# TMS-GNN
## Traffic-aware Multistep Graph Neural Network for bus passenger flow prediction

Baghbani, Asiye; Rahmani, Saeed; Bouguila, Nizar; Patterson, Zachary

**DOI**
10.1016/j.trc.2025.105107

**Publication date**
2025

**Document Version**
Final published version

**Published in**
Transportation Research Part C: Emerging Technologies

Transportation Research Part C 174 (2025) 105107

Contents lists available at ScienceDirect

# Transportation Research Part C

journal homepage: www.elsevier.com/locate/trc

# TMS-GNN: Traffic-aware Multistep Graph Neural Network for bus passenger flow prediction

Asiye Baghbani [a,b,*], Saeed Rahmani [c], Nizar Bouguila [b], Zachary Patterson [b]

[a] Concordia University, Montreal, Canada
[b] BusPas Inc, Montreal, Canada
[c] Delft University of Technology, Delft, The Netherlands

ARTICLE INFO

*Keywords:*
Passenger flow prediction
Graph neural network
Multistep prediction
Bus network

ABSTRACT

Bus network plays a critical role in urban transportation affecting the use of private vehicles, traffic congestion, and urban accessibility. The accurate prediction of bus passenger flow is key to improving transit passenger experience and increasing the efficiency of bus network operations. In line with recent advances in deep learning for passenger flow prediction, graph neural networks (GNNs) have become increasingly popular due to their ability to account for the network structure between stops. Existing GNN-based models for bus passenger flow prediction, however, face several limitations. First, they do not take into account some distinctive characteristics of bus networks, such as their coexistence with vehicular traffic and their high sensitivity to urban traffic conditions. Moreover, sequence prediction models that have been widely applied to multistep passenger flow prediction suffer from a critical issue, called “exposure bias.” This results in the propagation and accumulation of errors through prediction steps while making predictions for farther time horizons. To address these issues, this study presents the Traffic-Aware multistep Graph Neural Network (TMS-GNN) model with Scheduled Sampling, a graph-based deep-learning framework designed to forecast multistep bus passenger flows at individual stops across a bus network. The model takes into account factors such as bus stop connectivity, urban traffic impacts, and multi-dimensional temporal patterns; and addresses exposure bias by employing a curriculum learning strategy called Scheduled Sampling. The comparison between the proposed model and other popular baseline models on two real-world networks with different geographical and urban patterns in Canada and USA shows that TMS-GNN outperforms the baselines in both the overall network-wide task, as well as multistep prediction. Also, to verify the contribution of the proposed components of the model, an ablation study is conducted. The results of the ablation study validate the design choices as well.

## 1. Introduction

Buses form the backbone of public transportation in most cities, playing a crucial role in increasing accessibility and reducing traffic congestion and pollution in urban areas. The effectiveness of bus transportation systems depends on various factors, with bus ridership (or short-term passenger flows) being particularly important as it directly influences traveler experience and helps operators improve system performance. Given its significance, a wide variety of methods have been applied for short-term passenger

* Corresponding author at: Concordia University, Montreal, Canada.
*E-mail addresses:* asiye.baghbani@mail.concordia.ca (A. Baghbani), s.rahmani@tudelft.nl (S. Rahmani), nizar.bouguila@concordia.ca (N. Bouguila), zachary.patterson@concordia.ca (Z. Patterson).

https://doi.org/10.1016/j.trc.2025.105107
Received 18 March 2024; Received in revised form 13 March 2025; Accepted 13 March 2025
Available online 24 March 2025

flow prediction. Initial studies utilized classical time series analysis methods, such as autoregressive models (Cyril et al., 2018; Xue et al., 2015) and Kalman filters (Jiao et al., 2016; Chun-Hui and Song-Rui, 2011), but these methods were shown to be incapable of capturing nonlinear and complex patterns in passenger flow data (Zhai et al., 2018). Machine learning and deep learning models have emerged as alternative techniques with their great ability to model spatial and temporal dependencies (Xie et al., 2020). Among different deep learning techniques, Convolutional neural networks (CNNs), in particular, have been widely used for capturing spatial correlations in traditional deep networks (Zhai et al., 2018; Xue et al., 2023).

Despite their popularity, CNNs have a fundamental limitation when applied to bus networks as they assume correlations in Euclidean space (Wu et al., 2020). This assumption falls short in bus networks, where connectivity between stops plays a more important role than physical proximity in determining passenger flow patterns (Baghbani et al., 2023b; Jiang and Luo, 2022). Correlation between transit stops can vary significantly based on network topology. Proximate stops on disparate routes may exhibit minimal correlation, whereas geographically distant stops along a single line could demonstrate high correlation. This characteristic of bus networks suggests that *graph representations* offer a more accurate model for bus transport systems than traditional grid-based approaches like CNNs. Graph neural networks (GNNs) have been introduced as effective methods to enable applying traditional deep learning operations, such as convolutions, on graph-structured data (Wu et al., 2020; Liu and Meidani, 2023). Recent applications of GNNs to passenger flow prediction problems have shown promising results (Rahmani et al., 2023). These models have demonstrated improved accuracy compared to traditional approaches, particularly in capturing the spatial dependencies between the stops (Rahmani et al., 2023) or defining new relationships between regions (Luo et al., 2021; Lin et al., 2018; Liu and Meidani, 2023).

While these advancements are significant, several important limitations and unexplored areas persist in the current research on GNNs for passenger flow prediction. Firstly, applications of GNNs in passenger flow prediction have been mainly limited to rail-based public transportation systems, with bus systems either disregarded or modeled similarly to rail-based systems without considering their unique properties (Baghbani et al., 2023a; Rahmani et al., 2023). Bus transportation systems possess distinct operational and design characteristics compared to rail transit systems, necessitating a dedicated line of research (Liu and Chen, 2017; Liu et al., 2019). For instance, bus networks usually experience higher fluctuations of passenger flow, especially at unsaturated stops, with sporadic zero passenger counts. Although passenger flow sparsity is also observed in metro systems, bus networks are inherently more susceptible to this issue due to several key factors: Firstly, bus networks are designed for broad geographical coverage, extending services across areas with diverse population densities, which naturally includes zones with lower average ridership and increased potential for zero counts. Secondly, the high density of bus stops creates localized demand catchments, meaning individual stops are more likely to experience periods of low or no passenger boardings, especially outside peak hours. Thirdly, bus operations are more vulnerable to external disruptions, such as traffic and weather, leading to unpredictable ridership fluctuations and increased sparsity. Fourthly, bus networks feature operationally flexible and demand-responsive service schedules, which, while efficient, can result in periods of reduced service and thus higher sparsity. Finally, the role of bus networks in providing first/last mile connectivity often leads to routes designed for specific purposes and times, which can exhibit naturally lower ridership and contribute to data sparsity. These factors result in data sparsity that adds another layer of complexity to the modeling process and demands tailored solutions for bus networks.

However, the most significant distinction between bus and metro systems, arguably, lies in their interaction and overlap with road vehicular traffic. Unlike metro systems on dedicated tracks, bus routes often share roads with other vehicles, introducing high uncertainty and variability in operations. This integration affects travel times, schedule adherence, and passenger accumulation at stops, and consequently can adversely impact the short-term prediction of passenger flow. For instance, traffic congestion can lead to bus delays, resulting in higher passenger counts at subsequent stops and disrupting the typical passenger flow patterns across the network. Consequently, considering both the intrinsic factors of the bus network and the dynamic nature of urban traffic patterns is essential for predicting passenger flow in bus networks. Cui et al. (2019) attempted to incorporate traffic patterns in predictive GNN models, though their focus was on road traffic systems rather than bus networks. Despite this effort, their study only accounted for static traffic conditions, assuming free-flow conditions on all roads and overlooking the dynamic, often fluctuating nature of real-world traffic. Consequently, their model failed to capture the impact of varying traffic conditions on prediction accuracy, an oversight particularly significant for applications in urban environments with variable congestion patterns. To the best of our knowledge, current research utilizing GNNs for bus passenger flow prediction has similarly disregarded this important property of bus networks.

Secondly, most recent studies for passenger flow prediction have focused on single-step forecasting, which may not meet the diverse needs of transit operators and users. Travelers require short-term predictions (10–20 min) for decision-making, while operators need longer horizons (30–60 min) to implement measures. A precise multistep prediction system offering both short and mid-term predictions would be more beneficial. Although some recent studies have explored multi-step models (Ou et al., 2022; Hao et al., 2019), their proposed frameworks are prone to “exposure bias”, leading to error accumulation in longer prediction horizons (Sangiorgio and Dercole, 2020). While techniques to address exposure bias exist in the computer science literature, their application and empirical validation remain unexplored in passenger flow prediction, particularly in the context of bus passenger flows, which are highly stochastic and dependent on exogenous factors like traffic conditions.

Thirdly, prior research indicates that the relationships between spatial and temporal features in passenger flow change over time and are at multiple scales (Yin et al., 2023; He et al., 2022). This implies that accounting for fluctuations in temporal correlations among data points, as well as multiple relationships among historical patterns, is essential in passenger flow prediction. However, the incorporation of multi-dimensional temporal features in GNN-based bus passenger flow prediction models has been barely explored, and the effects of such features on the accuracy and robustness of these models have not been thoroughly evaluated in the current literature.

### 1.1. Objectives and contributions

The limitations and unexplored areas in the current research on GNNs for bus passenger flow prediction highlight the need for new approaches that consider the distinctive properties of bus transportation systems, such as their interaction with urban traffic patterns, while also incorporating robust multi-step forecasting and multi-dimensional temporal features. This study aims to address these gaps by developing a graph-based deep learning framework for predicting multi-step bus passenger flows at individual stops across an entire bus network. The proposed framework explicitly accounts for the impact of traffic conditions, the connectivity of bus stops, and multi-dimensional temporal patterns, while integrating multi-step predictions using both short-term and long-term historical data and applying Scheduled Sampling (Bengio et al., 2015) to prevent error propagation and accumulation in multi-step prediction. Furthermore, we perform an extensive evaluation based on two real-world datasets and conduct an ablation study to explore the role of different proposed model components. The main contributions of this study can be summarized as follows:

1. For the first time in bus passenger flow prediction, we have developed a graph-based deep learning model that explicitly models both the graph-structured nature of bus networks and their dynamic nature due to their co-existence with vehicular traffic. This unique design goes beyond the simplified assumptions often made in modeling passenger flow prediction in bus transportation systems, which ignore the impacts of road congestion and uncertainties on the network. The integration of traffic-aware reachability within a graph-based neural network structure is a significant contribution as it enables the model to reflect the complexities and dependencies that affect bus operations. We also conduct extensive investigations to prove the importance of considering such property of bus networks.
2. We address the challenge of error accumulation in multi-step bus passenger flow prediction by designing a specific multi-step prediction mechanism. This ensures that the model remains robust across different time horizons. The proposed approach enhances the model's ability to provide reliable multi-step forecasts, which are crucial for both short-term operational decisions and long-term planning. We further evaluate the effectiveness of the proposed strategy, which has not been previously explored in multi-step passenger flow prediction models.
3. We propose a novel fusion module to capture multi-scale temporal dependencies in historical data and evaluate its effectiveness in bus passenger flow prediction. This approach enables the model to learn from time-varying spatio-temporal patterns based on recent hourly, daily, and weekly passenger flow patterns. This multi-scale approach is particularly valuable in bus networks with sparse and highly fluctuating passenger flow patterns. We also conduct a comprehensive evaluation to identify the relevance and significance of different temporal patterns in the final performance of the proposed framework.
4. Given the scarcity of comprehensive cross-dataset evaluations and detailed component analyses in the literature, we evaluate the performance of the proposed framework against two large-scale datasets to demonstrate its robustness and generalizability. These evaluations include an ablation study and comparisons with popular and advanced baseline models. The results consistently demonstrate the superior performance of TMS-GNN in both datasets, underscoring its practical utility and robustness in diverse and dynamic urban transit environments.

With these contributions, this research seeks to enhance the accuracy and robustness of bus passenger flow prediction models, providing valuable insights and tools for transit operators to optimize their services and for users to improve their travel experience.

The remainder of the paper is organized as follows. In the next section, we overview the literature on passenger flow prediction. Next, we present the proposed methodology of the Traffic-Aware multistep Graph Neural Network (TMS-GNN) Model with Scheduled Sampling. Next, we discuss the case study networks and their characteristics, as well as the relationship between passenger flow and speed and delay, in the Data Description and Analysis section. Then in the Results and Discussion section, after applying the proposed model to this network, its performance is compared to popular and state-of-the-art baseline models in both overall and multistep scenarios. Moreover, the performance of the proposed model is compared in different network structures in the Ablation Studies section. The impact of other components of the model, such as the fusion module and the Scheduled Sampling method, is also examined. We conclude the study by summarizing the findings and suggesting future research topics.

## 2. Literature review

Passenger flow prediction approaches can be broadly classified into linear and non-linear methods (Zhai et al., 2018). Linear models, such as time series analysis, Kalman filtering, and linear regression, are based on the assumption of stability and linearity in the data. In contrast, non-linear models, including machine learning techniques, are designed to capture the complex, non-linear relationships often present in transportation data. In the following subsections, we overview these methods, identifying the current gaps in the literature on bus passenger flow prediction.

### 2.1. Linear methods

Linear models have been foundational in passenger flow prediction, with time series analysis methods, such as autoregressive models, being particularly popular. Researchers have applied these methods to various scenarios, from forecasting short-term flow at transportation hubs (Gu et al., 2011) to predicting inter-district travel demand (Cyril et al., 2018). More sophisticated approaches have been proposed by combining multiple time series models to capture different temporal patterns (Ma et al., 2014; Xue et al., 2015). A few studies have explored the impact of unusual temporal patterns, such as weekends (Ye et al., 2019). Kalman filtering is another popular method, which has been applied and enhanced in various ways. Chun-Hui and Song-Rui (2011) used Kalman filter for predicting the number of passengers at bus stops, while Jiao et al. (2016) proposed three improved Kalman filter models for rail transit. A real-time forecasting system was developed by Zhang et al. (2017) using an Extended Kalman Filter. Gong et al. (2014) introduced a comprehensive three-stage framework integrating seasonal ARIMA, event-based methods, and Kalman filtering for bus stop predictions.

Despite their widespread application and initial success, linear methods exhibit inherent limitations when confronted with the complex, dynamic nature of passenger flow. These approaches often struggle to capture sudden changes and non-linear patterns, particularly in short-term predictions where passenger behavior becomes more volatile and is influenced by numerous rapidly changing factors (Liu and Chen, 2017). Furthermore, the effectiveness of linear methods has been primarily demonstrated at individual stops or within limited regions. They face significant challenges when attempting to model network-wide passenger flows (Baghbani et al., 2023a). These limitations underscore the need for more capable modeling approaches that can better account for the multifaceted and nonlinear nature of transit networks.

### *2.2. Non-linear methods*

Nonlinear methods have emerged as powerful tools for addressing the complexities of passenger flow. Initial studies utilized traditional machine learning techniques, such as support vector machine (Chen et al., 2011; Sun et al., 2015), random forest and regression trees (Samaras et al., 2015); however, artificial neural networks (ANNs) became increasingly popular among nonlinear methods (Sonoda and Murata, 2017). ANNs have been utilized purely (Li et al., 2019; Tsai et al., 2009) or in combination with other methods, such as wavelet analysis (Zhao et al., 2011) and ARIMA (Teng and Chen, 2015). Despite these efforts, shallow ANNs have shown limitations in extracting complex features, modeling temporal dependencies, and handling large-scale data. Consequently, researchers have been motivated to investigate more sophisticated deep learning architectures to address these challenges in passenger flow prediction.

Liu and Chen (2017) were pioneers in utilizing deep networks by introducing a stacked autoencoder model. Subsequent studies employed a variety of architectures to capture the intricate spatio-temporal patterns in passenger flow data. Fully connected networks (Liu and Chen, 2017; Zhu et al., 2018; Gallo et al., 2019) have been used to learn complex non-linear relationships. Convolutional neural networks (CNNs) (Zhang et al., 2019; Liu et al., 2020b) have been applied to extract spatial features, while long short-term memory (LSTM) models (Hao et al., 2019; Liu et al., 2019; Lin et al., 2020; Yang et al., 2021; Nagaraj et al., 2022) have been utilized for their capacity to capture long-term temporal dependencies. Hybrid frameworks have also been developed, often incorporating attention mechanisms to focus on the most relevant features (Zhang et al., 2020b; Jing et al., 2020; Luo et al., 2020; Huang et al., 2023). These advanced architectures have demonstrated significant improvements in modeling complex spatio-temporal patterns in passenger flow data. However, a critical limitation persists: the reliance on Euclidean space for spatial correlations (Wu et al., 2020), which limits their applicability to bus networks, where spatial relationships are often defined by network topology rather than solely geographic proximity.

To address the limitations of Euclidean-based models, Graph Neural Networks (GNNs) have emerged as a promising approach for modeling complex topological structures in transportation systems (Wu et al., 2020; Rahmani et al., 2023). Initially applied to vehicular traffic flow prediction (Cui et al., 2019; Li et al., 2017; Yu et al., 2017; Zhao et al., 2019; Guo et al., 2019; Liu and Meidani, 2023), GNNs have recently been adapted for passenger flow prediction, offering a more suitable framework for capturing non-Euclidean spatial relationships in bus networks.

#### *2.2.1. Graph neural networks for passenger flow prediction*

Given their ability to capture complex spatial patterns within transit networks, Graph Neural Networks (GNNs) have naturally extended their success in vehicular traffic flow prediction to the domain of passenger flow forecasting. Han et al. (2019) developed a spatiotemporal graph convolutional neural network for predicting the passenger flows within the citywide metro network of Shanghai, China. Since then, various frameworks have been proposed to improve the performance of graph-based models: Multi-graphs and hyper-graphs have been used to model multidimensional and evolving spatial and temporal relationships (Zhang et al., 2020a; Liu et al., 2020a; He et al., 2022; Wang et al., 2021); LSTM modules have been integrated into GCN blocks to enable capturing more complex temporal dependencies (Chen et al., 2021; Ye et al., 2020; He et al., 2020; Bai et al., 2021); and attention-based models have been developed to improve the performance and flexibility of models in capturing both spatial and temporal correlations (Lu et al., 2021; Zou et al., 2024; Zeng and Tang, 2023). Although these studies have led to significant advancements in passenger flow prediction, their focus has been primarily on rail-based transport systems. Bus networks, however, present unique challenges as they coexist with road traffic and typically exhibit more sparse and highly fluctuating passenger flow values (Baghbani et al., 2023b).

Recognizing these nuances between rail-based and bus transport systems, Baghbani et al. (2023b) developed the bus network graph convolutional long short-term memory neural network model (BNG-ConvLSTM) to predict the passenger flow at the service-level of a bus network. Ma et al. (2020) aimed to exploit the multiple static spatial correlations by multi-graph fusion convolution operator, including adjacent relation, station functional zone similarity, and geographical distance. Luo et al. (2021) introduced a hashing multi-graph convolutional network to consider both physical adjacency and semantic similarities between bus stops. The mobility patterns of passengers were integrated into the prediction task to improve its accuracy by Kong et al. (2022). They construct a sharing-stop network to identify passengers with similar mobility patterns. Chen et al. (2022) developed an attention-based framework to capture both spatial and temporal correlation of historical ridership at bus stops. Zhai and Shen (2023) applied the popular diffusion convolutional recurrent neural network (DCRNN) (Li et al., 2017) for short-term bus passenger flow prediction to capture the impacts of distance decay between the stops.

To conclude, various aspects of passenger flow prediction in bus networks have been explored in the existing literature; however, there are open research areas. Firstly, an important property of bus networks, which is their overlap with vehicular traffic has not been considered and modeled. Among the limited studies that have incorporated traffic conditions into prediction tasks, such as (Cui et al., 2019), the approaches fall short by treating traffic as a static phenomenon rather than capturing the dynamic nature of traffic flow. Secondly, the focus of passenger flow prediction models has been on single-step prediction and the challenges of multi-step prediction have not been directly addressed in this domain. Thirdly, the current studies do not investigate the role of multi-dimensional temporal dependencies in the accuracy of predictions, which can play a crucial role in bus networks, considering the sparse and highly fluctuating nature of passenger flow in these systems. Finally, most of the current models are not assessed against different datasets and under extensive ablation studies, which limits the understanding of their generalizability and the relative importance of different model components in capturing the unique characteristics of bus passenger flow dynamics.

## 3. Methodology

The architecture of the TMS-GNN model is presented in Fig. 1. As can be seen, in this framework, the historical passenger flow data has been used across three different time scales: recent, daily, and weekly. Each of these time scales is processed by corresponding Bus Graph Convolution-LSTM (BGC-LSTM) modules, specifically designed to capture the patterns relevant to their specific time frames.

In these modules, initially, the input sequence data, including passenger flows and graph structures, is passed through the Bus Graph Convolution (BGC) cells. There are two types of graph structures: a static graph that represents the fixed connectivity between bus stops according to bus routes, and dynamic graphs that account for the Traffic-Aware Reachability of stops, adjusting based on real-time traffic conditions. This combination of graphs enables the model to accurately capture and represent the spatial relationships within the bus network. The output from the BGC module is subsequently processed by LSTM cells, which are responsible for capturing the temporal patterns in the data. This approach ensures that the TMS-GNN model can effectively capture the spatial and temporal features through BGC and LSTM cells, respectively.

Following this, the outputs from recent, daily, and weekly BGC-LSTM modules are integrated using the fusion module to generate the final prediction for each time step. Next, the model employs a technique known as Scheduled Sampling for multistep prediction, which enhances performance over extended prediction horizons by minimizing error accumulation during the prediction process. The iterative nature of the model is evident as it uses each prediction to inform the next, continuing this cycle until all the required number of future time steps is forecasted.

In this section, the proposed methodology is discussed in detail. First, the problem we aim to address is defined, followed by a detailed explanation of the graph structure in the proposed model. Moving forward, we explore the various modules within the TMS-GNN framework, starting with the BGC-LSTM module, which combines Bus Graph Convolution for spatial modeling and LSTM cells for temporal modeling. Then, we talk about how the different temporal historical dependencies have been captured by using the multi-component fusion module. Finally, and in the last part of this section, we discuss the incorporation of Scheduled Sampling for addressing the “exposure bias” issue in multistep prediction.

### 3.1. Problem definition: Multistep short-term passenger flow prediction

The aim is to predict bus passenger flows for all stops in the network for multiple time steps into the future by taking into account both spatial and temporal interdependencies. Concerning the spatial dependencies, we highlight the importance of both the connectivity between the stops in the bus network, as well as the distance between the stops on the road network while considering traffic conditions. Concerning temporal dependencies, the components include historical patterns of passenger flow over the past few hours, similar time spans in the previous days, and similar time spans on the same day within previous weeks.

Accordingly, the objective of the model is to establish a function $F(.)$ represented by (1), where $P_t \in R^N$ specifies a vector that represents passenger flow at all bus stops throughout the network at a specific time $t$. This function, considering spatial–temporal features, is devised to map the time steps of the historical graph of passenger flow to the subsequent graph of it in multiple forthcoming time intervals up to a prediction horizon ($T_P$).

$$F(\text{Historical Passenger Flow, Graph Structure}) = [P_{t+1}, P_{t+2}, \dots, P_{t+T_P}] \tag{1}$$

According to (1), the TMS-GNN model relies on two main inputs. First, it uses historical passenger flow data, which consists of time-series data representing the passenger flow at each bus stop in the network over past time intervals, capturing the temporal dynamics of the system. The second input is the graph structure of the bus network, including the connectivity and the Traffic-Aware Reachability matrices, which will be explained in the next section. Furthermore, the output of the TMS-GNN model is the predicted passenger flow at each bus stop across the entire network for multiple future time intervals.

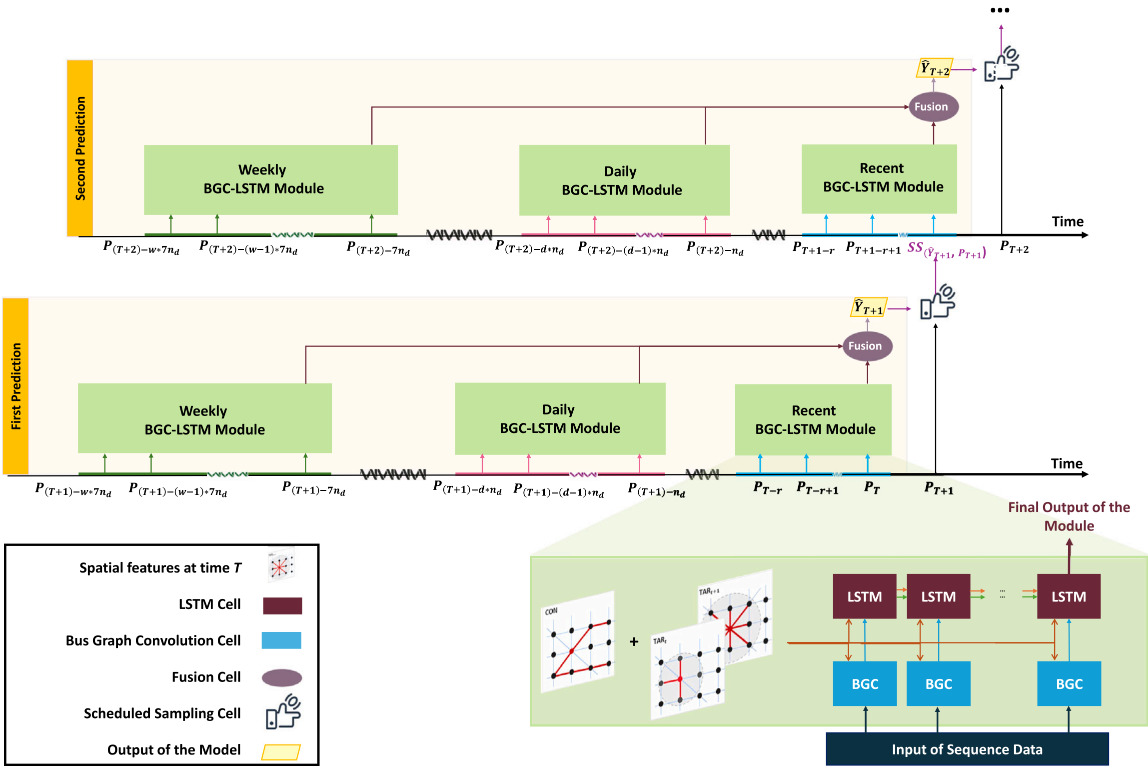



**Fig. 1.** The structure of the TMS-GNN model.

### 3.2. Building the bus network graph structure

A graph $G$ can be described as a collection of elements, known as vertices or nodes. The nodes are interconnected by relationships known as edges or links (Hamilton, 2020). In this definition, $V$ represents the set of nodes (vertices), and $E$ represents the set of edges (links). Each edge can be denoted as $(u, v) \in E$, where node $u \in V$ and node $v \in V$.

Graph networks are commonly established and visualized using an adjacency matrix. This matrix provides insight into graph structure by describing the connections between nodes. To build this matrix, nodes are arranged in a way that each row and column corresponds to a specific node. If there is an edge between the two nodes $u$ and $v$, which means $(u, v) \in E$, then $A[u, v] = 1$, otherwise, if $(u, v) \notin E$, $A[u, v] = 0$. It is worth mentioning that adjacency matrices can be also weighted, which means the values of cells can be any value based on the strength of connections between the two nodes on the graph.

Two distinct adjacency matrices are introduced in this study to define the spatial relationships among stops. The first matrix (which is called the Connectivity matrix ($CON$) hereinafter) is defined based on the sole connectivity among stops of a specific bus route, and the second one (Traffic-Aware Reachability ($TAR$) matrix) is determined based on the distance between the stops on the real road network considering the dynamic traffic situation. The term "dynamic traffic situation" refers to real-time traffic conditions, including factors such as congestion and travel times. These factors impact the value of cells within the Traffic-Aware Reachability matrix as will be discussed in Section 3.2.2.

#### 3.2.1. Connectivity matrix

The Connectivity matrix is a directed and non-weighted adjacency matrix, which aims to capture the graphical nature of the bus transport system and represents the network as the connection of stops using bus routes. This approach has been popular in other studies in the transportation and traffic forecasting domain (Han et al., 2019; Chen et al., 2021; Baghbani et al., 2023a; Rahmani et al., 2023). In this regard, if a stop on a specific bus route is situated before another stop on that route, it is considered connected to it, and the value in the Connectivity matrix from the preceding stop to the succeeding stop is set to 1. Eq. (2) shows how the Connectivity matrix is built:

$$CON_{i,j} = \begin{cases} 1, & \text{Stop } i \text{ is before stop } j \text{ in the same line} \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

#### 3.2.2. Traffic-aware reachability matrix

As was discussed in the introduction, the performance of bus services is affected by road traffic conditions since buses share parts or all of their routes with vehicular traffic, for instance on road links or at intersections. Accordingly, considering the traffic conditions and accounting for traffic congestion is essential for accurately modeling and capturing short-term passenger flow dynamics. In light of this, in addition to the Connectivity matrix, another matrix is defined, which is referred to as the Traffic-Aware Reachability henceforth.

To define this matrix, we first set a target time interval ($\Delta t_{interval}$), which serves as a threshold to determine reachability between stops. If two stops can be reached within $\Delta t_{interval}$ considering real-time traffic conditions, the cell value for those stops in the TAR matrix is equal to one. This reflects that traffic conditions allow for a potential dynamic interaction between these stops, influencing their reachability in the network at any given moment. The TAR matrix, therefore, serves as a mechanism to introduce traffic dynamics into the model, enabling it to adapt to real-time fluctuations in traffic flow. It is important to note that TAR does not directly encode static correlations but instead dynamically integrates traffic as a key factor affecting stop-to-stop interactions. This is particularly valuable for capturing the effects of traffic congestion on stops that are farther apart, where delays caused by traffic can significantly influence reachability.

The selection of $\Delta t_{interval}$ is guided by several interdependent factors, including the spatial layout of the network, the frequency of bus services, and the typical travel times between stops under varying traffic conditions. Importantly, $\Delta t_{interval}$ should be considered a hyperparameter that needs to be carefully chosen for each specific study and network. A well-chosen threshold ensures that the TAR matrix balances capturing dynamic interactions without overgeneralizing or excluding meaningful connections. This balance is critical for accurately reflecting how traffic influences the network's operational characteristics, providing the model with the ability to better learn and predict passenger flow dynamics.

The reachability between stops is further influenced by the static distances ($Dist_{i,j}$) between them and the bus speed ($S_t$). Although the distance is static, the speed of the bus is determined by the traffic situation on the network, which can fluctuate throughout the day. In this study, $S_t$, which denotes the average bus speed at time $t$, is calculated using the Automated Passenger Counter (APC) dataset. This dataset contains detailed records of the distances traveled and times taken by various buses across each time interval. Therefore, we calculate the speed for each path by dividing the distance by the travel time. Following this, we average these speeds across all paths to determine $S_t$. While $S_t$ reflects the real-time fluctuations in bus speeds at the moment of prediction, we assume that $S_t$ remains constant throughout the prediction horizon. This allows the *TAR* matrix to dynamically capture traffic conditions at the time of prediction, providing real-time insights while ensuring consistent application of $S_t$ during the entire prediction horizon. Accordingly, considering $S_t$ and also $Dist_{i,j}$ as the distance between stops $i$ and $j$, the values of the cells of the Traffic-Aware Reachability matrix for the Bus Network will be calculated as shown in (3):

$$(TAR_t)_{i,j} = \begin{cases} 1, & S_t \Delta t_{interval} - Dist_{i,j} \geq 0 \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

According to this definition, each element $(TAR_t)_{i,j}$ equals zero if the bus cannot reach from stop $i$ to stop $j$ in a given time interval $\Delta t_{interval}$ at time $t$ of day. The diagonal values in *TAR* are equal to one since every stop is self-reachable at zero amount of time.

### 3.3. Bus graph convolution LSTM (BGC-LSTM) module

The BGC-LSTM module is the core building block of the proposed TMS-GNN model and is specifically designed to capture spatial and temporal features within the bus network data. This module consists of two key components: the Bus Graph Convolution (BGC) and the Long Short-Term Memory (LSTM) cells. The BGC cell captures the spatial features by processing the connectivity and dynamic traffic data within the bus network. These spatial features are then fed into the LSTM cell, which captures temporal dependencies in passenger flow data. The upcoming subsections will provide an in-depth look at both the BGC and LSTM components within the BGC-LSTM module.

#### 3.3.1. Bus graph convolution structure: Modeling spatial dependencies

In the context of neural networks, the convolution layer is often used to extract spatial features from the input data, which may have a matrix structure that is either two-dimensional or three-dimensional. In this study, using this definition, to extract spatial information from input data represented in a bus graph structure, the Bus Graph Convolution (BGC) module has been introduced. This operation is defined as the Hadamard product of a trainable weight matrix, the Connectivity matrix, the Traffic-Aware Reachability matrix, and the historical passenger flow data. As the Traffic-Aware Reachability matrix is based on the traffic at the time of prediction, it changes at different times of the day. As a result, the BGC process is also dynamic. Using the formula in (4), the BGC is calculated as follows:

$$BGC_t = (TW_{bgc} \odot CON \odot TAR_t)P_t \tag{4}$$

where $BGC_t \in R^N$ is the extracted spatial features at time $t$. Eq. (4) consists of $TW_{bgc} \in R^{N\times N}$ as a trainable weight matrix, $CON$ as the Connectivity matrix, $TAR_t$ as the Traffic-Aware Reachability matrix at time $t$, and $P_t \in R^N$ as the vector of passenger flows for all stops in the network at time $t$. Also, $\odot$ is representing the Hadamard product operator.

#### 3.3.2. LSTM model structure: Modeling temporal dependencies

After the spatial modeling step using BGC cell, the extracted features will be imported into the temporal modeling cell. The backbone of the temporal module is comprised of several LSTM cells. LSTM is a type of recurrent neural network (RNN) architecture that addresses the problem of vanishing gradients encountered in traditional RNNs. They do so by incorporating a complex memory cell known as the "cell state". This internal memory enables LSTM cells to store and remember information over a prolonged period of time in the bus network data. This capability allows LSTM to learn patterns and relationships across multiple time steps effectively. Each LSTM cell has three interacting gates that control the flow of information through the LSTM cell in different ways. The gates are defined as follows:

*Input Gate*: The input gate determines which values should be added to the cell state from the sequence input. A sigmoid activation function is used to determine the importance of each input. Eq. (5) shows how the input gate is calculated:

$$ig_t = \sigma_g \left( TW_{ig} \cdot BGC_t + U_{ig} \cdot h_{t-1} + b_{ig} \right) \tag{5}$$

*Forget Gate*: By using a sigmoid activation function, the forget gate determines what information of the inputs should be discarded from the cell state (6).

$$fg_t = \sigma_g \left( TW_{fg} \cdot BGC_t + U_{fg} \cdot h_{t-1} + b_{fg} \right) \tag{6}$$

*Output Gate*: The output gate decides what information from the cell state should transfer to the hidden state. A hidden state represents the information encoded from just a previous time step. Output gate is calculated using (7):

$$og_t = \sigma_g \left( TW_{og} \cdot BGC_t + U_{og} \cdot h_{t-1} + b_{og} \right) \tag{7}$$

In (5) to (7), "·" has been used to represent the matrix multiplication operations. Moreover, $TW_{ig}$, $TW_{fg}$, and $TW_{og} \in R^{N\times N}$ are weight matrices that have been used to map graph convolution outputs to the defined gates. There is also another group of weights including $U_{ig}$, $U_{fg}$, and $U_{og} \in R^{N\times N}$ which are used for the preceding hidden state. Further, $b_{ig}$, $b_{fg}$, and $b_{og} \in R^{N}$ are used as bias vectors. Finally, $\sigma_g$ shows the gate activation function, which in this study is a sigmoid function.

Moreover, to define cell state, we should first consider that in this study, each bus stop is affected by the previous states of both itself and its neighbors, and we need to store information from both sources within the cell state. Therefore, first, and in order to store information regarding each stop itself, we use the input cell state, calculated using (8):

$$\tilde{C}_t = \tanh \left( TW_c \cdot BGC_t + U_c \cdot h_{t-1} + b_c \right) \tag{8}$$

In this equation, the weight matrix that is used to map the outputs of bus graph convolution cells to the input cell state is shown by $TW_C \in R^{N\times N}$, and $U_C \in R^{N\times N}$ is the weight matrix that corresponds to the previous hidden state. In addition, $b_C$ is the bias, and $tanh$ is the hyperbolic tangent function. Second, to store information regarding the neighbors of each stop in the graph, we define a cell state gate which is calculated by using (9)

$$C^*_{t-1} = TW_N \odot \left( CON \odot TAR_t \right) \cdot C_{t-1} \tag{9}$$

The value of $TW_N$ measures the impact of neighboring cell states by multiplying the Connectivity matrix and the Traffic-Aware Reachability matrix. Lastly, the final cell state $C_t$ and hidden state $h_t$ are defined as (10) and (11) respectively:

$$C_t = fg_t \odot C^*_{t-1} + ig_t \odot \tilde{C}_t \tag{10}$$

$$h_t = og_t \odot \tanh(C_t) \tag{11}$$

The output of the LSTM cells ultimately will be the hidden state $h_T$ at the last time step $T$ of the pattern in BGC-LSTM module, and that is the predicted value $\hat{y} = h_T$. Fig. 2 shows the structure of the LSTM cell in this study and how the output of the BGC step is used as the input of the LSTM cell.

### 3.4. Multi-component fusion module for historical trends capturing

The TMS-GNN model employs a multi-component fusion module that integrates BGC-LSTM modules, each dedicated to processing recent, daily, and weekly historical data. This module allows the model to effectively capture both short-term fluctuations and long-term periodic trends, resulting in more accurate and reliable passenger flow predictions. In this section, we begin by explaining how the temporal data components are constructed, followed by a detailed discussion of the fusion process.

#### 3.4.1. Temporal data components

To understand how these temporal patterns are defined and integrated within the model, it is important to understand the role each plays in the prediction process, as explained below. In this context, $t$ represents the current time step, and $t+1$ is the next time step for which the prediction is made. For simplicity, the explanation below focuses on single-step prediction; the extension to multistep prediction, using the Scheduled Sampling technique, is covered later in Section 3.5.

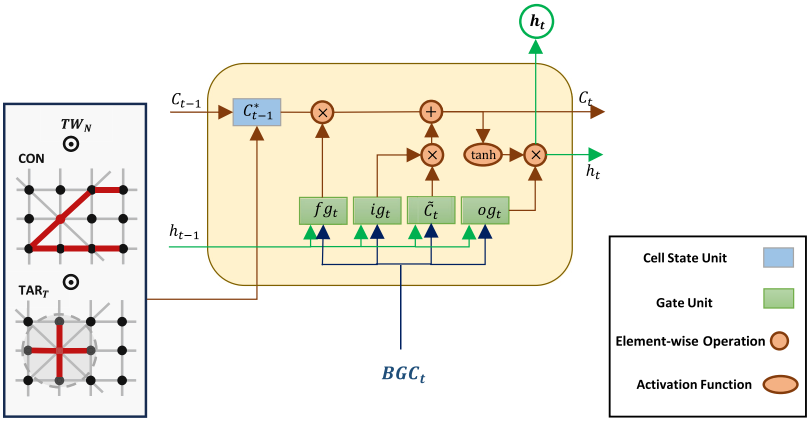



**Fig. 2.** The structure of an LSTM cell.

1. **Recent Data:** This data plays a crucial role in capturing the most immediate trends in passenger flow, accounting for short-term fluctuations caused by events or irregularities. This data is essential for adapting predictions to current conditions, as passenger flows at time $t+1$ often closely follow those from the recent past (Cui et al., 2019; Du et al., 2019; Han et al., 2019). The recent data sequence is represented as:

$$(P_{recent})_{(t+1)} = \{P_{(t-r)}, P_{(t-r+1)}, \dots, P_{(t)}\} \tag{12}$$

Here, $r$ indicates the number of previous time steps included.

2. **Daily Periodic Data:** Daily periodic data plays a key role in capturing recurring patterns within a day, such as peak passenger flows during morning and evening commutes. These patterns are vital for recognizing daily routines, which generally tend to occur at the same time each day. The daily data sequence is expressed as:

$$(P_{daily})_{(t+1)} = \{P_{(t+1-d*n_d)}, P_{(t+1-(d-1)*n_d)}, \dots, P_{(t+1-n_d)}\} \tag{13}$$

Here, $d$ indicates the number of days considered, and $n_d$ is the number of time intervals within each day.

3. **Weekly Periodic Data:** The proposed framework also includes segments at the same intervals as the prediction period on the same day of the week in the last few weeks. This component enables the model to account for long-term periodic trends as there may be some travel patterns that are unique to one particular day. For example, people may tend to do their shopping or recreational trips on certain days of the week and their business trips on other days. This means on a particular day of the week, for example, Friday, the travel behavior is likely to be similar to the pattern on the previous Fridays. As a result, the weekly periodic component of the passenger flow has been added to the model structure in order to capture these patterns of the data. The weekly data sequence is defined as follows:

$$(P_{weekly})_{(t+1)} = \{P_{(t+1-w*7n_d)}, P_{(t+1-(w-1)*7n_d)}, \dots, P_{(t+1-7n_d)}\} \tag{14}$$

Here, $w$ indicates the number of weeks considered.

### 3.4.2. Fusion process

Each of these temporal sequences – recent, daily, and weekly – is the input and processed through a dedicated BGC-LSTM module, which captures the spatial and temporal dependencies specific to that time scale. The outputs from these modules – denoted as $\hat{y}_{r(t+1)}$ for recent, $\hat{y}_{d(t+1)}$ for daily, and $\hat{y}_{w(t+1)}$ for weekly data – are then fused to produce the final prediction. The *TAR* matrix used for all of these moments is equal to the matrix at the time of prediction.

The fusion process is implemented by a two-layer convolutional neural network that uses Relu activation function, which integrates the information from all three time scales:

$$\hat{Y}_{t+1} = W_2 * RELU(W_1 * [\hat{y}_{r(t+1)} | \hat{y}_{d(t+1)} | \hat{y}_{w(t+1)}]) \tag{15}$$

Here, | denotes concatenation, and $W_1$ and $W_2$ are trainable weights that the model learns during training. This fusion mechanism ensures the model can dynamically adjust the impact of recent, daily, and weekly patterns, resulting in accurate and robust predictions. Importantly, the BGC-LSTM modules and the fusion mechanism are not trained independently but as integral components of the overall TMS-GNN framework. All components, including the BGC-LSTM modules, the fusion process, and the scheduled sampling mechanism, are trained in an end-to-end manner using backpropagation. The loss function is computed after the fusion step, as explained in detail in Section 3.6, ensuring that gradients are propagated through all components of the model simultaneously. This unified optimization strategy updates the weights across the BGC-LSTM modules and the fusion mechanism, allowing the network to dynamically learn how best to integrate recent, daily, and weekly patterns rather than treating them as independent features.

Fig. 1 visually represents, for each step of prediction, how recent, daily, and weekly data are processed through BGC-LSTM modules and then fused to form the final prediction. This integrated design enables the model to learn from diverse temporal patterns collaboratively, rather than in isolation.

### 3.5. Multistep prediction and scheduled sampling

The previous section discussed how we predict the passenger flow for the first time step. This section explains how the prediction time horizon can be extended into multiple future time steps. Assume that $t$ represents the current time, and we aim to predict the passenger flow over the next $p > 1$ time periods. Based on what was mentioned in the previous section, we need to build the recent, daily, and weekly periodic data in each prediction step and import them to the BGC-LSTM module. Accordingly, assume that we want to predict the value of passenger flow at $t + 2$ (then we can extend the methodology for the prediction of passenger flow after this time step). In order to build the daily and weekly datasets, according to (13) and (14), we need to look at the same time interval ($t + 2$) in the previous days (for daily periodic data) and the same day in the previous weeks (for weekly periodic data). These datasets can easily be built using historical data from the past days and weeks. Eqs. (16) and (17) are showing the values of them:

$$(P_{daily})_{(t+2)} = \{P_{(t+2-d*n_d)}, P_{(t+2-(d-1)*n_d)}, \dots, P_{(t+2-n_d)}\} \tag{16}$$

$$(P_{weekly})_{(t+2)} = \{P_{(t+2-w*7n_d)}, P_{(t+2-(w-1)*7n_d)}, \dots, P_{(t+2-7n_d)}\} \tag{17}$$

Based on the (12), the recent dataset also should have the structure like (18):

$$(P_{recent})_{(t+2)} = \{P_{(t+2-r)}, P_{(t+2-r+1)}, \dots, P_{(t)}, P_{(t+1)}\} \tag{18}$$

However, from (18), it can be seen that for predicting the passenger flow at $P_{(t+2)}$, the value of passenger flow at $P_{(t+1)}$ is needed. Accessing the $P_{(t+1)}$ value in the training phase is not an issue since we have the ground truth value for $P_{(t+1)}$ from the historical data. However, in the test phase, we do not have such a value from the ground truth data, and we only have access to the latest predicted value for $t + 1$ ($\hat{Y}_{(t+1)}$).

This inconsistency between the training and testing phases results in a mismatch between the two phases and can cause propagations of errors through prediction steps in the test phase. This is because if the predicted value for one step is accompanied by an error, it will be used as the input for the next step, and this error will be added to the error of the next step. This propagation of errors amplifies as the number of steps increases. To solve that, one solution is to use $\hat{Y}_{t+1}$ instead of $P_{(t+1)}$ during the training phase (similar to the testing phase). However, this strategy may not work because the propagation and accumulation of errors will then happen during the training phase.

To solve this issue, Bengio et al. (2015) proposed a curriculum learning strategy, called Scheduled Sampling, for sequence Prediction. This method aimed at bridging the gap between the training and testing phases in sequence prediction by utilizing a combination of the ground truth data and predicted values as the input for the model during the training phase. In Scheduled Sampling, the ground truth data is used in the initial stages of the training process but will be gradually replaced by the predicted values as the training phase progresses. Using this method, the time series data segment for the second (and subsequent) step of the prediction process will be built utilizing a weighted combination of both ground truth in the training phase and predicted value in the testing phase. In this study, considering $i$ as the number of the current epoch, we use the ground truth with a probability of $\epsilon_i$ and the prediction from the model with a probability of $(1 - \epsilon_i)$ for building the recent periodic dataset in the training phase. In this way, if the $\epsilon_i = 1$, the model will be trained using only the ground truth value, and if $\epsilon_i = 0$, it will be trained like the testing phase and will use only the predicted value.

Therefore, we will have the following recent sequence data as the input for predicting the second timestamp of the prediction period, which will be used in the fusion equation with daily and weekly datasets (15):

$$(P_{recent})_{(t+2)} = \{P_{(t+1-r)}, P_{(t+1-(r+1))}, \dots, P_{(t)}, (\epsilon_i P_{(t+1)} + (1 - \epsilon_i)\hat{Y}_{(t+1)})\} \tag{19}$$

This process will be repeated for the next prediction time steps. We share the parameters for all fusion components during multistep prediction, to prevent the model from over-smoothing. Different functions could be used to define how $\epsilon$ is changing over time as the training phase progresses. In this study, we define $\epsilon_i = k^i$ where $k < 1$.

Fig. 1 illustrates how Scheduled Sampling is embedded in the proposed framework. As is shown, after the outputs of BGC-LSTM sequence modules have been fused to make the passenger flow prediction for the first time step ahead (the yellow parallelogram), this prediction will be passed through the Scheduled Sampling module together with the ground truth input for the same time interval ($P_{T+1}$). These two inputs are combined using a dynamic weighting strategy as explained. Then, the output of the Scheduled Sampling cell will be used in making the recent periodic sequence to calculate the passenger flow for the second time step ahead. These steps will be repeated for each time step prediction.

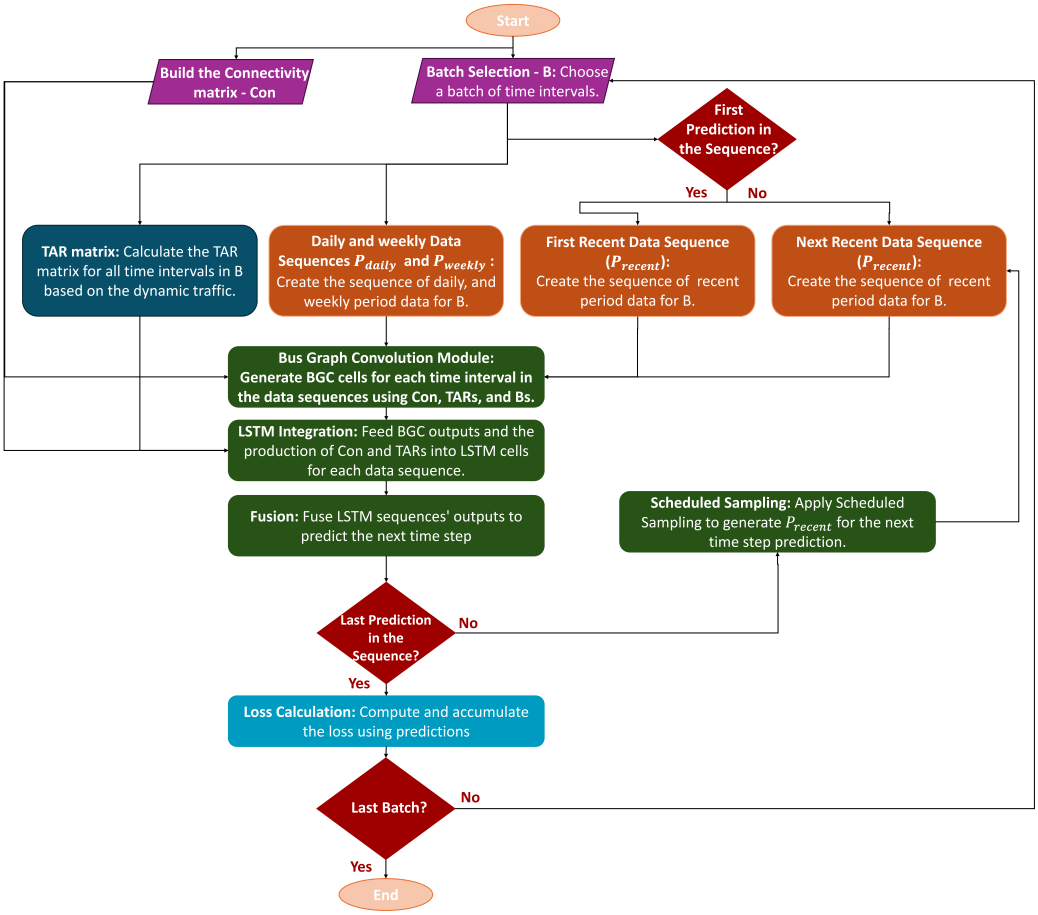



**Fig. 3.** The training procedure of the TMS-GNN model.

### 3.6. Loss function and training algorithm

After going through all the steps of prediction, the loss will be calculated using (20). In this equation, $p$ is considered as the number of prediction steps. The $L(.)$ term in this equation also represents the Mean Square Error (MSE), which is used to determine the residual between the predicted $\hat{Y}_T$ and actual values $Y_T$.

$$Loss = \sum_{n=1}^{p} L(Y_{(T+n)}, \hat{Y}_{(T+n)}) \tag{20}$$

The loss function is applied to the model's final predictions, optimizing the entire TMS-GNN framework in a unified manner. This includes the BGC-LSTM modules, the multi-component fusion module, and subsequent prediction layers. By applying the loss after the fusion and scheduled sampling steps, the entire pipeline is jointly trained, ensuring that each component contributes effectively to passenger flow forecasting. To have an overall overview of the proposed framework and the training process, the training algorithm is shown in Fig. 3. Before starting the training steps, the Connectivity matrix is built. The training algorithm begins with the Batch Selection, where a specific batch of time intervals is chosen. Using this batch, the *TAR* matrix is then calculated to reflect the dynamic nature of traffic within these intervals. At the same time, the first recent data sequence is generated, and after that, daily and weekly data sequences are generated for the selected batch. Following this, to capture spatial dependencies, the Bus Graph Convolution modules are calculated for each element of the sequences, and their outputs are imported into the LSTM cells. In the next step, the outputs from the LSTM cells for each sequence are fused to predict passenger flow for the next time step. After that, the Scheduled Sampling method is employed to create the recent period data sequence for subsequent time step predictions. The algorithm continues by iterating for the future time steps till it reaches the final step of the prediction. After completing the set of predictions, the Loss Calculation step computes the difference between the model's predictions and the actual data. The entire process will then be repeated for each batch.

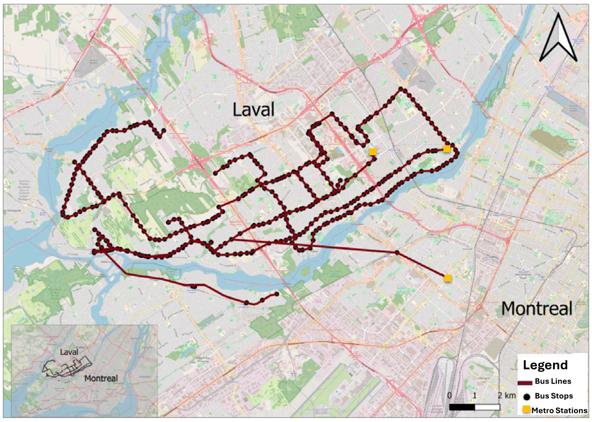



**Fig. 4.** The modeled network of bus stops — Laval, Canada.

## 4. Data description and analysis

Two datasets have been used in this study: APC data from Laval, Canada, and Ames, Iowa, USA. This section primarily leverages the Laval dataset to examine passenger flow dynamics, while also providing some insights from the Ames dataset. First, it examines the network's complexity and then investigates the correlation between traffic conditions, bus delays, and passenger flow fluctuations. Following that, it explores how bus network reachability fluctuates with daily traffic conditions, and finally, by analyzing daily, weekly, and monthly passenger flow trends, it attempts to understand the patterns of passenger flow within an urban bus network over time.

### *4.1. Datasets description*

As has been mentioned, in this study the two datasets from different and distinct geographic locations – Laval, Canada, and Ames, Iowa, USA – have been used to evaluate the effectiveness and generalizability of the proposed TMS-GNN model. Each dataset is organized as a graph, where the bus stops are represented as nodes, and the node feature is the passenger flow at each stop. The following are the details of each dataset:

**Laval Bus Network Dataset:** The primary dataset used for this study is from APC data collected in Laval, Greater Montreal area, Canada. Montreal is the second-largest city in Canada. One month of data is used, from mid-September to mid-October 2021. Laval has only three metro stations, and its public transportation system relies primarily on buses, making it an appropriate network for this study. In this study, we select an important and complex subset of the bus network in Laval, Canada. Fig. 4 provides a depiction of the network, highlighting its complex graph-structured nature. The bus lines are represented by red lines, and the stops are marked with dark dots, showing the intricate web of connections, which form the public transportation system in this area. The network on this map emphasizes that the spatial arrangement of the stops and the bus routes between them form a non-Euclidean graph-like topology, where passengers can only travel between two stops if those stops are connected via a route, regardless of their spatial distance. From another perspective, the number of passengers that are traveling between two stops can be regarded as information passing through the links of the graphs. The passenger flow at each bus stop, which forms the node features, is influenced by the number of passengers traveling to that stop (information passing) and those boarding and alighting at the stop. Therefore, modeling and predicting passenger flows using a graph-structured model seems intuitive. Further, another challenge of bus passenger flow prediction is that this passenger flow at each stop is itself affected by the arrival time of the bus, meaning that if the bus is arriving with some delay, the number of passengers is expected to accumulate, and therefore, the passenger flow at each node can be different from normal conditions. These delays may happen for several reasons, with traffic conditions and congested roads being the most important.

Table 1 displays some characteristics of the dataset, including the route number and its direction, the number of stops on each route, as well as the number of passengers. In total, there are 467 stops and 11 routes in this dataset. It is worth mentioning that only observations between 05:00 and 23:00 have been considered since bus routes and stops differ at midnight from those during the day. In addition, passenger counts are aggregated at 10-min time intervals. To generate the training and test datasets, 70% of the dataset was used for training, 20% for validation, and 10% for testing.

**Ames Bus Network Dataset:** In addition to the Laval dataset, we also used a dataset from the Ames bus network in Iowa, USA, to evaluate the generalizability and robustness of the proposed TMS-GNN model. This dataset, which is open-source and was collected in October 2022, provides a detailed snapshot of the bus network in Ames (Wilbur, 2023). Unlike Laval, Ames has distinct geographic and demographic characteristics. It has a dense population which is largely influenced by the presence of Iowa State University. The difference is evident in the higher number of bus stops (515) and the higher passenger volume (419,124 total passengers), despite

**Table 1**
A description of data used for the modeled network (Laval — September–October-2021).

| Route number and direction | No. of stops | No. of passengers |
|---|---|---|
| 26 - Westbound | 62 | 48,656 |
| 24 - Westbound | 48 | 21,622 |
| 46 - Northbound | 66 | 14,638 |
| 903 - Northbound | 56 | 9,227 |
| 60 - Eastbound | 61 | 7,581 |
| 66 - Westbound | 52 | 7,240 |
| 20 - Westbound | 56 | 6,763 |
| 36 - Westbound | 39 | 3,097 |
| 730 - Northbound | 6 | 1,044 |
| 730 - Southbound | 7 | 890 |
| 744 - Westbound | 14 | 385 |
| Total | 467 | 121,143 |

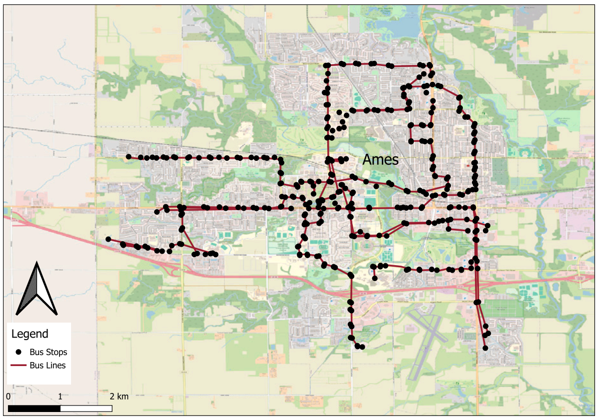



**Fig. 5.** The modeled network of bus stops — Ames, Iowa.

covering a smaller geographic area. Fig. 5 visually represents this bus network, illustrating the spatial distribution and connectivity of the routes and stops.

Table 2 also shows detailed information about the network, including the number of stops and passengers for each route. The Ames dataset captures data from 18 bus routes, with operational hours focused between 08:30 and 19:00, ensuring a stable network period for analysis. For this dataset too, for data preparation, 70% of it was used for the training phase, 20% for the validation phase, and 10% for the testing phase. In addition, passenger counts are aggregated at 10-min time intervals, allowing for precise tracking of passenger flow throughout the day.

### *4.2. Bus speed and delay analysis on passenger flows*

As buses share their routes with vehicular traffic, traffic conditions can have significant impacts on the performance of bus services. Moreover, traffic congestion can lead to longer travel times for bus passengers, uncertainty in bus schedules, reduction in user comfort, and so on. Therefore, it might also indirectly influence the choices of passengers, and thus affect the number of passengers across the bus transport system. The purpose of this section is to examine how delays and speed variations, which reflect traffic conditions in the network, can influence the passenger flow numbers in the network.

In order to explore the effects of delay and bus arrival reliability on passenger flow values, arrival delays of buses for all stops in the network in the Laval dataset are calculated for one day as a sample. These values are aggregated per hour in the form of box plots. Then, these box plots are depicted against hourly passenger flow values (Fig. 6). The larger interquartile ranges (IQR) in box plots are an indication of bus arrival unreliability. As can be seen in Fig. 6, when there is more delay and unreliability in bus arrivals, higher values of passenger flow have been observed. In addition, the Pearson correlation coefficient between hourly mean delays and hourly passenger flows is approximately 0.83, which supports the observations based on the box plots in Fig. 6, where greater passenger flow was associated with greater delays and unreliability in bus arrivals.

Moreover, the scatter plot in Fig. 7 shows the relationship between the 10-min average speed of arriving buses at different stops and the number of passengers recorded at those stops in the Laval bus network in the same period. The data points in this figure are colored according to their monthly average speed. The monthly average speed is an indication of the prevalent speed of buses at that stop, and the 10-min average speed is an indication of short-term fluctuations in the speed of buses at the same stop. With this,

**Table 2**
A description of data used for the modeled network (Ames, Iowa — October 2022).

| Route number and direction | No. of stops | No. of passengers |
|---|---|---|
| 4528 | 43 | 27,584 |
| 4529 | 42 | 28,380 |
| 4530 | 41 | 8,537 |
| 4531 | 42 | 9,410 |
| 4532 | 33 | 47,344 |
| 4533 | 22 | 2,280 |
| 4534 | 50 | 12,693 |
| 4535 | 43 | 14,950 |
| 4536 | 21 | 3,871 |
| 4538 | 31 | 12,728 |
| 4539 | 32 | 33,438 |
| 4540 | 34 | 2,085 |
| 4541 | 17 | 118,900 |
| 4542 | 18 | 27,108 |
| 4543 | 15 | 26,459 |
| 4570 | 16 | 36,408 |
| 4571 | 15 | 6,949 |
| Total | **515** | **419,124** |

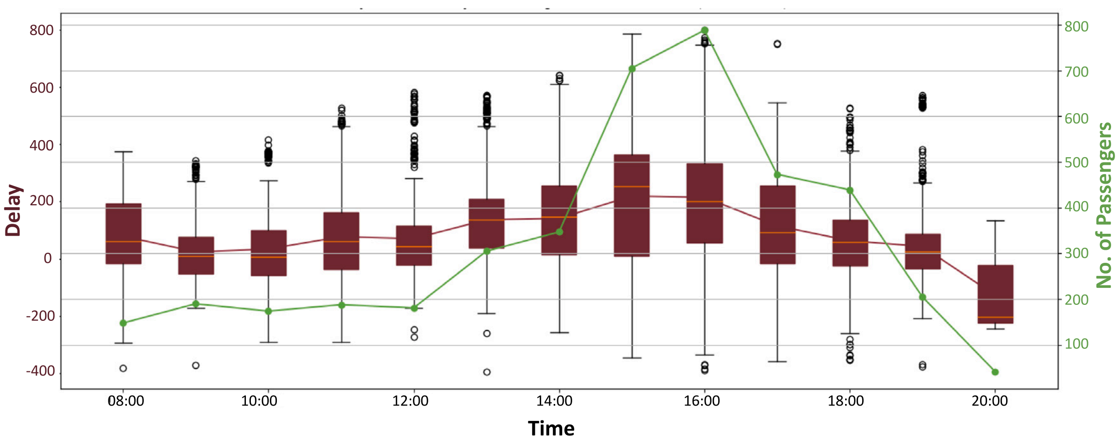



**Fig. 6.** Delay mean/variability and number of passengers in the Laval bus network for 2021-09-10.

any deviation (delays or early arrival) from the monthly average speed can be tracked. As observed from the data points, there is an inverse relationship between the number of passengers recorded and the recorded short-term speed for buses with similar monthly average speeds (points with similar colors). In other words, for the services with the same monthly average speed, as the short-term speed of the service is reduced (due to traffic congestion), the number of passengers recorded at stops served by those services shows an increase, and vice versa. This trend confirms that delays in bus services caused by slower traffic speeds significantly affect passenger flows. It underscores the importance of considering real-time traffic conditions when predicting passenger flows in bus networks, especially due to their overlap with vehicular traffic networks. Unlike static models such as the one proposed by Cui et al. (2019), our TMS-GNN model incorporates the dynamic Traffic-Aware Reachability (TAR) matrix to account for these fluctuations. The real-time adaptability of our model reflects an important difference from static traffic prediction methods that fail to capture such changes. This is a unique characteristic of the bus network, which is not pertinent in rail-based transportation systems that do not share their routes with other modes of travel.

In summary, these findings not only highlight the critical role of traffic conditions in determining passenger flows but also validate the necessity of using dynamic *TAR* matrices in our predictive model. By capturing these real-time traffic impacts, the TMS-GNN model can provide more accurate and actionable predictions for transit agencies, enabling them to better manage and respond to fluctuations in passenger demand.

### *4.3. Dynamic traffic conditions and different traffic-aware reachability matrices*

This subsection aims to investigate the intricacies of reachability within the bus network and its responsiveness to changing traffic conditions over the day. Fig. 8 presents a series of *TAR* matrices for different times of a sample day in the Laval bus network: early morning (05:00), morning peak (08:00), afternoon peak (17:00), and evening (20:00). Each matrix represents whether a specific

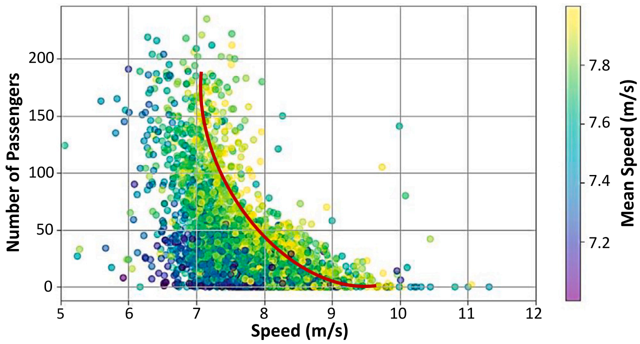



**Fig. 7.** The relationship between number of passengers and speed in the Laval bus network.

stop is reachable from other stops in the network. The matrices use a color-coding scheme where the dark blue indicates that a bus stop (represented by the numbers on the axes which are the stop IDs) is reachable from another stop within the network at the specified time, and the white color indicates that the stops are not reachable in that time span. These visualizations capture the dynamic nature of the network's connectivity, which varies with the time of day. The variation in the patterns of reachability suggests that as the day progresses, the accessibility of certain stops from others changes, potentially due to fluctuations in traffic conditions and travel speeds. In other words, the temporal changes in the reachability matrices underscore the complexity of public transportation systems, where the concept of reachability extends beyond simple physical distance to incorporate time-dependent variables such as traffic flow. During peak hours, as seen in the 08:00 and 17:00 matrices, there is a noticeable shift in reachability, likely due to increased congestion affecting bus travel speeds and schedules. In contrast, the matrices for the early morning and evening hours show higher reachability, implying less congestion and higher speeds in the network. These insights highlight the critical importance of considering variable traffic conditions when analyzing bus network dynamics and modeling passenger flows. Understanding these patterns is vital for transit planning, as it can lead to more accurate predictions of passenger numbers and better optimization of service routes to enhance efficiency and reliability.

Moreover, to better visualize the concept of reachability in the network and highlight its importance, Fig. 9 presents a visual analysis of accessibility within the Laval bus network, showing which stops can be reached from a particular origin stop at two different times. The origin stop is marked by an orange dot, while the red dots represent reachable stops, illustrating the extent of the network's coverage from the origin stop. Other stops within the network that are not reachable by the origin stop are indicated with gray dots. The shaded areas represent the zones of reachability from the origin stop, revealing how the network's accessibility pattern changes between the two times. As observed, in a non-rush hour, almost all stops in the network can be reached within a specified time, but during rush hour, this reachability is significantly different and more limited. This can affect the interdependencies between short-term passenger flows since in congested hours, the flow of passengers in the network is limited compared to less congested hours.

### *4.4. Daily, weekly, and monthly trends analysis in dataset*

To comprehensively understand the temporal patterns of passenger flow within the urban bus network, this section presents an analysis of the fluctuations in passenger flows over different time frames: daily, weekly, and monthly. As an example, we again consider the Laval bus network and Fig. 10(a) to Fig. 10(c) show the trend of passenger flows for the entire network over a specific day, week, and month within the studied period. Fig. 10(a) represents the passenger flow for October 14, 2021, in the period of 5:00 to 23:00. According to this figure, in general, it is more probable that adjacent time slots have similar values than farther ones, which indicates correlations between adjacent time slots and justifies using the most recent historical data as one of the temporal input sequences. Moreover, Fig. 10(b) shows the passenger flow over a week (2021-09-13 to 2021-09-19) which clearly illustrates repetitive trends, particularly on the weekdays and weekends. Similarly, repetitive trends can be found in Fig. 10(c), which displays the passenger flows on different days of the month. These figures and the identified trends within them indicate that passenger flow values show correlations across various time periods, which is consistent with the assumption made in the proposed TMS-GNN model and justifies the choice of recent, daily, and weekly temporal sequences. However, choosing the right number of intervals in the historical data is important for the model performance. Choosing very short sequences of historical data might lead to ignoring the historical patterns while choosing very long sequences of historical data can import irrelevant patterns into the model. Accordingly, based on observed patterns in the current dataset, different lengths of sequences were tested and the sequences that led to the best model performance were chosen as the final ones. As a result, for both case studies, the number of time intervals for the recent sequence is set to 12 ($r = 12$), which means two hours of historical data just before the target time slot are considered as the recent historical sequence. Also, three adjacent days before the target day, for which we are predicting the passenger flow values, are used to construct the daily historical sequence ($d = 3$). Similarly, the values of passenger flows on the same day of the target day in the last three weeks are considered for building the weekly historical sequence ($w = 3$). As was discussed in the methodology section, these historical sequences are then fused based on learnable weights within the proposed framework to let the model learn the importance of each of these temporal patterns for the final prediction.

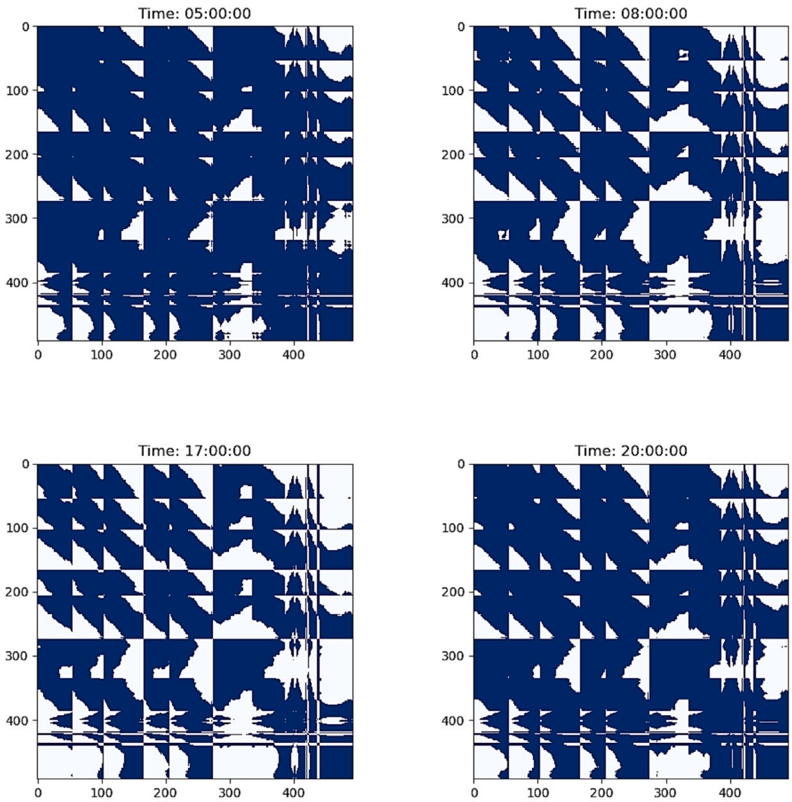
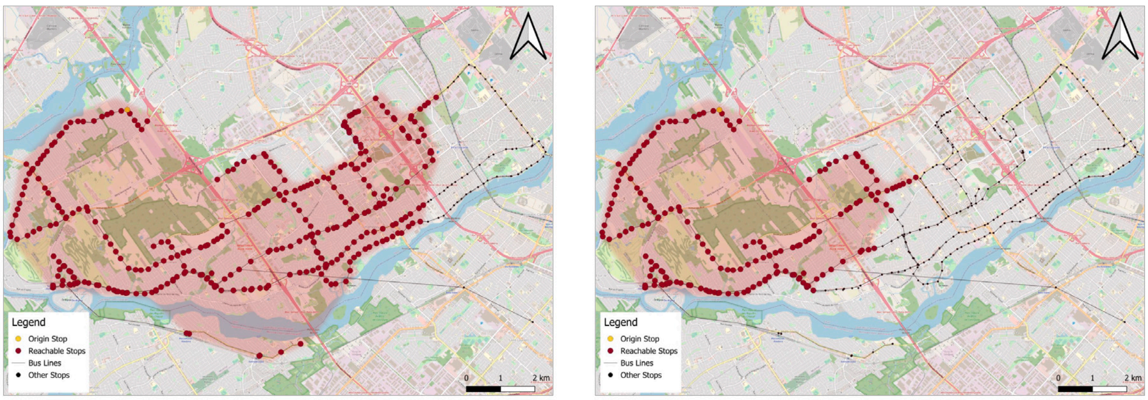



**Fig. 8.** Different *TAR* matrices during the day.

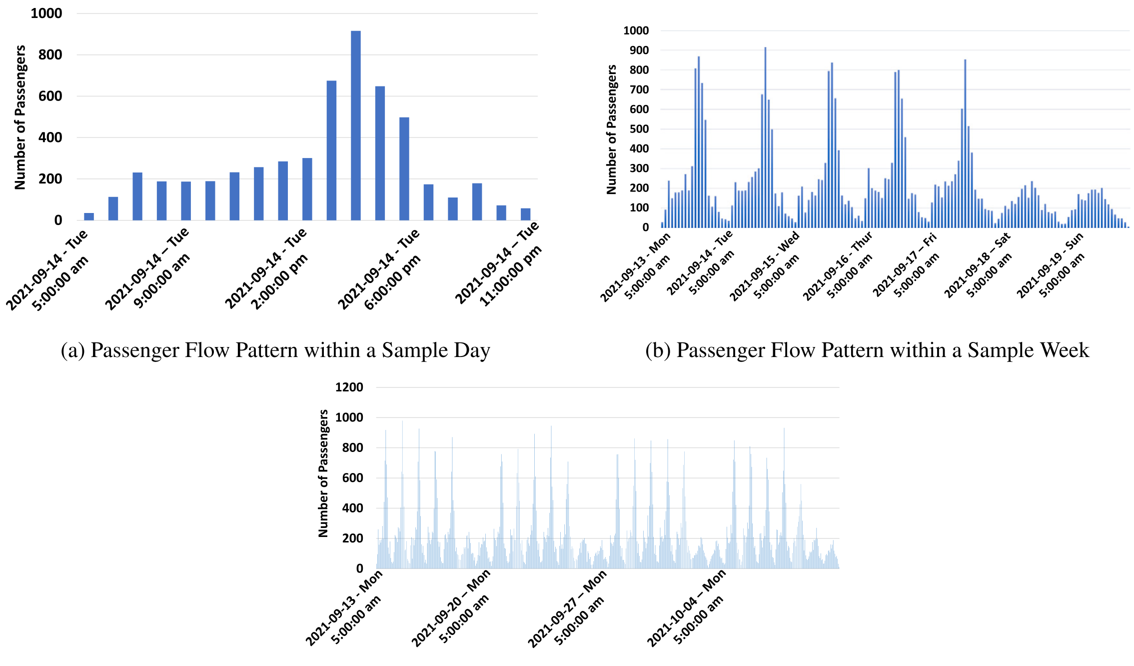



(a) Reachable Stops from a Particular Stop in a Non-Rush Hour (b) Reachable Stops from a Particular Stop in a Rush Hour

**Fig. 9.** Network accessibility at different time of day from a particular stop.

## 5. Results and discussion

This section includes the set of experiments designed to evaluate the performance of the proposed model in predicting passenger flow values for the Laval and Ames bus networks. This includes performance evaluation for both single and multistep prediction through comparison with popular baselines in the literature, the ablation study to verify the contribution of different proposed modules within TMS-GNN, as well as providing practical insights and actionable recommendations.




(a) Passenger Flow Pattern within a Sample Day

(b) Passenger Flow Pattern within a Sample Week

(c) Passenger Flow Pattern within a Sample Month

**Fig. 10.** The trend of passenger flow over different time periods.

### 5.1. Experimental design

In this section, the experimental setting and results for evaluating the proposed framework are discussed.

#### 5.1.1. Experimental setting and baseline models

We compare the performance of the TMS-GNN with several traditional and deep learning-based models that have been widely used to predict passenger flows. These models include:

- **HA:** This simple model predicts future values based on the average of past observations, providing a basic baseline for comparison.
- **MLP:** A fully connected feedforward neural network that learns patterns in the data by passing input through multiple layers of neurons which is equal to the number of nodes.
- **LSTM:** A type of recurrent neural network specifically designed to capture long-term dependencies in sequential data, making it effective for time series prediction.
- **STGCN:** This model incorporates Graph Convolutional Networks (GCN) for capturing spatial dependencies and Temporal Convolutional Networks (TCN) for capturing temporal patterns (Yu et al., 2017).
- **Graph WaveNet:** This model integrates graph convolution with dilated causal convolution, enabling it to effectively capture both local and global temporal dependencies in graph-structured data (Wu et al., 2019).
- **DDSTGCN:** This model utilizes Gated Temporal Convolutional Networks (Gated-TCN) for temporal modeling, coupled with GCN for spatial modeling, providing a robust approach for capturing dynamic spatial–temporal data (Sun et al., 2022).
- **DCRNN:** This model combines diffusion convolution for spatial modeling with a sequence-to-sequence architecture to capture temporal dependencies, making it highly effective for spatio-temporal forecasting tasks (Li et al., 2017).
- **STAEformer:** The STAEformer model uses vanilla transformers along both the temporal and spatial axes. It begins with an embedding layer, followed by transformers that operate as temporal and spatial layers, and concludes with a regression layer. This architecture allows it to effectively capture complex dependencies across both dimensions (Liu et al., 2023).

Furthermore, and as an experimental setting, in the training of the TMS-GNN model, RMSProp is used as the gradient descent optimization algorithm for the model, because it overcomes exploding and vanishing gradient problems. Mean squared error (MSE) is used for the loss function, and the initial learning rate used was $10^{(-5)}$. The batch size is set to 32 and the hidden layer size is equal to the number of nodes. We also implemented early stopping during training, with a patience of 10 epochs, to monitor performance on a validation set. This approach pauses training when improvements cease, preventing the model from becoming overly complex and ensuring it focuses on learning the most relevant patterns.

### 5.1.2. Evaluation metrics

The performance of the models in this study was evaluated using Mean Absolute Error (MAE) and Root Mean Square Error (RMSE). These metrics are calculated using (21) and (22).

$$MAE = \frac{1}{n}\sum_{T=1}^{n}|y_T - \hat{y}_T| \tag{21}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{T=1}^{n}(y_T - \hat{y}_T)^2} \tag{22}$$

In these equations, $n$ is the number of observations for each stop in the data set, $y_T$ is the actual value, and $\hat{y}_T$ represents the predicted value by the model. Flow predictions, along with MAE and RMSE, are evaluated for each time interval, which in this study is set to 10 min. This unit of measurement is consistent throughout the evaluation and provides clear and standardized metrics for assessing model performance.

## 5.2. Performance evaluation

As part of this section, we examine various scenarios in order to evaluate the performance of the proposed model. We begin by comparing the TMS-GNN's overall predictive capabilities with those of other baseline models. After that, we examine their performance in a multistep forecasting context, assessing their accuracy over various prediction horizons. Following that, we have a training time analysis for all models. Then, the training and validation loss analysis is provided and finally, we present a detailed comparison of the TMS-GNN's prediction performance at one crowded and one non-crowded stop over the course of a day, along with actual ground truth data to demonstrate the model's effectiveness in capturing passenger flow dynamics and fluctuations.

### 5.2.1. Overall forecasting performance

Table 3 shows the performance of the proposed model in comparison with other baseline models, evaluated with MAE and RMSE values for the test dataset for all 10-min intervals within the prediction horizon of 60 min. The reported MAE and RMSE represent the average of these metrics across all prediction intervals. As this table indicates, the TMS-GNN model performs better than all other baseline models in overall forecasting, with an MAE value of 1.882 and an RMSE value of 4.599 for the Laval dataset, and an MAE of 2.207 and an RMSE of 5.953 for the Ames dataset. This superior performance demonstrates the model's ability to generalize across different datasets and urban contexts.

Interestingly, graph-based models such as Graph Wavenet, DDSTGCN, and DCRNN show relatively high RMSE values, especially on the Ames dataset. This suggests that these models find it challenging to accurately capture the complexities of bus passenger flow in more crowded urban areas like Ames, Iowa. Moreover, one of the challenges with these graph neural network-based models is that they were initially designed for road traffic networks, where adjacent nodes are usually affected by each other due to phenomena like congestion waves propagating upstream — known as shockwaves in traffic flow theory. However, in bus networks, the passenger flow at one stop is not necessarily reflective of the passenger flow at adjacent stops, as passenger distribution is often significantly affected by major stops or unique destinations that disrupt normal flow patterns. Consequently, these graph-based models may struggle to accurately capture the localized and often isolated passenger dynamics typical of bus transit systems.

Moreover, the results show that LSTM has a closer performance to the proposed model on the Laval dataset, indicating the importance of temporal correlations in predicting passenger flows. In summary, the proposed TMS-GNN model not only considers multi-horizon temporal dependencies in passenger flow values but also captures the spatial dependencies between the passenger flows of different stops while taking into account the bus network structure and traffic conditions.

### 5.2.2. Multistep forecasting performance

To evaluate the performance of the model in multistep prediction, assisted with Scheduled Sampling, the values of passenger flow are predicted every 10 min until 60 min in the future for both Laval and Ames datasets. The same experiment is repeated for other baselines. Table 4 provides performance results at different time steps according to MAE and RMSE metrics.

Starting with DDSTGCN, in the Laval dataset, TMS-GNN generally outperforms it across most time intervals, particularly in short- to mid-term predictions. For instance, at the 10-min time interval, TMS-GNN achieves an MAE of 1.526 and an RMSE of 3.881, significantly better than DDSTGCN which has an MAE of 3.367 and an RMSE of 4.785. However, at the 50- and 60-min intervals, DDSTGCN slightly surpasses TMS-GNN in RMSE, though TMS-GNN maintains a better MAE, indicating more consistent accuracy in predicting absolute passenger counts. However, in the Ames dataset, TMS-GNN consistently outperforms DDSTGCN in both MAE and RMSE across all intervals, highlighting its robustness across different urban environments.

STGCN, which integrates graph convolution with temporal convolution, performs well at shorter intervals, specifically in the Laval dataset where it achieves an MAE of 2.310 and an RMSE of 4.535 at the 10-min interval. However, the performance of this

**Table 3**
Overall performance comparison between different models.

| Model | Laval dataset | | Ames dataset | |
|---|---|---|---|---|
| | MAE | RMSE | MAE | RMSE |
| HA | 2.273 | 5.209 | 2.636 | 6.764 |
| MLP | 4.034 | 6.265 | 4.642 | 9.669 |
| LSTM | 2.239 | 5.584 | 3.277 | 9.189 |
| Graph Wavenet | 3.974 | 6.617 | 5.028 | 11.202 |
| STGCN | 2.631 | 5.105 | 2.333 | 6.650 |
| DDSTGCN | 3.378 | 4.789 | 5.980 | 13.627 |
| DCRNN | 3.443 | 4.879 | 7.428 | 17.242 |
| STAEformer | 4.099 | 6.748 | 4.329 | 8.910 |
| TMS-GNN (Proposed Model) | **1.882** | **4.599** | **2.207** | **5.953** |

**Table 4**
Multistep performance comparison between different models for Laval and Ames datasets.

| Model | 10-min | | 20-min | | 30-min | | 40-min | | 50-min | | 60-min | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | MAE | RMSE | MAE | RMSE | MAE | RMSE | MAE | RMSE | MAE | RMSE | MAE | RMSE |
| Laval Dataset | | | | | | | | | | | | |
| HA | 2.248 | 5.146 | 2.236 | 5.137 | 2.256 | 5.184 | 2.288 | 5.239 | 2.300 | 5.269 | 2.306 | 5.279 |
| MLP | 3.212 | 4.961 | 3.732 | 5.815 | 3.986 | 6.200 | 4.208 | 6.534 | 4.435 | 6.882 | 4.634 | 7.199 |
| LSTM | 2.102 | 5.522 | 2.211 | 5.977 | 2.342 | 6.344 | 2.461 | 6.591 | 2.591 | 6.316 | 2.716 | 6.545 |
| Graph Wavenet | 3.910 | 6.554 | 3.964 | 6.665 | 3.971 | 6.651 | 3.991 | 6.629 | 3.995 | 6.622 | 4.017 | 6.582 |
| STGCN | 2.310 | 4.535 | 2.446 | 4.759 | 2.635 | 4.763 | 2.840 | 4.915 | 2.848 | 4.964 | 2.783 | 5.016 |
| DDSTGCN | 3.367 | 4.785 | 3.394 | 4.774 | 3.393 | 4.763 | 3.394 | 4.827 | 3.351 | **4.746** | 3.316 | **4.717** |
| DCRNN | 2.725 | 3.954 | 3.089 | 4.463 | 3.283 | 4.711 | 3.443 | 4.919 | 3.801 | 5.316 | 4.316 | 5.909 |
| STAEformer | 3.188 | 5.419 | 4.047 | 6.533 | 4.253 | 6.904 | 4.467 | 7.177 | 4.327 | 7.243 | 4.457 | 7.261 |
| TMS-GNN | **1.526** | **3.881** | **1.782** | **4.443** | **1.909** | **4.666** | **1.962** | **4.715** | **2.022** | 4.853 | **2.094** | 5.036 |
| Ames Dataset | | | | | | | | | | | | |
| HA | 2.719 | 7.114 | 2.698 | 7.032 | 2.641 | 6.861 | 2.599 | 6.625 | 2.570 | 6.464 | 2.587 | 6.490 |
| MLP | 3.136 | 5.567 | 4.678 | 9.314 | 5.003 | 11.099 | 4.950 | 10.536 | 4.816 | 9.801 | 5.264 | 11.694 |
| LSTM | 3.383 | 9.701 | 3.383 | 9.632 | 3.316 | 9.371 | 3.232 | 8.964 | 3.181 | 8.750 | 3.165 | 8.719 |
| Graph Wavenet | 4.928 | 11.008 | 5.052 | 11.118 | 4.958 | 11.006 | 4.954 | 11.072 | 5.087 | 11.449 | 5.189 | 11.562 |
| STGCN | 2.222 | 6.010 | 2.306 | 6.127 | 2.324 | 6.289 | 2.323 | 6.629 | 2.392 | 7.294 | 2.433 | 7.553 |
| DDSTGCN | 5.923 | 13.475 | 5.957 | 13.382 | 5.972 | 13.814 | 5.979 | 13.743 | 6.054 | 13.808 | 5.999 | 13.541 |
| DCRNN | 7.420 | 17.218 | 7422 | 17.220 | 7.393 | 17.196 | 7.433 | 17.255 | 7.439 | 17.264 | 7.458 | 17.300 |
| STAEformer | 4.171 | 8.494 | 4.302 | 8.859 | 4.345 | 9.126 | 4.328 | 8.964 | 4.379 | 8.946 | 4.449 | 9.071 |
| TMS-GNN | **2.119** | **5.758** | **2.235** | **6.069** | **2.242** | **6.104** | **2.229** | **5.988** | **2.212** | **5.906** | **2.205** | **5.893** |

model declines as the prediction horizon lengthens, particularly in the Ames dataset, where the complexity of the network might challenge its ability to capture longer-term dependencies, underscoring the STGCN model's limitations in handling more diverse and extensive transit systems.

Furthermore, LSTM, while effective for short-term predictions, is showing difficulty in capturing the complex spatial–temporal relationships necessary for accurate long-term forecasts as its RMSE rises significantly at longer intervals. This trend is even more pronounced in the Ames dataset, where LSTM's accuracy significantly declines, falling behind more sophisticated models that could better handle the complexities of larger and more varied transit networks.

Graph WaveNet, which integrates graph convolution with dilated causal convolution, generally underperforms compared to TMS-GNN and other advanced models. In the Laval dataset, for instance, the MAE is 3.910 and the RMSE is 6.554 at the 10-min interval, reflecting its relative inefficiency in capturing the complexities of passenger flow. This underperformance is also evident in the Ames dataset, where Graph WaveNet continues to struggle to adapt to the different network structures and passenger flow patterns.

Traditional models like HA and MLP show their limitations across both the Laval and Ames datasets. In Laval, HA achieves an MAE of 2.248 at the 10-min interval, but its performance diminishes at longer prediction horizons, highlighting the constraints of simpler approaches. Similarly, MLP consistently exhibits higher MAE and RMSE values across all intervals in both datasets, which shows its insufficiency in handling the complexities of urban transit networks.

Other models like DCRNN or STAEformer also did not match the performance of the proposed TMS-GNN model. In conclusion, although each model brings its own strengths to the table, the consistent and balanced performance of TMS-GNN across both datasets demonstrates its capability to handle the diverse and intricate challenges that arise in different urban transit networks, underscoring its effectiveness and reliability in passenger flow prediction. The observed differences between models across the Laval and Ames datasets further highlight the necessity for adaptability in predictive models and underscore the specific challenges each urban environment presents.

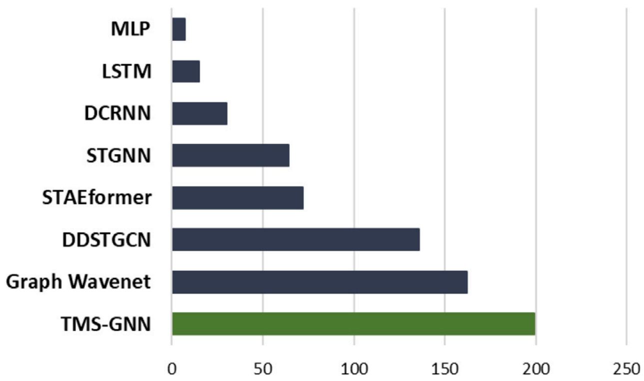



**Fig. 11.** Comparing computation training time per epoch.

### 5.2.3. *Training time analysis*

Fig. 11 shows a comparison of the training time per epoch for various models, including the proposed TMS-GNN, on the Laval subnetwork. Based on this figure, MLP and LSTM have the shortest training times, reflecting their simplicity but also their limited capacity for handling complex patterns. DCRNN, STGCN, and STAEformer require more time, indicating their added complexity, but they do not significantly outperform simpler models in terms of prediction accuracy. DDSTGCN and Graph Wavenet have even longer training times, indicating substantial computational demands without corresponding improvements in predictive performance.

Moreover, Fig. 11 highlights that TMS-GNN has the highest training time compared to other models, with about 199 s per epoch. While this may indicate a higher computational demand, it is important to note that TMS-GNN consistently outperforms other models in terms of predictive accuracy, particularly in both short- and long-term predictions across various datasets. This trade-off between training time and accuracy suggests that TMS-GNN, despite its longer training duration, provides significant advantages in capturing complex spatio-temporal patterns in bus passenger flow, making it a valuable model for applications where accuracy is critical. This analysis underscores the need to balance training time and model complexity when selecting a model for practical deployment in specific network scenarios. All the models were implemented on the OMEN GT13-0090 30L Gaming PC, equipped with an NVIDIA® GeForce RTX™ 3090, Intel Core i9-10850K, and HyperX® 32 GB DDR4-3200 XMP MHz RAM (96 GB). Despite the varying training times, the inference time for all models was relatively small, about 10–20 s, making them well-suited for real-time deployment. This efficiency highlights their potential for practical use in real-world scenarios, where quick and accurate predictions are necessary. In summary, the TMS-GNN with its relatively moderate training time and low inference time, offers a well-rounded solution for real-time passenger flow prediction.

### 5.2.4. *Analysis of training and validation loss*

Fig. 12 depicts the training loss (blue line) and validation loss (orange line) across epochs. Initially, both losses decrease sharply, reflecting effective learning by the model. The training loss continues its downward trajectory, suggesting that the model is fitting the training data well. The validation loss also decreases but at a slower rate and starts to flat out after around 150 epochs, suggesting the model's performance on unseen data is stabilizing.

To ensure the model does not overfit, early stopping was applied during training. This strategy monitors the validation loss and stops training once there is no further improvement, thus preventing the model from continuing to learn past the point of generalization. As a result, there is no significant uptick in the validation loss after the flattening phase, indicating that the model is not overfitting. Typically, overfitting is marked by a persistent decrease in training loss accompanied by an increase in validation loss, which is a sign that the model is beginning to memorize the training data rather than learning from it. The parallel decline of both loss curves in this scenario, with the added benefit of early stopping, implies that the model is effectively balancing its learning, ensuring that it works well on both training and unseen data.

Overall, the loss curves indicate that the model is well-regularized and has been trained effectively without overfitting. The minimal difference between the final training and validation losses implies that the model is capable of making accurate predictions on new data, demonstrating robust generalization capabilities.

### 5.2.5. *TMS-GNN's performance at crowded and non-crowded stops*

In this section, a closer look at the prediction performance of the proposed model is taken by zooming into two individual stops in the Laval network; one crowded stop and another less-crowded one. This helps to investigate the ability of the model to capture short-term fluctuations in passenger flow values, which is one of the most challenging tasks in passenger flow prediction. Fig. 13 depicts the combined trends of passenger flow predictions for the TMS-GNN and LSTM models – which is positioned as the first and one of the most effective models overall – versus actual recorded passenger flow at two different stops with different passenger flow patterns on October 7th, 2021. Fig. 13(a) represents a relatively crowded stop in the network. As can be seen, there are significant fluctuations (subsequent peaks and troughs) in passenger flow throughout the day, which makes the prediction task

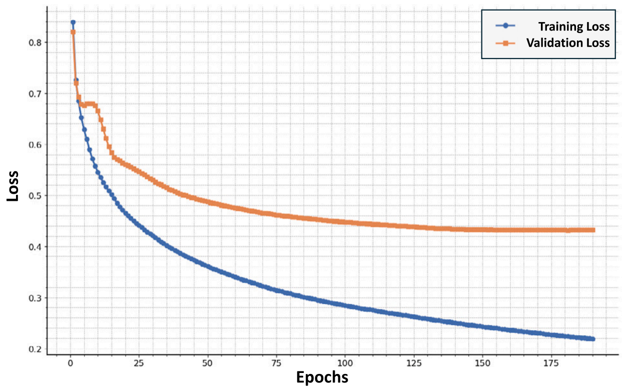



**Fig. 12.** Training and validation loss over epochs.

challenging. It appears that the LSTM model closely follows the actual data during low passenger flow periods (shown by the green line), but it diverges markedly during peak times, indicating that the model is unable to accurately predict situations when passenger flow is high. However, the TMS-GNN model's prediction (indicated by the orange line) and the real data (shown by the blue line) follow a similar trend in both low and high passenger flow periods, and the TMS-GNN model is more competent in capturing the fluctuations despite their seemingly uneven distribution. Moreover, two main megatrends (peaks) can be identified in the figure (one in the morning and one in the afternoon), related to morning and afternoon peak periods. The proposed model has captured these two main trends as well.

The second graph depicted in Fig. 13(b) illustrates the actual and predicted passenger flows of a non-crowded stop sample, which exhibits a less crowded profile with a significantly lower volume of passenger flow, peaking at around 25 passengers per hour. There are frequent zero values as the observed number of passengers at this stop. The high frequency of zero values makes the training step of the model challenging as the model has access to limited useful information to capture the trends in the dataset and therefore predict the non-zero value accurately. This has shown to be one of the challenges in dealing with passenger flow prediction in bus networks compared to other rail-based networks which usually have higher flows of passengers (Baghbani et al., 2023a; Lv et al., 2022). To address these challenges, the Traffic-Aware Reachability (*TAR*) matrix and the multi-component fusion module in TMS-GNN work in tandem to mitigate the effects of sparse data. The TAR matrix dynamically integrates real-time traffic conditions, ensuring that even sparse data scenarios benefit from a contextual understanding of stop-to-stop interactions influenced by traffic. Simultaneously, the fusion module synthesizes temporal patterns across different time scales — recent, daily, and weekly. In sparse contexts, where recent data may lack meaningful trends, the daily and weekly components fill this gap by capturing recurring behaviors, such as weekly peak periods or daily off-peak trends. Together, these components enable the model to effectively recognize and predict patterns, even in cases where sparse data might otherwise hinder performance. Therefore, as it has been shown in this stop, although both the LSTM model and the TMS-GNN model have been successful in capturing the overall trend in the passenger flow values, the non-zero values predicted by the LSTM model are significantly different from the actual values, while the TMS-GNN model shows a significantly better alignment with actual values both for zero and non-zero values. This shows the competency of the proposed model in accurately capturing the fluctuations in passenger flow.

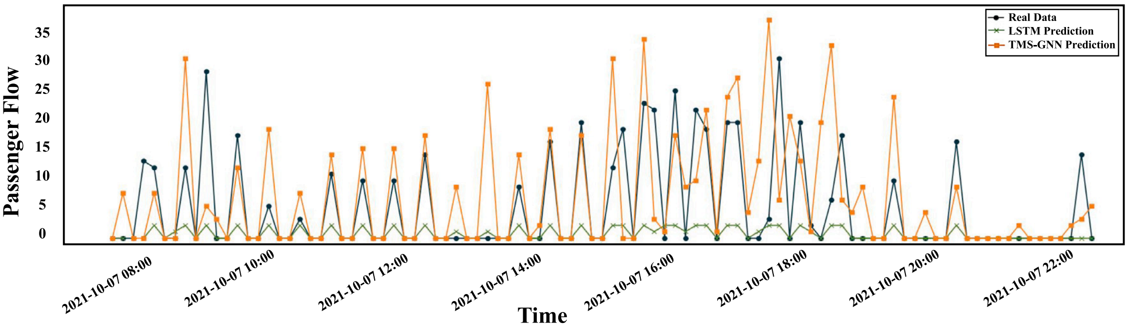
All in all, reviewing the two figures suggests that the TMS-GNN model consistently fits the real data more precisely than the LSTM model for both crowded and less-crowded stops. Based on the TMS-GNN model's ability to accurately predict passenger flows both at crowded and less crowded stops, it is of great importance as an optimization tool for bus scheduling and route planning. The forecasting of passenger flows with high precision allows transit authorities to allocate resources more efficiently, reduce wait times, and improve service reliability.

### *5.3. Ablation studies*

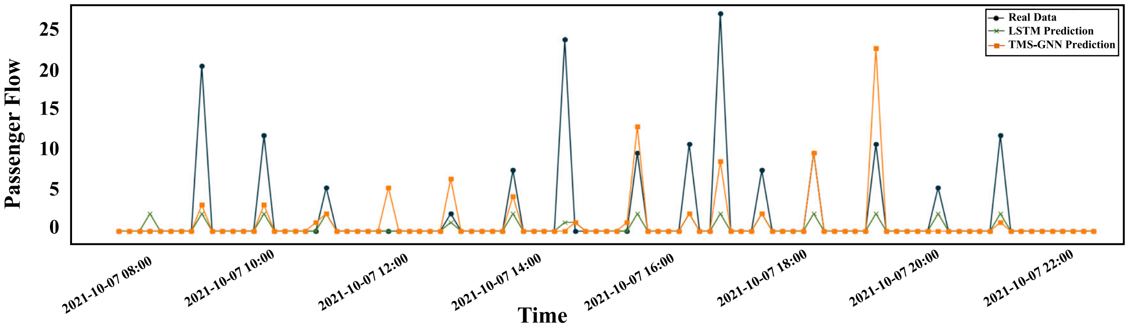
In this section, the results of the ablation studies are presented to verify the contribution of the different modules introduced in this study. First, the performance of the model is evaluated when removing some of the additional modules introduced in this study, including the *TAR* matrix, multi-component temporal fusion module, and the Scheduled Sampling module. Then, the effect of the *TAR* on the accuracy of passenger flow prediction is investigated by using different thresholds for constructing the *TAR* matrix.

#### *5.3.1. Contributions of different modules in the model*

The purpose of this section is to investigate the performance of the TMS-GNN model without some of its key components that are considered the novelties of this study. Several configurations of the proposed framework have been tested, and the results are summarized in Table 5. The tested configurations have been implemented on the Laval bus network and include:




(a) Passenger Flow Prediction Comparing to the Real Data in a Sample Crowded Stop




(b) Passenger flow Prediction Comparing to the Real Data in a Sample Non-Crowded Stop

**Fig. 13.** Comparison of prediction results and real data based on TMS-GNN and LSTM models.

1. **TMS-GNN with Connectivity Matrix Exclusively:** This variant of the model uses only the static connectivity matrix, which represents the fixed connections between bus stops based on bus routes. It does not consider the dynamic *TAR* matrix, which incorporates real-time traffic data, highlighting the importance of dynamic factors in public transportation prediction.
2. **TMS-GNN Using Static Traffic-Aware Reachability Only:** This configuration focuses on the static Traffic-Aware Reachability matrix without considering real-time adjustments based on traffic conditions. Here, the static TAR matrix reflects only average conditions rather than real-time traffic impacts, similar to static assumptions in studies like (Cui et al., 2019). The aim is to determine how effective static traffic data alone can be in predicting passenger flows, compared to dynamic traffic information.
3. **TMS-GNN Exclusively with Recent Data and No Fusion**: This version of the model uses only the most recent data to make passenger flow predictions, by not considering daily and weekly temporal patterns. The objective here is to assess how much short-term data alone can contribute to the model's performance without the integration of longer-term patterns.
4. **TMS-GNN Exclusively with Daily Data and No Fusion**: In this configuration, the model relies solely on daily data, focusing on capturing daily recurring patterns. This allows us to understand the model's performance when it is driven by day-to-day trends without the influence of recent or weekly data.
5. **TMS-GNN Exclusively with Weekly Data and No Fusion:** This version emphasizes weekly patterns, using data from the same days over previous weeks. It evaluates how well the model can predict passenger flows based solely on longer-term weekly trends.
6. **TMS-GNN without Scheduled Sampling:** This version omits the Scheduled Sampling technique, which is used to reduce exposure bias in multistep predictions. The goal is to assess how much this technique improves the model's performance over longer prediction intervals.

The results of the first configuration, which relies exclusively on the connectivity matrix and excludes the *TAR* matrix, show that this approach falls short in predicting passenger flows accurately. In another words, this variant of the model relies solely on the static connectivity matrix, representing fixed connections between bus stops based on routes, without incorporating the dynamic *TAR* matrix. The *TAR* matrix is crucial for introducing real-time traffic conditions into the model, particularly in scenarios where delays and traffic-induced changes in passenger flow dynamics are significant. Its absence leads to poorer predictive performance, as it fails to capture the dynamic interactions that the full model, integrating *TAR*, successfully represents.

In the second configuration, the *TAR* matrix is simplified to a static version, reflecting average traffic conditions rather than real-time fluctuations. The poorer performance of this static *TAR* matrix further emphasizes that only dynamic, real-time data can capture the true impacts of traffic fluctuations on passenger flows, contrasting sharply with static methods. To conclude, the results

**Table 5**
Performance comparison in ablation study.

| Model | MAE | RMSE | Training time (s) |
|---|---|---|---|
| TMS-GNN with only Connectivity Matrix | 2.757 | 5.765 | 11.0 |
| TMS-GNN with Static *TAR* | 2.603 | 5.449 | 11.5 |
| TMS-GNN Exclusively with Daily Data and No Fusion | 2.334 | 5.281 | 81.0 |
| TMS-GNN Exclusively with Weekly Data and No Fusion | 2.333 | 5.282 | 81.0 |
| TMS-GNN Exclusively with Recent Data and No Fusion | 2.186 | 5.568 | 140.5 |
| TMS-GNN without Scheduled Sampling | 1.986 | 5.272 | 178.0 |
| TMS-GNN | **1.882** | **4.559** | **199.5** |

for the first two ablated configurations in Table 5 support the assumption that dynamic traffic conditions have correlations with the values of passenger flow over the network and can be used as a good predictor in passenger flow prediction. Also, the amount of improvement after the inclusion of the *TAR* matrix in the model (around 45% improvement in MAE and 20% improvement in RMSE), suggests that these correlations have a significant impact on the prediction accuracy and may not be neglected. This validates the first assumption and contribution of the paper and highlights the important difference between bus transport networks and other rail-based public transport systems, such as urban trains and metro.

The ablation study also explores three configurations that each focus on a single temporal data component by removing the multi-component fusion module. In one configuration, the model only considers the most recent historical data, ignoring daily and weekly trends, which limits the model's ability to capture consistent fluctuations in passenger flows. Similarly, another configuration focuses solely on daily data, while a third uses only weekly data. All three configurations result in a decline in prediction accuracy compared to the full TMS-GNN model, which integrates recent, daily, and weekly data. This result supports the significance of multi-scale temporal fusion, further distinguishing our model from simpler, static frameworks with one time scale in the literature.

The last configuration examines the effect of Scheduled Sampling in the model by removing this component and instead using the model's own predictions from previous steps as input for subsequent predictions. The results indicate that Scheduled Sampling improves the model's accuracy by 5%–10% in MAE and 15% in RMSE, effectively reducing error accumulation in multistep forecasts. This feature is especially critical in multi-step prediction scenarios like bus passenger flow forecasting, a context often overlooked in single-step prediction models (like (Cui et al., 2019)). Additionally, the training time with Scheduled Sampling is only slightly longer than without it, and the test time remains consistently low for all configurations. This demonstrates that Scheduled Sampling enhances predictive accuracy without significantly impacting the model's efficiency.

Additionally, as shown in Table 5, the training time for each epoch of the fully-featured TMS-GNN model is slightly longer, primarily due to the incorporation of its advanced modules. However, this additional training time is reasonable as it has a significant improvement in accuracy and robustness compared to other models. Considering that the test time remains low across all configurations, the slight additional training time is a reasonable trade-off for the enhanced precision necessary for real-world applications.

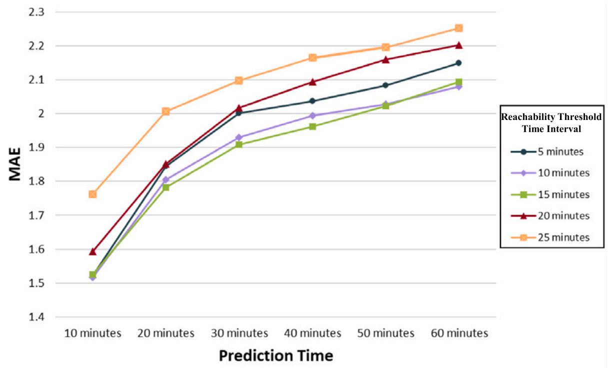
Overall, the conducted ablation study indicated that all features incorporated into the proposed model enhanced the model's performance. The *TAR* matrix enables the model to capture the uncertainty and dynamic nature of bus network performance due to its overlap with vehicular road traffic, the multi-component fusion module allows the model to capture more diverse and dynamic historical patterns (rather than relying on only one variations), and the Scheduled Sampling module helps the model to learn from its errors during the training phase in order to prevent the propagation of errors during the test phase in multistep passenger flow prediction.

#### 5.3.2. Sensitivity analysis on reachability threshold in TAR

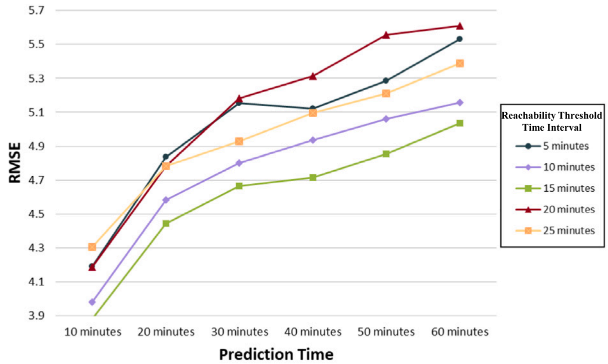
The *TAR* matrix is designed to dynamically incorporate traffic conditions, ensuring the model captures stop-to-stop interactions influenced by real-time delays and congestion. While the connectivity matrix reflects static relationships, the *TAR* matrix provides a dynamic perspective, particularly for stops that are farther apart, where traffic conditions significantly affect reachability. By incorporating this dynamic element, the *TAR* matrix ensures the model adapts to real-world operational dynamics. Specifying the time interval threshold for constructing the *TAR* matrix is crucial as it defines how different stops are reachable by other stops in the network at different times of the day. The reachability threshold, which is defined in terms of a time interval, specifies which stops are reachable from a specific stop considering the real-time traffic situations. Accordingly, it can depend on the road network structure, the distance between stops, and the travel times and frequencies of the buses on the network. Therefore, it is important to choose the right value for this threshold considering the specific characteristics of both bus and road networks.

As an example, we demonstrate the impact of selecting different time thresholds by analyzing the Laval subnetwork. Time intervals ranging from 5 to 25 min were explored for constructing the *TAR* matrix in this subnetwork. We chose this range by referencing several operational factors: (i) average bus headways in Laval generally range from 15–20 min; (ii) typical travel times between major stops often fall within this interval under varying congestion; and (iii) extremes below 5 min or above 25 min risked either ignoring meaningful connectivity or overgeneralizing the network. The results are depicted in Fig. 14. Lower MAE and RMSE values indicate higher accuracy in forecasting passenger flows.

As expected, changing the value of the threshold has a significant impact on the accuracy of the model. Choosing very low (5 min) or very high values (25 min) can drastically affect the performance of the model. Choosing a very low threshold can be interpreted as ignoring the varying relationships between the stops due to dynamic traffic situations since only a few stops are




(a) MAE Values for Different Models with Different TARs




(b) RMSE Values for Different Models with Different TARs

**Fig. 14.** Model performance for different models with different TARs in Laval subnetwork.

reachable in the specified thresholds. As Fig. 14 indicates, ignoring these relationships might be acceptable for very short-term predictions (10–20 min), but as the prediction steps go forward, the performance of the model deteriorates. As a conclusion, such a low threshold might align with situations where buses have very frequent headways or where nearby stops dominate passenger flows. However, this limited scope can overlook broader interactions that become significant over longer intervals, particularly during less frequent service hours. On the other hand, setting an overly large threshold can introduce two main problems. First, by labeling too many stops as reachable, the model may obscure important, stronger correlations among specific stops. Second, the model becomes less sensitive to traffic congestion because all those stops appear accessible despite real-world delays. As a result, the threshold must be carefully tuned – often through trial and error – to balance capturing meaningful interactions without overgeneralizing. This underscores how critical it is to account for varying relationships between stops under changing traffic conditions.

The graphs for both MAE and RMSE metrics indicate that for the Laval subnetwork, the model with a 15-min reachability threshold delivers the most consistent performance across different prediction times in this particular network based on both MAE and RMSE metrics. This suggests there is an optimal middle ground for the reachability threshold interval that balances responsiveness to immediate traffic conditions with the ability to accurately predict further into the future. The selected 15-min threshold strikes a balance between capturing realistic traffic-induced delays and the typical bus headways observed in the network. This choice ensures the *TAR* matrix reflects meaningful interactions between stops without over-simplifying or over-generalizing the dynamic relationships inherent in bus operations. Therefore, for all analyses of this subnetwork in this study, the threshold of 15 min was chosen.

### *5.4. Practical insights and actionable recommendations*

This section presents practical insights and actionable recommendations for transit agencies derived from the findings of the TMS-GNN model. The predictive capabilities of the TMS-GNN model can be used by transit agencies to make real-time adjustments to bus schedules, responding dynamically to fluctuating passenger flows. By accurately predicting passenger flow patterns, agencies can allocate additional buses during peak hours to meet high demand while reducing bus frequency during off-peak times. This optimization of resource allocation ensures both service reliability and operational cost efficiency, which directly contributes to reducing passenger wait times and increasing overall satisfaction. Moreover, with this information, transit agencies can redesign bus routes to improve coverage and efficiency. For instance, if certain stops experience high passenger flows at specific times, express routes can be created that respond to these peak demands, thereby reducing overall travel times and enhancing the quality of service.

Moreover, by integrating the TMS-GNN model with existing passenger information systems, transit agencies can offer real-time updates on bus occupancy levels. This enables passengers to make more informed decisions about which buses to take, potentially reducing overcrowding and improving travel comfort, which in turn enhances passenger satisfaction. In addition, the predictive capabilities of the model can be used to actively manage and balance passenger demand. For example, during peak hours, the model can recommend alternative routes or transportation modes to passengers, helping to distribute the flow more evenly across the network. This approach not only improves the passenger experience but also supports the maintenance of operational efficiency.

Furthermore, the ability of the proposed TMS-GNN model to make multistep predictions supports strategic planning by providing passenger flow predictions across multiple time intervals. Transit agencies can leverage these forecasts to make informed decisions on resource allocation and service adjustments. With access to precise predictions over these intervals, agencies can better prepare for fluctuations in demand and adapt their services to meet passenger needs more effectively and efficiently. In cases of unforeseen events, such as road closures or extreme weather, the TMS-GNN model enables transit agencies to quickly assess the impact on passenger flows and make necessary service adjustments. This capability enhances the agencies' ability to maintain reliable service even in adverse conditions, ensuring both continuity of operations and passenger safety.

All in all, the TMS-GNN model has significant potential to improve bus transit operations through its advanced predictive capabilities. By optimizing dynamic bus scheduling, route planning, real-time information sharing, and demand management, the model enhances both operational efficiency and passenger experience. Additionally, it has the capability to plan for and respond to emergencies in a timely manner. These practical insights and recommendations aim to guide agencies in using this innovative tool to its fullest potential, ultimately leading to better service delivery and improved urban mobility.

## 6. Conclusion

This study develops a novel graph-based deep learning framework, called TMS-GNN, for predicting multistep passenger flows for bus transportation networks. This study enjoys three main novelties. Firstly, in order to consider the effects of road traffic conditions on bus operations and passenger flow prediction, a Traffic-Aware Reachability matrix is defined and incorporated into the framework. Secondly, in order to address the well-known issue of "Exposure Bias" in sequence prediction models, a curriculum learning strategy, called Scheduled Sampling, is adopted so that the model learns how to deal with its propagating errors during the testing phase when generating multiple-step predictions. Finally, in order to consider the time-varying temporal correlations in the modeling framework, a novel fusion mechanism is introduced in order to capture the recent, daily, and weekly historical patterns in passenger flows.

The performance of the TMS-GNN model was thoroughly evaluated against various baseline models using two distinct datasets from different countries. The first dataset represents the bus network in Laval, Canada, comprising 467 stops. The second dataset is from Ames, Iowa, USA, which features 515 bus stops and presents a contrasting set of geographical and urban characteristics. This diverse evaluation highlights the robustness and adaptability of the TMS-GNN model across varying transit environments. The proposed model outperformed the other candidate models in most cases, particularly in terms of Mean Absolute Error (MAE) and Root Mean Square Error (RMSE). Moreover, the overall performance of the TMS-GNN model across all time intervals was superior, with consistently lower MAE and RMSE values, demonstrating its robustness and accuracy in predicting passenger flows. The ablation study also revealed that the inclusion of the Traffic-Aware Reachability matrix, the Scheduled Sampling strategy, and the fusion mechanism for temporal modeling improved the performance of the model. A comparison of the performance of the model with different network structures has also been conducted, which indicates that considering the connectivity between stops is also crucial for capturing the spatial dependencies among passenger flows at different stops.

While the TMS-GNN framework has shown to be capable of making robust predictions for short-term passenger flows at the network level, there are some limitations that need further work. The model's reliance on high-quality, real-time data could be mitigated by implementing transfer learning techniques, which would extend its applicability across various environments. Additionally, addressing the computational complexity through optimization techniques will make the model more accessible for transit agencies with limited technological resources. Enhancing the model's robustness to sudden changes in traffic conditions, such as road closures or major public events, is also another critical area for future improvement.

Furthermore, there is room for integrating additional factors such as weather conditions, events, and constructions into the model to enhance its accuracy. Moreover, advanced graph structures such as hypergraphs and nested graph models can be used in order to separate nodes with similar characteristics at a deeper level. Furthermore, future research can also examine how the spatial correlation between stops varies between historical periods using the origin destinations matrix. The consideration of bus frequency rather than general time intervals in temporal modeling is another aspect that may be of interest. Field tests can also be conducted to see if this model can be used in a practical way.

Finally, future studies could consider incorporating open-source trajectory datasets, such as those presented by Ammourah et al. (2024), to further enhance the model's understanding of traffic conditions and interactions between buses and other vehicles. These datasets could provide detailed insights into road traffic dynamics, complementing the Traffic-Aware Reachability matrix and improving predictive accuracy.

## CRediT authorship contribution statement

**Asiye Baghbani:** Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Saeed Rahmani:** Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Conceptualization. **Nizar Bouguila:** Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Methodology, Investigation, Formal analysis, Conceptualization. **Zachary Patterson:** Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Methodology, Investigation, Formal analysis, Conceptualization.

## Acknowledgments

We are grateful to BusPas Inc. and Mitacs Accelerate Program for supporting this research, as well as to Société de Transport de Laval for providing data for this study.

## References

Ammourah, Rami, Beigi, Pedram, Fan, Bingyi, Hamdar, Samer H, Hourdos, John, Hsiao, Chun-Chien, James, Rachel, Khajeh-Hosseini, Mohammdreza, Mahmassani, Hani S, Monzer, Dana, et al., 2024. Introduction to the third generation simulation dataset: Data collection and trajectory extraction. Transp. Res. Rec. 03611981241257257.

Baghbani, Asiye, Bouguila, Nizar, Patterson, Zachary, 2023a. Short-term passenger flow prediction using a bus network graph convolutional long short-term memory neural network model. Transp. Res. Rec. 2677 (2), 1331–1340.

Baghbani, Asiye, Rahmani, Saeed, Bouguila, Nizar, Patterson, Zachary, 2023b. Predicting passenger flow using graph neural networks with scheduled sampling on bus networks. In: 2023 IEEE 26th International Conference on Intelligent Transportation Systems. ITSC, IEEE, pp. 3073–3078.

Bai, Lei, Yao, Lina, Wang, Xianzhi, Li, Can, Zhang, Xiang, 2021. Deep spatial–temporal sequence modeling for multi-step passenger demand prediction. Future Gener. Comput. Syst. 121, 25–34.

Bengio, Samy, Vinyals, Oriol, Jaitly, Navdeep, Shazeer, Noam, 2015. Scheduled sampling for sequence prediction with recurrent neural networks. Adv. Neural Inf. Process. Syst. 28.

Chen, Tao, Fang, Jie, Xu, Mengyun, Tong, Yingfang, Chen, Wentian, 2022. Prediction of public bus passenger flow using spatial–temporal hybrid model of deep learning. J. Transp. Eng. Part A: Syst. 148 (4), 04022007.

Chen, Pengfei, Fu, Xuandi, Wang, Xue, 2021. A graph convolutional stacked bidirectional unidirectional-LSTM neural network for metro ridership prediction. IEEE Trans. Intell. Transp. Syst..

Chen, Qian, Li, Wenquan, Zhao, Jinhuan, 2011. The use of LS-SVM for short-term passenger flow prediction. Transport 26 (1), 5–10.

Chun-Hui, Zhang, Song-Rui, Sun-Yang, 2011. Kalman filter-based short-term passenger flow forecasting on bus stop. J. Transp. Syst. Eng. Inf. Technol. 11 (4), 154.

Cui, Zhiyong, Henrickson, Kristian, Ke, Ruimin, Wang, Yinhai, 2019. Traffic graph convolutional recurrent neural network: A deep learning framework for network-scale traffic learning and forecasting. IEEE Trans. Intell. Transp. Syst. 21 (11), 4883–4894.

Cyril, Anila, Mulangi, Raviraj H., George, Varghese, 2018. Modelling and forecasting bus passenger demand using time series method. In: 2018 7th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions). ICRITO, IEEE, pp. 460–466.

Du, Bowen, Peng, Hao, Wang, Senzhang, Bhuiyan, Md Zakirul Alam, Wang, Lihong, Gong, Qiran, Liu, Lin, Li, Jing, 2019. Deep irregular convolutional residual LSTM for urban traffic passenger flows prediction. IEEE Trans. Intell. Transp. Syst. 21 (3), 972–985.

Gallo, Mariano, De Luca, Giuseppina, D'Acierno, Luca, Botte, Marilisa, 2019. Artificial neural networks for forecasting passenger flows on metro lines. Sensors 19 (15), 3424.

Gong, Min, Fei, Xiang, Wang, Zhi Hu, Qiu, Yun Jie, 2014. Sequential framework for short-term passenger flow prediction at bus stop. Transp. Res. Rec. 2417 (1), 58–66.

Gu, Y., Han, Y., Fang, X., 2011. Method of hub station passenger flow forecasting based on ARMA model. J. Transp. Inf. Saf. 29 (2), 5–9.

Guo, Shengnan, Lin, Youfang, Feng, Ning, Song, Chao, Wan, Huaiyu, 2019. Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In: Proceedings of the AAAI Conference on Artificial Intelligence. 33, pp. 922–929.

Hamilton, William L., 2020. Graph Representation Learning. Morgan & Claypool Publishers.

Han, Yong, Wang, Shukang, Ren, Yibin, Wang, Cheng, Gao, Peng, Chen, Ge, 2019. Predicting station-level short-term passenger flow in a citywide metro network using spatiotemporal graph convolutional neural networks. ISPRS Int. J. Geo- Inf. 8 (6), 243.

Hao, Siyu, Lee, Der-Horng, Zhao, De, 2019. Sequence to sequence learning with attention mechanism for short-term passenger flow prediction in large-scale metro system. Transp. Res. Part C: Emerg. Technol. 107, 287–300.

He, Yuxin, Li, Lishuai, Zhu, Xinting, Tsui, Kwok Leung, 2022. Multi-graph convolutional-recurrent neural network (MGC-rnn) for short-term forecasting of transit passenger flow. IEEE Trans. Intell. Transp. Syst. 23 (10), 18155–18174.

He, Yuxin, Zhao, Yang, Wang, Hao, Tsui, Kwok Leung, 2020. GC-LSTM: A deep spatiotemporal model for passenger flow forecasting of high-speed rail network. In: 2020 IEEE 23rd International Conference on Intelligent Transportation Systems. ITSC, IEEE, pp. 1–6.

Huang, Hao, Mao, Jiannan, Lu, Weike, Hu, Guojing, Liu, Lan, 2023. DEASeq2Seq: An attention based sequence to sequence model for short-term metro passenger flow prediction within decomposition-ensemble strategy. Transp. Res. Part C: Emerg. Technol. 146, 103965.

Jiang, Weiwei, Luo, Jiayun, 2022. Graph neural network for traffic forecasting: A survey. Expert Syst. Appl. 207, 117921.

Jiao, Pengpeng, Li, Ruimin, Sun, Tuo, Hou, Zenghao, Ibrahim, Amir, 2016. Three revised kalman filtering models for short-term rail transit passenger flow prediction. Math. Probl. Eng. 2016.

Jing, Yun, Hu, Hongtao, Guo, Siye, Wang, Xuan, Chen, Fangqiu, 2020. Short-term prediction of urban rail transit passenger flow in external passenger transport hub based on LSTM-LGB-DRS. IEEE Trans. Intell. Transp. Syst..

Kong, Xiangjie, Wang, Kailai, Hou, Mingliang, Xia, Feng, Karmakar, Gour, Li, Jianxin, 2022. Exploring human mobility for multi-pattern passenger prediction: A graph learning framework. IEEE Trans. Intell. Transp. Syst. 23 (9), 16148–16160.

Li, Haiying, Wang, Yitang, Xu, Xinyue, Qin, Lingqiao, Zhang, Hanyu, 2019. Short-term passenger flow prediction under passenger flow control using a dynamic radial basis function network. Appl. Soft Comput. 83, 105620.

Li, Yaguang, Yu, Rose, Shahabi, Cyrus, Liu, Yan, 2017. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. arXiv preprint arXiv:1707.01926.

Lin, Lei, He, Zhengbing, Peeta, Srinivas, 2018. Predicting station-level hourly demand in a large-scale bike-sharing network: A graph convolutional neural network approach. Transp. Res. Part C: Emerg. Technol. 97, 258–276.

Lin, Ciyun, Wang, Kang, Wu, Dayong, Gong, Bowen, 2020. Passenger flow prediction based on land use around metro stations: a case study. Sustainability 12 (17), 6844.

Liu, Lijuan, Chen, Rung-Ching, 2017. A novel passenger flow prediction model using deep learning methods. Transp. Res. Part C: Emerg. Technol. 84, 74–91.

Liu, Lingbo, Chen, Jingwen, Wu, Hefeng, Zhen, Jiajie, Li, Guanbin, Lin, Liang, 2020a. Physical-virtual collaboration modeling for intra-and inter-station metro ridership prediction. IEEE Trans. Intell. Transp. Syst..

Liu, Hangchen, Dong, Zheng, Jiang, Renhe, Deng, Jiewen, Deng, Jinliang, Chen, Quanjun, Song, Xuan, 2023. Spatio-temporal adaptive embedding makes vanilla transformer sota for traffic forecasting. In: Proceedings of the 32nd ACM International Conference on Information and Knowledge Management. pp. 4125–4129.

Liu, Yang, Liu, Zhiyuan, Jia, Ruo, 2019. DeepPF: A deep learning based architecture for metro passenger flow prediction. Transp. Res. Part C: Emerg. Technol. 101, 18–34.

Liu, Yang, Lyu, Cheng, Liu, Xin, Liu, Zhiyuan, 2020b. Automatic feature engineering for bus passenger flow prediction based on modular convolutional neural network. IEEE Trans. Intell. Transp. Syst. 22 (4), 2349–2358.

Liu, Tong, Meidani, Hadi, 2023. Heterogeneous graph neural networks for data-driven traffic assignment. arXiv preprint arXiv:2310.13193.

Lu, Yuhuan, Ding, Hongliang, Ji, Shiqian, Sze, NN, He, Zhaocheng, 2021. Dual attentive graph neural network for metro passenger flow prediction. Neural Comput. Appl. 33 (20), 13417–13431.

Luo, Dan, Zhao, Dong, Ke, Qixue, You, Xiaoyong, Liu, Liang, Ma, Huadong, 2021. Spatiotemporal hashing multigraph convolutional network for service-level passenger flow forecasting in bus transit systems. IEEE Internet Things J. 9 (9), 6803–6815.

Luo, Dan, Zhao, Dong, Ke, Qixue, You, Xiaoyong, Liu, Liang, Zhang, Desheng, Ma, Huadong, Zuo, Xingquan, 2020. Fine-grained service-level passenger flow prediction for bus transit systems based on multitask deep learning. IEEE Trans. Intell. Transp. Syst. 22 (11), 7184–7199.

Lv, Wanjun, Lv, Yongbo, Ouyang, Qi, Ren, Yuan, 2022. A bus passenger flow prediction model fused with point-of-interest data based on extreme gradient boosting. Appl. Sci. 12 (3), 940.

Ma, Jingyan, Gu, Jingjing, Zhou, Qiang, Wang, Qiuhong, Sun, Ming, 2020. Dynamic-static-based spatiotemporal multi-graph neural networks for passenger flow prediction. In: 2020 IEEE 26th International Conference on Parallel and Distributed Systems. ICPADS, IEEE, pp. 673–678.

Ma, Zhenliang, Xing, Jianping, Mesbah, Mahmoud, Ferreira, Luis, 2014. Predicting short-term bus passenger demand using a pattern hybrid approach. Transp. Res. Part C: Emerg. Technol. 39, 148–163.

Nagaraj, Nandini, Gururaj, Harinahalli Lokesh, Swathi, Beekanahalli Harish, Hu, Yu-Chen, 2022. Passenger flow prediction in bus transportation system using deep learning. Multimedia Tools Appl. 81 (9), 12519–12542.

Ou, Junjie, Sun, Jiahui, Zhu, Yichen, Jin, Haiming, Liu, Yijuan, Zhang, Fan, Huang, Jianqiang, Wang, Xinbing, 2022. STP-TrellisNets+: Spatial-temporal parallel TrellisNets for multi-step metro station passenger flow prediction. IEEE Trans. Knowl. Data Eng..

Rahmani, Saeed, Baghbani, Asiye, Bouguila, Nizar, Patterson, Zachary, 2023. Graph neural networks for intelligent transportation systems: A survey. IEEE Trans. Intell. Transp. Syst..

Samaras, Patroklos, Fachantidis, Anestis, Tsoumakas, Grigorios, Vlahavas, Ioannis, 2015. A prediction model of passenger demand using AVL and APC data from a bus fleet. In: Proceedings of the 19th Panhellenic Conference on Informatics. pp. 129–134.

Sangiorgio, Matteo, Dercole, Fabio, 2020. Robustness of LSTM neural networks for multi-step forecasting of chaotic time series. Chaos Solitons Fractals 139, 110045.

Sonoda, Sho, Murata, Noboru, 2017. Neural network with unbounded activation functions is universal approximator. Appl. Comput. Harmon. Anal. 43 (2), 233–268.

Sun, Yanfeng, Jiang, Xiangheng, Hu, Yongli, Duan, Fuqing, Guo, Kan, Wang, Boyue, Gao, Junbin, Yin, Baocai, 2022. Dual dynamic spatial-temporal graph convolution network for traffic prediction. IEEE Trans. Intell. Transp. Syst. 23 (12), 23680–23693.

Sun, Yuxing, Leng, Biao, Guan, Wei, 2015. A novel wavelet-SVM short-time passenger flow prediction in Beijing subway system. Neurocomputing 166, 109–121.

Teng, Jing, Chen, Sijin, 2015. Modified bus passenger flow forecasting model based on integrating arima with neural network. In: COTA International Conference of Transportation Professionals 2015. Elsevier, pp. 1300–1310.

Tsai, Tsung-Hsien, Lee, Chi-Kang, Wei, Chien-Hung, 2009. Neural network based temporal feature models for short-term railway passenger demand forecasting. Expert Syst. Appl. 36 (2), 3728–3736.

Wang, Jingcheng, Zhang, Yong, Wei, Yun, Hu, Yongli, Piao, Xinglin, Yin, Baocai, 2021. Metro passenger flow prediction via dynamic hypergraph convolution networks. IEEE Trans. Intell. Transp. Syst. 22 (12), 7891–7903.

Wilbur, Keith, 2023. CyRide Automatic Passenger Counter Data, October 2021 - March 2023. Iowa State University, http://dx.doi.org/10.25380/Iastate.20346636.V2, Dataset.

Wu, Zonghan, Pan, Shirui, Chen, Fengwen, Long, Guodong, Zhang, Chengqi, Philip, S Yu, 2020. A comprehensive survey on graph neural networks. IEEE Trans. Neural Netw. Learn. Syst. 32 (1), 4–24.

Wu, Zonghan, Pan, Shirui, Long, Guodong, Jiang, Jing, Zhang, Chengqi, 2019. Graph wavenet for deep spatial-temporal graph modeling. arXiv preprint arXiv:1906.00121.

Xie, Peng, Li, Tianrui, Liu, Jia, Du, Shengdong, Yang, Xin, Zhang, Junbo, 2020. Urban flow prediction from spatiotemporal data using machine learning: A survey. Inf. Fusion 59, 1–12.

Xue, Rui, Sun, Daniel, Chen, Shukai, 2015. Short-term bus passenger demand prediction based on time series model and interactive multiple model approach. Discrete Dyn. Nat. Soc. 2015 (1), 682390.

Xue, Qiuchi, Zhang, Wei, Ding, Meiling, Yang, Xin, Wu, Jianjun, Gao, Ziyou, 2023. Passenger flow forecasting approaches for urban rail transit: a survey. Int. J. Gen. Syst. 1–29.

Yang, Xin, Xue, Qiuchi, Ding, Meiling, Wu, Jianjun, Gao, Ziyou, 2021. Short-term prediction of passenger volume for urban rail systems: A deep learning approach based on smart-card data. Int. J. Prod. Econ. 231, 107920.

Ye, Yinna, Chen, Li, Xue, Feng, 2019. Passenger flow prediction in bus transportation system using ARIMA models with big data. In: 2019 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery. CyberC, IEEE, pp. 436–443.

Ye, Jiexia, Zhao, Juanjuan, Ye, Kejiang, Xu, Chengzhong, 2020. Multi-stgcnet: A graph convolution based spatial-temporal framework for subway passenger flow forecasting. In: 2020 International Joint Conference on Neural Networks. IJCNN, IEEE, pp. 1–8.

Yin, Du, Jiang, Renhe, Deng, Jiewen, Li, Yongkang, Xie, Yi, Wang, Zhongyi, Zhou, Yifan, Song, Xuan, Shang, Jedi S, 2023. MTMGNN: Multi-time multi-graph neural network for metro passenger flow prediction. GeoInformatica 27 (1), 77–105.

Yu, Bing, Yin, Haoteng, Zhu, Zhanxing, 2017. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. arXiv preprint arXiv:1709.04875.

Zeng, Jie, Tang, Jinjun, 2023. Combining knowledge graph into metro passenger flow prediction: A split-attention relational graph convolutional network. Expert Syst. Appl. 213, 118790.

Zhai, Huawei, Cui, Licheng, Nie, Yu, Xu, Xiaowei, Zhang, Weishi, 2018. A comprehensive comparative analysis of the basic theory of the short term bus passenger flow prediction. Symmetry 10 (9), 369.

Zhai, Xubin, Shen, Yu, 2023. Short-term bus passenger flow prediction based on graph diffusion convolutional recurrent neural network. Appl. Sci. 13 (8), 4910.

Zhang, Jinlei, Chen, Feng, Guo, Yinan, Li, Xiaohong, 2020a. Multi-graph convolutional network for short-term passenger flow forecasting in urban rail transit. IET Intell. Transp. Syst. 14 (10), 1210–1217.

Zhang, Hao, He, Jie, Bao, Jie, Hong, Qiong, Shi, Xiaomeng, 2020b. A hybrid spatiotemporal deep learning model for short-term metro passenger flow prediction. J. Adv. Transp. 2020.

Zhang, Kunpeng, Liu, Zijian, Zheng, Liang, 2019. Short-term prediction of passenger demand in multi-zone level: Temporal convolutional neural network with multi-task learning. IEEE Trans. Intell. Transp. Syst. 21 (4), 1480–1490.

Zhang, Jun, Shen, Dayong, Tu, Lai, Zhang, Fan, Xu, Chengzhong, Wang, Yi, Tian, Chen, Li, Xiangyang, Huang, Benxiong, Li, Zhengxi, 2017. A real-time passenger flow estimation and prediction method for urban bus transit systems. IEEE Trans. Intell. Transp. Syst. 18 (11), 3168–3178.

Zhao, Shu-Zhi, Ni, Tong-He, Wang, Yang, Gao, Xiang-Tao, 2011. A new approach to the prediction of passenger flow in a transit system. Comput. Math. Appl. 61 (8), 1968–1974.

Zhao, Ling, Song, Yujiao, Zhang, Chao, Liu, Yu, Wang, Pu, Lin, Tao, Deng, Min, Li, Haifeng, 2019. T-GCN: A temporal graph convolutional network for traffic prediction. IEEE Trans. Intell. Transp. Syst. 21 (9), 3848–3858.

Zhu, Huaizhong, Yang, Xiaoguang, Wang, Yizhe, 2018. Prediction of daily entrance and exit passenger flow of rail transit stations by deep learning method. J. Adv. Transp. 2018.

Zou, Linmu, Wang, Zijia, Guo, Rui, 2024. Real-time prediction of transit origin–destination flows during underground incidents. Transp. Res. Part C: Emerg. Technol. 163, 104622.