understanding of spatiotemporal dependencies in near real-time sensor data through three specific objectives:

1. Evaluate the effectiveness of different deep learning architectures in capturing spatiotemporal dependencies within urban sensor networks, with particular focus on computational efficiency for real-time applications.
2. Quantify the minimum data requirements (in terms of both spatial and temporal resolution) needed to reliably detect and predict spatiotemporal patterns in urban mobility data.
3. Develop and validate metrics for assessing the quality of spatiotemporal predictions, considering both local and global dependency structures.

These objectives directly address current gaps in our understanding of how to effectively leverage deep learning approaches for real-time spatiotemporal analysis while maintaining computational feasibility and ensuring reliable predictions. The research will contribute to both the theoretical understanding of spatiotemporal dependencies in sensor networks and the practical implementation of these insights in urban monitoring systems.

3 Overview of Methods

The methodology for assessing spatiotemporal dependencies in near real-time sensor data will follow a systematic approach implemented in Python using PyTorch for deep learning components. The investigation will proceed through three main phases: data preprocessing and representation, model development and training, and performance evaluation.

In the initial phase, sensor data will be processed to create appropriate spatiotemporal

representations. This will involve constructing graph structures where nodes represent individual sensors and edges represent potential relationships between them, such as physical proximity or correlated behaviour patterns. The spatial relationships will be encoded using adjacency matrices, while temporal dependencies will be captured through time-series windows of appropriate length. A key consideration will be determining the optimal window size that balances computational efficiency with the capture of relevant temporal patterns.

Sensors will be edge nodes in the graph, and the edges will be weighted based on the distance of the shortest path between the sensors from the road/path network. This will form a fully connected graph. This may cause issues as the best time complexity that we can expect is $O(V + E)$ where V are vertices (nodes) and E are edges. This simplifies to $O(V^2)$ (quadratic time) in a fully connected graph. Graph sparsification techniques such as edge pruning or edge sampling may be necessary to reduce the computational complexity of the model.

The model development phase will implement and compare different deep learning architectures, focusing particularly on their ability to capture spatiotemporal dependencies. The base implementation will utilise PyTorch's neural network modules, extended with specialised libraries such as PyTorch Geometric or DGL (Deep Graph Library) for graph-based operations. Two primary model variants will be developed: a baseline model using traditional deep learning approaches (such as LSTM) and an advanced model incorporating graph neural network components. This comparison will help quantify the benefits of explicitly modelling spatial relationships in the prediction task.Performance evaluation will focus on two key metrics: prediction accuracy and computational efficiency. Standard regression metrics such as Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) will assess prediction accuracy. Computational efficiency will be measured through training time, inference time, and memory usage. The evaluation will also include an analysis of how prediction accuracy varies with different spatial and temporal resolutions of input data, helping establish minimum data requirements for reliable predictions.

To ensure reproducibility and facilitate future development, all code will be version controlled using Git and documented following standard Python documentation practices. The methodology will include regular validation checks using hold-out test sets and cross-validation procedures to ensure robust evaluation of model performance. This methodological approach directly addresses the research objectives by providing quantitative measures of spatiotemporal dependency capture, establishing minimum data requirements, and evaluating computational feasibility for real-time applications. The implementation will be structured to allow for future extensions and adaptations based on initial findings and emerging requirements.

References

Barr, S. L., Johnson, S., Ming, X., Peppa, M., Dong, N., Wen, Z., Robson, C., Smith, L., James, P., Wilkinson, D., Heaps, S., Laing, Q., Xiao, W., Dawson, R. & Ranjan, R. (2020), 'FLOOD-PREPARED: A NOWCASTING SYSTEM FOR REAL-TIME IMPACT ADAPTION TO SURFACE WATER FLOODING IN CITIES', *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences* **VI-4-W2-2020**, 9–15.

Dong, G., Tang, M., Wang, Z., Gao, J., Guo, S., Cai, L., Gutierrez, R., Campbel, B., Barnes, L. E. & Boukhechba, M. (2023), 'Graph Neural Networks in IoT: A Survey', *ACM Trans. Sen. Netw.* **19**(2), 47:1–47:50.

Hu, Y., Zhou, Y., Song, J., Xu, L. & Zhou, X. (2023), 'Citywide Mobile Traffic Forecasting Using Spatial-Temporal Downsampling Transformer Neural Networks', *IEEE Transactions on Network and Service Management* **20**(1), 152–165.

Jin, G., Wang, M., Zhang, J., Sha, H. & Huang, J. (2022), 'STGNN-TTE: Travel time estimation via spatial–temporal graph neural network', *Future Generation Computer Systems* **126**, 70–81.

Karkouch, A., Mousannif, H., Al Moatassime, H. & Noel, T. (2016), 'Data quality in internet of things: A state-of-the-art survey', *Journal of Network and Computer Applications* **73**, 57–81.

Károly, A. I., Galambos, P., Kuti, J. & Rudas, I. J. (2021), 'Deep Learning in Robotics: Survey on Model Structures and Training Strategies', *IEEE Transactions on Systems, Man, and Cybernetics: Systems* **51**(1), 266–279.

Keskes, O. & Noumeir, R. (2021), 'Vision-Based Fall Detection Using ST-GCN', *IEEE Access* **9**, 28224–28236.

Kim, T., Kim, J., Kim, J. & Oh, S. (2024), 'Optimization of number of wireless temperature sensors using clustering algorithm for deep learning algorithm-based Kimchi quality prediction', *Journal of Food Engineering* **367**.

Kipf, T. N. & Welling, M. (2017), 'Semi-Supervised Classification with Graph Convolutional Networks'.

Kontolati, K., Goswami, S., Em Karniadakis, G. & Shields, M. D. (2024), 'Learning nonlinear operators in latent spaces for real-time predictions of complex dynamics in physical systems', *Nature Communications* **15**(1), 5101.

Qi, D. & Majda, A. J. (2020), 'Using machine learning to predict extreme events in complex systems', *Proceedings of the National Academy of Sciences* **117**(1), 52–59.

Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M. & Monfardini, G. (2009), 'The Graph Neural Network Model', *IEEE Transactions on Neural Networks* **20**(1), 61–80.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł. & Polosukhin, I. (2017), 'Attention is All you Need'.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P. & Bengio, Y. (2018), 'Graph Attention Networks'.

Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C. & Yu, P. S. (2021), 'A Comprehensive Survey on Graph Neural Networks', *IEEE Transactions on Neural Networks and Learning Systems* **32**(1), 4–24.

Xia, T., Lin, J., Li, Y., Feng, J., Hui, P., Sun, F., Guo, D. & Jin, D. (2021), '3DGCN: 3-Dimensional Dynamic Graph Convolutional Network for Citywide Crowd Flow Prediction', *ACM Trans. Knowl. Discov. Data* **15**(6), 110:1–110:21.

Xiao, S., Wang, S., Huang, Z., Wang, Y. & Jiang, H. (2022), 'Two-stream transformer network for sensor-based human activity recognition', *Neurocomputing* **512**, 253–268.

Yu, C., Ma, X., Ren, J., Zhao, H. & Yi, S. (2020), 'Spatio-Temporal Graph Transformer Networks for Pedestrian Trajectory Prediction'.

Zhang, D., Xiao, F., Shen, M. & Zhong, S. (2021), 'DNEAT: A novel dynamic node-edge attention network for origin-destination demand prediction', *Transportation Research Part C: Emerging Technologies* **122**, 102851.

Zheng, Y., Zhang, D., Yang, L. & Zhou, Z. (2019), Fall detection and recognition based on GCN and 2D Pose, *in* '2019 6th International Conference on Systems and Informatics (ICSAI)', pp. 558–562.

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', *Proceedings of the AAAI Conference on Artificial Intelligence* **35**(12), 11106–11115.