WEST GCN-LSTM: Weighted Stacked Spatio-Temporal Graph Neural Networks for Regional Traffic Forecasting

Theodoros Theodoropoulos[a,b,*], Angelos-Christos Maroudis[c], Uwe Zdun[b], Antonios Makris[c,a] and Konstantinos Tserpes[c,a]

[a] *Harokopio University of Athens, Omirou 9, Athens, 17778, Greece*
[b] *University of Vienna, Software Architecture Research Group, Vienna, 1090, Austria*
[c] *National Technical University of Athens, Heroon Polytechniou 9, Athens, 15780, Greece*

ARTICLE INFO

Keywords:
Graph Neural Networks
Traffic Forecasting
Urban Mobility

ABSTRACT

Regional traffic forecasting is a critical challenge in urban mobility, with applications to various fields such as the Internet of Everything. In recent years, spatio-temporal graph neural networks have achieved state-of-the-art results in the context of numerous traffic forecasting challenges. This work aims to expand upon the conventional spatio-temporal graph neural network architectures in a manner that may facilitate the inclusion of information regarding the examined regions and the populations that traverse them to establish a more efficient prediction model. The end-product of this scientific endeavor is a novel spatio-temporal graph neural network architecture for regional traffic forecasting referred to as WEST (WEighted STacked) GCN-LSTM. Furthermore, the aforementioned information is included via two novel dedicated algorithms, the Shared Borders Policy and the Adjustable Hops Policy. Through information fusion and distillation, the proposed solution significantly outperforms its competitors in an experimental evaluation of 19 forecasting models across several datasets. Finally, an additional ablation study determined that each component of the proposed solution enhances its overall performance.

1. Introduction

Regional traffic forecasting is an emerging challenge in the domain of urban mobility that is important in various fields such as smart cities (1), edge computing (2), the Internet of Things (3), wireless networks (4), personalized recommender systems (5), epidemiology modeling (6), and many more. However, the importance of regional traffic forecasting continues to grow in the context of the Internet of Everything (IoE) paradigm (7), which is characterized by an intricate web of relationships among people, things, data, and processes. Regional traffic forecasting refers to predicting future traffic conditions across diverse geographic areas characterized by grid-based or non-uniform partitioning and over multiple periods, which may span from a few minutes to several hours. This process is conducted using dedicated forecasting models.

Despite its numerous beneficial applications in the fields above, regional traffic forecasting poses a complex problem, as it requires accurately forecasting traffic conditions across different areas over an extended period. This complexity arises from the intricate and interlinked two-fold nature of traffic systems that manifest spatial and temporal characteristics (8). Spatially, traffic conditions in one region can be influenced by events occurring in neighboring or distant areas, requiring an understanding of the spatial dependencies between different regions. In other words, traffic conditions in one area can have a cascading effect on neighboring regions, making it imperative to model and account for the interdependencies among regions accurately. Temporally, traffic patterns undergo dynamic changes influenced by time cycles of varying lengths (9), demanding models that capture both short-term fluctuations and long-term trends.

Thus, any attempt at constructing regional traffic forecasting models should be designed to incorporate information regarding the topology of the various regions and the populations that traverse them. Thankfully, such information can become readily available via technologies such as advanced traffic sensor networks (10) and integrated geographic information systems (11). These technologies can continuously monitor, document, and archive spatial and time information across multiple instances. By combining advanced modeling techniques, real-time data streams, and domain-specific knowledge, it is possible to create robust and accurate forecasting systems capable of handling the intricacies of regional traffic (12).

In recent years, spatio-temporal graph neural networks, such as GCN-LSTM (13; 14), have demonstrated state-of-the-art performance in a wide range of traffic forecasting problems (15; 16), primarily due to their intrinsic nature of effectively incorporating contextual information. However, only a few of these endeavors focus on regional traffic forecasting. Furthermore, the use of spatio-temporal graph neural networks for regional traffic forecasting has been quite limited in the sense that prior attempts focus on capturing either the spatial or temporal aspects of this problem.

* This project has received funding from the European Union's Horizon 2020 research and innovation programmes under grant agreements No 101135775 (PANDORA), No 101016509 (CHARITY) and No 777695 (MASTER). The work reflects only the authors' view, and the EU Agency is not responsible for any use that may be made of the information it contains.

theodoros.theodoropoulos@univie.ac.at (T. Theodoropoulos); amaroudis@hua.gr (A. Maroudis); uwe.zdun@univie.ac.at (U. Zdun); antoniosmakris@mail.ntua.gr (A. Makris); tserpes@mail.ntua.gr (K. Tserpes)

ORCID(s): 0000-0002-4618-4891 (T. Theodoropoulos)

More specifically, in terms of encapsulating the temporal aspect, the vast majority of prior works leverage either the distance between the centers of regions (17; 18; 19; 20; 21; 22) or the neighboring statuses of regions (23; 24; 25). On the other hand, in terms of the temporal aspect, prior works mainly focus on traffic pattern similarity matrices (26; 27; 28). Unfortunately, this fragmented approach is bound to result in suboptimal representations of the examined environment and, consequently, in suboptimal regional traffic forecasting accuracy.

Motivated by this fact, our work aims at expanding upon the GCN-LSTM architecture in a manner that may facilitate the incorporation of a wider range of information regarding the various populations (temporal aspect), as well as the regions that they traverse (spatial aspect), to establish more refined and accurate prediction models, through information fusion and distillation. This scientific endeavor results in a novel spatio-temporal graph neural network architecture for regional traffic forecasting that leverages weighted stacked graph convolution. This architecture is referred to as WEST (WEighted STacked) GCN-LSTM. Furthermore, the information above is incorporated via two novel algorithms, the Shared Borders Policy and the Adjustable Hops Policy. This paper analyzes the proposed architecture and algorithms in great detail and evaluates the efficiency of the proposed solution.

More specifically, the remainder of the paper is structured in the following manner: Section 2 explores the corresponding scientific literature based on various traffic forecasting models. Section 3 establishes the problem formulation used throughout this paper. Section 4 showcases the proposed solution. Section 5 describes the experimental process undertaken to evaluate the efficiency of the proposed solution. Finally, Section 6 summarizes the merits and findings of this work and proposes potential future research directions.

2. Literature Review

Depending on the examined regional traffic forecasting scenario, the moving entities' characteristics and the regions they traverse may vary significantly. Nevertheless, regardless of which type of scenario is being considered, these two aspects constitute the cornerstone of regional traffic forecasting. As such, any attempt at conducting regional traffic forecasting should be carried out within a spatio-temporal framework capable of encapsulating the dynamic nature of the moving entities that are bound to manifest.

This view of traffic evolution as a dynamic system led earlier attempts at traffic forecasting to consider using Recurrent Neural Networks (RNNs) (29). While RNNs have been particularly effective for modeling sequential data exhibiting dynamic behavior, their efficacy wanes in storing prolonged information, attributed to the vanishing / exploding gradient phenomenon that appears in long sequence learning (30). To surmount this limitation, Long Short-Term Memory (LSTM) networks were used in the context of regional traffic forecasting (31). LSTMs excel in capturing temporal intricacies and long-term dependencies. However, while these models proficiently handle sequential data and unveil temporal patterns within mobility tasks, their prowess in the temporal domain stands in contrast to their limitations in comprehensively encapsulating the spatial aspects of the problem.

Encoder-decoders (EDs) are composite Deep Learning (DL) architectures capable of mitigating this limitation. They are designed to handle variable-length input and output sequences, making them ideal for sequence-to-sequence predictions due to their structure. The encoder converts variable-length input into a fixed-shape state, and the decoder, using the encoder's states, generates the output based on gathered information. While the role of the encoder is to encapsulate the spatial underlying dependencies, the decoder aims to capture the various temporal patterns and thus is usually based on some form of RNN model. Notable encoder-decoder architectures that have been examined in the frame of traffic forecasting scenarios include the LSTM ED (32), THE BD-LSTM ED (33), the CNN-LSTM (34), the Hybrid LSTM ED (35), and the Hybrid LSTM ATT ED (36).

More specifically, the findings we derived while authoring the latter of these works motivated us to focus our efforts on establishing a more advanced solution for regional traffic forecasting. This work showcased that ED architectures, when leveraged for multi-step regional traffic forecasting in a single region, manage to outperform their competitors. However, the situation changes drastically when simultaneously exploring multiple regions in a multi-step manner due to the dramatic increase in the complexity of the input and output sequences. During such multi-regional scenarios, ED architectures seem to lose their clear competitive advantage against other approaches, such as the ones that are based on Linear Regression (LR) (37) and Machine Learning (ML) (38) approaches, depending on the underlying characteristics of the dynamic systems that derive from the corresponding regional traffic scenario. Significant increases in the complexity of the input and output sequences seem to construct a performance plateau that disproportionally affects each forecasting approach. This observation shows the need to introduce regional traffic forecasting mechanisms whose efficiency is not jeopardized by increases in the underlying complexity of the problem but, instead, through information fusion and refinement, manage to consistently rise above the aforementioned plateau, regardless of the underlying system dynamics.

The pursuit of effective methodologies to address challenges inherent in processing data originating from non-Euclidean domains has recently garnered significant attention. At the forefront of these endeavors lie Graph Neural Networks (GNNs) (39), renowned for their adeptness in resolving problems that are intricately intertwined with spatial aspects (40). This prowess emanates from their inherent capacity to harness and exploit the spatial attributes of data about a given problem. Over time, foundational GNN architectures have undergone extensions to augment their inherent attributes, thus resulting in several variations. The most notable of these variations are recurrent graph neural networks (41), graph convolutional networks (GCN) (42), graph autoencoders (43), and spatio-temporal graph neural networks (44). Spatio-temporal graph neural networks have been proven quite successful in traffic forecasting (45) since their architecture enables them to capture spatial and temporal dependencies simultaneously. This can be achieved by leveraging graph convolutions to model spatial dependencies alongside RNNs to encapsulate temporal dependencies (46) in a manner aligned with the ED paradigm.

The modus operandi of GNNs is based on iteratively aggregating information from the neighbors and updating the representations of nodes. A dedicated adjacency matrix dictates the neighbors of each node. In most cases, the node representation is updated based on the direct neighbors of the node, which are called 1-hop neighbors. However, there have been works (47; 48) that advocate for extending information aggregation to K-hop format to enhance the model's expressive power (49). In the context of spatio-temporal graph neural networks, there has been only a single instance (50) where the incorporation of K-hop information aggregation was examined. However, in the context of the aforementioned work, K was regarded as a parameter tuned by trial-and-error. In our work, however, K is calculated using information regarding the moving speed of the populations and the topology of the regions. This design choice enables the proposed solution to encapsulate the temporal aspects of regional traffic forecasting concerning the moving speed of the various populations.

Although there have been numerous works that examine the use of GNNs in the context of traffic forecasting (51; 52; 53), only a rather small fraction of them focus on the use of graph neural networks for regional traffic forecasting (54). The vast majority of these works can be categorized based on the way they construct the graphs' adjacency matrices (55). Most of these works (17; 18; 19; 20; 21; 22) choose to construct the adjacency matrices based on various distance-related metrics (in most cases, the distance between the centers of the regions). Other works (26; 27; 28) propose the construction of adjacency matrices based on traffic pattern similarity matrices. Finally, the last category includes solutions (23; 24; 25) that leverage binary adjacency matrices based on whether or not the involved nodes are neighboring. An overview of this analysis is depicted in Table 1. In our work, we expand upon the latter category by proposing a weighted adjacency matrix to encapsulate the lengths of borders being shared between the various regions. This design choice enables the proposed solution to encapsulate the spatial aspects of regional traffic forecasting regarding the topology of the involved regions in a more refined manner. Furthermore, all of the aforementioned GNN-based solutions for regional traffic forecasting are designed to focus on either the spatial (distance of centers, neighboring status) or the temporal (traffic) aspects of this problem. However, in our work, we propose a solution that is capable of optimally capturing both the spatial and the temporal aspects of regional traffic forecasting.

Our work aims to introduce an advanced forecasting model that, through information refinement and fusion, can produce more accurate regional traffic predictions. To achieve this goal, we extend spatio-temporal graph neural networks in a manner that is aligned with the regional traffic forecasting paradigm. To do so, we propose a novel spatio-temporal graph neural network architecture for regional traffic forecasting, incorporating weighted stacked convolutions. Weighted stacked convolutions require the calculation of the weighted adjacency matrix and the number of graph convolution layers K. To that end, we also propose two novel policies (Shared Borders Policy & Adjustable Hops Policy) that leverage information regarding the speed of the populations and the topography of the regions that they traverse to calculate the adjacency matrix and K.

What differentiates our proposed solution from these prior works in regional traffic forecasting is the fact that they were designed in a manner that leverages the incorporation of information regarding the spatial / temporal aspect only in regards to the construction of the corresponding adjacency matrix. As such, they can incorporate only one of these aspects. On the other hand, our proposed solution is designed in a manner that incorporates the spatial aspect in the construction of the adjacency matrix, and the temporal aspect in regards to the number of graph convolution layers K (Adjustable Hops Policy). In other words, our proposed approach expands upon the existing feature aggregation approaches in the frame of using spatio-temporal GNNs for regional traffic forecasting by incorporating both the adjacency matrix (spatial information) and the number of graph convolution layers K, while prior works use only the adjacency matrix to incorporate only either spatial or temporal information.

Furthermore, our proposed solution builds upon the neighboring statuses of regions approach by introducing a weighted adjacency matrix that represents the shared border lengths between different regions. This design decision enhances the proposed solution's ability to capture the spatial characteristics of regional traffic forecasting with greater precision in relation to the topology of the regions involved (Shared Borders Policy). Out of the aforementioned works, only a small number of them leverage weighted graph convolution, but they apply them in the frame of the distance between centers or the traffic pattern similarity approaches. To the best of our knowledge, our work is the first one that uses weighted graph convolution in the frame of the neighboring status of regions approach. Hence, our proposed solution does not only leverage a wider range of information but also more refined forms of information (compared to the corresponding scientific literature) for encapsulating the underlying environment. As a result, the proposed solution is designed in a manner that shall be capable of providing superior results in terms of regional traffic forecasting.

Reference	Spatial Aspect	Temporal Aspect
(17)	Distance between Centers	-
(18)	Distance between Centers	-
(19)	Distance between Centers	-
(20)	Distance between Centers	-
(21)	Distance between Centers	-
(22)	Distance between Centers	-
(23)	Neighboring status of Regions (binary)	-
(24)	Neighboring status of Regions (binary)	-
(25)	Neighboring status of Regions (binary)	-
(26)	-	Traffic pattern similarity
(27)	-	Traffic pattern similarity
(28)	-	Traffic pattern similarity
Proposed (ours)	**Distance between Centers**, **Neighboring Status of Regions (weighted)**	**Speed of involved entities**

Table 1
Categorization of related works on regional traffic forecasting using spatio-temporal GNNs based on their corresponding spatial and temporal aspects.

3. Problem Formulation

A large portion of this paper is dedicated to evaluating models to forecast population flow estimates for regions across multiple future periods. We refer to this challenge as multi-step regional traffic forecasting. A region refers to a specific area within a larger geographic or urban context of particular interest for a certain purpose or analysis. These regions are represented by the set $Regions = \{Region_1, Region_2.., Region_n\}$, where $Region_n$ indicates the n^{th} Region with $1 \leq n \leq N$, where N corresponds to the number of Regions. Each $Region_n$ is characterized by its Borders which are represented by the set $B_n = \{b_1, b_2.., b_b\}$, where b_b indicates the b^{th} Border, where $1 \leq b \leq B$, and by its $Center_n$ which is represented by a 2-tuple that corresponds to its x & y coordinates. The average distance between the various $Center_n$ 2-tuples is denoted by D. Furthermore, each Border is regarded as a line segment, which is characterized by a 4-tuple that corresponds to the x & y coordinates of the two endpoints of the line segment. The various types of populations are represented by the set $Populations = \{Population_1, Population_2.., Population_p\}$, where p_p indicates the p^{th} Population with $1 \leq p \leq P$, where P corresponds to the number of different Populations. The distinction between Population types is made based on their ability to traverse the area, and thus, each of these Populations is characterized by an average moving speed ($Speed_p$). The notations used in the context of this work are summarized in Table 2.

In time-series analysis (56), the multi-step formulation involves predicting future values of a time series by forecasting multiple time steps. This method differs from the single-step approach, which only estimates the next point in time. Furthermore, the multivariate formulation in time series analysis involves creating a model for target variables that relies on multiple predictor variables. These predictor variables are interdependent and display temporal dependencies over time, which may be impacted by exogenous inputs and noise. This method can be mathematically represented as a system of equations, where the target and predictor variables are modeled as stochastic processes that vary with time.

In the context of the present challenge, the output vector's dimensional space is denoted by R^{N*U}, wherein N represents the number of Regions for which we intend to predict traffic at a given time point t, and U denotes the number of future steps over which we aim to make these projections. Similarly, we define the input vector's dimensional space as $R^{N'*U'}$, where N' corresponds to the number of the Regions whose population variations exhibit reliance on those of N, and U' signifies the number of preceding time steps that contribute to the retrospective observation window (look-back window). Notably, in our modeling, the value of N' is equal to that of N.

In further elaboration, we focus on a particular time point t_i and consider the input vector X as follows:

$$X = \{x_{i-U'+1}, ..., x_{i-l'}, ..., x_i\}, l' \in U', \tag{1}$$

wherein $x_{i-l'} = \{Region^1_{t_{i-l'}}, Region^2_{t_{i-l'}}, ..., Region^{n'}_{t_{i-l'}}\}$ represents the Population of each Region $n \in N'$ at the time $t_{i-l'}$. Similarly, we model the output vector Y, which is characterized as follows:

$$Y = \{y_{i+1}, ..., y_{i+l}, ..., y_U\}, l \in U, \tag{2}$$

wherein $y_{i+l} = \{Region^1_{t_{i+l}}, Region^2_{t_{i+l}}, ..., Region^n_{t_{i+l}}\}$ represents the population of each Region $n \in N$ at the time t_{i+l}.

Since our work aims to expand upon spatio-temporal graph neural networks, it is vital to convert the prior problem formulation to graph format. There have been numerous variations in terms of types of graphs. Arguably, the most significant distinction among these variations is whether the considered graph structures are static or dynamic. Dynamic graphs can be classified into Discrete-Time Dynamic Graphs (DTDG) (57) and Continuous-Time Dynamic Graphs (CTDG) (58). We have chosen the DTDG approach to formulate regional traffic dynamically. According to the DTDG paradigm, a dynamic graph is defined as a sequence of snapshots of a static graph. Each snapshot corresponds to a specific time-step t, the duration of which is referred to as t_{window}. These snapshots construct a temporal continuum that enables the emergence of temporal patterns and phenomena. Furthermore, each one of the aforementioned static graphs consists of multiple nodes and edges that encapsulate the underlying spatial relations. In the context of this work, each graph corresponds to an area in two-dimensional space divided into N Regions that are being traversed by P Populations at each time-step t.

Given an undirected graph $\overline{G}$ that consists of N nodes and E edges. The graph nodes correspond to the Regions, and the graph's edges correspond to how likely it is for a member of a Population to move from one Region to another within the time frame of a singular time step t. The following two matrices can describe this graph:

- A weighted **Adjacency Matrix A** $\in \mathbb{R}^{N\times N}$ that incorporates edge weights wij.
- A **Feature Matrix Z** $\in \mathbb{R}^{N\times F}$, where F corresponds to the dimension of each Feature Vector.

The Feature Matrix can be viewed as the total of the Feature Vectors. Each of the N rows of the Feature Matrix corresponds to a Feature Vector that describes node-level features. In the context of this work, each node (Region) is described by a Feature Vector whose dimension is equal to $U^{'}$, which are the traffic values recorded at the corresponding Region during the last $U^{'}$ time-steps, and thus $F = U^{'}$. Furthermore, each of the $U^{'}$ columns of the Feature Matrix Z corresponds to a different time-step t of the input sequence. This formulation enables instances of the Feature Matrix to be modeled as time-series data. The Adjacency Matrix A is static and, thus, remains unchanged throughout the various time steps since it models the statistical possibility of moving from one Region to another. On the other hand, the Feature Matrix Z is dynamic and is different for each time-step t. Subsequently, snapshots of the Feature Matrix Z are conceptually equivalent to the aforementioned input vector $X \in \mathbb{R}^{N\times U^{'}}$.

Table 2
Notations used in this paper.

Notations	Descriptions
$\mathbb{R}^n$	n-dimensional euclidean space
N	number of Regions
$Regions$	set of Regions
$Border_n$	set of $Region_n$'s Borders
$Center_n$	Center of $Region_n$
D	the average distance between the Centers
P	number Populations
$Populations$	set of Populations
$Speed_p$	average speed of $Population_p$
U	number of input time-steps
U'	number of prediction time-steps
t_{window}	duration of each time-step
F	dimension of Feature Vector
Z	Feature Matrix
D	Average distance between two Region Centers
$\overline{G}$	Undirected Graph
A	Adjacency Matrix
I	Identity Matrix
w	Edge Weights
K	number of stacked GCN layers
W	learnable weight matrix
$\|X\|$	the number of elements in a given set X

4. Proposed Solution

This work aims to expand upon the GCN-LSTM architecture, depicted in Fig. 1, in a manner that may enable the incorporation of information regarding the various Populations, as well as the Regions that they traverse, to produce more refined and accurate prediction models.

Towards achieving this goal, the proposed solution consists of three components that are closely intertwined with each other. These components are the following:

- the **WEST GCN-LSTM**
- the **Shared Borders Policy**
- the **Adjustable Hops Policy**

The proposed WEST GCN-LSTM is a novel architecture that extends the conventional GCN-LSTM architecture by facilitating multiple weighted stacked graph convolution layers based on a weighted Adjacency Matrix A and the number of stacked graph convolution layers that is denoted by K for regional traffic forecasting. According to our solution, A and K are calculated using the Shared Borders Policy and the Adjustable Hops Policy, respectively. The Shared Borders Policy is designed to leverage information regarding the Borders of the Regions. In contrast, the Adjustable Hops Policy is designed to leverage information regarding the Speed of Populations, the Centers of the Regions, and the number of prediction steps. This section explains these three components in more detail.

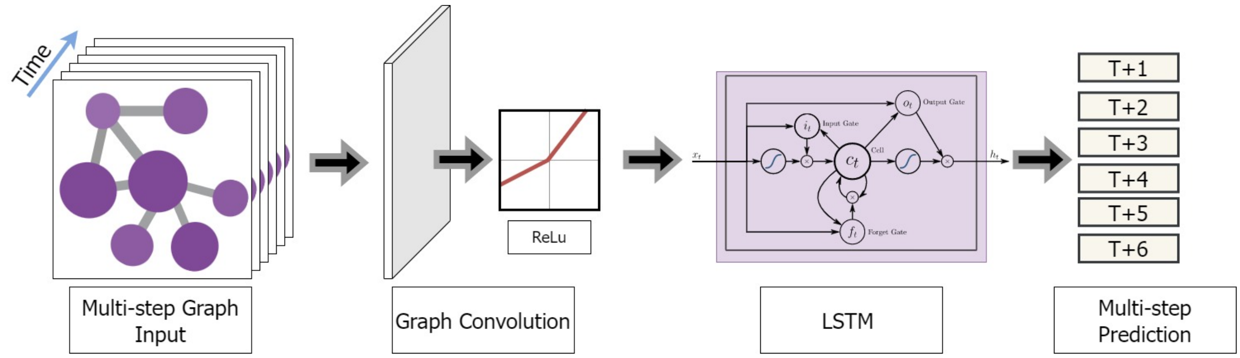

Figure 1: GCN-LSTM model architecture.

4.1. WEST GCN-LSTM

WEST GCN-LSTM is a spatio-temporal graph neural network based on the ED architectural paradigm. As the name suggests, the encoder is based on weighted stacked GCNs, and the decoder is based on LSTMs. To establish the foundations of the proposed solution, it is necessary to delve briefly into the mechanisms of the WEST GCN-LSTM model.

4.1.1. WEighted STacked (WEST) Graph Convolutional Networks (GCN)

GCNs leverage the graph structure to aggregate information from neighboring nodes. In this step, each node collects information from its neighbors, including itself, to update its feature representation. This is achieved through a linear transformation and aggregation process. After aggregating neighbor information, a non-linear activation function (ReLU) is applied to generate the aggregated representation h. It is paramount to note that the number of stacked GCN layers equals the number of hops in terms of neighbors that the aggregation process can encompass during each iteration. For instance, when a GCN is constructed with just a single convolution layer, nodes can exclusively access their immediate neighbors for aggregating representation data. In the frame of weighted stacked graph convolution, the convolutional layers are influenced by the weighted Adjacency Matrix A, while the quantity of stacked graph convolution layers is denoted as K. Thus, in the context of weighted stacked graph convolutions, h is calculated in the following manner:

$$h^{(0)} = (Z) \qquad \text{for } k = 0 \tag{3}$$

$$h^{(k)} = \sigma(\hat{D}^{-\frac{1}{2}}\hat{A}\hat{D}^{-\frac{1}{2}}h^{(k-1)}W^{(k-1)}) \qquad \text{for } k = 1, \dots, K \tag{4}$$

,where $h^{(k)}$ corresponds to the aggregated representation after k convolutional layers, and $h^{(0)} = Z$ corresponds to the Feature Matrix. Furthermore, $\hat{A}$ is equal to $A + I$, and its purpose is to incorporate self-connections, $\hat{D}$ is the diagonal Degree Matrix of $\hat{A}$, W signifies a dedicated learnable weight matrix and σ represents the ReLU function.

The use of $\hat{A} = A + I$ ensures the inclusion of self-loops, which are crucial for retaining a node's original features during the aggregation process. This step prevents the features from being entirely diluted by contributions from neighboring nodes during message passing. Additionally, normalizing the adjacency matrix through $\hat{D}^{-\frac{1}{2}}\hat{A}\hat{D}^{-\frac{1}{2}}$ addresses imbalances in node degree, ensuring that information aggregation remains equitable across nodes regardless of their connectivity. This normalization aligns with principles from spectral graph theory, enhancing the stability and effectiveness of the learning process.

The weighted stacking mechanism in WEST-GCN introduces further flexibility by allowing dynamic adjustment of layer-specific weights, represented by $W^{(k-1)}$. These learnable weights enable the model to capture complex dependencies across layers, which is particularly valuable in graphs with hierarchical or heterogeneous structures. Furthermore, stacking K layers extends the model's receptive field, allowing nodes to aggregate information from neighbors up to K hops away. This capability is essential for capturing global structural patterns but introduces a tradeoff, as excessive stacking can lead to over-smoothing, where node representations across the graph become indistinguishable. The balance of K is, therefore, critical to model performance.

The application of the ReLU activation function (σ) introduces the necessary non-linearity into the model, enabling it to learn complex feature representations. Without this non-linearity, the model would be limited to performing linear transformations, which would significantly restrict its expressive power. Additionally, initializing $h^{(0)}$ as the feature matrix Z serves as a logical starting point, reflecting the raw input features before they are iteratively refined through graph convolution. The progression of $h^{(k)}$ across layers demonstrates how the model systematically integrates both local and global graph structures into the feature representations. The weighted adjacency and explicit control over layer stacking are deliberate design choices that enhance both the interpretability and performance of the model. Weighted adjacency allows for the incorporation of domain-specific insights, such as edge importance, into the graph structure, while the modular stacking mechanism facilitates customization for a wide range of applications.

4.1.2. Long Short Term Memory (LSTM)

LSTM networks, similar to their precursor, employ the Hidden State mechanism to facilitate the representation of dynamic temporal behavior. The unique aspect of LSTM networks is their utilization of the Cell State structure. This architecture introduces Cell State manipulation through regulatory mechanisms known as Gates. Each LSTM node encompasses three gate-related elements, all incorporating sigmoid layers to ensure differentiability within the $0-to-1$ range. The sigmoid activation function scales values to facilitate data importance assessment and decision-making regarding retention or omission. Gate structures incorporate two sets of weight matrices, denoted as W and U, associated with Hidden State and input, and additional matrices for Cell State. The input X_t corresponds to timestamp t. Gates employ these matrices, along with input and prior Hidden State ($hidden_{t-1}$).

The Forget Gate determines which historical information from past timestamps is to be excluded from the Cell State. Its output is computed using Eq. 5. The Input Gate evaluates the significance of recent input, updating Cell State using Eq. 6. Cell State calculation employs the $\overline{\mathrm{C}}$ vector, generated as per Eq. 7, with tanh activation mitigating gradient issues. The Cell State update process is described in Eq. 8, combining the output of the Forget Gate and the Input Gate with $\overline{\mathrm{C}}$. The Output Gate computes the subsequent hidden state using Eq. 9. The new Hidden State is calculated using Eq. 10. Updated Cell State and Hidden State are then propagated to subsequent LSTM nodes for the next time-step (59).

$$\text{forget}_t = \text{sigmoid}(X_t \cdot W_f + \text{hidden}_{t-1} \cdot U_f) \tag{5}$$

$$\text{input}_t = \text{sigmoid}(X_t \cdot W_i + \text{hidden}_{t-1} \cdot U_i) \tag{6}$$

$$\overline{\mathrm{C}} = \tanh(X_t \cdot W_c + \text{hidden}_{t-1} \cdot U_c) \tag{7}$$

$$C_t = \text{forget}_t \cdot C_{t-1} + \text{input}_t \cdot \overline{\mathrm{C}}_t \tag{8}$$

$$\text{output}_t = \text{sigmoid}(X_t \cdot W_o + \text{hidden}_{t-1} \cdot U_o) \tag{9}$$

$$\text{hidden}_t = \text{output}_t \cdot \tanh(C_t) \tag{10}$$

The use of the sigmoid activation function across all gate operations (Eqs. 5, 6, and 9) ensures differentiability while constraining outputs to the $[0,1]$ range. This bounded range facilitates probabilistic weighting, allowing the network to dynamically assess the importance of features. This capability is essential for mimicking binary decisions while maintaining the smoothness required for gradient-based optimization. Additionally, the hyperbolic tangent activation in Eq. 7 normalizes values to the $[-1,1]$ range, thereby mitigating the vanishing gradient problem that plagued earlier architectures like vanilla RNNs.

The gate mechanisms—Forget Gate, Input Gate, and Output Gate— described in Eqs. 5 through 10, provide a modular framework for memory manipulation across time steps. The Forget Gate (Eq. 5) adaptively determines which parts of the previous state should persist, ensuring that irrelevant information is discarded. This dynamic management of memory is critical in tasks like time-series forecasting, where the importance of past and current contexts changes continuously. The Cell State plays a pivotal role by acting as a conduit for gradient flow during backpropagation. By incorporating contributions from both past states (C_{t-1}) and current computations (C_t), the LSTM ensures stable and efficient training even for long sequences.

4.1.3. WEST GCN-LSTM

We combined weighted stacked GCN and LSTM layers to predict regional traffic. This fusion of spatial and temporal DL layers is called WEST GCN-LSTM. Weighted stacked GCNs constitute the encoder and are designed to extract structural characteristics of the input sequence, producing an aggregated representation. This process is carried out in the following manner:

$$h^{\text{encoder}} = \text{WEST GCN}_{\text{encoder}}(Z, A) \tag{11}$$

Here, h^{encoder} represents the aggregated representation after applying weighted stacked graph convolution, A is the weighted Adjacency Matrix of the graph, and Z is the Feature Matrix. This representation is then fed as input to the LSTM part of the model, thus capturing temporal patterns at the graph snapshot level. The LSTM part of the model, acting as a decoder, produces the desired predictions in the following manner:

$$Y = \text{LSTM}_{\text{decoder}}(h^{\text{encoder}}) \tag{12}$$

$\text{LSTM}_{\text{decoder}}$ refers to the LSTM network that takes the aggregated representation from the encoder as input to produce an output that is then passed through a dense layer to generate the multi-step predictions. In the frame of multi-step time-series forecasting, the WEST GCN-LSTM model takes as input a sequence of graph signals, where each signal corresponds to a different time-step and is represented as a graph signal on a fixed graph. The goal is to predict the future values of the time series based on the graph signals of previous time steps.

4.2. Shared Borders Policy

The graph's edges play an integral role in representing the spatial relations between the Regions by dictating which nodes shall partake in the feature aggregation process. The authors of this paper propose a novel approach by introducing the Shared Borders policy. The Shared Borders policy is based on the reasonable assumption that the greater the lengths of the shared borders are, the more statistically likely it is for a larger percentage of a Population to traverse them.

According to this approach, the Adjacency Matrix A shall be constructed based on the lengths of the Borders that are being shared between each pair of Regions (nodes). The process of calculating the length of the shared Borders between two Regions is presented in Alg. 1. In case the two

Regions are not neighbors, then the corresponding matrix elements shall be equal to 0. Furthermore, the adjacency matrix's diagonal elements shall equal the length of the perimeter of the perspective Region. By doing so, we ensure that the ongoing traffic of a particular Region will be the main factor in predicting its corresponding future state. At the same time, the rest of the Regions will influence the prediction results to a degree associated with the Borders that they share with the aforementioned Region. The Shared Borders Policy is presented in Alg. 2. Upon the construction of the Adjacency Matrix A using the Shared Borders Policy, normalizing the resulting values in the $0 - to - 1$ range is required.

The main idea behind the incorporation of the Shared Borders Policy is that by enabling only the neighboring Regions to partake in the feature aggregation process, we can establish a distilled version of the spatial correlations that are inherent in topological structures. By doing so, we are alleviating part of the complexity that would otherwise be imposed on the forecasting model. Furthermore, by incorporating the aforementioned weights, we can establish a more refined encapsulation of the spatial dependencies that manifest.

4.3. Adjustable Hops Policy

The incorporation of a wide range of features is of paramount importance in the context of enabling the WEST GCN-LSTM model to conduct accurate predictions. In the previous subsection, we showcased how the lengths of the shared Borders can be leveraged to construct the adjacency matrix A. In this subsection, we shall focus on showcasing how the Speed of a Population can be leveraged to calculate K, which refers to the number of stacked graph convolution layers incorporated at each constructed WEST GCN-LSTM model.

Let us consider three distinct parameters. The first one is the time between two consecutive predictions or observations, corresponding to the selected time duration t_{window}. The second one is the average speed of a specific Population, referred to as $Speed_p$. The third one is the average distance D that has to be traversed to transit from the Center of one Region to the Center of one of its neighboring Regions.

When the prediction process is being implemented on the basis of a singular Population, one is able to select the appropriate t_{window} accordingly in a manner that facilitates the establishment of consistent observations. For instance, if a population moves at a significantly high speed, the chosen time between two consecutive observations can be decreased. Unfortunately, when multiple Populations occupy a space, each one moving at a different Speed, a problem arises since, to formulate comparative analyses between them, it is necessary to utilize the same t_{window} parameter across the various Populations. The WEST GCN-LSTM model, when leveraging the aforementioned Shared Borders Policy, receives as input only the Regional Traffic that corresponds to neighboring Regions. In other words, when using a conventional GCN-LSTM model, the observed Populations should be able to traverse one Region at most during one time-step. As a result, in cases where a subset of the Population is able to traverse multiple Regions within a singular time step, a GCN-LSTM model that is using the Shared Borders Policy shall lose its advantage since there will be a potentially significant loss of information.

To mitigate this issue, we propose the Adjustable Hops Policy. According to the paradigm of GCNs, the number of graph convolution layers that are being deployed within a singular model corresponds to the number of aggregation hops that are being conducted each time the convolution process takes place. So, in the case of one convolution layer, only the first-degree neighbors are taken into consideration; in the case of two convolution layers, the first and second-degree neighbors are taken into consideration, etc.

Algorithm 1 LengthOfSharedBorder Algorithm

Description: Algorithm for calculating the length of the overlapping section between two line segments in two-dimensional space.
Input: The endpoints of the two line segments: $(x_1, y_1), (x_2, y_2)$ for the first line segment and $(x_3, y_3), (x_4, y_4)$ for the second line segment.
Output: The $length$ of the overlapping segment between the two line segments.
Begin algorithm
1. If $x_1 > x_2$:
$x_1, x_2 = x_2, x_1$
$y_1, y_2 = y_2, y_1$
2. If $x_3 > x_4$:
$x_3, x_4 = x_4, x_3$
$y_3, y_4 = y_4, y_3$
3. Calculate the slopes of both line segments using the formulas: $slope_1 \leftarrow \frac{y_2-y_1}{x_2-x_1}$ and $slope_2 \leftarrow \frac{y_4-y_3}{x_4-x_3}$. If $slope_1 = slope_2$, then the two line segments are parallel. If they are indeed parallel, proceed to step 4, else $length = 0$, and proceed to step 8.
4. If $\max(x_1, x_3) \leq \min(x_2, x_4)$, then the two line segments overlap. In that case, proceed to step 5, else $length = 0$ and proceed to step 8.
5. Calculate x_e and x_s, which are x coordinates of the endpoints of the overlapping segment, by using the following formulas:
$x_s = \max(x_1, x_3)$
$x_e = \min(x_2, x_4)$
6. Calculate y_e and y_s, which are y coordinates of the endpoints of the overlapping segment, by using the following formulas:
$y_s = y_1 + \frac{(x_s-x_1)\cdot(y_2-y_1)}{x_2-x_1}$
$y_e = y_3 + \frac{(x_e-x_3)\cdot(y_4-y_3)}{x_4-x_3}$
7. Calculate the $length$ of the overlapping segment by using the following formula:
$length = \sqrt{(x_e - x_s)^2 + (y_e - y_s)^2}$
8. Return $length$.
End algorithm

Algorithm 2 Shared Borders Policy Algorithm.

Input: The N Regions, each represented as a list of 4-tuples. The size of each list of 4-tuples is equal to the number of edges that a particular Region has. Each 4-tuple corresponds to the x & y coordinates of the vertices connected by that particular edge.
Output: The weighted Adjacency Matrix A.
Begin algorithm
1. For each pair of lists (Regions) i, j:
2. Initialize $L \leftarrow 0$
3. For each pair of 4-tuples (edges) k, l:
4. $length \leftarrow LengthOfSharedBorder(tuple_k, tuple_l)$
5. $L \leftarrow L + length$
6. $A_i, j \leftarrow L$
7. Return A
End algorithm

The Adjustable Hops Policy commences by calculating the average distance between the Centers of the Regions and the identification of the Population with the lowest Speed. The parameter t_{window} is then adjusted in a manner that this specific Population shall be capable of traversing at most one Region during each observation time-step of duration equal to t_{window}. This Population shall be used as the baseline, and the dedicated GCN-LSTM model that corresponds to it shall have only one GCN layer since its members shall be able to traverse a single Region at most during each time-step. Using the same t_{window} parameter, calculate how many Regions at most each of the remaining Populations can traverse during a single time-stem. This number of Regions is equal to the number of GCN layers that shall be utilized for each of the corresponding Populations, which is denoted by K. The specifics of the Adjustable Hops Policy are presented in Alg. 3. It is worth mentioning that this policy results in the establishment of a dedicated prediction model for each distinct K that emerges across the various populations. Furthermore, in multi-step forecasting scenarios, like the ones that shall be explored in the next section of this work, t_{window} should be adjusted to take into account the U prediction steps that are being considered and thus need to be multiplied by $\frac{U}{2}$.

Algorithm 3 Adjustable Hops Policy Algorithm.

Input: The $Speed_p$ values, each one corresponding to one of the P Populations. The N 2-tuples correspond to the x & y coordinates of a $Center_n$ of a Region. The U' value corresponds to the number of prediction steps.
Output: The array K of size P, each element of which corresponds to the maximum number of Regions, members of each corresponding Population can traverse during a singular time-step, whose duration is equal to T.
Begin algorithm
1. Initialize $T \leftarrow 0$
2. Initialize $MinSpeed \leftarrow Speed_0$
3. Initialize $D \leftarrow 0$
4. For each pair of 2-tuples ($Center_n$) k, l:
5. $D = D + \sqrt{(x_l - x_k)^2 + (y_l - y_k)^2}$
6. $D = \frac{D}{N^2}$
7. For each $Speed$ value p:
8. If $Speed_p < MinSpeed$, then: $MinSpeed \leftarrow Speed_p$
9. If $U' > 1$:
10. $t_{window} \leftarrow \frac{D}{Minspeed} \cdot \frac{U'}{2}$
11. else:
12. $t_{window} \leftarrow \frac{D}{Minspeed}$
13. For each $Speed$ value p:
14. $K_p = \text{round}\left(\frac{Speed_p \cdot t_{window}}{D}\right)$
15. Return K
End algorithm

4.4. Component Synergy

The synergy among the components of the proposed solution is evident in how each module complements the others to ensure accurate spatio-temporal predictions. As depicted in Fig. 2, the architecture begins with the spatial representation of Regions and Populations. Regions, defined by their Borders, form the foundation for the Shared Borders Policy, which is responsible for constructing the weighted Adjacency Matrix. This Adjacency Matrix reflects the spatial relationships by encoding the length of shared Borders between Regions, while self-connections are weighted based on regional perimeters. The Adjacency Matrix is then normalized and used to generate the Degree Matrix. Furthermore, the Feature Matrix is constructed using information regarding the Borders, and the location of members of the involved Populations.

The Adjacency Matrix, the Degree Matrix, and the Feature Matrix are leveraged by the WEST GCN, which aggregates features from neighboring nodes through multiple weighted stacked GCN layers. The range of this aggregation is dynamically controlled by the Adjustable Hops Policy, which determines the number of GCN layers. This policy relies on Population information, such as the average speed and the location of Population members, as well as the Centers of the Regions and the number of prediction steps. By adjusting the propagation range, the Adjustable Hops Policy ensures that the spatial aggregation accounts for both the mobility dynamics of the population and the temporal requirements of the prediction task. As such, the Adjacency Matrix, the Degree Matrix, the Feature Matrix, and the number of GCN layers constitute the input of the WEST GCN-LSTM forecasting model.

The collaboration between the Shared Borders Policy and the Adjustable Hops Policy ensures that spatial and temporal information is seamlessly integrated. The adjacency matrix created by the Shared Borders Policy encapsulates localized spatial dependencies, while the Adjustable Hops Policy tailors the depth of feature aggregation to capture broader spatial patterns as needed. These aggregated spatial features are then fed into the LSTM decoder, which models

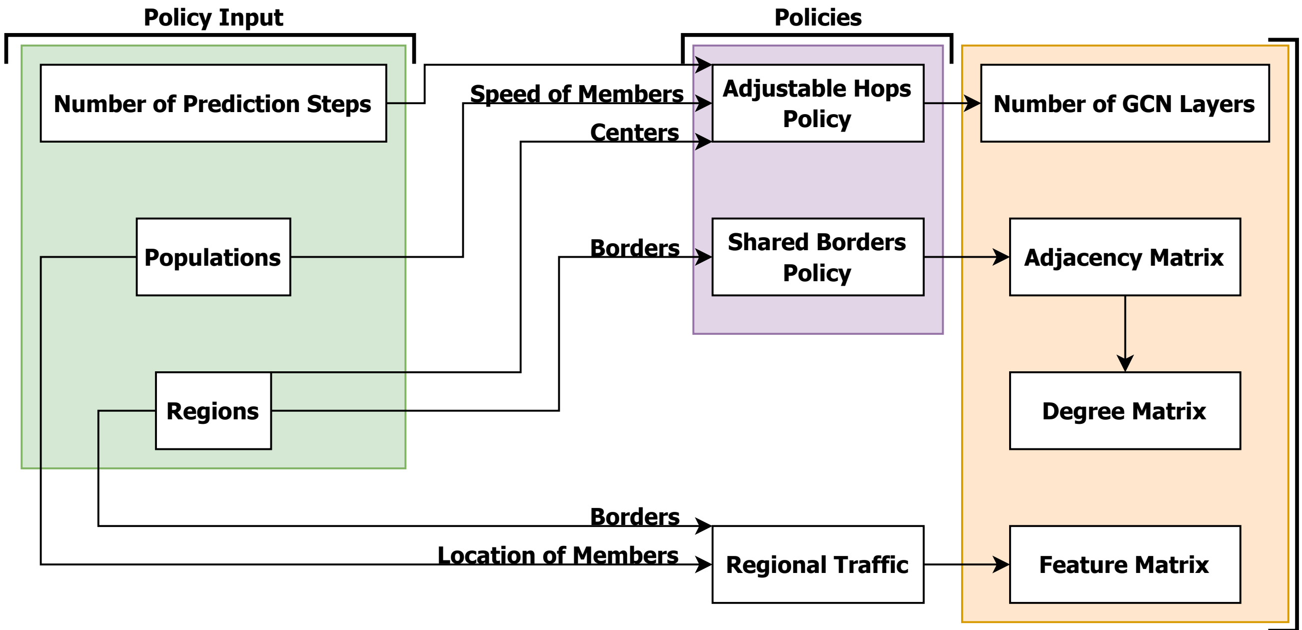

Figure 2: An overview of the proposed solution. The green rectangle corresponds to the information used by the policies (purple rectangle) in order to establish the input of the WEST GCN-LSTM forecasting model (orange rectangle).

temporal relationships over sequential graph snapshots. This decoder processes the output of the WEST GCN and predicts regional traffic by considering the temporal evolution of spatial features.

The interaction between Regions and Populations also plays a crucial role. Regional Borders define the static spatial structure, while Populations introduce dynamic elements such as their location and speed. Together, these factors influence the adjacency matrix and the depth of the GCN layers, creating a feedback loop between spatial and temporal components. This synergy allows the architecture to produce multi-step predictions that are both spatially and temporally informed, leveraging the complementary strengths of the Shared Borders Policy, Adjustable Hops Policy, and the WEST GCN-LSTM model.

5. Experimental Evaluation

This section is dedicated to evaluating the efficiency of the proposed solution. To achieve this goal, the experimental evaluation consists of 19 forecasting models. The forecasting models were designed and implemented via Python 3.9.13 and Tensorflow 2.9.1. Additionally, the Hardware Backend used for training and inference is an i5-11400 CPU and an NVIDIA GeForce RTX 3060 GPU.

Furthermore, the experiments were conducted using a combination of real and synthetic datasets. The synthetic ones were constructed using the Simulation of Urban MObility (SUMO) (60) framework. SUMO is a versatile traffic simulator with the capability to handle extensive mobility networks. It incorporates various modes of transportation, including pedestrians, and is equipped with many tools for generating diverse mobility scenarios. The simulator exhibits realistic features of pedestrian mobility, such as pedestrian-pedestrian interactions in close proximity, reasonable walking speeds, and natural movement patterns. Additionally, SUMO enables pedestrians to interact safely by implementing features such as collision avoidance.

To maintain consistency throughout the experimental evaluation and across the two datasets, the authors of this work have chosen to implement the following format selections for both the experimental protocol and the parameter choices that supported the assessment of the models. The space that each dataset covers was divided into 6 Regions. Each of the two datasets was split using the 80/20% ratio for training and testing, respectively. In the context of multi-step forecasting, 6 time-steps were considered for the input and output sequences.

5.1. Benchmark Datasets

To gain a comprehensive understanding of the proposed solution's capabilities, as well as of the experimental process that was employed to evaluate it, it is of paramount importance to delve deeper into the specifics of the datasets that were used. These datasets are a subset of the Berlin (Cycling) dataset and three Central Park (Pedestrian) datasets. The pedestrian ones were generated using SUMO.

As part of our study, we conducted a simulation of pedestrian traffic in the New York City area over seven days. Specifically, we focused on Central Park and the adjoining urban districts, modeling different traffic patterns. The generated dataset encompasses the movement attributes of $200,000 - 230,000$ individuals on a daily basis, delineating their position and velocity per second. Central Park and the adjoining urban districts offer pedestrians the opportunity to take breaks and explore the different attractions or smart city features available to them, as well as to temporarily pause at points of interest such as interactive public art installations, historical landmarks, or food trucks and markets. Additionally, given the park's function as a sports venue, the simulation incorporates variable pedestrian speeds that simulate jogging or running. The distribution of speeds aims to realistically capture the motility characteristics of pedestrians based on factors such as age, type of activity, and geospatial context. The t_{window} for this dataset is equal to 5 minutes. This dataset is referred to as *Central Park*.

Furthermore, the experimental evaluation process includes two additional datasets: *Central Park (Low)* and *Central Park (High)*. These two datasets are derived from the Central Park dataset by categorizing the moving entities based on their Speed. More specifically, Central Park (Low) consists of moving entities that can traverse at most one Region during a single time-step ($K = 1$). In contrast, the Central Park (High) dataset consists of moving entities that can traverse at most two Regions during a single time-step ($K = 2$). In other words, the aforementioned Central Park dataset is the amalgamation of the Central Park (Low) and Central Park (High) datasets. It is worth mentioning that these two datasets contain a similar total number of recorded moving entities (Central Park (High) contains about 4% more moving entities compared to Central Park (Low)).

The *Berlin* Cycling Dataset (https://www.kaggle.com/datasets/phisinger/bike-counting-berlin), procured by the Berlin administration in Berlin city, Germany, provides a comprehensive understanding of the long-term developments in bicycle traffic, including trends and seasonal fluctuations, through the use of Automatic Permanent Counting Points (APCPs). The dataset includes 9 years of bicycle traffic counts (2012-2020) collected using induction loops and sensors at APCPs. Bicycles passing over the detection cross-sections of the counting points are counted as they cause changes in the induced electromagnetic field, which are subsequently analyzed by the sensor and recorded as counting pulses. For roads with separate bicycle traffic guidance in each direction of travel, one counting station is established for each direction of travel. In contrast, for cross-sections with shared bicycle traffic guidance in both directions of travel, one counting station is installed for both directions of travel. The continuous counting process identifies bicycles based on specific geometries detected by the sensors, while other vehicles are filtered out. The t_{window} for this dataset equals 1 hour.

To ensure a rigorous and diverse assessment of the proposed solution, the experimental evaluation process deliberately incorporates these datasets to account for varying conditions. By using datasets with distinct characteristics—such as the Central Park datasets, which simulate high-resolution pedestrian dynamics with 5-minute temporal intervals, and the Berlin Cycling dataset, which captures broader, long-term trends with 1-hour time steps—the evaluation spans a wide range of temporal and spatial complexities. These variations in dataset conditions, including differences in population movement speeds, regional sizes, and traffic patterns, help validate the adaptability and robustness of the solution. The Central Park datasets, with their detailed modeling of pedestrian behaviors, contrast sharply with the Berlin dataset's aggregated cycling data, emphasizing the proposed solution's capability to handle scenarios of differing granularities and modalities. This comprehensive approach ensures that the evaluation not only reflects real-world applicability but also rigorously tests the solution under diverse and challenging scenarios.

5.2. Region Construction

The datasets mentioned above can be leveraged to provide information regarding the Regions and the corresponding Regional Traffic that manifests during each time-step. Regarding Regional Traffic, both datasets enable direct access to the numerous recorded geolocation points throughout the simulation. However, choosing an appropriate partitioning approach is contingent upon the specific context and challenges encountered. While these datasets provide valuable information regarding numerous recorded geolocation points throughout the examined periods, they do not incorporate any notion of distinct spatial Regions. Thus, we had to construct the required Regions using the following approach. More specifically, we applied a k-means clustering algorithm to the various geolocation points in each training dataset. The k variable that corresponds to the k-means clustering algorithm was selected to be equal to $N = 6$. The coordinates of the created centroids were selected to serve as the corresponding Centers of the N Regions. Finally, dedicated Voronoi diagrams were created for each dataset using these N Centers. The representation of the Voronoi diagram, for a set of n points $P = \{(x_i, y_i)\}$, is achieved by using the following data structures:

- A list V of Voronoi regions, where each region is associated with one of the input points.
- A list E of Voronoi edges that define the boundaries between regions.

To calculate each Voronoi cell, one needs to find the bisectors between point (x_i, y_i) and all other points in P, and then to clip the edges to the bounding box. Then, to merge any overlapping edges in E to form the complete Voronoi diagram. This approach enables the examined Regions to encompass diverse geographic areas, exhibit varying population densities, and feature distinct mobility patterns, while maintaining a constant size throughout the experimentation process. On top of that, the selection of datasets encompasses two distinct urban mobility scenarios (pedestrian and cycling), with the aim of embracing a broader spectrum of urban applications and assessing the models' capacity and generalization power across various vector input spaces that exhibit diverse statistical properties and characteristics. These properties and characteristics include different temporal granularities in the context of the chosen t_{window}, the size of the Regions, the trend, seasonality, and volume of regional traffic. For instance, Fig. 3 presents a comprehensive and

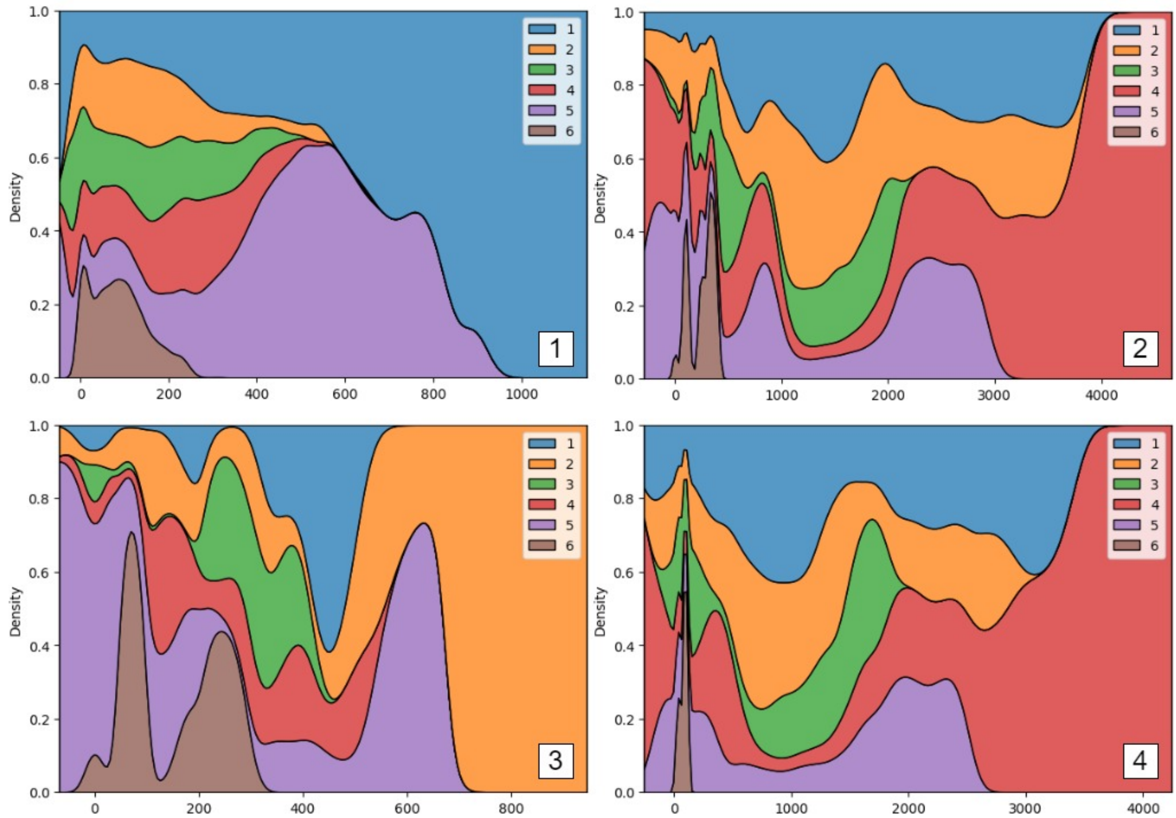

Figure 3: Densities of the regional traffic volumes that correspond to the (1) Berlin, (2) Central Park, (3) Central Park (Low), and (4) Central Park (High) datasets. Each of the 6 Regions is assigned a different color.

nuanced portrayal of the different densities of the regional traffic volume that corresponds to each examined dataset using normalized Kernel Density Estimate (KDE) plots. KDE plots are a method for visualizing the distribution of observations in a dataset by representing the data using a continuous probability density curve in a given number of dimensions. As the figure indicates, there is a significant diversity in regional traffic volume distribution across the various datasets and underlying Regions. This type of diversity of experimental conditions is vital in evaluating the efficiency of the proposed solution in a robust manner.

5.3. Assumptions, Limitations, and Potential Biases

To ensure a thorough and transparent evaluation of our proposed solution, it is crucial to acknowledge the assumptions, limitations, and potential biases that may influence the results. In the Berlin Cycling dataset, the population count is derived from APCPs using induction loops and sensors. While these provide an estimation of the number of cyclists in each zone, the actual number of individuals may differ due to potential inaccuracies in detection. Factors such as sensor malfunctions, environmental interference, and the inability to differentiate between closely spaced bicycles can introduce variability in the recorded counts.

The model assumes that the time required for an individual to traverse from one region to its neighboring region is directly proportional to the distance between the centers of the respective regions. This assumption simplifies mobility dynamics but may not accurately capture variations in actual movement patterns influenced by factors such as traffic congestion, pedestrian density, and terrain variations. Additionally, the calculation of the weighted adjacency matrix A relies on the Shared Borders Policy, which considers the direct spatial connectivity between regions. However, the model does not explicitly account for real-world obstacles such as buildings, traffic lights, and restricted access areas that could influence movement pathways. As a result, the adjacency representation may not fully reflect real-world navigation constraints, potentially impacting the accuracy of trajectory predictions.

The pedestrian datasets utilized in this study are generated using the SUMO framework. Although SUMO is a robust traffic simulator capable of modeling diverse mobility scenarios, it remains a synthetic representation of reality. The simulation parameters, such as pedestrian speed distributions, interaction mechanisms, and environmental settings, are approximations that may not precisely replicate real-world behaviors. Moreover, biases in the simulation settings, including assumptions regarding pedestrian decision-making and movement preferences, could introduce systematic variations in the dataset.

By recognizing these assumptions and limitations, we aim to provide a balanced and objective assessment of our proposed solution.

5.4. Benchmark Models

Although ED architectures, when leveraged for multi-step regional traffic forecasting in a single region, manage to outperform their competitors, the situation shifts dramatically when exploring multiple regions simultaneously in a multi-step fashion, as the complexity of input and output sequences increases significantly. In such multi-regional scenarios, ED architectures appear to lose their distinct competitive edge compared to other methods, such as those based on Linear Regression (LR) (37) and Machine Learning (ML) (38), depending on the specific characteristics of the dynamic systems arising from the corresponding regional traffic conditions. The sharp rise in input-output complexity seems to create a performance plateau that unevenly impacts different forecasting approaches. As such, toward evaluating the efficiency of the proposed solution, on top of the aforementioned diversity of experimental conditions (in terms of complexity and system characteristics), we have included a diverse ensemble of peer competitors that are based on the LR, ML, and ED paradigms. By following this approach, we are able to not only evaluate the efficiency of the proposed solution against other state-of-the-art ED architectures but also against numerous notable LR and ML architectures in order to guarantee that the superiority of the proposed solution is established across a wide range of experimental conditions.

In terms of LR models, we have included LASSO (61), Ridge (62), Elastic Net (63), Lasso Lars (64). LASSO (Least Absolute Shrinkage and Selection Operator) applies L1 regularization, enforcing sparsity by shrinking some regression coefficients to zero, effectively selecting only the most relevant features. Ridge regression, in contrast, utilizes L2 regularization to penalize large coefficients, reducing overfitting and mitigating multicollinearity. Elastic Net combines L1 and L2 penalties, offering a balance between feature selection and coefficient shrinkage, making it particularly useful when dealing with highly correlated predictors. Lasso Lars efficiently computes LASSO solutions by incrementally updating coefficients, making it computationally advantageous for high-dimensional data.

In terms of ML models, we have included KNN (65), Decision Trees (66), Tree Regression (67), Bagged Decision Trees (68), Random Forest (69), Extra Trees Regressor (70)). Nearest Neighbors (KNN) is a non-parametric model that predicts outputs by averaging the k-nearest training instances in feature space, making it highly dependent on the choice of k and the distance metric used (65). Decision trees build hierarchical structures by recursively partitioning data based on feature thresholds, offering an interpretable yet often overfitting-prone model. Tree regression, a specialized form of decision trees for continuous outputs, divides the feature space into homogeneous regions to improve predictive accuracy. To enhance the robustness of decision trees, bagged decision trees employ bootstrap aggregation (bagging), training multiple trees on different subsets of data and averaging their predictions to reduce variance and improve generalization. Random Forest, an extension of bagging, further diversifies tree predictions by selecting random subsets of features at each split, enhancing model stability and accuracy. Extra Trees Regressor, similar to Random Forest, introduces additional randomness in feature selection and split threshold determination, further reducing variance and improving model robustness, particularly in high-dimensional spaces.

In terms of ED models, we have included the LSTM ED (32), the BD-LSTM ED (33), the CNN-LSTM (34), the Hybrid LSTM ED (35), and the Hybrid LSTM ATT ED (36). LSTM ED is a sequence-to-sequence architecture that leverages LSTMs units for both encoding and decoding time-series data. The encoder processes an input sequence over time and compresses it into a fixed-size context vector, which serves as an internal representation of the sequence. The decoder then utilizes this context vector to generate an output sequence, predicting future time steps based on the learned patterns. This model is particularly effective in capturing long-range dependencies, making it useful for time-series forecasting and sequence modeling tasks. Bidirectional LSTM ED extends the standard LSTM ED by incorporating Bidirectional LSTM layers in the encoder, allowing the model to process input sequences in both forward and backward directions. This bidirectional structure enables the encoder to capture both past and future dependencies simultaneously, leading to a richer internal representation. The decoder, typically composed of unidirectional LSTM units, then generates the predicted sequence using this enhanced context vector. By leveraging information from both temporal directions, the Bidirectional LSTM ED improves accuracy in capturing complex sequential patterns and dependencies. CNN-LSTM ED combines CNNs for feature extraction with LSTMs for sequence modeling. Since CNNs are effective at automatic feature learning, this model applies 1D convolution layers to capture spatial dependencies in the input sequence, followed by max-pooling to condense relevant features. The extracted features are then processed by LSTM layers to capture temporal dependencies, allowing the model to learn both spatial and sequential patterns efficiently. Hybrid LSTM ED integrates bidirectional and unidirectional LSTM layers to capture richer temporal dependencies. The encoder consists of a bidirectional LSTM followed by a unidirectional LSTM, allowing the model to leverage both past and future information at each time step. The decoder mirrors this structure, enhancing the ability to reconstruct the underlying temporal motifs from the encoded representation and improving sequence-to-sequence prediction performance. Hybrid LSTM Attention ED extends the Hybrid LSTM model by incorporating self-attention layers in both the encoder and decoder. The encoder first applies a bidirectional LSTM, followed by an attention mechanism that enhances important temporal features before passing them to a unidirectional LSTM. Similarly, the decoder utilizes an attention layer to refine the output sequence, ensuring that significant temporal dependencies are emphasized, thereby improving predictive accuracy and interpretability.

Furthermore, the benchmark models ensemble includes the three variations of the GCN-LSTM model used in regional traffic forecasting. These variations were constructed based on the aforementioned categorization of related works on regional traffic forecasting using spatio-temporal GNNs based on their corresponding spatial and temporal aspects, presented in Table 1. The first one constructs the adjacency matrices based on the distance between the nodes of the graph (GCN-LSTM (centers)), the second one creates the adjacency matrices based on traffic pattern similarity matrices (GCN-LSTM (traffic)), and the third one establishes binary adjacency matrices based on whether or not the involved nodes are neighboring (GCN-LSTM (binary)). The latter of these variations also serves as part of an ablation study that investigates the performance of the proposed solution by removing certain components to understand the component's contribution to the overall system. As stated before, alongside the rest of the peer competitors, the proposed WEST GCN-LSTM is also tested against a GCN-LSTM model that uses binary graph convolution based on whether or not the examined Regions are neighboring. Furthermore, aside from the fully implemented WEST GCN-LSTM that leverages both proposed policies, we also explore a version of the WEST GCN-LSTM denoted as WE GCN-LSTM that solely relies on the Shared Borders Policy. The WE GCN-LSTM is suitable for Regional Traffic forecasting scenarios that do not incorporate the velocities of the various Populations, such as the Berlin dataset. Aside from the aforementioned dataset, we also considered a version of the Central Park dataset that does not include the velocities of the Populations. This ablation study serves as the third and final pillar that guarantees the robustness of the experimental evaluation process. Following the aforementioned ablation study, the experiments are designed to evaluate the efficiency of the proposed solution in two distinct contexts: one that leverages both of the proposed policies (WEST GCN-LSTM) and one that uses only the Shared Borders Policy (WE GCN-LSTM).

The implementation of all RL & ML models was performed using scikit-learn. The implementation of all ED models was performed using the corresponding scientific literature as a reference. The hyper-parameters of all examined forecasting models were set using contemporary tools such as KerasTuner. More specifically, for LASSO, Ridge, Elastic Net, and Lasso Lars, we optimized alpha (regularization strength), the L1 ratio (for Elastic Net), the fit intercept, and the solver. For KNN we optimized the number of neighbors, the distance metric, and the weight function. For Decision Trees, Tree Regression, Bagged Decision Trees, Random Forest, and Extra Trees Regressor, we optimized the max depth, the min samples split, the min samples leaf, the max features, and the criterion. Finally, for the ED models, we optimized the number of LSTM units, the dropout rate, the learning rate, and the batch size.

5.5. Evaluation Metrics

The proposed model's performance is assessed using Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE) as evaluation metrics. These metrics are suitable for evaluating predictions of continuous numbers.

MAE measures the average magnitude of errors in a set of predictions as it is given in Eq. 13 where y_i are the ground truth values and $\hat{y}_i$ are the predictions. At the same time, n is used to denote the total number of predictions in the evaluated/given set.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{13}$$

MSE is a commonly used metric for evaluating the average squared difference between observed values and predicted values in regression problems and is given in Eq. 14.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{14}$$

Finally, RMSE is the squared version of MSE; it is used as a measure for the standard deviation of the prediction errors, and it is given in Eq. 15.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{15}$$

Combining MAE, MSE, and RMSE provides a more comprehensive evaluation of a predictive model. Each metric captures different aspects of the model's performance, and using all three can offer a more nuanced understanding. MAE measures the average absolute errors. MSE measures the average squared errors, giving more weight to larger errors. RMSE is the square root of MSE, providing an interpretable metric in the same unit as the data. The lower these metrics are, the more efficient the corresponding forecasting model is.

5.6. Experimental Results

The evaluation process is conducted as a comparative analysis against peer competitors. More specifically, Table 3 displays the results of the WE GCN-LSTM model, compared against the various peer competitors in terms of MSE, RMSE, and MAE. These results correspond to the Berlin dataset and Central Park dataset. The various forecasting models were trained and hyper-tuned independently for each of these datasets. The displayed MSE, RMSE, and MAE values represent the average results (corresponding to the 6 prediction time-steps and Regions) for each combination of forecasting models and datasets.

Table 3
Experimental results for the use of the Shared Borders Policy (WE GCN-LSTM).

Model	Berlin			Central Park		
	MSE	RMSE	MAE	MSE	RMSE	MAE
LASSO	6813.74	82.576	55.856	28521.76	168.795	84.334
Ridge	6814.95	82.579	55.862	28550.85	168.974	84.462
Elastic Net	6813.86	82.576	55.855	28538.94	168.857	84.362
Lasso Lars	13969.78	118.304	91.777	19989.29	141.357	60.033
KNN	3040.22	55.111	33.134	148802.20	385.707	192.281
Decision Tree	5827.81	76.231	44.214	140607.77	375.077	181.793
Tree Regression	6746.09	82.151	47.549	143243.34	378.228	185.868
Bagged Decision Trees	3062.17	55.336	34.217	16767.66	409.477	194.014
Random Forest Reg	2742.04	52.215	34.337	198535.42	446.687	202.097
Extra Trees Regressor	2934.09	54.107	35.935	206976.72	454.603	183.070
LSTM ED	3678.18	60.565	42.507	145619.28	381.796	158.338
BD-LSTM ED	2802.66	52.886	36.302	169972.55	412.567	199.281
CNN-LSTM	3243.21	56.903	36.566	127684.06	355.516	123.458
Hybrid LSTM ED	5063.64	71.291	49.423	49496.91	221.573	158.938
Hybrid LSTM ATT ED	5872.85	76.589	51.105	59756.66	244.397	169.515
GCN-LSTM (traffic)	2515.62	50.156	31.744	14717.40	121.342	46.102
GCN-LSTM (centers)	2914.56	53.969	33.730	21114.36	145.459	56.010
GCN-LSTM (binary)	2332.97	48.263	30.354	16376.95	127.857	49.175
WE GCN-LSTM (ours)	**1989.07**	**44.599**	**26.639**	**11718.71**	**108.253**	**42.594**

Table 4
Experimental results for using both policies (WEST GCN-LSTM).

Model	Central Park					
	Low			High		
	MSE	RMSE	MAE	MSE	RMSE	MAE
LASSO	1957.84	44.275	27.170	38499.57	196.233	87.155
Ridge	1956.19	44.318	27.219	38694.22	196.735	87.312
Elastic Net	1956.68	44.289	27.187	38545.71	196.424	87.243
Lasso Lars	2288.14	47.848	34.168	32977.04	181.691	68.060
KNN	5153.61	71.738	47.242	211533.68	460.262	157.145
Decision Tree	7798.36	88.195	52.955	220273.80	468.947	171.820
Tree Regression	7892.61	88.865	55.491	205569.80	453.318	159.357
Bagged Decision Trees	4199.69	64.813	41.567	215214.93	464.097	168.351
Random Forest Reg	3947.21	62.900	39.293	216241.91	464.933	167.393
Extra Trees Regressor	3393.43	58.297	35.494	180371.66	424.022	152.694
LSTM ED	1401.66	37.422	27.564	63112.45	251.224	146.177
BD-LSTM ED	1440.24	37.943	27.442	57015.36	238.590	138.722
CNN-LSTM	1264.56	35.571	25.222	47348.78	217.745	122.168
Hybrid LSTM ED	1637.35	40.471	31.594	70813.06	266.018	165.205
Hybrid LSTM ATT ED	1705.97	41.316	31.202	93655.52	305.773	180.976
GCN-LSTM (traffic)	1105.94	33.224	27.186	17656.01	132.887	57.776
GCN-LSTM (centers)	1426.83	37.783	28.987	23634.92	153.532	73.110
GCN-LSTM (binary)	1179.33	34.329	27.917	20463.79	142.843	63.485
WEST GCN-LSTM (ours)	**802.87**	**28.335**	**22.861**	**12106.60**	**110.030**	**47.904**

Furthermore, Table 4 displays the results of the WEST GCN-LSTM model, compared against the various peer competitors in terms of MSE, RMSE, and MAE. These results correspond to the Central Park (Low) & (High) datasets.

5.7. Discussion

Before we proceed to analyze the experimental results, it is of paramount importance to delve deeper into the intricacies of each dataset. As depicted in Fig. 3, the regional traffic volume of the Berlin dataset is slightly larger compared to Central Park (Low). Furthermore, the volume of regional traffic of the Central Park and Central Park (High) datasets is quite similar. At the same time, both of them are significantly larger compared to the Regional Traffic (Low) dataset. This is because, in the context of the conducted experiments, fast-moving entities tend to emerge in a rather aperiodic manner, thus creating sudden bursts in the volume of regional traffic, contrary to slower-moving entities closely associated with periodic phenomena. Across all four datasets, a couple of regions monopolize the larger regional traffic volume values. However, in the case of the Central Park and the Central Park (High) datasets, we witness that four regions are associated with quite large volumes of regional traffic. In the context of their corresponding traffic, this inequality among Regions constitutes one of the most significant challenges that the various forecasting models have to overcome. More specifically, in the frame of the conducted experiments, we came across two types of regional traffic inequality, which are referred to as major inequality that corresponds to the Berlin and the Central Park (Low) datasets and minor inequality that corresponds to the Central Park and the Central Park (High) datasets. The manifestation of each type of regional traffic inequality highly depends on the moving speed of the various entities.

The experimental results indicate a plethora of interesting insights that are worth exploring. Let us begin our exploratory analysis by not considering the forecasting models based on the GCN-LSTM paradigm. When doing so, we see that for the Berlin dataset ML and ED solutions manage to perform the best, for the Central Park and Central Park (High) datasets LR solutions manage to outperform their competitors, and for the Central Park (Low) dataset ED solutions achieve the best scores. In other words, the more advanced ED solutions are better equipped to handle scenarios of major regional traffic inequality. In contrast, the quite simplistic LR solutions outperform their competitors in scenarios characterized by minor regional traffic inequality. This is because LR models are less sensitive to input fluctuations, and as such, they tend to converge to non-optimal solutions. Thus, in the case of minor regional traffic inequality, the extent to which they take into account the couple of Regions that monopolize traffic volume does not significantly alter the basis upon which they conduct their predictions and consequently produce superior results.

When we include forecasting models based on the GCN-LSTM paradigm, we see that these models either outperform or perform very closely to the best of their competitors across all examined datasets. This showcases the fact that the information filtering process, intertwined with graph convolution, can mitigate the rather negative impact that regional traffic inequalities have on the various forecasting models. However, the relative performance of each of these models, when compared to its peers, is closely associated with the reasoning behind each information distillation process and the characteristics of each dataset. The GCN-LSTM (centers) model's performance highly depends on the region's topology and the assumption that the optimal way to represent the relationship between two regions is by their proximity to each other. Since, in our experiments, we did not consider homogeneously shaped Regions, this approach produced the worst results in terms of prediction accuracy compared to its peers. Nevertheless, it produced results similar to the best LR, ML, and ED models. On the contrary, the performance of the GCN-LSTM (traffic) model depends on the traffic correlations among the various Regions. Real data exhibit a wide range of statistical properties within entity motion patterns. Consequently, the traffic-based relations among Regions that the GCN-LSTM (traffic) is designed to exploit are more complex and harder to encapsulate in the case of real data. As a result, the GCN-LSTM (traffic) model managed to produce good results in the case of the Central Park, Central Park (Low), and Central Park (High) datasets, while it performed less optimally in the case of the Berlin dataset. Last but not least, the GCN-LSTM (binary) model managed to perform, relatively to its peers, in the most consistent manner across all datasets. It outperformed the GCN-LSTM (centers) model across all datasets and the GCN-LSTM (centers) in the case of the Berlin dataset. This serves as a quite significant indication that the choice to construct the adjacency matrices based on the neighboring status of the various Regions is a robust approach that manages to provide satisfying results, regardless of the characteristics of each examined dataset.

As stated before, the proposed solution is an attempt at refining the aforementioned neighbor-based approach. So, let us examine the performance of the proposed solution using Fig. 4. Fig. 4 displays the performance of the proposed solution, of the best LR, ML, ED, and GCN-LSTM solutions, as well as of the GCN-LSTM (binary) model, in a comparative manner that is based on the use of normalized RMSE values. The same process was also performed for the MSE and the MAE results, and despite some minor differences, the overall conclusions remained the same. Thus, we decided not to include them as part of this discussion. The proposed solution clearly outperformed all of its competitors across all examined datasets. More specifically, for the Berlin dataset, the proposed solution outperformed the best LR model by 46%, the best ML model by 15%, the best ED model by 15%, and the best GCN-LSTM model by 8%. Furthermore, for the Central Park dataset, the proposed solution outperformed the best LR model by 23%, the best ML model by 71%, the best ED model by 51%, and the best GCN-LSTM model by 10%. For the Central Park (Low) dataset, the proposed solution outperformed the best LR model by 36%, the best ML model by 51%, the best ED model by 20%, and the best GCN-LSTM model by 15%. Finally, for the Central Park (High) dataset, the proposed solution outperformed the best LR model by 39%, the best ML model by 73%, the best ED model by 49%, and the best GCN-LSTM model by 16%.

The aforementioned experimental results indicate that the proposed solution can outperform its competitors by a quite significant margin, in a consistent manner, across all examined datasets. However, it is paramount to also examine the proposed solution in a manner aligned with the ablation study paradigm to showcase the effect of each part of the proposed solution on its overall performance. The proposed solution outperformed the GCN-LSTM (binary) model by 8% for the Berlin dataset, by 14% for the Central Park dataset, by 17% for the Central Park (Low) dataset, and by 22% for the Central Park (High) dataset. As mentioned

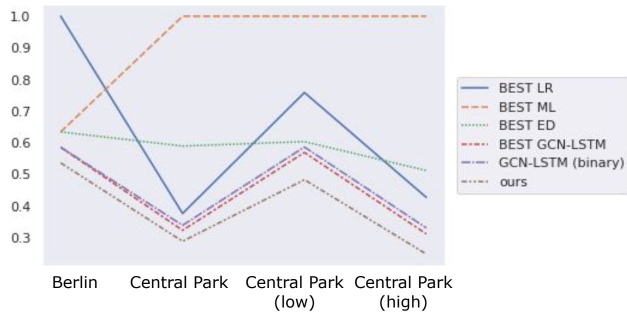

Figure 4: Normalized RMSE values that correspond to the proposed solution (ours), to GCN-LSTM (binary), as well as to the best LR, ML, ED, and GCN-LSTM solutions.

earlier, the Berlin and Central Park datasets were used to test a version of the proposed solution that uses only the Shared Borders Policy (WE GCN-LSTM). Thus, the fact that the proposed solution that leverages both policies (WEST GCN-LSTM) outperformed the GCN-LSTM (binary) by a greater margin when compared to the margin by which WE GCN-LSTM surpassed GCN-LSTM (binary), showcases that the fully implemented proposed solution that uses both policies is more efficient. The same principle applies when comparing the proposed solution against its competitors. The WE GCN-LSTM outperformed the second-best model by 8% for the Berlin dataset and by 10% for the Central Park dataset. In contrast, the WEST GCN-LSTM model outperformed the second-best model by 15% for the Central Park (Low) dataset, and by 16% for the Central Park (High) dataset. Since the Central Park dataset is the amalgamation of the Central Park (Low) and the Central Park (High) datasets, experimental consistency, in regards to the comparison between the WE GCN-LSTM and the WEST GCN-LSTM models, is achieved via the use of combinations of datasets that entail identical regional traffic characteristics, as well as across datasets that exhibit different types of regional and traffic-based characteristics.

6. Conclusions and Future Research

This work aims to introduce an advanced prediction model aimed at enhancing the accuracy of regional traffic forecasting through the refinement and fusion of information. Towards achieving this goal, we extend spatio-temporal graph neural networks in alignment with the regional traffic forecasting paradigm. Our approach proposes a novel architecture for spatio-temporal graph neural networks for regional traffic forecasting, incorporating weighted stacked graph convolutions. Implementing weighted stacked convolutions necessitates the calculation of a weighted adjacency matrix (denoted as A) and determining the number of graph convolution layers (denoted as K). To address this, we introduce two novel algorithms that utilize information about the speed of the populations and the topography of the regions they traverse to compute the adjacency matrix and K. In the context of this work, these policies are referred to as the Shared Borders Policy and the Adjustable Hops Policy. This implementation design enables the proposed solution to encapsulate the spatial and temporal characteristics intertwined with regional traffic forecasting in an optimal and more refined manner.

To evaluate the efficiency of the proposed solution, we conducted numerous experiments. The proposed solution significantly outperformed its competitors in the frame of an experimental evaluation consisting of 19 forecasting models across all examined datasets. This is because the proposed solution manages to simultaneously encapsulate both temporal and spatial aspects of regional traffic forecasting through information fusion. Furthermore, through information distillation, the proposed solution is also capable of mitigating the dire ramifications of regional traffic inequality. Finally, an additional ablation study concluded that each one of the three parts of the proposed solution serves to boost the performance of the proposed solution.

In terms of future research, there are various directions to potentially enhance the performance of the proposed solution even further. One of these directions derives from the fact that the experimental evaluation process did not account for various obstacles that may affect the resulting weighted Adjacency Matrix A calculated based on the Shared Borders Policy. Thus, a more refined version of Alg. 1 that considers such obstacles could result in a more efficient prediction model. Furthermore, it is worth examining the use of hybrid approaches for constructing the weighted Adjacency Matrix A. These hybrid approaches could stem from combining several methodologies already examined in the frame of the corresponding scientific literature and the proposed solution.

Acknowledgment

This project has received funding from the European Union's Horizon 2020 research and innovation programmes under grant agreements No 101135775 (PANDORA), No 101016509 (CHARITY) and No 777695 (MASTER). The work reflects only the authors' view, and the EU Agency is not responsible for any use that may be made of the information it contains.

References

[1] T. Verma, M. Sirenko, I. Kornecki, S. Cunningham, N. A. Araújo, Extracting spatiotemporal commuting patterns from public transit data, Journal of Urban Mobility 1 (2021) 100004.

[2] W. Shi, J. Cao, Q. Zhang, Y. Li, L. Xu, Edge computing: Vision and challenges, IEEE internet of things journal 3 (5) (2016) 637–646.

[3] A. Zanella, N. Bui, A. Castellani, L. Vangelista, M. Zorzi, Internet of things for smart cities, IEEE Internet of Things journal 1 (1) (2014) 22–32.

[4] C. Schindelhauer, Mobility in wireless networks, in: International Conference on Current Trends in Theory and Practice of Computer Science, Springer, 2006, pp. 100–116.

[5] P. K. Singh, P. K. D. Pramanik, A. K. Dey, P. Choudhury, Recommender systems: an overview, research trends, and future directions, International Journal of Business and Systems Research 15 (1) (2021) 14–52.

[6] C. Ilin, S. Annan-Phan, X. H. Tai, S. Mehra, S. Hsiang, J. E. Blumenstock, Public mobility data enables covid-19 forecasting and

management at local and global scales, Scientific reports 11 (1) (2021) 13531.

[7] T. Snyder, G. Byrd, The internet of everything, Computer 50 (06) (2017) 8–9.

[8] L. Alessandretti, P. Sapiezynski, S. Lehmann, A. Baronchelli, Multi-scale spatio-temporal analysis of human mobility, PloS one 12 (2) (2017) e0171686.

[9] H. Barbosa, M. Barthelemy, G. Ghoshal, C. R. James, M. Lenormand, T. Louail, R. Menezes, J. J. Ramasco, F. Simini, M. Tomasini, Human mobility: Models and applications, Physics Reports 734 (2018) 1–74.

[10] L. Po, F. Rollo, C. Bachechi, A. Corni, From sensors data to urban traffic flow analysis, in: 2019 IEEE International Smart Cities Conference (ISC2), IEEE, 2019, pp. 478–485.

[11] C. Cao, N. S.-N. Lam, Understanding the scale and resolution effects in remote sensing and gis, in: Scale in remote sensing and GIS, Routledge, 2023, pp. 57–72.

[12] S. A. Kashinath, S. A. Mostafa, A. Mustapha, H. Mahdin, D. Lim, M. A. Mahmoud, M. A. Mohammed, B. A. S. Al-Rimy, M. F. M. Fudzee, T. J. Yang, Review of data fusion methods for real-time and multi-sensor traffic flow analysis, IEEE Access 9 (2021) 51258–51276.

[13] D. Liu, S. Hui, L. Li, Z. Liu, Z. Zhang, A method for short-term traffic flow forecasting based on gcn-lstm, in: 2020 International Conference on Computer Vision, Image and Deep Learning (CVIDL), IEEE, 2020, pp. 364–368.

[14] T. Theodoropoulos, A. Makris, I. Kontopoulos, J. Violos, P. Tarkowski, Z. Ledwoń, P. Dazzi, K. Tserpes, Graph neural networks for representing multivariate resource usage: A multiplayer mobile gaming case-study, International Journal of Information Management Data Insights 3 (1) (2023) 100158.

[15] S. Wu, Spatiotemporal dynamic forecasting and analysis of regional traffic flow in urban road networks using deep learning convolutional neural network, IEEE Transactions on Intelligent Transportation Systems 23 (2) (2022) 1607–1615. doi:10.1109/TITS.2021.3098461.

[16] G. Jin, F. Li, J. Zhang, M. Wang, J. Huang, Automated dilated spatio-temporal synchronous graph modeling for traffic prediction, IEEE Transactions on Intelligent Transportation Systems 24 (8) (2022) 8820–8830.

[17] K. Chen, F. Chen, B. Lai, Z. Jin, Y. Liu, K. Li, L. Wei, P. Wang, Y. Tang, J. Huang, et al., Dynamic spatio-temporal graph-based cnns for traffic flow prediction, IEEE Access 8 (2020) 185136–185145.

[18] H. Peng, H. Wang, B. Du, M. Z. A. Bhuiyan, H. Ma, J. Liu, L. Wang, Z. Yang, L. Du, S. Wang, et al., Spatial temporal incidence dynamic graph neural networks for traffic flow forecasting, Information Sciences 521 (2020) 277–290.

[19] Q. Zhou, J.-J. Gu, C. Ling, W.-B. Li, Y. Zhuang, J. Wang, Exploiting multiple correlations among urban regions for crowd flow prediction, Journal of Computer Science and Technology 35 (2020) 338–352.

[20] H. Yang, X. Zhang, Z. Li, J. Cui, Region-level traffic prediction based on temporal multi-spatial dependence graph convolutional network from gps data, Remote Sensing 14 (2) (2022) 303.

[21] Y. Wang, A. Zhao, J. Li, Z. Lv, C. Dong, H. Li, Multi-attribute graph convolution network for regional traffic flow prediction, Neural Processing Letters (2022) 1–27.

[22] C. Li, H. Zhang, Z. Wang, Y. Wu, F. Yang, Multigraph aggregation spatiotemporal graph convolution network for ride-hailing pick-up region prediction, Wireless Communications and Mobile Computing 2022 (2022).

[23] B. Wang, X. Luo, F. Zhang, B. Yuan, A. Bertozzi, P. Brantingham, Graph-based deep modeling and real time forecasting of sparse spatio-temporal data (2018), arXiv preprint arXiv:1804.00684.

[24] H. Shi, Q. Yao, Q. Guo, Y. Li, L. Zhang, J. Ye, Y. Li, Y. Liu, Predicting origin-destination flow via multi-perspective graph convolutional network, in: 2020 IEEE 36th International conference on data engineering (ICDE), IEEE, 2020, pp. 1818–1821.

[25] G. Yeghikyan, F. L. Opolka, M. Nanni, B. Lepri, P. Lio, Learning mobility flows from urban features with spatial interaction models and neural networks, in: 2020 IEEE International Conference on Smart Computing (SMARTCOMP), IEEE, 2020, pp. 57–64.

[26] H. Qiu, Q. Zheng, M. Msahli, G. Memmi, M. Qiu, J. Lu, Topological graph convolutional network-based urban traffic flow and density prediction, IEEE transactions on intelligent transportation systems 22 (7) (2020) 4560–4569.

[27] J. Sun, J. Zhang, Q. Li, X. Yi, Y. Liang, Y. Zheng, Predicting citywide crowd flows in irregular regions using multi-view graph convolutional networks, IEEE Transactions on Knowledge and Data Engineering 34 (5) (2020) 2348–2359.

[28] S. Wang, H. Miao, H. Chen, Z. Huang, Multi-task adversarial spatial-temporal networks for crowd flow prediction, in: Proceedings of the 29th ACM international conference on information & knowledge management, 2020, pp. 1555–1564.

[29] J. Van Lint, S. P. Hoogendoorn, H. J. van Zuylen, Freeway travel time prediction with state-space neural networks: Modeling state-space dynamics with recurrent neural networks, Transportation Research Record 1811 (1) (2002) 30–39.

[30] S. Hochreiter, Y. Bengio, P. Frasconi, J. Schmidhuber, et al., Gradient flow in recurrent nets: the difficulty of learning long-term dependencies (2001).

[31] X. Ma, Z. Tao, Y. Wang, H. Yu, Y. Wang, Long short-term memory neural network for traffic speed prediction using remote microwave sensor data, Transportation Research Part C: Emerging Technologies 54 (2015) 187–197.

[32] S. Du, T. Li, Y. Yang, X. Gong, S.-J. Horng, An lstm based encoder-decoder model for multistep traffic flow prediction, in: 2019 International Joint Conference on Neural Networks (IJCNN), IEEE, 2019, pp. 1–8.

[33] S. Nadeeshan, A. S. Perera, Multi-step bidirectional lstm for low frequent bus travel time prediction, in: 2021 Moratuwa Engineering Research Conference (MERCon), IEEE, 2021, pp. 462–467.

[34] M. Cao, V. O. Li, V. W. Chan, A cnn-lstm model for traffic speed prediction, in: 2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring), IEEE, 2020, pp. 1–5.

[35] T. Theodoropoulos, A.-C. Maroudis, J. Violos, K. Tserpes, An encoder-decoder deep learning approach for multistep service traffic prediction, in: 2021 IEEE Seventh International Conference on Big Data Computing Service and Applications (BigDataService), IEEE, 2021, pp. 33–40.

[36] J. Violos, T. Theodoropoulos, A.-C. Maroudis, A. Leivadeas, K. Tserpes, Self-attention based encoder-decoder for multistep human density prediction, Journal of urban mobility 2 (2022) 100022.

[37] X. Su, X. Yan, C.-L. Tsai, Linear regression, Wiley Interdisciplinary Reviews: Computational Statistics 4 (3) (2012) 275–294.

[38] M. I. Jordan, T. M. Mitchell, Machine learning: Trends, perspectives, and prospects, Science 349 (6245) (2015) 255–260.

[39] M. Gori, G. Monfardini, F. Scarselli, A new model for learning in graph domains, in: Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005., Vol. 2, 2005, pp. 729–734 vol. 2. doi:10.1109/IJCNN.2005.1555942.

[40] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, S. Y. Philip, A comprehensive survey on graph neural networks, IEEE transactions on neural networks and learning systems 32 (1) (2020) 4–24.

[41] H. Dai, Z. Kozareva, B. Dai, A. Smola, L. Song, Learning steady-states of iterative algorithms over graphs, in: International conference on machine learning, PMLR, 2018, pp. 1106–1114.

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

[43] Z. Hou, X. Liu, Y. Cen, Y. Dong, H. Yang, C. Wang, J. Tang, Graphmae: Self-supervised masked graph autoencoders, in: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2022, pp. 594–604.

[44] K.-H. N. Bui, J. Cho, H. Yi, Spatial-temporal graph neural network for traffic forecasting: An overview and open research issues, Applied Intelligence 52 (3) (2022) 2763–2774.

[45] Y. Liu, S. Rasouli, M. Wong, T. Feng, T. Huang, Rt-gcn: Gaussian-based spatiotemporal graph convolutional network for robust traffic

prediction, Information Fusion 102 (2024) 102078.

[46] L. Zhao, Y. Song, C. Zhang, Y. Liu, P. Wang, T. Lin, M. Deng, H. Li, T-gcn: A temporal graph convolutional network for traffic prediction, IEEE transactions on intelligent transportation systems 21 (9) (2019) 3848–3858.

[47] S. Abu-El-Haija, B. Perozzi, A. Kapoor, N. Alipourfard, K. Lerman, H. Harutyunyan, G. Ver Steeg, A. Galstyan, Mixhop: Higher-order graph convolutional architectures via sparsified neighborhood mixing, in: international conference on machine learning, PMLR, 2019, pp. 21–29.

[48] G. Nikolentzos, G. Dasoulas, M. Vazirgiannis, k-hop graph neural networks, Neural Networks 130 (2020) 195–205.

[49] J. Feng, Y. Chen, F. Li, A. Sarkar, M. Zhang, How powerful are k-hop message passing graph neural networks, Advances in Neural Information Processing Systems 35 (2022) 4776–4790.

[50] J. Zhao, Z. Yan, X. Chen, B. Han, S. Wu, R. Ke, k-gcn-lstm: A k-hop graph convolutional network and long–short-term memory for ship speed prediction, Physica A: Statistical Mechanics and its Applications 606 (2022) 128107.

[51] W. Jiang, J. Luo, M. He, W. Gu, Graph neural network for traffic forecasting: The research progress, ISPRS International Journal of Geo-Information 12 (3) (2023) 100.

[52] G. Jin, Y. Liang, Y. Fang, Z. Shao, J. Huang, J. Zhang, Y. Zheng, Spatio-temporal graph neural networks for predictive learning in urban computing: A survey, IEEE Transactions on Knowledge and Data Engineering 36 (10) (2023) 5388–5408.

[53] G. Jin, L. Liu, F. Li, J. Huang, Spatio-temporal graph neural point process for traffic congestion event prediction, in: Proceedings of the AAAI conference on artificial intelligence, Vol. 37, 2023, pp. 14268–14276.

[54] W. Jiang, J. Luo, Graph neural network for traffic forecasting: A survey, Expert Systems with Applications 207 (2022) 117921.

[55] F. Li, J. Feng, H. Yan, G. Jin, F. Yang, F. Sun, D. Jin, Y. Li, Dynamic graph convolutional recurrent network for traffic prediction: Benchmark and solution, ACM Transactions on Knowledge Discovery from Data 17 (1) (2023) 1–21.

[56] Z. Shao, F. Wang, Y. Xu, W. Wei, C. Yu, Z. Zhang, D. Yao, T. Sun, G. Jin, X. Cao, et al., Exploring progress in multivariate time series forecasting: Comprehensive benchmarking and heterogeneity analysis, IEEE Transactions on Knowledge and Data Engineering (2024).

[57] B. Yu, H. Yin, Z. Zhu, Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting, in: Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, International Joint Conferences on Artificial Intelligence Organization, 2018. doi:10.24963/ijcai.2018/505.
URL https://doi.org/10.24963%2Fijcai.2018%2F505

[58] E. Rossi, B. Chamberlain, F. Frasca, D. Eynard, F. Monti, M. Bronstein, Temporal graph networks for deep learning on dynamic graphs (2020). doi:10.48550/ARXIV.2006.10637.
URL https://arxiv.org/abs/2006.10637

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

[60] M. Behrisch, L. Bieker, J. Erdmann, D. Krajzewicz, SUMO – Simulation of Urban MObility: An Overview, in: S. . U. o. O. Aida Omerovic, R. I.-R. T. P. Diglio A. Simoni, R. I.-R. T. P. Georgiy Bobashev (Eds.), Proceedings of SIMUL 2011, The Third International Conference on Advances in System Simulation, ThinkMind, Barcelona, 2011.
URL http://www.thinkmind.org/index.php?view=instance&instance=SIMUL+2011

[61] J. Haworth, T. Cheng, Graphical lasso for local spatio-temporal neighbourhood selection, in: Proceedings the GIS Research UK 22nd Annual Conference. Presented at the GISRUK, 2014, pp. 425–433.

[62] S. Kundu, M. S. Desarkar, P. Srijith, Traffic forecasting with deep learning, in: 2020 IEEE Region 10 Symposium (TENSYMP), IEEE, 2020, pp. 1074–1077.

[63] W. Liu, Z. Dou, W. Wang, Y. Liu, H. Zou, B. Zhang, S. Hou, Short-term load forecasting based on elastic net improved gmdh and difference degree weighting optimization, Applied Sciences 8 (9) (2018) 1603.

[64] O. Gkountouna, D. Pfoser, A. Züfle, Traffic flow estimation using probe vehicle data, in: 2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA), IEEE, 2020, pp. 579–588.

[65] L. Cai, Y. Yu, S. Zhang, Y. Song, Z. Xiong, T. Zhou, A sample-rebalanced outlier-rejected k-nearest neighbor regression model for short-term traffic flow forecasting, IEEE access 8 (2020) 22686–22696.

[66] W. Alajali, W. Zhou, S. Wen, Y. Wang, Intersection traffic prediction using decision tree models, Symmetry 10 (9) (2018) 386.

[67] X. Zhan, S. Zhang, W. Y. Szeto, X. Chen, Multi-step-ahead traffic speed forecasting using multi-output gradient boosting regression tree, Journal of Intelligent Transportation Systems 24 (2) (2020) 125–141.

[68] H. Xia, X. Wei, Y. Gao, H. Lv, Traffic prediction based on ensemble machine learning strategies with bagging and lightgbm, in: 2019 IEEE International Conference on Communications Workshops (ICC Workshops), IEEE, 2019, pp. 1–6.

[69] J. Evans, B. Waterson, A. Hamilton, Forecasting road traffic conditions using a context-based random forest algorithm, Transportation planning and technology 42 (6) (2019) 554–572.

[70] S. M. Mastelini, F. K. Nakano, C. Vens, A. C. P. de Leon Ferreira, et al., Online extra trees regressor, IEEE Transactions on Neural Networks and Learning Systems (2022).