

Received 9 November 2022, accepted 26 December 2022, date of publication 4 January 2023, date of current version 10 January 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3234214



## RESEARCH ARTICLE

# Graph Neural Network for Air Quality Prediction: A Case Study in Madrid

DITSUHI ISKANDARYAN<sup>ID</sup>, FRANCISCO RAMOS, AND SERGIO TRILLES<sup>ID</sup>

Institute of New Imaging Technologies (INIT), Universitat Jaume I, 12071 Castelló de la Plana, Spain

Corresponding author: Ditsuhi Iskandaryan (iskandar@uji.es)

The work of Ditsuhi Iskandaryan was supported by Universitat Jaume I through the Predoctoral Program PINV2018 under Grant PREDOC/2018/61. The work of Sergio Trilles was supported in part by the Juan de la Cierva-Incorporación Postdoctoral Program of the Ministerio de Ciencia e Innovación-Spanish Government under Grant IJC2018-035017-I funded by MCIN/AEI/10.13039/501100011033, and in part by “ERDF, a way of making Europe,” through the European Union.

**ABSTRACT** Air quality monitoring, modelling and forecasting are considered pressing and challenging topics for citizens and decision-makers, including the government. The tools used to achieve the above goals vary depending on the opportunities provided by technological development. Much attention is currently being paid to machine learning and deep learning methods, which, compared to domain knowledge methods, often perform better in terms of capturing, computing and processing multidimensional information and complex dependencies. The technique introduced in this work is an Attention Temporal Graph Convolutional Network based on a combination of Attention, a Gated Recurrent Unit and a Graph Convolutional Network. In the framework of the current study, it is initially suggested to use the presented approach in the domain of air quality prediction. The proposed method was tested using air quality, meteorological and traffic data obtained from the city of Madrid for the periods January-June 2019 and January-June 2022. The evaluation metrics, including Root Mean Square Error, Mean Absolute Error and Pearson Correlation Coefficient, confirmed the proposed model’s advantages compared with the reference models (Temporal Graph Convolutional Network, Long Short-Term Memory and Gated Recurrent Unit).

**INDEX TERMS** Graph neural network, attention temporal graph convolutional network, spatiotemporal prediction, nitrogen dioxide prediction, air quality prediction.

## I. INTRODUCTION

Air pollution’s consequences seriously impact the world’s population’s health and the ecosystem by affecting the single element and components of them. Air pollution is the fourth biggest global risk factor for human health [1]. It is responsible for about 16% of all deaths worldwide [2], in particular, 1.6 million death in China [3]. The World Health Organization (WHO) air quality guidelines report that about 90% of the world’s citizens live in areas where air pollution exceeds established thresholds [4]. Monitoring and improving air quality are considered one of today’s biggest challenges. Advanced knowledge of air quality concentrations can enable decision-makers to take appropriate action in order to reduce air pollution and its detrimental impacts. At its core,

air quality is extremely complex and influenced by many factors.

Several approaches are employed to model and forecast air quality. These approaches can be divided into two categories, namely, domain knowledge-based and data-driven. Studies have indicated that the first group has limitations in capturing non-linear dependencies. They mainly simplify the existing relationship between concentration and affected factors [5]. The second category, which includes machine learning techniques, has proven to be efficient in collecting and processing complex dependencies across scales from high-dimensional datasets, including interactions and non-linear relationships and intrinsic features that control and form pollution [6], [7]. Air pollution has spatial and temporal dependencies, i.e. the concentration depends on many factors, including local climatic conditions and air pollutants, which fluctuate over time. Thus, it is vital to conduct a spatiotemporal analysis in order to capture and process all the above dependencies.

The associate editor coordinating the review of this manuscript and approving it for publication was Giacomo Fiumara<sup>ID</sup>.

The main objective of the current study is to conduct spatiotemporal prediction of nitrogen dioxide ( $\text{NO}_2$ ) in the city of Madrid using air quality, meteorological and traffic data. Some of our previous studies were also devoted to solving the above tasks [8], [9]. However, the approaches implemented in those works require the input to have a Euclidean or grid-like structure. Reconstructing input data as a grid is a tedious and time-consuming task. Also, depending on the distribution of the data, the resulting grid may contain many empty cells without any data, which is likely to degrade the performance of the predictive analysis. One of the most advanced methods proposed in this research is an Attention Temporal Graph Convolutional Network (A3T-GCN), which processes non-Euclidean structured data as input. A3T-GCN allows efficient collection of spatiotemporal information through a model architecture based on a combination of Attention, Gated Recurrent Unit (GRU) and Graph Convolutional Network (GCN). The first two blocks are responsible for collecting temporal information and the GCN is responsible for spatial dependencies. Furthermore, the flexibility of a graph neural network enables it to be applied in many areas, especially those where the data is at a non-Euclidean distance. The following are the significant contributions addressed within the scope of this work:

- We conducted spatiotemporal prediction of  $\text{NO}_2$  using a graph neural network, namely A3T-GCN.
- We performed a predictive analysis at different time intervals, including the next 1-12 hours, 12-24 hours, 24-36 hours and 36-48 hours.
- We compared the proposed method with reference methods (Temporal Graph Convolutional Network (TGCN), Long Short-Term Memory (LSTM) and GRU) in terms of defined evaluation metrics (Root Mean Square Error (RMSE), Mean Absolute Error (MAE), Pearson Correlation Coefficient (R)).

The remainder of the paper is structured as follows. Section II introduces the latest studies in air quality prediction from a spatiotemporal perspective. The case study along with the data employed is described in Section III. Section IV defines the concept of a graph neural network and provides details of the construction of the proposed method. A thorough explanation of the experimental analyses and the results obtained can be found in Section V. Finally, Section VI summarises the results and proposes further research directions.

## II. RELATED WORK

Many methods have been used to implement air quality prediction in the spatiotemporal dimension. For example, in the following study [7] LSTM integrated with Multi-Output and Multi-Index of Supervised Learning was proposed. It was evaluated using the datasets for Beijing from 1 January 2016 to 31 December 2017. Zhao et al. [10] proposed the spatiotemporal method called Spatiotemporal Convolutional Neural Network combined with LSTM, which was implemented on data generated by a three-dimensional structure called Relevance Data Cube using a clustering

algorithm, time sliding windows and correlation analysis of factors. The authors, in their analysis, followed the concept of the regional air quality prediction problem. Abirami and Chitra [11] implemented a hierarchical deep learning model, DL-Air, which consisted of three blocks, including encoder, Spatiotemporal Association Analysis LSTM and decoder components. Another model, named Ensemble LSTM [12] was applied to predict particulate matter with a diameter of less than 2.5 micrometres ( $\text{PM}_{2.5}$ ). It consists of Ensemble Empirical Mode Decomposition (EEMD) and LSTM. EEMD is responsible for multi-modal feature extraction and estimated integration, and LSTM is responsible for multi-modal feature learning. Huang et al. [13] to predict  $\text{PM}_{2.5}$  implemented a method consisting of Empirical Mode Decomposition (EMD) and GRU.

In addition, many authors have recently been drawn to the advantages of a graph neural network in areas such as traffic flow prediction [14], [15], [16], [17], parking availability prediction [18], pedestrian trajectories prediction [19], [20], urban vehicle emission prediction [21], wind speed prediction [22], weather prediction [23] and solar irradiance prediction [24]. The air quality domain, among others, has also benefited from these advantages, and various authors have used graph neural networks to forecast air quality. Han et al. [25] proposed the Self-Supervised Hierarchical Graph Neural Network based on cities → functional zones → regions hierarchical graph network to perform fine-grained air quality prediction implemented on datasets for the Beijing-Tianjin-Hebei and the Pearl River Delta urban agglomerations. Ram et al. [26] proposed a Dual GCN (DGCN) and LSTM network combined with a wireless sensor network and Internet of Things (IoT) to perform Air Quality Index (AQI) predictions; in particular, DGCN helps to process the data from the sensors that were later learned by the graph LSTM. Xiao et al. [27] offered a Dual-path Dynamic Directed GCN (DP-DDGCN) based on the combination of dual-path dynamic directed graph blocks and GRU. Ouyang et al. [28] proposed a Spatiotemporal Dynamic GCN (ST-DGCN) based on a time-varying dynamic adjacency matrix to predict  $\text{PM}_{2.5}$ . Ge et al. [29] offered a Multi-scale Spatiotemporal GCN (MST-GCN) including a multi-scale block, several spatiotemporal blocks and a fusion block to forecast air quality. Wang et al. [30] used Attentive Temporal GCN to model inter-station relationships (spatial adjacency, functional similarity and temporal pattern similarity) to predict air quality. To forecast nationwide city air quality, Chen et al. [31] proposed the group-aware graph neural network using the Chinese city air quality dataset. Xu et al. [32] performed air quality forecasting based on a hierarchical graph neural network; in particular, city-level and station-level graphs were constructed using the Yangtze River Delta city group's dataset. The authors developed two strategies, upper delivery and lower updating, to implement the inter-level interactions and introduce a message-passing mechanism to implement the intra-level interactions. Another study is devoted to comparing graph-based and

non-graph-based models for PM<sub>2.5</sub> prediction under distribution shift [33]. Le [34] for efficient learning spatiotemporal characteristics of air quality values, and related factors Spatiotemporal Graph Convolutional Recurrent Neural Network was used.

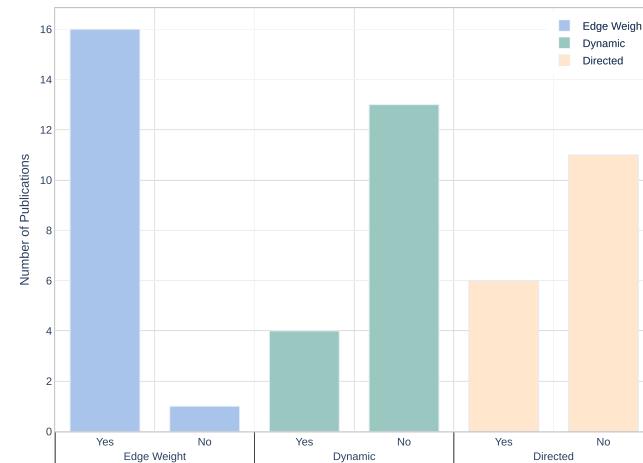
Zhao et al. [35] introduced a novel model based on the combination of air quality spatiotemporal network and GCN for PM<sub>2.5</sub> prediction. Gao and Li [36] proposed a graph-based LSTM model to perform spatiotemporal prediction of PM<sub>2.5</sub> concentration. Zhang et al. [37] used a Temporal Attention network with domain-specific graph regularisation in order to improve PM<sub>2.5</sub> prediction. Wang et al. [38] developed a new model called PM<sub>2.5</sub>-GNN to capture fine-grained and long-term influences in the PM<sub>2.5</sub> process. Zhao and Zettsu [39] proposed Multi-Attention Spatiotemporal Graph Networks to predict the concentration of PM<sub>2.5</sub>, ground-level ozone (O<sub>3</sub>) and particulate matter with a diameter of less than 10 micrometres (PM<sub>10</sub>). Qi et al. [40] implemented spectral GCN combined with LSTM using historical data for the last 24 h to forecast the PM<sub>2.5</sub> concentration for the next 1 h, 2 h, 4 h, 8 h, 12 h, 24 h, 48 h and 72 h. Huang et al. [41] implemented a Spatio-Attention embedded Recurrent Neural Network to predict air quality using Beijing's air quality datasets; to capture spatial patterns, a self-loop-normalised adjacency matrix was used. Lin et al. [42] proposed the Geo-context based Diffusion Convolutional Recurrent Neural Network to predict PM<sub>2.5</sub>. The geo-context segment was implemented by building a graph that allowed information to be collected in the spatial dimension, and a Diffusion Convolutional Recurrent Neural Network was responsible for collecting information in the temporal dimension.

The overall picture of the publications related to the implementation of a graph neural network for air quality prediction can be seen in Table 1. The following features were extracted from each work, including *Year*, *Method*, *Edge Weight*, *Dynamic/Static*, *Directed/Undirected*, *Target*, *Dataset* and *Evaluation Metric*.

**Year:** year of publication of the works. As can be seen, interest in the topic began quite recently, since 2018, in particular, the main peak came in 2021, when ten out of eighteen extracted works were published in 2021.

**Method:** implemented methods to perform the prediction. As shown, most studies involve GCN combined with Recurrent Neural Network (RNN) such as GRU or LSTM. Recently, the integration of the attention-based network is also increasing.

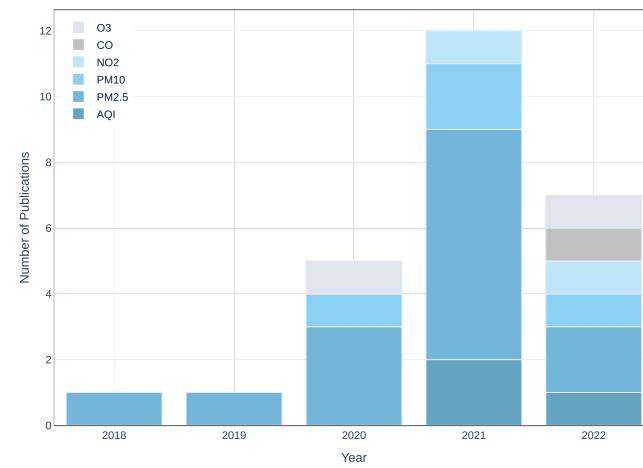
**Edge Weight, Dynamic/Static, Directed/Undirected:** to find out more information about the structure of the graph, information about the edge weight, dynamics and direction was extracted. Figure 1 shows the distribution of publications for each feature. It is noticeable that most of the papers used graphs consisting of weighted edges (seventeen out of eighteen). In terms of dynamic status, most of them are static (fourteen out of eighteen), and in terms of direction, most studies used undirected graphs (twelve out of eighteen).



**FIGURE 1.** Number of publications in terms of edge weights (Yes-with weights, No-without weights), dynamic (Yes-dynamic, No-static) and direction (Yes-directed, No-undirected).

**Target:** are predictable pollutants. The following pollutants were taken into account: PM<sub>2.5</sub> (fourteen papers), PM<sub>10</sub> (four papers), AQI (three papers), NO<sub>2</sub> (two papers), O<sub>3</sub> (two papers), carbon monoxide (CO (one paper)). The most used pollutant was PM<sub>2.5</sub>.

Figure 2 shows the distribution of the prediction target over time. It can be seen that PM<sub>2.5</sub> has been in use since 2018. From 2020, additional targets are included, and in 2022, studies attempt to predict all the aforementioned prediction targets.



**FIGURE 2.** Number of publications per prediction target throughout the years.

**Dataset:** datasets used for predictive analysis. The following datasets are used: air quality (eighteen papers), spatial (the location of air quality monitoring stations; (eighteen papers)), meteorological (seventeen papers), points of interest (five papers), traffic (two papers), road network (two papers) and geographic data (land uses, roads, water areas, buildings; (one paper)). Air quality datasets and spatial datasets were most commonly used, which is logical, since the main task was air

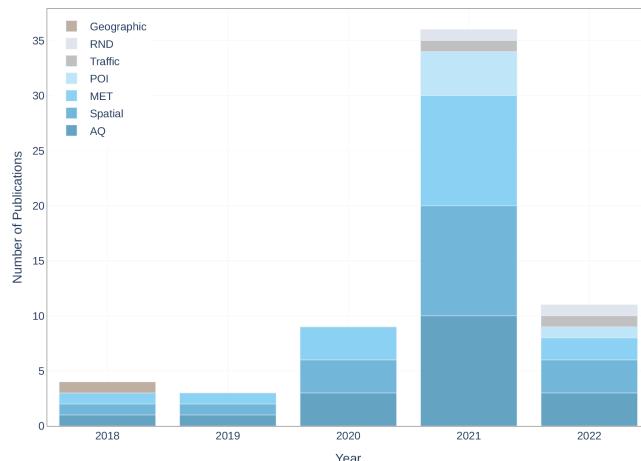
**TABLE 1.** Features of the papers devoted to the implementation of graph neural networks for air quality prediction (\*).

Work	Year	Method	Edge Weight	Dynamic/Static	Directed/Undirected	Target	Dataset	Evaluation Metric
[25]	2022	SSH-GNN	Yes	Static	Undirected	AQI	AQ, MET, Traffic, POI, RND	MAE, RMSE
[26]	2022	DGCN with Graph LSTM	No	Static	Directed	PM <sub>2.5</sub> , PM <sub>10</sub> , NO <sub>2</sub> , CO, O <sub>3</sub>	AQ	MAE, RMSE, ACC
[27]	2022	DP-DDGCN	Yes	Dynamic	Directed	PM <sub>2.5</sub>	AQ, MET	MAE, RMSE
[28]	2021	ST-DGCN	Yes	Dynamic	Undirected	PM <sub>2.5</sub>	AQ, MET	MAE, RMSE
[29]	2021	MST-GCN	Yes	Static	Undirected	PM <sub>2.5</sub>	AQ, MET, POI, RND	ACC, MAE, RMSE
[30]	2021	ATGCN	Yes	Static	Undirected	Not Specified	AQ, MET, POI	MAE, RMSE
[31]	2021	GAGNN	Yes	Static	Undirected	AQI	AQ, MET	MAE, RMSE
[32]	2021	HGNN	Yes	Dynamic	Directed	AQI	AQI, MET, POI	MAE, RMSE
[33]	2021	Naive, LR, ARIMA, MLP, GCN, STGCN, Temporal, ASTGCN	Yes	Static	Undirected	PM <sub>2.5</sub>	AQ, MET	RMSE
[34]	2021	Spatiotemporal GCRNN	Yes	Static	Undirected	PM <sub>2.5</sub> , PM <sub>10</sub>	AQ, MET, Traffic	RMSE, R <sup>2</sup> , spRMSE
[35]	2021	AQSTN-GCN	Yes	Static	Undirected	PM <sub>2.5</sub>	AQ, MET	MAE, RMSE, MAPE, MSE, R <sup>2</sup>
[36]	2021	GLSTM	Yes	Static	Directed	PM <sub>2.5</sub>	AQ, MET	RMSE, MAE, MAPE, IA
[41]	2021	SpAttRNN	Yes	Static	Undirected	PM <sub>2.5</sub> , PM <sub>10</sub> , NO <sub>2</sub>	AQ, POI	RMSE, MAE, SMAPE
[37]	2020	TA-GCN	Yes	Static	Directed	PM <sub>2.5</sub>	AQ, MET	Train loss, Test loss, Validation loss, RMSE, MAE, CSI, POD, FAR
[38]	2020	PM2.5-GNN	Yes	Dynamic	Directed	PM <sub>2.5</sub>	AQ, MET	Train loss, Test loss, Validation loss, RMSE, MAE, CSI, POD, FAR
[39]	2020	MASTGN	Yes	Static	Undirected	PM <sub>2.5</sub> , PM <sub>10</sub> , O <sub>3</sub>	AQ, MET	MAE, SMAPE
[40]	2019	GC-LSTM	Yes	Static	Undirected	PM <sub>2.5</sub>	AQ, MET	MAE, RMSE, IA, RR, FAR, R <sup>2</sup>
[42]	2018	GC-DCRNN	Yes	Static	Undirected	PM <sub>2.5</sub>	AQ, MET, Geographic	MAE, RMSE

\* **Method:** SSH-GNN—Self-Supervised Hierarchical Graph Neural Network, DGCN—Dual Graph Convolutional Network, DP-DDGCN—Dual-path Dynamic Directed Graph Convolutional Network, ST-DGCN—Spatiotemporal Dynamic Graph Convolution Neural Network, MST-GCN—Multi-scale Spatiotemporal Graph Convolution Network, ATGCN—Attentive Temporal Graph Convolutional Network, GAGNN—Group-aware Graph Neural Network, HGNN—Hierarchical Graph Networks, LR—Linear Regression, ARIMA—Autoregressive Integrated Moving Average, MLP—Multilayer Perceptron, GCN—Graph Convolutional Neural Network, STGCN—Spatiotemporal Graph Convolutional Network, ASTGCN—Attention based Spatiotemporal Graph Convolution Network, GCRNN—Graph Convolutional Recurrent Neural Network, AQSTN—Air Quality Spatiotemporal Network, SpAttRNN—Spatio-Attention embedded Recurrent Neural Network, TA-GCN—Temporal Attention-GCN, MASTGN—Multi-attention Spatiotemporal Graph Networks, GC-LSTM—Graph Convolution Network embedded Long Short-term Memory, GC-DCRNN—Geo-context based Diffusion Convolutional Recurrent Neural Network; **Target:** AQI—Air Quality Index, PM<sub>2.5</sub>—Particulate Matter with a diameter of less than 2.5 micrometres, PM<sub>10</sub>—Particulate Matter with a diameter of less than 10 micrometres, NO<sub>2</sub>—Nitrogen Dioxide, CO—Carbon Monoxide, O<sub>3</sub>—Ozone; **Dataset:** MET—Meteorological, POI—Point of Interest, RND—Road Network Data; **Evaluation Metric:** MAE—Mean Absolute Error, RMSE—Root Mean Square Error, ACC—Accuracy, R<sup>2</sup>—Coefficient of Determination, spRMSE—Spatiotemporal RMSE, MAPE—Mean Absolute Percentage Error, MSE—Mean Square Error, IA—Index of Agreement, CSI—Critical Success Index, POD—Probability of Detection, FAR—False Alarm Rate, SMAPE—Symmetric Mean Absolute Percentage Error, RR—Recall Rate.

quality forecasting, and since the main focus of the work was on the implementation of a graph neural network, the location of monitoring stations is the main base for constructing the graph. The next most commonly used dataset is meteorological data, due to the strong correlation between air quality and meteorological data.

It is also very interesting to see the distribution of the datasets in chronological order. Air quality, spatial and meteorological data are included for all years (Figure 3). In recent years, in particular 2021 and 2022, the analysis began to include also points of interest, traffic and road network data.



**FIGURE 3.** Number of publications per dataset throughout the years.

**Evaluation Metric:** is a metric for measuring model performance. The following metrics were used, including RMSE (seventeen papers), MAE (sixteen papers), Coefficient of Determination (three papers), False Alarm Rate (three papers), Accuracy (two papers), Mean Absolute Percentage Error (two papers), Index of Agreement (two papers), Critical Success Index (two papers), Probability of Detection (two papers), Symmetric Mean Absolute Percentage Error (two papers), train loss (two papers), test loss (two papers), validation loss (two papers), Spatiotemporal RMSE (one paper), Mean Square Error (one paper), and Recall Rate (one paper).

Compared to the above studies, the following are the main contributions made by the current work: 1) Including traffic, air quality and meteorological data; 2) Conducting spatiotemporal prediction of NO<sub>2</sub> in different time intervals, including 1-12 h, 12-24 h, 24-36 h and 36-48 h; 3) Comparing the proposed method with reference methods in terms of evaluation metrics that measure model performance (RMSE, MAE, R). Furthermore, since the suggested A3T-GCN method is primarily used to predict traffic speed [43], the current research can be considered a somewhat groundbreaking contribution to the domain of air quality prediction.

### III. DATA

The dataset used in this study consists of air quality, meteorological and traffic data from the period January-June 2019 (training set) and January-June 2022 (testing set), and the

location of air quality and meteorological monitoring stations and traffic measurement points of the city of Madrid. The data were obtained from the Open Data portal of Madrid City Council.<sup>1</sup> In addition, there are twenty-four air quality control stations, twenty-six meteorological control stations and more than 4,000 traffic measurement points. Note that the location of the traffic measurement points is not fixed, and the Open Data portal of Madrid City Council provides the location files for each month. The files for January-June 2019 and January-June 2022 were used in the current analysis. In Figure 4, for illustration purposes, only locations for January 2019 are used, which is also indicated in the caption.

The datasets used in the current work are presented below (Table 2 shows summary statistics of each type of data for the periods used in the analyses):

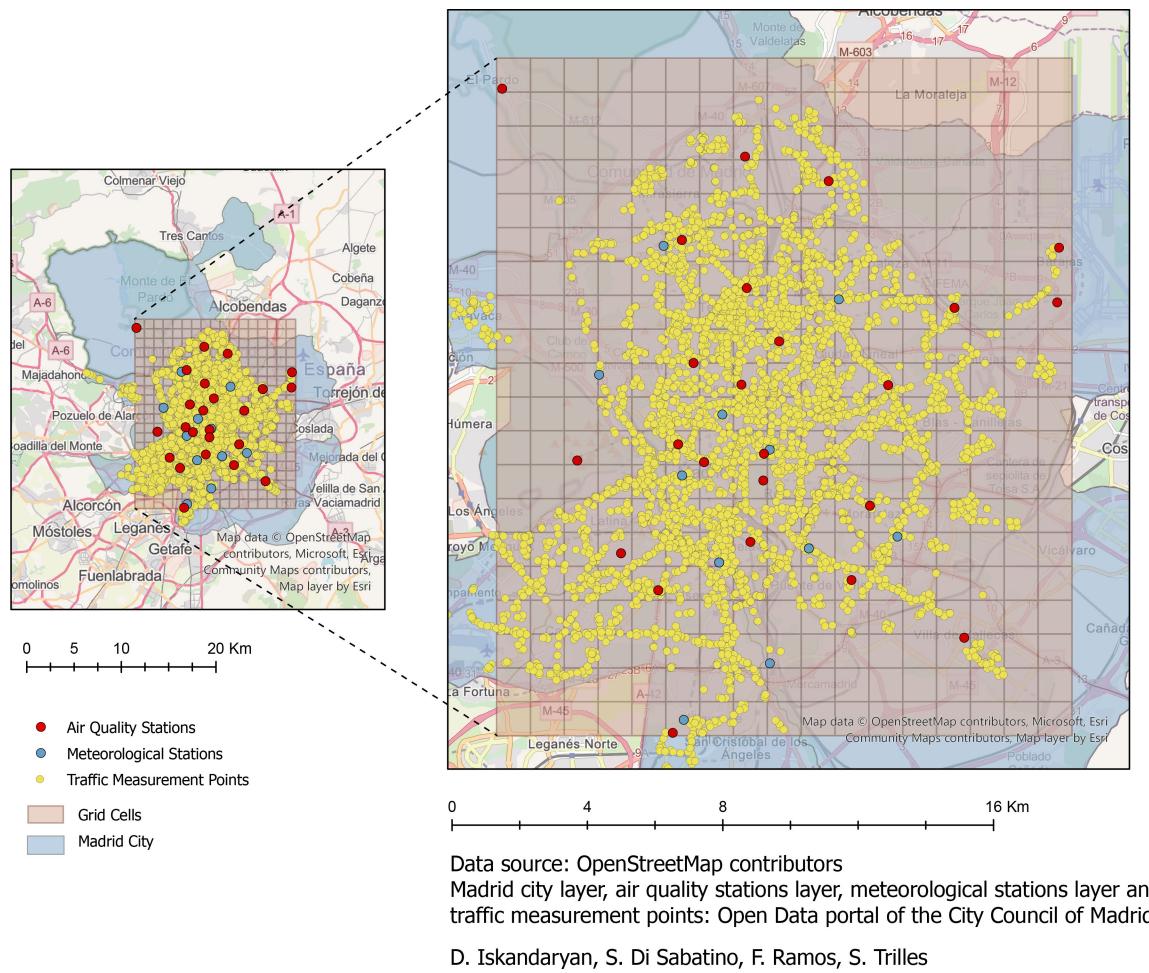
- Air Quality Data - NO<sub>2</sub> ( $\mu\text{g}/\text{m}^3$ ).
- Meteorological Data - wind speed (m/s), wind direction, temperature (°C), relative humidity (%), barometric pressure (mb), solar irradiance (W/m<sup>2</sup>).
- Traffic Data - since the attributes of the traffic data can be specific to a certain area, the selected traffic attributes are shown below with their definitions for the city of Madrid.
  - Intensity - the intensity of the measurement point in a period of 15 minutes (vehicles/hour).
  - Occupancy time - measurement point occupancy time in a period of 15 minutes (%). For example, a 50% occupancy in a 15-minute period means that vehicles have been positioned over the detector for 7 minutes and 30 seconds.
  - Load - vehicle loading in a 15-minute period. This parameter represents an estimate of the degree of congestion, calculated from an algorithm that uses intensity and occupancy as variables, with certain correction factors. It establishes the degree of road use in a range from 0 (empty) to 100 (collapse).
  - Average traffic speed - an average speed of the vehicles in a period of 15 minutes (km/h). Only for M30 intercity measuring points.

Although the traffic data is recorded every 15 minutes, since the NO<sub>2</sub> and meteorological data are at hourly rates, the traffic data was filtered, and only hourly records were selected (for example, with entries at 13:00, 13:15, 13:30, 13:45 and 14:00, we selected the entries at 13:00 and 14:00 and the same logic was applied for the entire period).

It should be noted that the original datasets also contained ultraviolet radiation (Mw/m<sup>2</sup>) and precipitation (l/m<sup>2</sup>), which were removed after exploratory data analysis. Regarding wind direction, it was converted to the following categories: north, east, south, west, southwest, northeast, southeast, northwest, and then underwent the implementation of One Hot Encoder<sup>2</sup> [44].

<sup>1</sup>Portal de datos abiertos del Ayuntamiento de Madrid: <https://bit.ly/3FFRIQM>

<sup>2</sup>One Hot Encoder: <https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.OneHotEncoder.html>



**FIGURE 4.** Air quality stations, meteorological stations, traffic measurement points (January 2019) and grid cell segments in the area defined in the city of Madrid.

**TABLE 2.** Summary statistics of the periods January-June 2019 and January-June 2022 for each data type.

Phenomena	Descriptors	January-June 2019	January-June 2022
Nitrogen dioxide ( $\mu\text{g}/\text{m}^3$ )	Mean (SD) Median [Min,Max]	36.63 (30.86) 27.0 [0.0, 328.0]	27.96 (23.66) 20.0 [0.0, 625.0]
Wind speed (m/s)	Mean (SD) Median [Min,Max]	1.33 (1.04) 1.09 [0.0, 8.75]	1.25 (0.89) 1.05 [0.0, 8.18]
Temperature ( $^{\circ}\text{C}$ )	Mean (SD) Median [Min,Max]	13.21 (7.81) 12.3 [-3.0, 40.6]	13.86 (8.36) 12.2 [-2.0, 40.6]
Humidity (%)	Mean (SD) Median [Min,Max]	49.57 (20.86) 48.0 [0.0, 100]	54.90 (20.24) 55.0 [0.0, 100]
Barometric pressure (mb)	Mean (SD) Median [Min,Max]	943.81 (17.89) 944.0 [0.0, 962.0]	944.53 (17.09) 945.0 [0.0, 967.0]
Solar irradiance ( $\text{W}/\text{m}^2$ )	Mean (SD) Median [Min,Max]	223.96 (302.56) 14.0 [0.0, 1103.0]	205.91 (287.15) 14.0 [0.0, 1403.0]
Intensity (vehicles/hour)	Mean (SD) Median [Min,Max]	315.18 (303.05) 252.62 [0.0, 3712.87]	270.58 (258.70) 221.17 [0.0, 2787.15]
Occupancy time (%)	Mean (SD) Median [Min,Max]	5.04 (5.26) 3.68 [0.0, 55.47]	4.46 (4.76) 3.07 [0.0, 41.62]
Load	Mean (SD) Median [Min,Max]	17.51 (14.15) 16.0 [0.0, 93.36]	15.68 (12.80) 14.3 [0.0, 73.74]
Average traffic speed (km/h)	Mean (SD) Median [Min,Max]	0.37 (1.30) 0.0 [0.0, 35.29]	0.06 (0.55) 0.0 [0.0, 52.0]

Since the monitoring stations and measurement points are different for each dataset, the datasets were combined

spatially and temporally based on a defined grid (with a width and height of 1,000 metres within the following

extent: Top -4,486,449.725263 metres; Bottom -4,466,449.725263 metres; Left -434,215.234430 metres; Right -451,215.234430 metres). Regarding the projected coordinate system, EPSG: 25830, ETRS89/UTM zone 30N was used (EPSG: European Petroleum Survey Group, ETRS89: European Terrestrial Reference System 1989, UTM: Universal Transverse Mercator).<sup>3</sup> The grid was created with the help of ArcPy package,<sup>4</sup> specifically with the *CreateFishnet* function.<sup>5</sup>

Altogether, there are 340 cells (20 by 17) that cover 56.27% of the area of the city of Madrid (Figure 4). The reason for selecting this area was to have a minimum extent to encompass all air quality monitoring stations. The value of each cell consists of the values of NO<sub>2</sub>, meteorological and traffic attributes obtained from assigned stations covered by that cell at a certain time. Cells with no stations or measurement points were assigned a value of zero, whereas cells with more than one station or measurement point were assigned the mean value of all existing values in the given cells. It can be observed that only twenty-four cells out of 340 have NO<sub>2</sub> data. After combining all the features, the twenty-four cells mentioned above were selected for further analysis, particularly for use as input for the proposed method working with non-Euclidean distances. As a result, the input data has the dimensions mentioned in Table 3.

**TABLE 3. Dimension of each set.**

Set	Dimension ( $x \times y \times z$ ) <sup>(*)</sup>
Training Set	4344 $\times$ 24 $\times$ 18
Testing Set	4343 $\times$ 24 $\times$ 18

\*  $x$  – Number of samples;  $y$  – Number of stations;  $z$  – Number of the features (NO<sub>2</sub>, wind speed, temperature, humidity, barometric pressure, solar irradiance, intensity, occupancy time, load, average traffic speed, north, east, south, west, southwest, northeast, southeast, northwest). Note that features include wind directions after the implementation of One Hot Encoder.

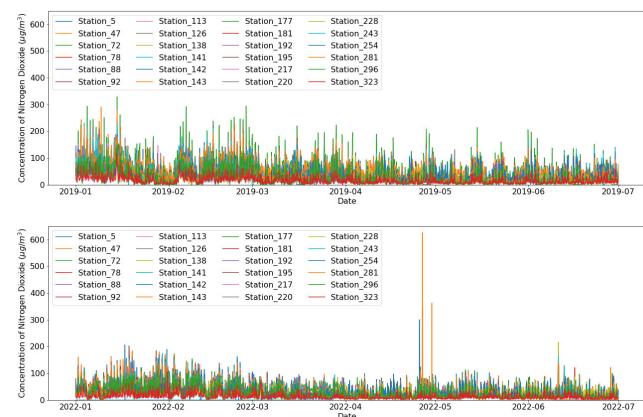
Regarding NO<sub>2</sub>, Figure 5 shows the time series of NO<sub>2</sub> during January-June 2019 and January-June 2022. It can be seen that it decreases over time, which might be attributed to domestic heating use throughout the winter. Moreover, the overall concentration during 2022 is lower than that for the same period in 2019, which can be explained by the constraints enforced to control the spread of coronavirus disease 2019 (COVID-19). From Figure 6 it can be seen that maximum values during January-June 2019 are detected around 300  $\mu\text{g}/\text{m}^3$  and the highest concentration was detected at the station with id 72 (328  $\mu\text{g}/\text{m}^3$  at the following time: 2019-01-14 19:00:00). During the 2022 period, the greatest value was identified in the station with id 47 (625  $\mu\text{g}/\text{m}^3$  at the following time: 2022-04-26 14:00:00), which most probably is an outlier.

Another exploratory analysis was performed to detect the correlation between the time series of NO<sub>2</sub> at the stations

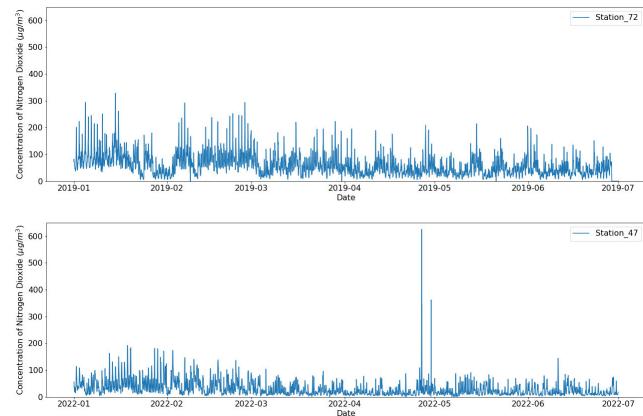
<sup>3</sup>Projected coordinate system: <https://epsg.io/25830>

<sup>4</sup>ArcPy package: <https://bit.ly/3UPYKjy>

<sup>5</sup>Create Fishnet (Data Management): <https://bit.ly/3Rn62Zj>



**FIGURE 5. Time series of the concentration of nitrogen dioxide at all the stations during January-June 2019 (top) and January-June 2022 (bottom) in the city of Madrid.**



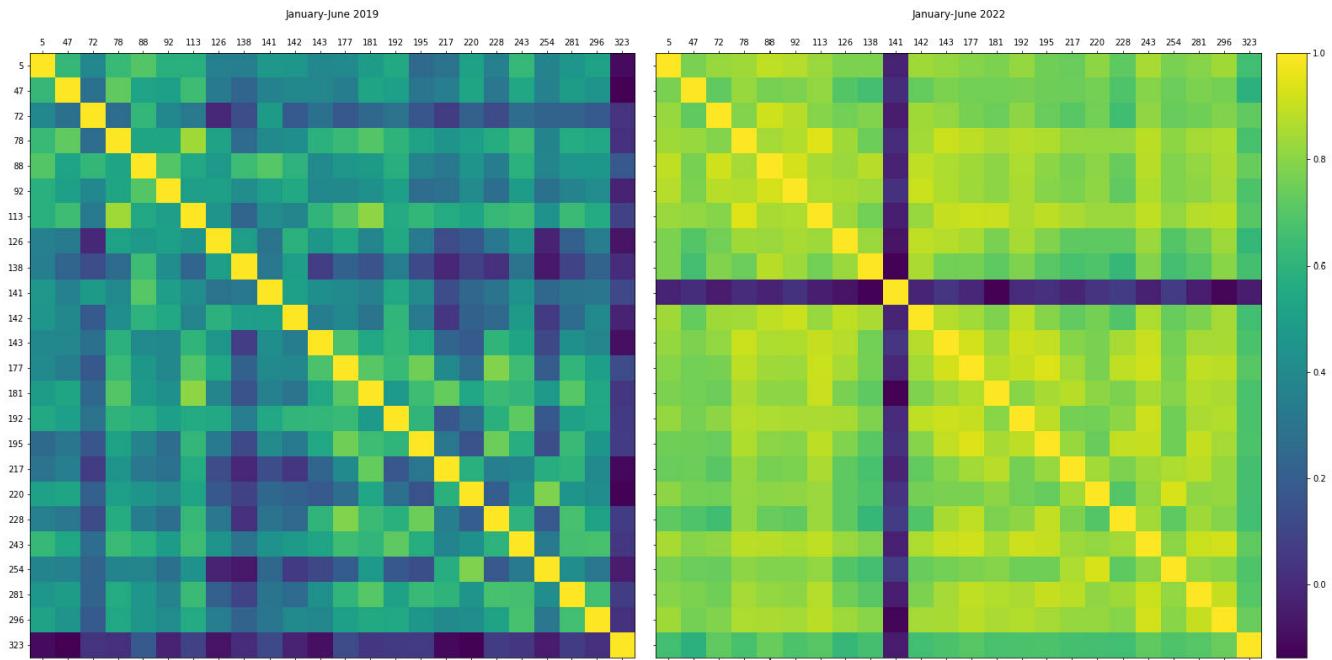
**FIGURE 6. Time series of the concentration of nitrogen dioxide at stations with maximum values for each period in the city of Madrid (top: the station with id 72 during January-June 2019; bottom: the station with id 47 during January-June 2022).**

(Figure 7). During the 2019 period, it can be seen that the stations are correlated, except for the station with id 323, which has a lower correlation than the others. This can be explained by the station's location, which is relatively remote from the others. Furthermore, during 2022, in addition to the station with id 323, the station with id 141 was also found to be less correlated. Looking at Figure 8, which shows the time series of NO<sub>2</sub> concentration at station 141, it can be seen that there is no data for the end of January, as well as for the entire period of March and April, which may be due to a sensor malfunction.

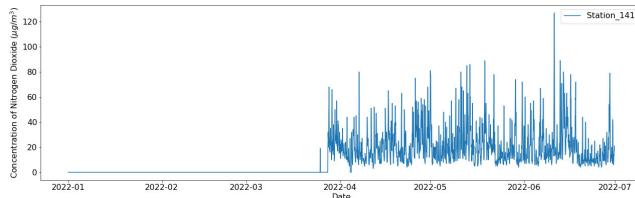
#### IV. METHODOLOGY

##### A. GRAPH NEURAL NETWORK

A graph can be designated as  $G = (V, E)$ , where  $|V| = N$  is the number of nodes and  $|E| = N^e$  is the number of edges.  $A \in R^{N \times N}$  is the adjacency matrix [45]. The graphs are categorised into directed and undirected, based on the connection of the edges. In the case of directed graphs, edges are directed from one node to another, while edges in undirected



**FIGURE 7.** Correlation between the time series of nitrogen dioxide at the stations during January-June 2019 (left) and January-June 2022 (right) in the city of Madrid.



**FIGURE 8.** Time series of the concentration of nitrogen dioxide at the station with id 141 during January-June 2022.

graphs are connected between nodes without specifying the direction. Undirected graphs are sometimes considered to be two directed edges.

Another classification divides them into weighted and unweighted graphs. An unweighted graph only shows whether two nodes are connected or not (in the case of being connected, it is assigned 1, otherwise 0). In contrast, the weighted graph provides additional information about the connected edges; for example, in this work, the distance between stations is assigned as weights.

Regarding learning tasks, there are three categories: node-level (node classification, node regression, node clustering), edge-level (edge classification, link prediction) and graph-level (graph classification, graph regression, graph matching). The graph considered in the scope of this study is an undirected weighted graph, and the learning task is a node regression since the main objective of this work is to predict the concentration of NO<sub>2</sub> in each station in a given time interval.

## B. ATTENTION TEMPORAL GRAPH CONVOLUTIONAL NETWORK

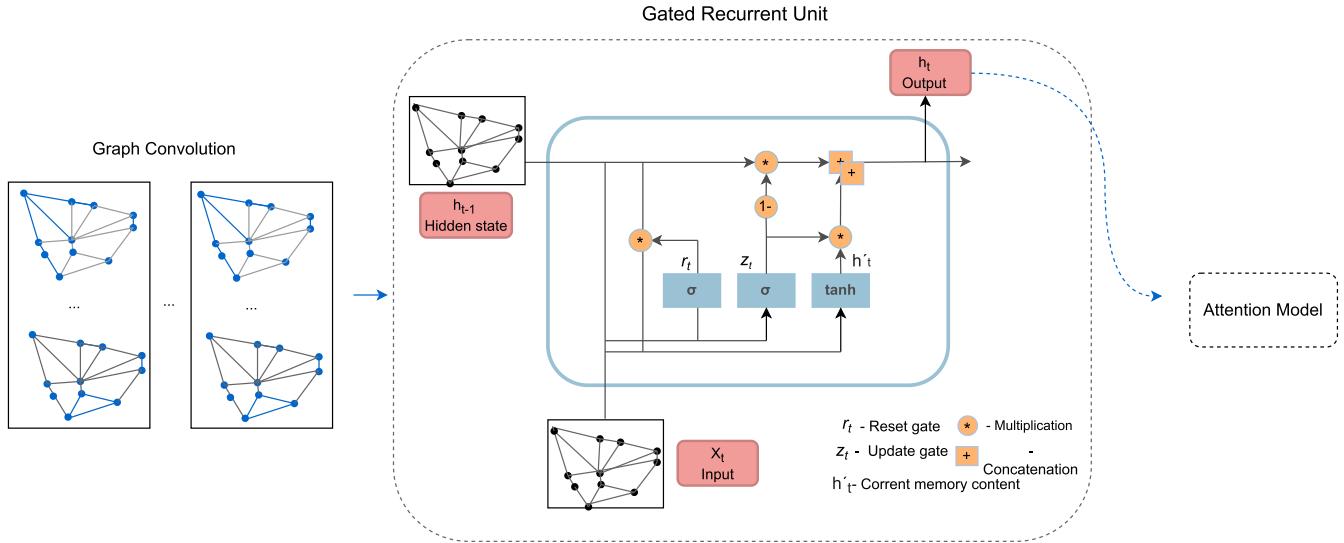
The model proposed in the current work is the A3T-GCN, which is the combination of the Attention, GRU and GCN methods (Figure 9) [43]. The GRU and Attention mechanism are responsible for temporal aggregation and GCN deals with spatial aggregation.

*Graph Convolutional Networks:* There are two types of GCNs: Spatial GCN and Spectral GCN [46]. To learn graphs, spatial GCN uses spatial features. It defines convolutions on spatially close neighbours. It generates  $v_i$  node's representation by aggregating its own features  $X_i$  and neighbours' features  $X_j$ . As an aggregation function is used mean, sum or max functions. Afterwards, a non-linear transformation is applied to the outputs. While in the case of spectral GCN, it defines graph convolutions using filters from the perspective of graph signal processing. Spectral GCN is a combination of the following steps: 1) converting the graph into the spectral domain with the help of eigendecomposition, 2) applying eigendecomposition to the specified kernel, 3) multiplying spectral graph and spectral kernel, and 4) returning the results in the original spatial domain.

The one used in this work was Spectral GCN, which can be defined as the multiplication of a filter  $g_\theta$  with signal  $x$  in the Fourier domain.

$$g_\theta(L) * x = U g_\theta(U^T x) \quad (1)$$

where  $\theta$  is a model parameter,  $L$  is the graph Laplacian matrix (Eq. 2) and  $U$  is the eigenvector of the normalised



**FIGURE 9.** Architecture of the proposed A3T-GCN model.

Laplacian matrix.

$$I_N - D^{-1/2}AD^{-1/2} = U\lambda U^T \quad (2)$$

where  $I_N \in R^{N \times N}$  is the identity matrix,  $D \in R^{N \times N}$  is the diagonal degree matrix and  $\lambda$  is the diagonal matrix of the eigenvalues of the Laplacian matrix.

*Gated Recurrent Unit:* GRU is a type of recurrent neural network introduced by Cho et al. [47]. It can be defined with the following equations:

$$z_t = \sigma(W^{(z)}x_t + U^{(z)}h_{t-1}) \quad (3)$$

$$r_t = \sigma(W^{(r)}x_t + U^{(r)}h_{t-1}) \quad (4)$$

$$h'_t = \tanh(Wx_t + r_t \odot Uh_{t-1}) \quad (5)$$

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot h'_t \quad (6)$$

where  $x_t$  is the input vector at the current time step,  $z_t$  is the update gate,  $r_t$  is the reset gate,  $h'_t$  is the current memory content,  $h_{t-1}$  is the hidden state at the previous time step,  $h_t$  is the hidden state at the current time step and  $\odot$  is the Hadamard product.

*Attention:* The Attention model focuses on a few relevant things in the complex input while ignoring others in networks. Bahdanau et al. [48] proposed the attention mechanism in order to overcome the drawbacks of RNN, in particular, the inability to remember longer sequences. The equation defining the attention model is shown below.

$$c_i = \sum_{j=1}^{T_x} a_{ij}h_j \quad (7)$$

where  $c_i$  is the context vector,  $a_{ij}$  is the weights and  $h_j$  is the hidden state.  $a_{ij}$  can be calculated with the following equations.

$$a_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T_x} \exp(e_{ik})} \quad (8)$$

$$e_{ij} = a(s_{i-1}, h_j) \quad (9)$$

where  $e_{ij}$  is the output score of a feedforward neural network (alignment scores).

There are several categories of the Attention mechanism. Soft Attention is the type used in this study (soft Attention refers to the function that varies smoothly over its domain and, as a result, can be differentiated).

### C. EVALUATION METRICS

To evaluate the performance of the models, the following metrics were used in the ongoing work, including RMSE, MAE and R. RMSE measures the geometric difference between the estimated and actual values, and is susceptible to significant errors; MAE measures the average magnitude of the errors, and R measures the strength and the direction of a linear relationship between two variables.

$$RMSE = \left( \frac{1}{n} \sum_{i=1}^n (E_i - A_i)^2 \right)^{1/2} \quad (10)$$

$$MAE = \frac{1}{n} \sum_{i=1}^n |E_i - A_i| \quad (11)$$

$$R = \frac{\sum_{i=1}^n (E_i - \bar{E}_i)(A_i - \bar{A}_i)}{\sqrt{\sum_{i=1}^n (E_i - \bar{E}_i)^2 \sum_{i=1}^n (A_i - \bar{A}_i)^2}} \quad (12)$$

where  $n$  is the number of instances and  $E_i$  and  $A_i$  are the estimated and actual values. Regarding RMSE and MAE, the lower the value is, the better the prediction will be. In terms of R, the values range from  $-1$  to  $+1$ , where  $-1$  means a perfectly linear negative correlation,  $+1$  means a perfectly linear absolute correlation, and  $0$  means no correlation.

### D. GRAPH CONSTRUCTION

Following the definition of the graph structure, in this work, air quality stations will be considered graph nodes. All stations are interconnected, forming graph edges, and the distances between them will be considered edge weights.

The distance between nodes was calculated using the  `arcpy.analysis.GenerateNearTable6` function. It should be mentioned that to create the adjacency matrix, the original distance between two nodes was converted to 1/distance (Eq.13), so if the distance is large, the division will be smaller, and this will give little weight to a certain edge, which matches the graph logic since closer nodes have more influence on each other than remote nodes.

$$A_{ij} = \begin{cases} \frac{1}{d_{ij}}, & i \neq j \\ 0, & \text{otherwise} \end{cases} \quad (13)$$

where  $d_{ij}$  is the distance between  $i$  and  $j$  stations.

Regarding node features, all variables associated with each station will be considered node features; in our case, for each time  $t$  the node features can be assigned as  $X_t \in R^{N \times M}$ , where  $N$  is the number of nodes and  $M$  is the features. Figure 10 shows the graph constructed on the basis of air quality stations. It consists of 24 nodes and 276 edges (connecting each pair of nodes). The numbers on the nodes are the identifier of each cell of the grid that was initially given, which contains a certain station. Algorithm 1 shows the procedure for creating a graph network on the map.

#### Algorithm 1 Creating Graph Network on the City of Madrid

```

Input: Grid with 340 Cells (20*17)
function EXTRACT CELLS HAVING AIR QUALITY
STATIONS(data)
    return 24 cells including air quality stations
end function
function CALCULATE CENTROIDS OF THE
EXTRACTED CELLS AND CREATE SEPARATE
FEATURE CLASS7(24 extracted cells)
    return feature class of centroids
end function
function DRAW NETWORK BETWEEN EACH PAIR
OF POINTS(feature class of centroids)
    return Draw a Network between each
pair of points with all combinations using the
arcpy.management.XYToLine8 function
end function
Output: Figure 10

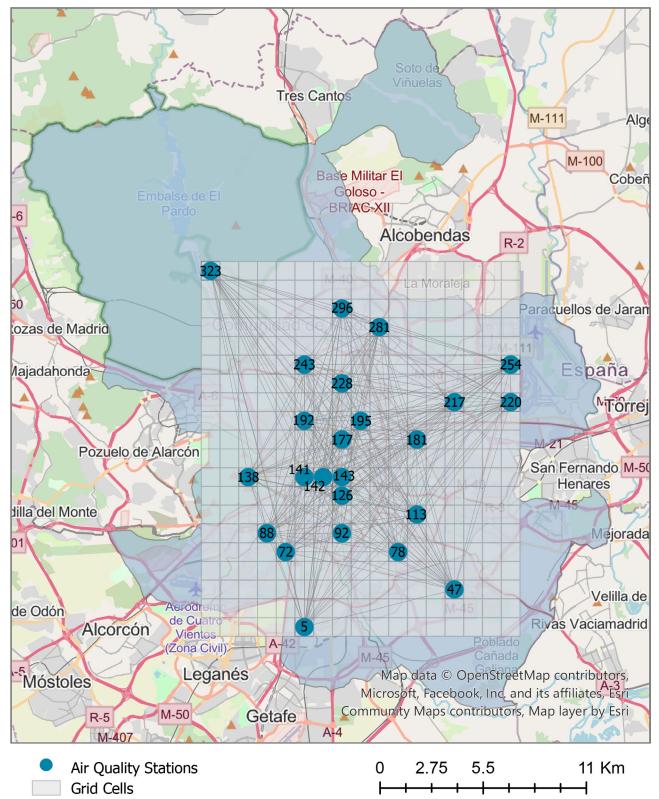
```

The prediction of NO<sub>2</sub> was performed on the basis of different time granularities, in particular, using the previous 12 hours to predict the concentration in the next  $T$  hours. The following time intervals have been defined as the value of  $T$ : 1-12 h, 12-24 h, 24-36 h and 36-48 h. In the mathematical expression, the procedure above can be defined as a function of the air quality stations network  $G$  and the feature

<sup>6</sup>Generate Near Table (Analysis): <https://pro.arcgis.com/en/pro-app/latest/tool-reference/analysis/generate-near-table.htm>

<sup>7</sup>Find the centroid of polygons using the Calculate Geometry Attributes tool in ArcGIS Pro: <https://bit.ly/3EhGGbV>

<sup>8</sup>XY To Line: <https://bit.ly/3UoRJFK>



**FIGURE 10. Graph network of the air quality stations placed in the city of Madrid.**

matrix  $X$  (Eq. 14).

$$[X_{t+1}^a, \dots, X_{t+T}^a] = f(G; (X_{t-n}, \dots, X_{t-1}, X_t)) \quad (14)$$

where  $T$  is the next hours,  $n$  is the previous hours,  $X_t^a$  is the concentration of NO<sub>2</sub> at time  $t$ , and  $X_t$  is a combination of NO<sub>2</sub>, meteorological and traffic data. Each sample of the input has the following structure:

Data( $x = [24, 18, T]$ ,  $\text{edge\_index} = [2, 552]$ ,  
 $\text{edge\_attr} = [552]$ ,  $y = [24, T]$ ,  $\text{batch} = [64]$ )

where 24 is the number of nodes, 18 is the number of features of each node,  $T$  is equal to 12, [2, 552] from  $\text{edge\_index}$  refers to the fact that every edge was considered twice ( $276 \times 2 = 552$ ). Algorithm 2 shows the procedure of input data preparation for the graph neural network.

The parameters and settings applied in the analysis are summarised in Table 4. Regarding the structure of the proposed model, it consists of three graph convolutional layers (the output of the layers is used in GRU: update gate, reset gate and hidden state) followed by learnable transformations.

Regarding the reference models, we used a TGCN, LSTM and GRU. Below is demonstrated the architecture of each of them.

**Algorithm 2** Data Preparation for Graph Neural Network

**Input:** Data - [Hourly NO<sub>2</sub>, Meteorological and Traffic data]; Period-[01.01.2019-30.06.2019; 01.01.2022-30.06.2022]

```

1: function MERGE THE DATA SPATIALLY AND TEMPORALLY(data)
2:   for each hour ∈ Period do
3:     Create grid with CreateFishnet function (ArcPy library)
4:     Add field to the created grid
5:     for each item i ∈ Data do
6:       i spatial join with grid
7:       input the mean of the values of each corresponding cell to the field
8:     end for
9:   end for
10:  return .csv files for each hour including NO2, Meteorological and Traffic data [4344 and 4343 .csv files for January-June 2019 and January-June 2022, respectively with the following dimension: 4344 × 340 × 18: (January-June 2019); 4343 × 340 × 18 (January-June 2022)]
11: end function
12: function EXTRACT CELLS HAVING AIR QUALITY STATIONS(data)
13:  return extract cells or rows including NO2, Meteorological and Traffic data, where air quality monitoring stations exist [4344 × 24 × 18: (January-June 2019); 4343 × 24 × 18 (January-June 2022)]
14: end function
15: function CREATE ADJACENCY MATRIX(location of Air Quality stations (nodes))
16:  calculate distance between each pair of air quality stations using arcpy.analysis.GenerateNearTable
17:  return adjacency matrix filled with inverse distance (1/distance) between each pair of nodes
18: end function
19: function NORMALISE DATA(data)
20:  return normalised data using Z-Score method
21: end function
22: function GENERATING DATASET FOR GNN BASED ON THE DEFINED TIME INTERVAL: [1-12, 12-24, 24-36 AND 36-48] (T=12)(data)
23:  return Dataset with 8687 samples (Training Set-4344, Testing Set-4343; each sample-Data(x=[24, 18, T], edge_index=[2, 552], edge_attr=[552], y=[24, T], batch=[64]))
24: end function
```

**Output:**Dataset with 8687 samples (each sample: Data(x=[24, 18, T], edge\_index=[2, 552], edge\_attr=[552], y=[24, T], batch=[64]))

**TGCN** [49]: is a combination of GCN and GRU. The architecture is composed of three graph convolutional layers (the output of the layers is used in GRU: update gate, reset

**TABLE 4.** Details of the experimental settings.

Parameter	Value
Number of records	8687
Time interval (h)	1
Training set	January-June 2019: 4344
Testing set	January-June 2022: 4343
Prediction length (T, h)	[1-12, 12-24, 24-36, 36-48]
History length (T, h)	12
Number of stations	24
Training epochs	100
Learning rate	0.1
Batch size	64
Hidden units	256
Optimiser	Adam
Loss function	Mean Squared Error

gate and hidden state) followed by learnable transformations. Regarding the hidden unit, it was assigned to 256.

**LSTM** [50]: It consists of the fully connected layer with 432 units (24\*18), followed by the LSTM layer with 512 units, and then the models were finalised with another fully connected layer with 24 units (representing NO<sub>2</sub> for all stations).

**GRU** [47]: It consists of the fully connected layer with 432 units (24\*18), followed by the GRU layer with 512 units, and then the models were finalised with another fully connected layer with 24 units (representing NO<sub>2</sub> for all stations).

It should be mentioned that the analysis was performed in the Google Colab cloud service<sup>9</sup> using the PyTorch Geometric Temporal library [51] (the source code can be accessed at the GitHub repository<sup>10</sup>).

**V. EXPERIMENTAL ANALYSIS AND RESULTS**

This section presents a detailed explanation of the experimental analysis and results obtained. The workflow is shown in Figure 11. In the first stage, the data were integrated with the spatiotemporal dimension in a defined area, followed by the graph construction, which has already been introduced in the previous chapter. The next block is feature engineering, consisting of transformation and scaling steps.

**Transformation:** this step refers to generating data based on the defined time interval. The defined time interval is 1-12 h, 12-24 h, 24-36 h and 36-48 h.

**Scaling:** before converting the data into a graph construction, the input data were standardised (the standardisation is also called Z-score; Eq. 15).

$$X' = \frac{X - \mu}{\sigma} \quad (15)$$

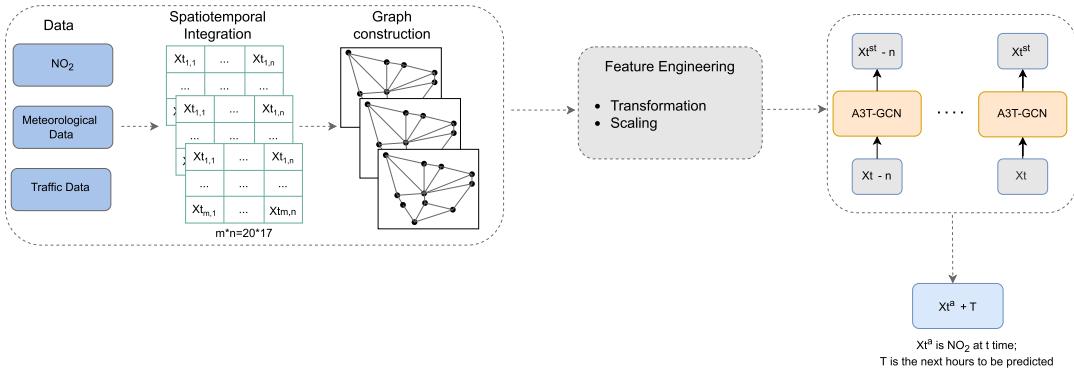
where  $\mu$  is the mean, and  $\sigma$  is the standard deviation.

The next block is the architecture of the proposed model that returns the results of the experimental analysis demonstrated in the next part.

This part illustrates the output of the analysis. The results of the analysis are shown in Table 5 (the best results are

<sup>9</sup>Google Colab: <https://colab.research.google.com/notebooks/intro.ipynb>

<sup>10</sup>GitHub repository: [https://github.com/Ditsuhi/GNN\\_Air\\_Quality](https://github.com/Ditsuhi/GNN_Air_Quality)



**FIGURE 11.** Workflow of the proposed methodology.

**TABLE 5.** Performance evaluation metrics of A3T-GCN, TGCN, LSTM and GRU in terms of time granularities.

Method	Hours	RMSE ( $\mu\text{g}/\text{m}^3$ )	MAE ( $\mu\text{g}/\text{m}^3$ )	R
A3T-GCN	1-12	<b>19.14</b>	<b>15.33</b>	<b>0.59</b>
	12-24	19.85	15.91	0.52
	24-36	21.85	17.87	0.47
	36-48	21.54	17.69	0.46
TGCN	1-12	<b>21.48</b>	<b>16.24</b>	<b>0.49</b>
	12-24	22.73	17.35	0.42
	24-36	23.76	18.48	0.40
	36-48	23.38	18.28	0.40
LSTM	1-12	<b>22.33</b>	<b>16.70</b>	<b>0.57</b>
	12-24	23.16	17.43	0.53
	24-36	26.38	19.87	0.46
	36-48	24.78	18.79	0.46
GRU	1-12	22.52	17.27	<b>0.56</b>
	12-24	<b>22.29</b>	<b>16.97</b>	0.54
	24-36	25.38	19.41	0.47
	36-48	23.45	17.45	0.31

indicated in bold). Algorithm 3 provides the pseudo-code of the  $\text{NO}_2$  prediction procedure.

Looking at the results in Table 5, first of all, it can be seen that for the A3T-GCN, TGCN and LSTM models, the time interval of 1-12 hours is superior to other time intervals in terms of all three evaluation metrics, and in the case of GRU, the leading time interval is 12-24 hours in terms of RMSE and MAE, and 1-12 hours in terms of R.

Regarding individual model performance, the A3T-GCN outperformed all three reference models. Especially, in terms of RMSE, the proposed method ( $19.14 \mu\text{g}/\text{m}^3$ ) outperformed TGCN ( $21.48 \mu\text{g}/\text{m}^3$ ) by 10.89%, LSTM ( $22.33 \mu\text{g}/\text{m}^3$ ) by 14.29%, and GRU ( $22.29 \mu\text{g}/\text{m}^3$ ) by 14.13%. In terms of MAE, the A3T-GCN ( $15.33 \mu\text{g}/\text{m}^3$ ) outperformed TGCN ( $16.24 \mu\text{g}/\text{m}^3$ ) by 5.6%, LSTM ( $16.70 \mu\text{g}/\text{m}^3$ ) by 8.2%, and GRU ( $16.97 \mu\text{g}/\text{m}^3$ ) by 9.7%. In terms of R, the A3T-GCN (0.59) outperformed TGCN (0.49) by 16.95%, LSTM (0.57) by 3.39%, and GRU (0.56) by 5.08%.

Looking at R, it can be noticed that the values are in the range of 0.49 to 0.59. Although the proposed method outperforms the reference methods, further improvements can be made. Regarding RMSE and MAE, their units match

### Algorithm 3 $\text{NO}_2$ Prediction

**Input:** Dataset with 8687 samples (Training Set-4344, Testing Set-4343; each sample-Data( $x=[24, 18, T]$ ,  $\text{edge\_index}=[2, 552]$ ,  $\text{edge\_attr}=[552]$ ,  $y=[24, T]$ ,  $\text{batch}=[64]$ ))

**function** CREATE MODEL

2:   **return** A3T-GCN architecture based on the settings from Table 4

**end function**

4: **function** EVALUATE MODEL

**return** error estimated with evaluation metrics (RMSE, MAE, R)

6: **end function**

**function** EVALUATE REFERENCE MODELS (TGCN, LSTM AND GRU)(models with defined parameters)

8:   **return** error estimated with evaluation metrics (RMSE, MAE, R)

**end function**

**Output:** RMSE, MAE, R for A3T-GCN, TGCN, LSTM and GRU (Table 5)

with the unit of the target variable ( $\text{NO}_2: \mu\text{g}/\text{m}^3$ ). Therefore, based on the results obtained (RMSE- $19.14 \mu\text{g}/\text{m}^3$ , MAE- $15.33 \mu\text{g}/\text{m}^3$ ), the proposed method can be considered sufficient compared with the mean values of  $\text{NO}_2$  (36.69 and 27.96 for the period 2019 and 2022, respectively).

It is important to mention that when comparing only reference methods between them, it can be noticed that TGCN outperforms the other two methods (LSTM and GRU). Especially in terms of RMSE, TGCN ( $21.48 \mu\text{g}/\text{m}^3$ ) outperformed LSTM ( $22.33 \mu\text{g}/\text{m}^3$ ) by 3.81%, and GRU ( $22.29 \mu\text{g}/\text{m}^3$ ) by 3.63%. In terms of MAE, TGCN ( $16.24 \mu\text{g}/\text{m}^3$ ) outperformed LSTM ( $16.70 \mu\text{g}/\text{m}^3$ ) by 2.75%, and GRU ( $16.97 \mu\text{g}/\text{m}^3$ ) by 4.3%. Since TGCN is also a graph-based method, based on these findings, the advantage of a graph-based method with the ability to capture spatial dependencies in addition to temporal dependencies can be highlighted.

## VI. CONCLUSION AND FUTURE WORK

The spatiotemporal prediction of air quality has attracted the attention of many authors. Having access to a large amount of data creates challenges for their proper integration. New technologies with constant development offer new approaches to conducting spatiotemporal analysis. The approach proposed in this study is A3T-GCN. The main goal is to predict NO<sub>2</sub> using data from Madrid air quality monitoring stations combined with meteorological and traffic data from January 2019 to June 2019 (training set) and January 2022 to June 2022 (testing set). RMSE, MAE and R were applied as evaluation metrics. The results highlighted the superiority of the proposed method over the reference methods (TGCN, LSTM and GRU). In terms of RMSE, the A3T-GCN outperformed TGCN by 10.89%, LSTM by 14.29%, and GRU by 14.13%. In terms of MAE, the A3T-GCN outperformed TGCN by 5.6%, LSTM by 8.2%, and GRU by 9.7%. In terms of R, the A3T-GCN outperformed TGCN by 16.95%, LSTM by 3.39%, and GRU by 5.08%.

Regarding future work, it would be interesting to implement this approach in other regions and see if it can be generalised in terms of spatial dimension. The final performance is likely to be affected by the spatial peculiarities of different regions, as well as the distance between stations, the number of stations, and the available features. Another extension can be related to the approach to graph construction. Considering that an undirected graph was used in this work, it would be advantageous to use a directed graph, since the importance of the node V<sub>i</sub> on V<sub>j</sub> is different from that of V<sub>j</sub> on V<sub>i</sub>. It would also be preferable to consider the topology, buildings and infrastructure connecting the two nodes in relation to the weighted edges that were created by the inverse distance between the two nodes. The constraints related to the construction of the graph can be addressed in future work. Additionally, it is advisable to change the architecture of the proposed method and stack more layers, which will probably improve the results. Another improvement could be to include data for a more extended period.

Regarding the limitations, the proposed method requires a relatively long time for the data to converge. As regards experimental analysis, it is worth mentioning the limitation associated with Google Colab Pro as it has limitations in terms of resources<sup>11</sup>.

## REFERENCES

- [1] P. Rafaj, G. Kiesewetter, T. Güll, W. Schöpp, J. Cofala, Z. Klimont, and P. Purohit, "Outlook for clean air in the context of sustainable development goals," *Global Environ. Change*, vol. 53, pp. 1–11, Nov. 2018.
- [2] P. J. Landrigan, R. Fuller, N. J. R. Acosta, O. Adeyi, R. Arnold, A. B. Baldé, and R. Bertollini, "The Lancet commission on pollution and health," *Lancet*, vol. 391, no. 10119, pp. 462–512, 2018.
- [3] R. A. Rohde and R. A. Müller, "Air pollution in China: Mapping of concentrations and sources," *PLoS ONE*, vol. 10, no. 8, Aug. 2015, Art. no. e0135749.
- [4] *State of Global Air 2019*, Health Effects Institute, Boston, MA, USA, 2019.
- [5] D. A. Vallero, *Fundamentals of Air Pollution*. New York, NY, USA: Academic, 2014.
- [6] J. Fan, Q. Li, J. Hou, X. Feng, H. Karimian, and S. Lin, "A spatiotemporal prediction framework for air pollution based on deep RNN," *ISPRS Ann. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. 4, p. 15, Jan. 2017.
- [7] D. Seng, Q. Zhang, X. Zhang, G. Chen, and X. Chen, "Spatiotemporal prediction of air quality based on LSTM neural network," *Alexandria Eng. J.*, vol. 60, no. 2, pp. 2021–2032, Apr. 2021.
- [8] D. Iskandaryan, F. Ramos, and S. Trilles, "Comparison of nitrogen dioxide predictions during a pandemic and non-pandemic scenario in the city of Madrid using a convolutional LSTM network," *Int. J. Comput. Intell. Appl.*, vol. 21, no. 2, Jun. 2022, Art. no. 2250014.
- [9] D. Iskandaryan, F. Ramos, and S. Trilles, "Bidirectional convolutional LSTM for the prediction of nitrogen dioxide in the city of Madrid," *PLoS ONE*, vol. 17, no. 6, Jun. 2022, Art. no. e0269295.
- [10] G. Zhao, G. Huang, H. He, H. He, and J. Ren, "Regional spatiotemporal collaborative prediction model for air quality," *IEEE Access*, vol. 7, pp. 134903–134919, 2019.
- [11] S. Abirami and P. Chitra, "Regional air quality forecasting using spatiotemporal deep learning," *J. Cleaner Prod.*, vol. 283, Feb. 2021, Art. no. 125341.
- [12] Y. Bai, B. Zeng, C. Li, and J. Zhang, "An ensemble long short-term memory neural network for hourly PM<sub>2.5</sub> concentration forecasting," *Chemosphere*, vol. 222, pp. 286–294, May 2019.
- [13] G. Huang, X. Li, B. Zhang, and J. Ren, "PM<sub>2.5</sub> concentration forecasting at surface monitoring sites using GRU neural network based on empirical mode decomposition," *Sci. Total Environ.*, vol. 768, May 2021, Art. no. 144516.
- [14] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," 2017, *arXiv:1707.01926*.
- [15] M. Li and Z. Zhu, "Spatial-temporal fusion graph neural networks for traffic flow forecasting," in *Proc. AAAI Conf. Artif. Intell.*, May 2021, vol. 35, no. 5, pp. 4189–4196.
- [16] F. Zhou, Q. Yang, K. Zhang, G. Trajcevski, T. Zhong, and A. Khokhar, "Reinforced spatiotemporal attentive graph neural networks for traffic forecasting," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 6414–6428, Jul. 2020.
- [17] X. Wang, Y. Ma, Y. Wang, W. Jin, X. Wang, J. Tang, C. Jia, and J. Yu, "Traffic flow prediction via spatial temporal graph neural network," in *Proc. Web Conf.*, Apr. 2020, pp. 1082–1092.
- [18] W. Zhang, H. Liu, Y. Liu, J. Zhou, and H. Xiong, "Semi-supervised hierarchical recurrent graph neural network for city-wide parking availability prediction," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, 2020, pp. 1186–1193.
- [19] H. Zhou, D. Ren, H. Xia, M. Fan, X. Yang, and H. Huang, "AST-GNN: An attention-based spatio-temporal graph neural network for interaction-aware pedestrian trajectory prediction," *Neurocomputing*, vol. 445, pp. 298–308, Jul. 2021.
- [20] A. Mohamed, K. Qian, M. Elhoseiny, and C. Claudel, "Social-STGCNN: A social spatio-temporal graph convolutional neural network for human trajectory prediction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 14424–14432.
- [21] Z. Xu, Y. Kang, Y. Cao, and Z. Li, "Spatiotemporal graph convolution multifusion network for urban vehicle emission prediction," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 8, pp. 3342–3354, Aug. 2021.
- [22] M. Khodayar and J. Wang, "Spatio-temporal graph deep neural network for short-term wind speed forecasting," *IEEE Trans. Sustain. Energy*, vol. 10, no. 2, pp. 670–681, Apr. 2019.
- [23] M. Ma, P. Xie, F. Teng, T. Li, B. Wang, S. Ji, and J. Zhang, "HiSTGNN: Hierarchical spatio-temporal graph neural networks for weather forecasting," 2022, *arXiv:2201.09101*.
- [24] M. Khodayar, S. Mohammadi, M. E. Khodayar, J. Wang, and G. Liu, "Convolutional graph autoencoder: A generative deep neural network for probabilistic spatio-temporal solar irradiance forecasting," *IEEE Trans. Sustain. Energy*, vol. 11, no. 2, pp. 571–583, Apr. 2020.
- [25] J. Han, H. Liu, H. Xiong, and J. Yang, "Semi-supervised air quality forecasting via self-supervised hierarchical graph neural network," *IEEE Trans. Knowl. Data Eng.*, early access, Feb. 9, 2022, doi: 10.1109/TKDE.2022.3149815.
- [26] R. S. Ram, K. Venkatachalam, M. Masud, and M. Abouhawwash, "Air pollution prediction using dual graph convolution LSTM technique," *Intell. Autom. Soft Comput.*, vol. 33, no. 3, pp. 1639–1652, 2022.

<sup>11</sup>Drawbacks of Google Colab: <https://bit.ly/3xJVcof> <https://bit.ly/3Ok7s6g>

- [27] X. Xiao, Z. Jin, S. Wang, J. Xu, Z. Peng, R. Wang, W. Shao, and Y. Hui, “A dual-path dynamic directed graph convolutional network for air quality prediction,” *Sci. Total Environ.*, vol. 827, Jun. 2022, Art. no. 154298.
- [28] X. Ouyang, Y. Yang, Y. Zhang, and W. Zhou, “Spatial-temporal dynamic graph convolution neural network for air quality prediction,” in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2021, pp. 1–8.
- [29] L. Ge, K. Wu, Y. Zeng, F. Chang, Y. Wang, and S. Li, “Multi-scale spatiotemporal graph convolution network for air quality prediction,” *Int. J. Speech Technol.*, vol. 51, no. 6, pp. 3491–3505, Jun. 2021.
- [30] C. Wang, Y. Zhu, T. Zang, H. Liu, and J. Yu, “Modeling inter-station relationships with attentive temporal graph convolutional network for air quality prediction,” in *Proc. 14th ACM Int. Conf. Web Search Data Mining*, Mar. 2021, pp. 616–634.
- [31] L. Chen, J. Xu, B. Wu, Y. Qian, Z. Du, Y. Li, and Y. Zhang, “Group-aware graph neural network for nationwide city air quality forecasting,” 2021, *arXiv:2108.12238*.
- [32] J. Xu, L. Chen, M. Lv, C. Zhan, S. Chen, and J. Chang, “HighAir: A hierarchical graph neural network-based air quality forecasting method,” 2021, *arXiv:2101.04264*.
- [33] Y. Liu, J. Ma, P. Dhillon, and Q. Mei, “A new benchmark of graph learning for PM<sub>2.5</sub> forecasting under distribution shift,” in *Proc. Workshop Graph Learn. Benchmarks Web Conf. (GLB)*. New York, NY, USA: ACM, 2021, p. 6.
- [34] V.-D. Le, T.-C. Bui, and S.-K. Cha, “Spatiotemporal graph convolutional recurrent neural network model for citywide air pollution forecasting,” *TechRxiv*, Jul. 2021, doi: 10.36227/techrxiv.14958552.v1.
- [35] G. Zhao, H. He, Y. Huang, and J. Ren, “Near-surface PM<sub>2.5</sub> prediction combining the complex network characterization and graph convolution neural network,” *Neural Comput. Appl.*, vol. 33, no. 24, pp. 17081–17101, Dec. 2021.
- [36] X. Gao and W. Li, “A graph-based LSTM model for PM<sub>2.5</sub> forecasting,” *Atmos. Pollut. Res.*, vol. 12, no. 9, 2021, Art. no. 101150.
- [37] J. Zhang, Z. Wang, Y. Liu, R. Gai, L. Hu, and Q. Zou, “Temporal attention with domain-specific graph regularization for PM<sub>2.5</sub> forecasting,” in *Proc. Int. Conf. Artif. Intell. Comput. Eng. (ICAICE)*, Oct. 2020, pp. 510–517.
- [38] S. Wang, Y. Li, J. Zhang, Q. Meng, L. Meng, and F. Gao, “PM<sub>2.5</sub>-GNN: A domain knowledge enhanced graph neural network for PM<sub>2.5</sub> forecasting,” in *Proc. 28th Int. Conf. Adv. Geograp. Inf. Syst.*, Nov. 2020, pp. 163–166.
- [39] P. Zhao and K. Zettsu, “MASTGN: Multi-attention spatio-temporal graph networks for air pollution prediction,” in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2020, pp. 1442–1448.
- [40] Y. Qi, Q. Li, H. Karimian, and D. Liu, “A hybrid model for spatiotemporal forecasting of PM<sub>2.5</sub> based on graph convolutional neural network and long short-term memory,” *Sci. Total Environ.*, vol. 664, pp. 1–10, May 2019.
- [41] Y. Huang, J. J.-C. Ying, and V. S. Tseng, “Spatio-attention embedded recurrent neural network for air quality prediction,” *Knowl.-Based Syst.*, vol. 233, Dec. 2021, Art. no. 107416.
- [42] Y. Lin, N. Mago, Y. Gao, Y. Li, Y.-Y. Chiang, C. Shahabi, and J. L. Ambite, “Exploiting spatiotemporal patterns for accurate air quality forecasting using deep learning,” in *Proc. 26th ACM SIGSPATIAL Int. Conf. Adv. Geograp. Inf. Syst.*, Nov. 2018, pp. 359–368.
- [43] J. Bai, J. Zhu, Y. Song, L. Zhao, Z. Hou, R. Du, and H. Li, “A3T-GCN: Attention temporal graph convolutional network for traffic forecasting,” *ISPRS Int. J. Geo-Inf.*, vol. 10, no. 7, p. 485, 2021.
- [44] D. Iskandaryan, S. Di Sabatino, F. Ramos, and S. Trilles, “Exploratory analysis and feature selection for the prediction of nitrogen dioxide,” *AGILE, GISci. Ser.*, vol. 3, p. 6, Jun. 2022. [Online]. Available: <https://agile-giss.copernicus.org/articles/3/6/2022/>
- [45] J. L. Gross and J. Yellen, *Handbook of Graph Theory*. Boca Raton, FL, USA: CRC Press, 2003.
- [46] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, “Spectral networks and locally connected networks on graphs,” 2013, *arXiv:1312.6203*.
- [47] K. Cho, B. van Merriënboer, D. Bahdanau, and Y. Bengio, “On the properties of neural machine translation: Encoder-decoder approaches,” 2014, *arXiv:1409.1259*.
- [48] D. Bahdanau, K. Cho, and Y. Bengio, “Neural machine translation by jointly learning to align and translate,” 2014, *arXiv:1409.0473*.
- [49] L. Zhao, Y. Song, C. Zhang, Y. Liu, P. Wang, T. Lin, M. Deng, and H. Li, “T-GCN: A temporal graph convolutional network for traffic prediction,” *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 9, pp. 3848–3858, Sep. 2020.
- [50] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [51] B. Rozemberczki, P. Scherer, Y. He, G. Panagopoulos, A. Riedel, M. Astefanoaei, O. Kiss, F. Beres, G. López, N. Collignon, and R. Sarkar, “PyTorch geometric temporal: Spatiotemporal signal processing with neural machine learning models,” in *Proc. 30th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2021, pp. 4564–4573.



**DITSUHI ISKANDARYAN** received the master’s degree (Erasmus Mundus) in geospatial technologies. She is currently pursuing the Ph.D. degree in computer science with Universitat Jaume I, Spain. She worked as a Junior Specialist at the Department of Water State Cadaster and Balance, Armenian State Hydrometeorological and Monitoring Service, a Senior Laboratory Assistant at the Department of GIS and Remote Sensing, Centre for Ecological-Noosphere Studies (CENS), and a GIS Specialist at the Department of GIS, Centre for Geodesy and Cartography. Her research interest includes analyzing geospatial data using machine learning technologies.



**FRANCISCO RAMOS** received the Ph.D. degree (Hons.) from Universitat Jaume I (UJI), Spain, in 2008. He is currently a Senior Lecturer with the Department of Computer Languages and Systems, UJI. He also teaches the Master’s Science degree in geospatial technologies and the Director of the master’s degree in mobile programming and videogames at UJI. He is the CTO and the Co-Founder of Emotional Apps, an enterprise which combines technology and emotions to create innovative mobile apps. This company has been featured on the full-frame show (CCTV America), in TIME magazine or Yahoo News, to name but a few. His research interests include mobile interaction, computer graphics, visualization, and GIS.



**SERGIO TRILLES** received the Ph.D. degree in integration of geospatial information from Universitat Jaume I, in 2015. He is currently a Postdoctoral Fellow with Universitat Jaume I, where he held a Juan de la Cierva-Incorporacion Fellowship. He is the author of more than 50 peer-reviewed journal and conference publications. His research interests include geospatial fields, the Internet of Things (sensors), interoperability, geoprocessing, and web mapping.