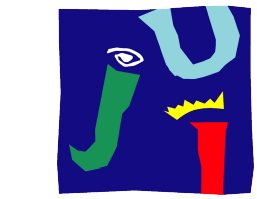


Study and Prediction of Air Quality in Smart Cities through Machine Learning Techniques Considering Spatiotemporal Components

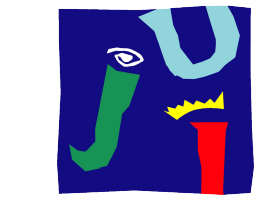
Doctoral Thesis

Ditsuhi Iskandaryan

Supervisors: Dr. Francisco Ramos
Dr. Sergio Trilles

A dissertation presented for the degree of Doctor of Computer Science

Castelló de la Plana (Spain)
February 2023



Programa de Doctorado en Informática

Escuela de Doctorado de la Universitat Jaume I

ESTUDIO Y PREDICCIÓN DE LA CALIDAD DEL AIRE EN CIUDADES INTELIGENTES MEDIANTE TÉCNICAS DE APRENDIZAJE AUTOMÁTICO CONSIDERANDO COMPONENTES ESPACIOTEMPORALES

Memoria presentada por Ditsuhi Iskandaryan para optar al grado de doctor por la Universitat Jaume I

Autor
Ditsuhi Iskandaryan

Directores
Dr. Francisco Ramos y Dr. Sergio Trilles

DITSUHI|ISKANDARYAN Digitally signed by DITSUHI|ISKANDARYAN Date: 2023.02.17 10:29:51 +01'00'

JOSE FRANCISCO|RAMOS|ROMERO Digitally signed by JOSE FRANCISCO|RAMOS|ROMERO Date: 2023.02.17 12:04:43 +01'00'

SERGIO|TRILLES|OLIVER Firmado digitalmente por SERGIO|TRILLES|OLIVER Fecha: 2023.02.17 10:32:02 +01'00'

Castelló de la Plana (Spain)
Febrero 2023

Financial Support

This thesis has been realised with the financial support of the following institutions:

Predoctoral contract:

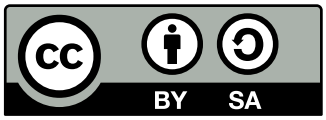
- Ayuda predoctoral para la formación de personal investigador FPI-UJI, dentro del Plan de Promoción de la Investigación de la UJI 2018 (Ref. PREDOC/2018/61). Universitat Jaume I. 1st September 2019 - 9th December 2022.

Research stay:

- University of Bologna, Bologna, Italy (1st September 2021 - 31st December 2021). Financed by Beca para realizar estancias temporales en otros centros de investigación, para el personal docente e investigador de la Universidad del Plan de Promioción de la Investigación de la UJI 2020 (Ref. E-2020-14).

External fundings:

- ValidanT project (GV/2020/035 Sergio Trilles). 1st January 2020 - 31st December 2021. Funded by Generalitat Valenciana.
- Trust4IoE project (PID2019-104065GA-I00) from the Spanish Ministry of Science and Innovation. 1st June 2020 - 31st May 2023. Funded by MCIN/AEI/10.13039/501100011033 and, as appropriate, by MCIN/AEI/10.13039/501100011033 and by "ERDF, a way of making Europe", by the European Union.



Acknowledgments

I would like to acknowledge the predoctoral programme PINV2018 - Universitat Jaume I (PREDOC/2018/61) and the Pla de promoció de la investigació a l'UJI (E-2020-14) for their financial support. I am very thankful for the opportunity to pursue my PhD, as well as to carry out my research stay at the University of Bologna.

This journey would not have been possible without the support and encouragement of my supervisors Dr. Francisco Ramos and Dr. Sergio Trilles. I am extremely grateful for their patience, motivation, constructive feedback and guidance that led me to accomplish this dissertation and become an independent researcher.

I would like to express my deepest gratitude to Dr. Joaquín Huerta for his continued support, encouragement and willingness to resolve any issues that arise.

Special thanks to Estefania Aguilar for her assistance during her tenure at GEOTEC, especially her endless patience and support during the PhD application and post-admission processes.

My sincere appreciation goes out to Prof. Silvana Di Sabatino and her research group, especially Dr. Erika Brattich, Dr. Francesca Di Nicola and Dr. Leonardo Aragão, for their immense knowledge, insightful comments, and encouragement throughout my stay in Bologna.

Also, I would like to thank the staff and all the colleagues in the GEOTEC research group for the amazing working environment and the interesting discussions.

Big thanks to my friends for their encouragement, invaluable advice and support during challenging situations.

Finally, I would like to express my love and gratitude to my parents, Nikolay Iskandaryan and Svetlana Khachaturyan, my brother Davit, and my sisters Lusine and Mariam for their trust, motivation, encouragement and love.

Abstract

Air quality is one of the top concerns for science, government, and society stakeholders. Elevated concentrations of certain pollutants above defined thresholds can cause many diseases, including heart disease, stroke, chronic obstructive pulmonary disease and lung cancer. Information and knowledge about air quality can assist in effectively monitoring and controlling pollutant concentrations, reducing or preventing the harmful impacts and consequences associated with it. Various methodologies and procedures have been incorporated and deployed in the air quality domain to acquire and understand this information. However, the complexity of air quality dependence on various components beyond the temporal dimension as well as the spatial dimension creates additional challenges.

The current dissertation proposes machine learning and deep learning technologies that are capable of capturing and processing multidimensional information and complex dependencies, in particular, spatiotemporal dependencies controlling the formation of air quality. The first key contribution of the current dissertation is a meta-review of air quality prediction using machine learning and deep learning technologies, and the current state-of-the-art of the domain, which served as an introduction and guide to the further directions of our research. The second contribution is the incorporation of air quality, meteorological and traffic data of the study area (the city of Madrid) in spatiotemporal dimensions over the defined area. The third contribution is the exploratory analysis of these datasets to detect existing interconnections and reveal features that have a significant impact on the air quality forecast. The fourth contribution is the implementation of various feature engineering techniques, including feature selection and outlier detection approaches, which, along with exploratory analysis, are acknowledged as potential strategies to aid in improving model performance. Finally, a fifth contribution is the implementation of spatiotemporal air quality forecasting methods that have been

evaluated in the city of Madrid under various defined scenarios.

Overall, the following components come together to generate and formulate the novelty of the current work: spatiotemporal forecast of the defined prediction target (nitrogen dioxide); incorporation and integration of air quality, meteorological and traffic data with their features/variables in spatiotemporal dimensions within a certain spatial extent and temporal interval; the consideration of coronavirus disease 2019 as an external key factor impacting air quality level; and provision of the code and data implemented to incentivise and guarantee reproducibility.

Keywords: *air quality prediction*, *air pollution*, *machine learning*, *deep learning*, *spatiotemporal prediction*, *ConvLSTM*, *BiConvLSTM*, *A3T-GCN*, *feature selection*, *outlier detection*.

Resumen

La calidad del aire es una de las principales preocupaciones de la ciencia, de los gobernantes y la sociedad en general. Las concentraciones elevadas de ciertos contaminantes por encima de los umbrales definidos pueden causar diferentes daños en la salud humana, incluidas las enfermedades cardíacas, daños cerebrovasculares, enfermedad pulmonar obstructiva crónica o cáncer de pulmón. Una mayor información y conocimiento sobre la calidad del aire pueden ser de ayuda para monitorear y controlar de manera efectiva las concentraciones de contaminantes, reduciendo o previniendo los impactos nocivos y las consecuencias asociadas con ellos. Hasta el momento se han incorporado y desplegado varias metodologías y procedimientos en el dominio de la calidad del aire para adquirir y comprender esta información. Sin embargo, la complejidad y la dependencia de la calidad del aire sobre las dimensiones espacial y temporal, hace que su predicción no sea una tarea trivial y genere nuevos desafíos.

La tesis actual propone tecnologías de aprendizaje automático y aprendizaje profundo capaces de capturar y procesar información multidimensional y dependencias complejas, en particular, dependencias espaciotemporales que mejoran la predicción de la calidad del aire. La primera contribución clave de este trabajo es una meta-revisión del estado del arte de la predicción de la calidad del aire utilizando tecnologías de aprendizaje automático y profundo, que sirvió como punto de partida y guía a lo largo de la investigación realizada. La segunda aportación es la incorporación y preparación de los datos de calidad del aire, meteorológicos y de tráfico del área de estudio (la ciudad de Madrid) con las dimensiones espaciotemporales y el área delimitada. La tercera contribución es el análisis exploratorio de estos conjuntos de datos para detectar interconexiones existentes y revelar características que tienen un impacto significativo en el pronóstico de la calidad del aire. La cuarta contribución es la implementación de varias técnicas de ingeniería de

características, incluidos los enfoques de selección de características y detección de valores atípicos, que, junto con el análisis exploratorio, se reconocen como estrategias potenciales para ayudar a mejorar el rendimiento de los modelos de aprendizaje automático. Finalmente, una quinta contribución es la implementación de modelos de predicción espaciotemporal de la calidad del aire siendo estos evaluados sobre la ciudad de Madrid y diferentes escenarios definidos.

En general, las novedades del trabajo actual son: estudio de las componentes espaciotemporal para la predicción de la calidad del aire (dioxido de nitrogeno); integración de datos de calidad del aire, meteorológicos y de tráfico con sus características/variables en una determinada extensión espacial e intervalo temporal; adaptación al efecto externo generado por la pandemia del Covid19 sobre el nivel de calidad del aire; y provisión de los datos y código implementados para incentivar y garantizar la reproducibilidad.

Keywords: *predicción de la calidad del aire*, *la contaminación del aire*, *aprendizaje automático*, *aprendizaje profundo*, *predicción espaciotemporal*, *ConvLSTM*, *BiConvLSTM*, *A3T-GCN*, *selección de características*, *detección de valores atípicos*.

Index

Index of figures

Acronyms

ACM	Association for Computing Machinery
ANN	Artificial Neural Network
ARIMA	Autoregressive Integrated Moving Average
A3T-GCN	Attention Temporal Graph Convolutional Network
AQHI	Air Quality Health Index
AQI	Air Quality Index
BC	Black Carbon
BiConvLSTM	Bidirectional Convolutional Long Short-Term Memory
CAQI	Common Air Quality Index
CI	Confidence Interval
CNN	Convolutional Neural Network
CO	Carbon Monoxide
CO_2	Carbon Dioxide
ConvLSTM	Convolutional Long Short-Term Memory
COPD	Chronic Obstructive Pulmonary Disease
COVID-19	Coronavirus Disease 2019
CSV	Comma Separated Values

DAQI	Daily Air Quality Index
DCRNN	Diffusion Convolutional Recurrent Neural Network
DL	Deep Learning
DGCN	Dual Graph Convolutional Network
DNN	Deep Neural Network
DP-DDGCN	Dual-Path Dynamic Directed Graph Convolutional Network
ELM	Extreme Learning Machine
EU	European Union
GC-DCRNN	Geo-context based Diffusion Convolutional Recurrent Neural Network
GCN	Graph Convolutional Network
GNN	Graph Neural Network
GRU	Gated Recurrent Unit
HNO_3	Nitric Acid
IA	Index of Agreement
IAQL	Individual Air Quality Index
ICT	Information and Communication Technologies
IDW	Inverse Distance Weighting
IEEE	Institute of Electrical and Electronics Engineers
iForest	Isolation Forest
IoT	Internet of Things
KNN	K-Nearest Neighbor
LightGBM	Light Gradient Boosted Machine

LOF	Local Outlier Factor
LR	Linear Regression
LSTM	Long Short-Term Memory
LSTM-FC	Fully connected LSTM
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
MI	Mutual Information
ML	Machine Learning
MLP	Multilayer Perceptron
MLPNN	Multi-Layer Perceptron Neural Networks
MLR	Multiple Linear Regression
MM-SVM	Multi-Task Learning SVM
mRMR	Maximum Relevance — Minimum Redundancy
MSE	Mean Square Error
MST-GCN	Multi-scale Spatiotemporal Graph Convolutional Network
N_2	Molecular Nitrogen
NEVFMA	Network Emissions/Vehicle Flow Management Adjustment
NH_3	Ammonia
NN	Neural Network
NNI	Nearest Neighbour Interpolation
NO	Nitrogen Oxide
NO_x	Nitrogen Oxides
NO_2	Nitrogen Dioxide

NRMSE	Normalised Root Mean Square Error
O	Oxygen
O_2	Molecular Oxygen
O_3	Ground-level Ozone
OH	Hydroxyl Radical
$PM_{0.1}$	Particulate Matters less than 0.1 micrometers in diameter
PM_1	Particulate Matters less than 1 micrometers in diameter
$PM_{2.5}$	Particulate Matters less than 2.5 micrometers in diameter
PM_{10}	Particulate Matters less than 10 micrometers in diameter
PN_{10}	Particles Number less than 10 nanometers
POI	Point of Interest
PRISMA	Preferred Reporting Items for Systematic Reviews and Meta-Analyses
R	Pearson Correlation Coefficient
R^2	Coefficient of Determination
RMSE	Root Mean Square Error
RNN	Recurrent Neural Network
RF	Random Forest
RR	Relative Risk
SDG	Sustainable Development Goal
SHAP	SHapley Additive exPlanations
SO_2	Sulfur Dioxide
SpAttRNN	Spatio-Attention embedded Recurrent Neural Network

SPM	Suspended Particulate Matter
SSH-GNN	Self-Supervised Hierarchical Graph Neural Network
STGCRNN	Spatiotemporal Graph Convolutional Recurrent Neural Network
ST-DGCN	Spatiotemporal Dynamic Graph Convolutional Network
SVM	Support Vector Machine
TGCN	Temporal Graph Convolutional Network
VOC	Volatile Organic Compounds
UFP	Ultrafine Particle
UN	United Nations
US EPA	United States Environmental Protection Agency
UV	Ultraviolet
WHO	World Health Organization
WoS	Web of Science
WSN	Wireless Sensor Network
XGBoost	Extreme Gradient Boosting

Chapter 1

Introduction[1,2]

Air pollution is defined as any substance in the air that can contaminate the environment (e.g., people, animals)[3]. Air pollution's consequences seriously impact the world's population's health and the ecosystem by affecting the single element and components of them. Regarding human health impact, the following effects should be mentioned: asthma, pneumonia, bronchitis, chronic obstructive pulmonary disease (COPD), cardiovascular diseases, and cancer. 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]. Regarding environmental impact, acid rain, haze, eutrophication, also global climate change can be included in this list [5].

Moreover, considering the importance of reducing air pollution, this is also reported by the United Nations (UN) Sustainable Development Goals (SDGs), which consists of 17 goals and 169 subsidiary targets, i.e., air pollution is mentioned in

[1]The part of this chapter previously appeared as a book chapter in the Book of Academic Press. The original citation is as follows: Iskandaryan, Ditsuhi, Francisco Ramos, and Sergio Trilles. "Application of deep learning and machine learning in air quality modeling." In *Current Trends and Advances in Computer-Aided Intelligent Environmental Data Engineering*, pp. 11-23. Academic Press, 2022.

[2]The publications supporting this dissertation with all highlighted contributions and improvements can be found in Appendix A.

[3]Terms of Environment: Glossary, Abbreviations and Acronyms: `https://bit.ly/3SD4dc1`. [Online; accessed 15-February-2023]

two targets: SDG 3 and SDG 11 which focus on good health, sustainable cities and communities, respectively[4]. Furthermore, a study and brief analysis conducted by Longhurst et al. [6], demonstrates that, although there is no individual goal dedicated to air pollution and its management, air pollution has a direct impact on each of the goals. Another study confirming this belief was carried out by Zhao et al. [7], which analysed the impact of air quality, in particular the impact of the actual reduction of industrial sulfur dioxide (SO_2) emissions on the SDGs in China. They determined that in China from 2005 to 2015 the actual reduction in industrial SO_2 emissions contributes 3.5% -12.3% of the actual change in the certain SDGs, including SDG 3, 4, 9, 12, 15, 16, and 17.

The above examples demonstrate the necessity of air quality monitoring, forecasting, and control. Among these three stages, forecasting air quality is the main target of the current work, which will allow decision-makers to control air quality within a range of acceptable thresholds, and as a result, prevent negative consequences caused by poor air quality. To achieve these goals, first of all, it is important to understand what are the main factors and sources causing air pollution. Manisalidis et al. [5] categorised the source of air pollution into the following categories (Figure 1.1): 1) *major sources* (e.g., power stations, refineries, and petrochemicals, the chemical and fertiliser industries); 2) *indoor area sources* (e.g., domestic cleaning activities, dry cleaners, and printing shops); 3) *mobile sources* (e.g., automobiles, cars, and railways); and 4) *natural sources* (e.g., forest fire, volcanic erosion, and dust storms). As an example of natural sources, in particular, dust storms, the world's largest source of dust, the Saharan dust can be mentioned, which has repeatedly led to numerous human casualties and environmental damage (the last significant exposure was recorded in March 2022[5,6]). Wang et al. [8] measured the impact of Saharan dust on air quality and health impacts in Europe over the period 2016–2017, and the results showed that 41,884 deaths per year were attributable to dust exposure in the countries studied, in particular, in Spain, Italy, and Portugal, dust accounts for 44%, 27% and 22% of total particulate matters less than 10 micrometers in diameter (PM_{10})-related

[4]Transforming Our World: The 2030 Agenda for Sustainable Development: `https://bit.ly/3SPIIF3`. [Online; accessed 15-February-2023]

[5]Widespread dust intrusion across Europe: `https://bit.ly/3y1CsBT`. [Online; accessed 15-February-2023]

[6]Severe Weather Europe: `https://bit.ly/3UNW9qh`. [Online; accessed 15-February-2023]

deaths, respectively.

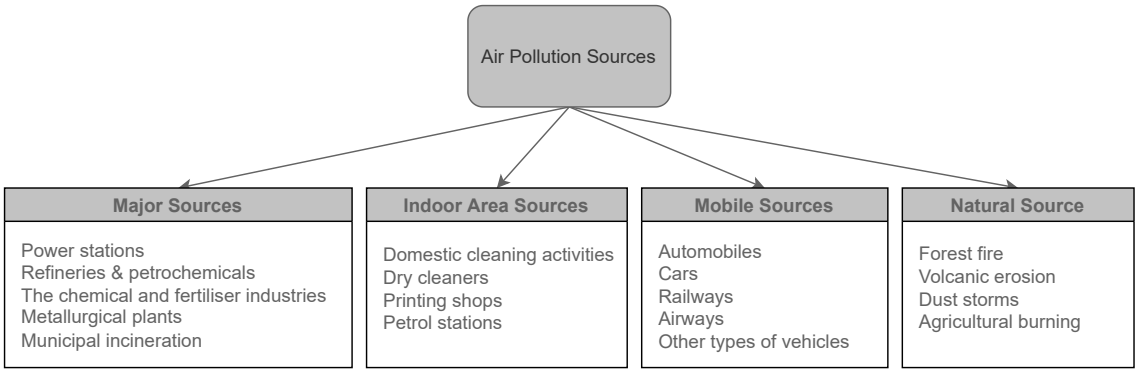


Figure 1.1: The categories of air pollution sources.

Based on sources of emission or generation, pollutants are categorised as primary and secondary pollutants. A primary pollutant is an air pollutant emitted directly from particular sources (sources: sandstorms, volcanic eruptions, industrial and vehicle emissions), such as SO_2, carbon monoxide (CO), nitrogen oxides (NO_x), ammonia (NH_3), and hydrogen chloride (HCl). A secondary pollutant is an air pollutant resulting from the chemical or physical interaction of primary pollutants with other atmospheric substances, such as ground-level ozone (O_3), hydrogen peroxide (H_2O_2) and sulfuric acid (H_2SO_4). It is important to mention about nitrogen dioxide (NO_2), which is the prediction target of this dissertation. According to Vallero [9], NO_2 is a secondary pollutant, however, as a small amount is emitted directly from vehicles, it is also considered as a primary pollutant[7]. The effect of NO_2 is tremendous, both on the environment and on health, causing various respiratory infections, acid rain, etc. (the detailed explanation is stated in Section 3.1).

A smart city can help to address the complexity of urban challenges, including the problems caused by air pollution. The principal goal of a smart city, along with Information and Communication Technologies (ICT), is to make the problem visible, measure it, provide intelligent solutions to mitigate it and raise awareness of air quality among environmental managers and citizens with the advent of innovative tools.

Currently installed sensors and the Internet of Things (IoT) devices [10] make it possible to obtain data on the concentration of various pollutants which, after

[7]Nitrogen dioxide: `https://bit.ly/3F40U8d`. [Online; accessed 15-February-2023]

further processing and interpretation, enable accurate monitoring [11]. Based on the results obtained and following the standards established by the WHO or European Union (EU) Air Quality Directive for concentration thresholds[8], appropriate decisions can be made and further actions can be implemented. Here there are several examples to support the latter argument: the city of Hamburg has taken over the Seaharbor shore connection from Siemens, which allows the ships' generators to be turned off while they are in port and receive electricity from the mainland[9]; the city of Beijing uses the Smog Free Tower, which purifies 30,000 cubic meters of air every hour using 1,170 watts of energy[10]; or another example is the Network Emissions/Vehicle Flow Management Adjustment (NEVFMA) project, which through the use of an air quality monitoring product provides real-time pollution data that has been integrated with the Aimsun traffic management system[11].

Regarding the air quality prediction target, it should be pointed out that it includes both air quality indices and pollutants forecasts. Some of air quality indices focus on a single pollutant, while others take a multi-pollutant approach by utilising various aggregation approaches. The most widely used indices are: United States Environmental Protection Agency (US EPA) Air Quality Index (AQI), Canada Air Quality Health Index (AQHI), Common Air Quality Index (CAQI), Daily Air Quality Index (DAQI), France Air Quality Index (ATMO Index) [12].

Several approaches have been implemented to model and forecast the above air quality indices and pollutants. The most common forecasting models are dispersion, photochemical, statistical and Machine Learning (ML) models [9].

Dispersion Models: predict concentrations at certain downwind receptor locations, i.e., characterise the atmospheric processes that cause the dispersion and movement of emitted pollutants from the source (e.g., industrial plants, vehicular traffic). The box model, Gaussian plume model, and Gaussian puff model are types of dispersion models. To perform the dispersion model many factors have been considered, such as meteorological features, source item (stack height, gas

[8]Outdoor air quality in urban areas: `https://bit.ly/2GQWfLd`. [Online; accessed 15-February-2023]

[9]Shore connection for berthed ships: SIHARBOR: `https://sie.ag/2XOMoOP`. [Online; accessed 15-February-2023]

[10]SMOG FREE TOWER: `https://bit.ly/2wwdcmL`. [Online; accessed 15-February-2023]

[11]Network Emissions/Vehicle Flow Management Adjustment (NEVFMA): `https://bit.ly/3BX7FXQ`. [Online; accessed 15-February-2023]

exit, and velocity), and the topology of source and receptor items.

Photochemical Models: are applicable at many spatial scales. Using mathematical equations describing the chemical and physical processes in the atmosphere, it models changes in pollutant concentrations. The atmosphere is modelled with a three-dimensional grid composed of many grid cells (each cell is typically 4 kilometers by 4 kilometers). It calculates the concentrations in each cell by simulating the movement of air into and out of the cells by advection and dispersion, and by simulating the vertical mixing of contaminants between layers. The following are the types of photochemical models: the Lagrangian trajectory model and the Eulerian grid model.

Statistical Models: used statistical data analysis to determine concentrations. Compared to the other two models, statistical models do not simulate the physical relationship between emissions and environmental concentrations. These methods investigate relationships and correlations between features. Autoregressive Integrated Moving Average (ARIMA), Linear Regression (LR), Multiple Linear Regression (MLR) are examples of statistical methods.

Machine Learning Models: the above models have limitations in capturing non-linear dependencies. They mainly simplify the existing relationship between concentration and affected factors. To overcome the drawbacks, ML models with their subset, named Deep Learning (DL) (stated in Section 3.5), have been implemented to forecast air quality. Studies have indicated and confirmed the significant advantages of ML models over traditional approaches as they can efficiently capture, compute and process complex dependencies across scales from the high-dimensional datasets, including interactions and non-linear relationships and intrinsic features that control and form pollution. For example, Peng et al. [13] demonstrated the superiority of Neural Network (NN) methods by comparing several models, including MLR, Multi-Layer Perceptron Neural Networks (MLPNN) and Extreme Learning Machine (ELM) to forecast O_3, particulate matters less than 2.5 micrometers in diameter ($PM_{2.5}$) and NO_2 in Canada. Another study also confirming the above belief was carried out by Neto et al. [14] where the main objective was to predict $PM_{2.5}$ and PM_{10} in Finland and Brazil. It is worth mentioning, that apart from advantages, ML models also face difficulties and challenges, such as computational costs, overfitting/underfitting the training data, nonrepresentative training data and the lack of interpretability [15].

1.1 Motivation

Given the impact of air pollution on the health and environment, it is crucial to monitor, predict, and control pollutant concentrations. However, due to the complexity of air quality dependence on various factors and phenomena, there are additional difficulties in achieving the above goals. A thorough study of these factors and phenomena should be carried out at the initial stages in order to identify all existing dependencies related to air quality, to find out which factors are more related, what to exclude and what to include in further analysis. These dependencies are connected both in the temporal and spatial dimensions[12] [16, 17]. Dependencies in a temporal dimension refer to temporal relationships of a variable's value at time t and $t-1$. Spatial dependence refers to the spatial relationship of the values of a variable for pairs of locations at a specified distance apart, such that they are more (or less) similar than randomly related pairs of observations. The difference between temporal and spatial dependences is related to directionality, i.e., temporal dependence is unidirectional, while spatial dependence is multidirectional (temporal dependence - past observations can only affect present or future observations but not inversely; spatial dependence - observation in a spatial unit can influence and be influenced by observations in multiple spatial units). Air pollution is one example of a spatiotemporal phenomenon, 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.

Before performing a spatiotemporal analysis, in the first stage, it is necessary to obtain data on the factors that formed and controlled air quality. Data collection can be difficult for several reasons, such as the quality of recorded sources, the availability and density of monitoring stations, and the characteristics of the study area. The next crucial step is to combine together the data obtained from different sources in spatial and temporal dimensions. Another key consideration is to select techniques that will best serve to tackle the problems identified. With all of these considerations in mind, the strategies for performing more accurate analyses are proposed in the scope of this dissertation.

[12] Spatio-Temporal Analysis: `https://bit.ly/3WciEGd`. [Online; accessed 15-February-2023]

1.2 Research Questions and Objectives

This research work has been designed and developed to achieve the defined principal goals and objectives, guided by research questions. The following are the principal research questions addressed by this work:

- ***RQ1***: Which ML approaches have been used in the domain of air quality prediction and how effective are these approaches in reducing air pollution by predicting air quality?
- ***RQ2***: What main components, such as dataset types, prediction targets, and evaluation metrics, have been included in the process of air quality forecasting?
- ***RQ3***: How strong is the correlation between features? Which feature/variables have the highest impact on the performance of ML models?
- ***RQ4***: How well do feature engineering methods improve the accuracy of predictive models?
- ***RQ5***: Does the inclusion of geospatial factors (i.e., the location of air quality and meteorological monitoring stations and traffic measurement points) and spatiotemporal dependencies in predictive models lead to better results?

The research objectives are listed below:

- ***RO1***: Explore and review the most related studies on air quality prediction using ML techniques: Related chapter 2.
- ***RO2***: Detect and observe the ML approaches, the main features employed to predict air quality in the smart city domain: Related chapter 2.
- ***RO3***: Implement detailed exploratory analysis to discover the correlation between dependent and target variables: Related chapter 3.
- ***RO4***: Examine which features/variables significantly affect the performance of predictive models, and select the best combination of the relevant features: Related chapters 3 and 5.

- ***RO5***: Incorporate various data sources, including air quality, meteorological and traffic datasets, as well as the location of air quality and meteorological monitoring stations and traffic measurement points, and process them by implementing different feature engineering techniques: Related chapter 3.
- ***RO6***: Develop and evaluate different ML methods, focusing on the process and computation of spatial and temporal dependencies: Related chapters 3, 4, 5 and 6.

1.3 Research Contributions

The contributions received as a result of the implementation of the proposed approaches can be summarised and defined as follows:

- Meta-review of air quality prediction using ML technologies, current state-of-the-art of the domain, which served as an introduction and guide to the further directions of our research.
- Inclusion and combination of air quality, meteorological, and traffic data in spatiotemporal dimensions with the purpose to perform air quality prediction.
- Exploratory data analysis of datasets to identify relationships between features/variables and highlight those that have a strong impact on air quality prediction. Exploratory data analysis was performed both from a physical point of view and using ML technology.
- Feature engineering approaches such as feature selection and outlier detection techniques, that have significantly improved the performance of the models.
- The implementation of spatiotemporal air quality forecasting methods, including Convolutional Long Short-Term Memory (ConvLSTM), Bidirectional Convolutional Long Short-Term Memory (BiConvLSTM) and Attention Temporal Graph Convolutional Network (A3T-GCN) (which belong to the DL subset and are detailed in Section 3.5.3), that have been evaluated in a real city with real data under various defined scenarios. It should be mentioned that the BiConvLSTM and A3T-GCN implementation for air quality prediction, and the ConvLSTM implementation for NO_2 prediction, are the first

time proposed by the current research, making this research a somewhat groundbreaking contribution to the domain of air quality prediction. The city of Madrid is used as a scenario to perform predictive analysis defined in the framework of this dissertation.

1.4 Thesis Structure

The dissertation begins by presenting the background and importance of the topic, the motivations, the research questions and the objectives, based on which the work was constructed and developed. Afterwards, the main contributions were highlighted. The rest of the work is structured as follows:

- [Chapter 2] It discusses the most recent developments and current state of the art in the field of air quality prediction using ML techniques. The key features of selected papers are extracted, and comparisons and analyses are provided.

- [Chapter 3] The employed datasets, their transformation and generation into the format required for the analysis are defined in detail. Additionally, the exploratory analysis of the datasets is provided to reveal all existing relationships and linkages between distinct data types. Furthermore, the feature engineering techniques with their workflow are presented, which serve as a data preprocessing step before implementing the predictive analysis. The chapter is finalised with a detailed description of the proposed methods along with the key components of their architectures.

- [Chapter 4] One of the most advanced methods, ConvLSTM is introduced. Particularly, a comparison of NO_2 prediction for pandemic and non-pandemic periods with different temporal granularities in the city of Madrid is provided using the ConvLSTM on historical NO_2 and meteorological data.

- [Chapter 5] An extended version of the ConvLSTM, called BiConvLSTM, is proposed and developed and further compared with the reference models. The reported analysis comprised traffic data in addition to NO_2 and meteorological data. Additionally, feature selection approaches are discussed with an emphasis on and comparison of their peculiarities. A further contribution

is the implementation of approaches to the transformation of cyclic data and the selection of a superior approach.

- [Chapter 6] The new advanced technique based on Graph Neural Networks (GNNs), called A3T-GCN, is introduced. A comparison of the proposed method with reference methods (Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU)) is given to predict NO_2. Furthermore, the importance of outlier detection techniques is highlighted.
- [Chapter 7] The dissertation wraps up with the key concluding notes and proposes further research directions.

Chapter 2

State of the Art[1]

Predicting air quality is one of the most pressing global challenges. To understand how to contribute to the domain, it is essential to thoroughly examine the existing studies devoted to air quality prediction using ML and DL algorithms. Particularly, the following contributions can be highlighted in this chapter:

- We selected the relevant studies on air pollution prediction in smart cities using ML methods;
- We compared and analysed the approaches and features implemented in the domain to provide a comprehensive overview;
- We introduced the studies related to the employment of one subclass of DL techniques, namely a GNN for air quality prediction, and analysed the main components in terms of implemented GNN architecture.

[1]The part of this chapter previously appeared as articles in the Journals Applied Sciences, Atmosphere and IEEE Access, and an article in the Conference of EnviroInfo. The original citations are as follows: Iskandaryan, Ditsuhi, Francisco Ramos, and Sergio Trilles. ”Air quality prediction in smart cities using machine learning technologies based on sensor data: a review.” *Applied Sciences* 10, no. 7 (2020): 2401; Iskandaryan, Ditsuhi, Francisco Ramos, and Sergio Trilles. ”Features exploration from datasets vision in air quality prediction domain.” *Atmosphere* 12, no. 3 (2021): 312; Iskandaryan, Ditsuhi, Francisco Ramos, and Sergio Trilles. ”Graph Neural Network for Air Quality Prediction: A Case Study in Madrid.” *IEEE Access* 11 (2023): 2729-2742; and Iskandaryan, Ditsuhi, Francisco Ramos, and Sergio Trilles. ”Spatiotemporal Prediction of Nitrogen Dioxide Based on Graph Neural Networks.” *Environmental Informatics*, pp. 111-128. Springer, Cham, 2023.

This chapter is composed of two sections. The first section presents the procedure for selecting studies related to air quality prediction using ML methods and analysing the extracted features and components. The second section introduces the studies on the prediction of air pollution concentrations carried out using GNN. A detailed description of each section is given below.

2.1 Machine Learning Models for Air Quality Prediction

This section focuses on providing a broad overview, screening and analysing relevant works on air quality prediction using ML methods, comparing applicable approaches to find existing trends and research advancements, as well as analysing the relevant studies from the perspective of the datasets used. Furthermore, those datasets and external factors (e.g. precipitation, wind direction, traffic intensity or population density) that affect air quality should be estimated and integrated as input to models to improve air quality forecasting. To solve the aforementioned tasks the following questions were defined:

1. Which ML approaches are used to predict air quality in the smart city domain?
2. Which features are the most used to define ML models?
3. How do the suggested methods handle diverse types of data?
4. What types of datasets are used to improve air quality predictions?
5. What dataset characteristics are important for efficient and effective air quality forecasting?

The aforementioned questions were addressed by defining a search strategy and implementing inclusion/exclusion criteria. First, to select relevant studies, the Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) statement [18] was used as a guideline. Figure 2.1 shows PRISMA flow diagram with four phases, including *Identification, Screening, Eligibility and Included*. The papers were queried in Association for Computing Machinery (ACM), Institute of Electrical and Electronics Engineers (IEEE) Xplore, Web of Science (WoS) databases using the following query: (“machine learning”) AND (“prediction” OR

"forecast") AND ("air quality" OR "air pollution"), which was being applied to title, abstract and keywords. In the first step, all papers published until September 28, 2020 (search date) were chosen, yielding a total of 1,214 papers. Afterwards, duplicated and non-empirical manuscripts were eliminated. Following that, title, abstract, keyword screening, and full-text assessment were carried out based on the inclusion/exclusion criteria stated in Table 2.1.

Table 2.1: Inclusion and exclusion criteria.

Inclusion Criteria	**Exclusion Criteria**
Papers written in English	Non-English written papers
Publications in scientific journals	Non-reviewed papers, editorials, presentations
Publications focused on outdoor air pollution	Publications focused on indoor air pollution
Additional dataset together with air quality data	Using only air quality data
Analysis with implementation of ML techniques	Analysis without implementation of ML techniques
Models applied for forecasting purpose	Works without forecasting models

Following the analysis of the manuscripts, the exploration and observation of the obtained results are introduced by extracting the following components from the selected studies: *Year*, *Study Area*, *Prediction Target*, *Dataset Type*, *Data Rate*, *Period (Days)*, *Open Data*, *Algorithm*, *Time Granularity* and *Evaluation Metric*.

Each component of Table B.1 in Appendix B was observed in terms of dataset types to find out which dataset features were used in each research work, and the findings are displayed below.

Dataset Type: includes types of data which were used to perform analysis. After reviewing the selected studies the followings dataset types were extracted (Figure 2.2): '*MET*': meteorological data, '*Spatial*': topographical characteristics, the locations of the stations, '*Temporal*': includes the day of the month, day of the week, the hour of the day, '*AOD*': aerosol optical depth, '*Social Media*': microblog data, '*Traffic*', '*PBL Height*': planetary boundary layer height, '*Land Use*', '*BEV*':

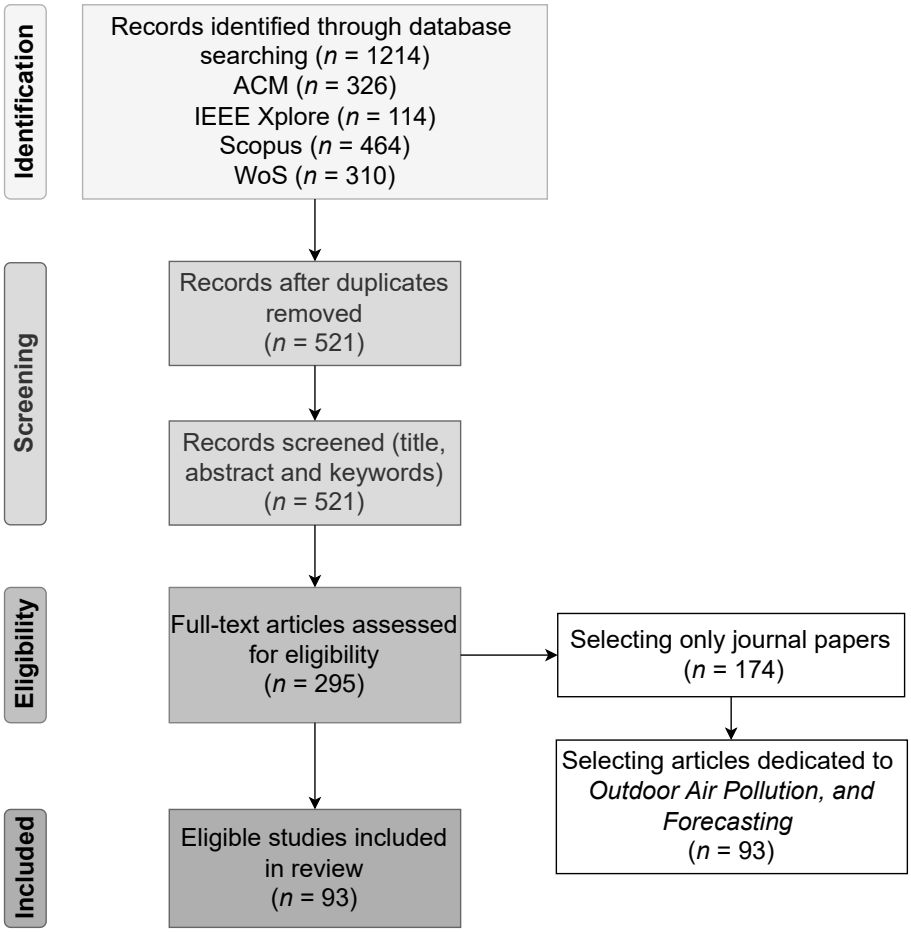


Figure 2.1: Preferred Reporting Items for Systematic Reviews and Meta-Analyses flow diagram for the review (*n* is the number of papers).

built environment variables, '*UV Index*': ultraviolet index, '*SP*': sound pressure, '*PD*': population density, '*Human Movements*': floating population and estimated traffic volume, '*Altitude*', '*OMI-SO_2*': satellite-retrieved SO_2 from Ozone Monitoring Instrument-SO_2, '*PPS*': pollution point source, '*TS*': transportation source, '*WFD*': weather forecast data, '*POI Distribution*': point of interest distribution, '*FAPE*': factory air pollution emission, '*RND*': road network distribution, '*Elevation*', '*AEI*': anthropogenic emission inventory, '*NDVI*': normalised difference vegetation index, '*Chemical*': chemical component forecast data (organic carbon, black carbon, sea salt, etc.), and '*Emission*'.

Out of the twenty-six dataset types, meteorological data is the most widely used, appearing in eighty-eight publications (Figure 2.2). '*Temporal*', '*Spatial*', '*Traffic*',

'*AOD*' and '*Land Use*' datasets are the next relatively more common dataset types.

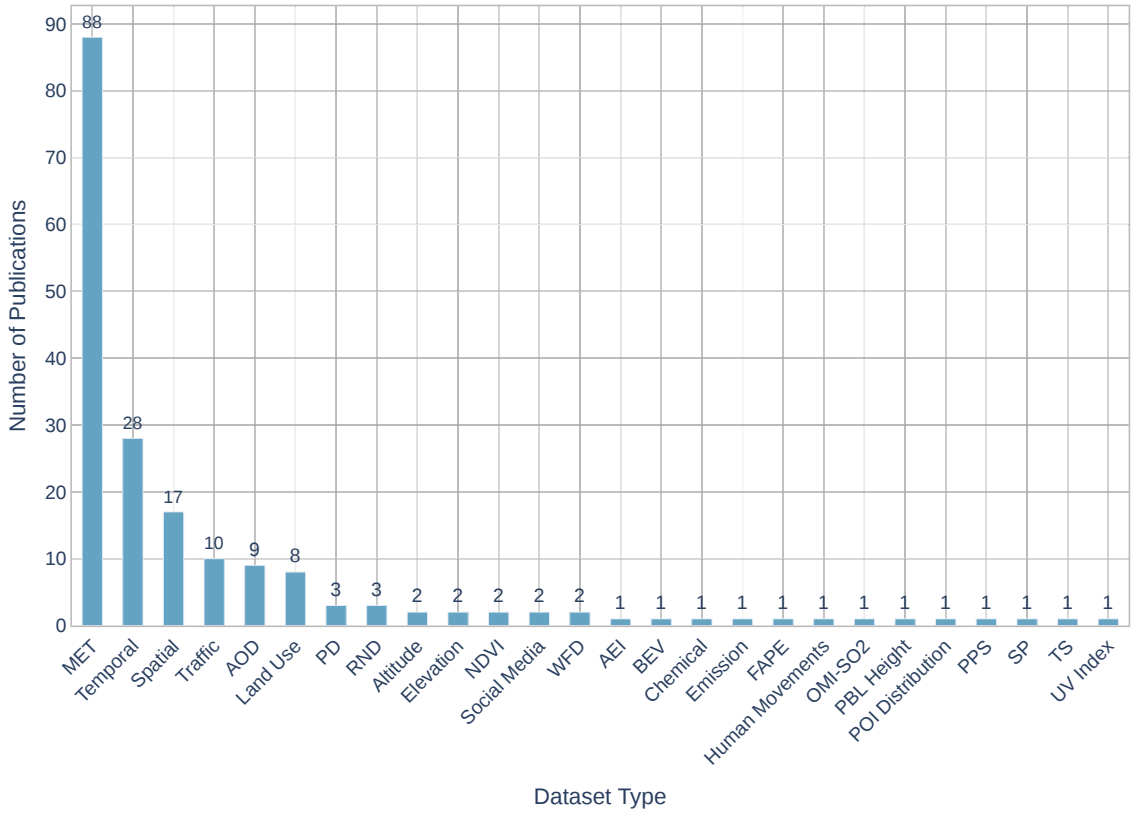


Figure 2.2: The number of publications per each dataset type.

Figure 2.2 shows the number of publications for each dataset type; however, it is also essential to see the number of publications for dataset combinations. Thirty combinations were formed from the aforementioned dataset types used in the publications. The number of publications for each dataset combination is shown in Table 2.2. The most frequently seen combination is meteorological data combined with air quality data, which appears in forty-five papers. There are twenty-three dataset combinations, each of which only appears in one publication; hence they have been grouped as *Others* for ease of analysis.

Year: includes years of publications. Figure 2.3 demonstrates the distribution of the used dataset combinations over time, together with the number of publications for each published year, and it may be used to track progress over time.

Since 2016, there has been an increase in the use of intense dataset combinations, especially in 2019 and 2020, which can be attributed to the rise of

Table 2.2: The number of publications of dataset combinations.

Dataset Combinations	Number of Publications
MET	45
MET, Temporal	11
MET, Spatial, Temporal	5
Spatial	2
MET, AOD	2
MET, Traffic	2
MET, Social Media	2
Others	23

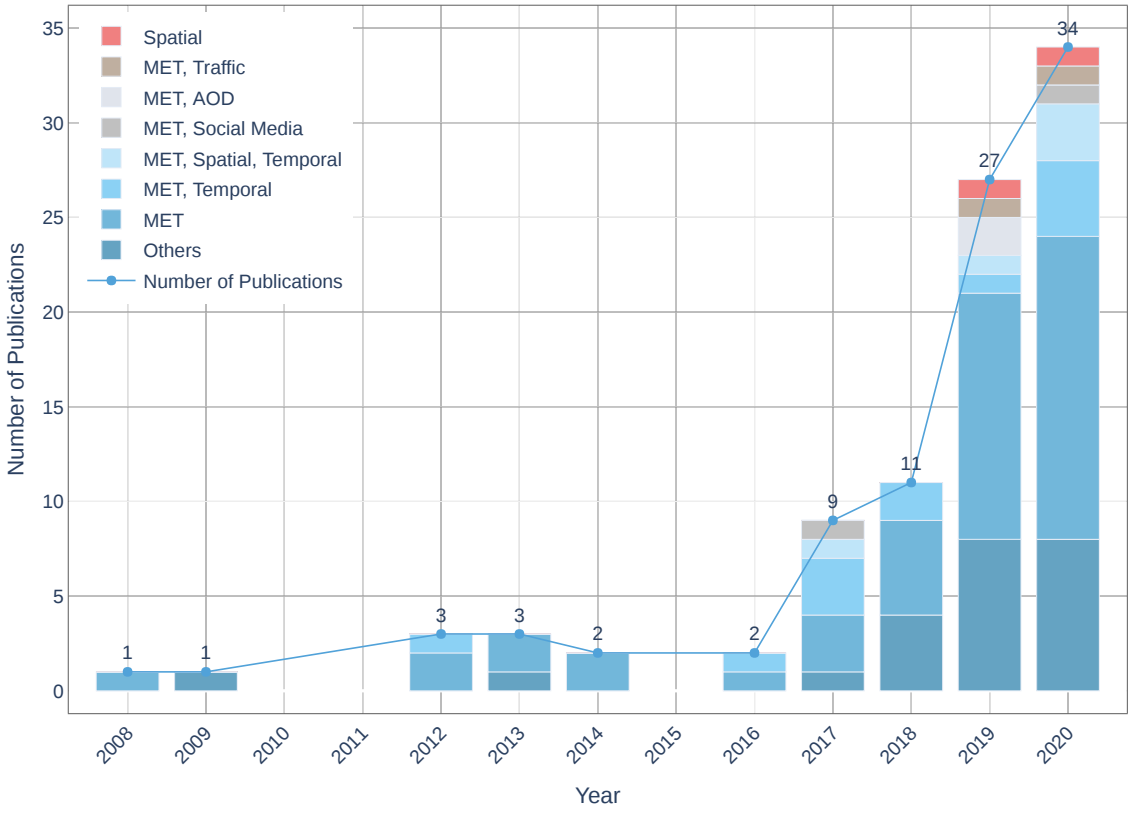


Figure 2.3: The distribution of the dataset combinations throughout the years.

smart cities and open data portals as scientific concepts. However, throughout the entire period, only meteorological data was dominant. The increase in the number of manuscripts can be attributed to the open data movement promoted by

the governments [19].

Study Area: are the countries used as a study area in the papers. In the majority of the papers (forty), China was a study area. Here is a list of the remaining countries, along with their number of publications: USA-six; Taiwan-six; India-four; Iran-four; South Korea-four; UK-three; Canada-two; Ecuador-two; Egypt-two; Europe-two; France-two; Italy-two; Kuwait-two; Saudi Arabia-two; Turkey-two; Germany-one; Jordan-one; Mongolia-one; Poland-one; Qatar-one; Slovenia-one; Spain-one; Thailand-one; and Tunisia-one. In addition to this examination, knowing the dataset combinations for each study area could be useful. The distribution of dataset combinations in terms of the study area is depicted in Figure 2.4. China was a study area in the papers with the majority of dataset combinations (China with '*MET*' is the dominating combination (twenty-one papers)).

Prediction Target: is a pollutant or air quality index that a particular study is intended to predict. In general, seventeen prediction targets were used: $PM_{2.5}$, O_3, NO_x, PM_{10}, AQI, SO_2, CO, ultrafine particle (UFP) or particulate matters less than 0.1 micrometers in diameter ($PM_{0.1}$), AQHI, Individual Air Quality Index (IAQL), NH_3, particle number concentrations (PNCs [particle number concentration is the total number of particles per unit volume of air[2]]), particles number less than 10 nanometers (PN_{10}), black carbon (BC), suspended particulate matter (SPM) and carbon dioxide (CO_2).

Figure 2.5 presents the distribution of dataset combinations in terms of prediction target, and it can be observed, that the prediction target can be an individual pollutant, as well as an air quality index. However, the prevailing targets are individual pollutants, particularly, $PM_{2.5}$, O_3, NO_x, and PM_{10}, which may be explained by the dangers of those pollutants and the need to detect and control them. Moreover, according to the US EPA, air quality in a certain area is defined by the above-mentioned pollutants [9]. The most commonly used prediction target, $PM_{2.5}$ (forty-eight papers), was applied in publications with all combinations, especially with '*MET*', which was the most commonly used combination by researchers (twenty-one papers). It is noteworthy, that technological advancements have made it possible to observe finer particles ($PM_{0.1}$, PN_{10} [20, 21]), which are more hazardous and easier to inhale.

[2]Particle Numbers and Concentrations Network: `https://bit.ly/39HqALZ`. [Online; accessed 15-February-2023]

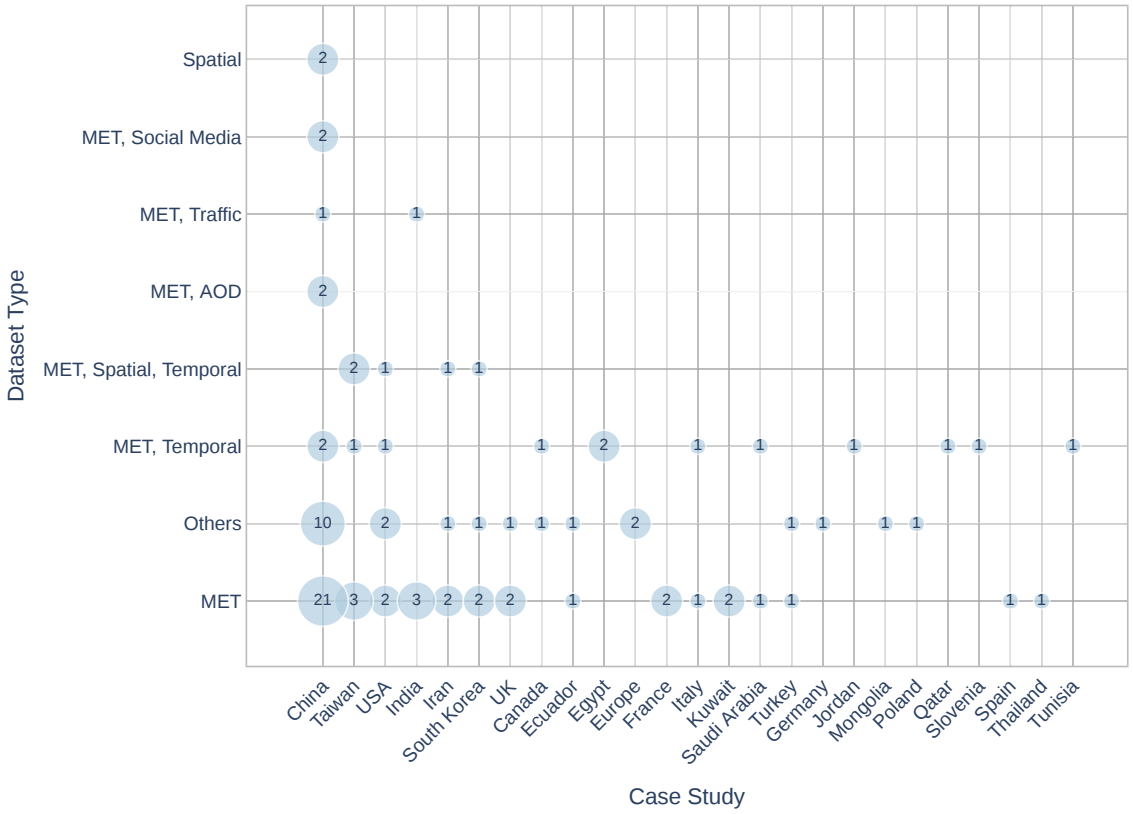


Figure 2.4: The number of publications of dataset combinations in terms of the study area.

It can be also interesting to look at the distribution of the study areas in terms of prediction targets. To illustrate the aforementioned observation, the prediction targets were categorised into the following groups considering the number of publications for each combination (country-prediction target combination), including $PM_{2.5}$, O_3, NO_x, PM_{10}, AQI, SO_2, CO, and Others (UFP, particulate matters less than 1 micrometers in diameter (PM_1), AQHI, IAQL, NH_3, PNCs, PN_{10}, BC, SPM and CO_2). The dominant study area is China for all prediction targets (Figure 2.6), particularly, the dominant combination is China with $PM_{2.5}$ (twenty-six publications).

Data Rate: is the timespan during which the sensors deliver data. Figure 2.7 shows the distribution of dataset combinations in terms of data rate. Overall, biweekly (one paper), daily (twenty papers), hourly (fifty-six papers), minutely (three papers), secondly (one paper), 15min (one paper), 5min (one paper), and 5s (one paper) data rates were used in the studies, and nine studies did not provide

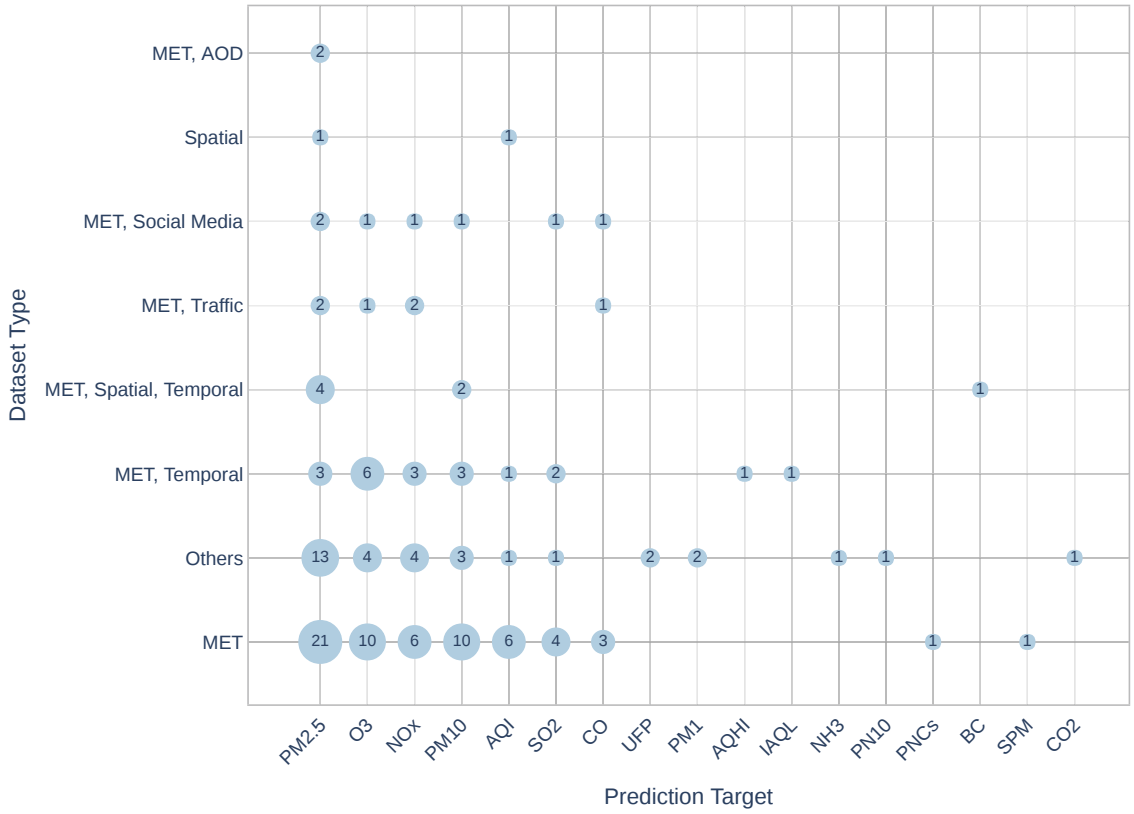


Figure 2.5: The number of publications of dataset combinations in terms of prediction target.

information about data rate. It can be shown that the most frequently used hourly data rate is used in publications with all combinations, especially with '*MET*', which was the most commonly used combination by researchers (thirty-two papers).

Period (Days): is the duration of the data collection (the number of days). The summary statistics of these days reveals a *mean* of days of 1300.63 days (*Std.Dev*: 1484.68) and a *median* of 731 days (*Min*: 3 and *Max*: 8023). In nine publications, the most widely used timeframe is 365 days. Combining this feature with the data rate makes it possible to estimate the volume of data used in the study (obviously, it cannot ensure data quality, since the data may contain noisy samples).

Open Data: provides information about data accessibility. Considering the role of reproducibility nowadays, the availability of the dataset used in the papers was also examined. However, reproducibility does not apply only to data; it also

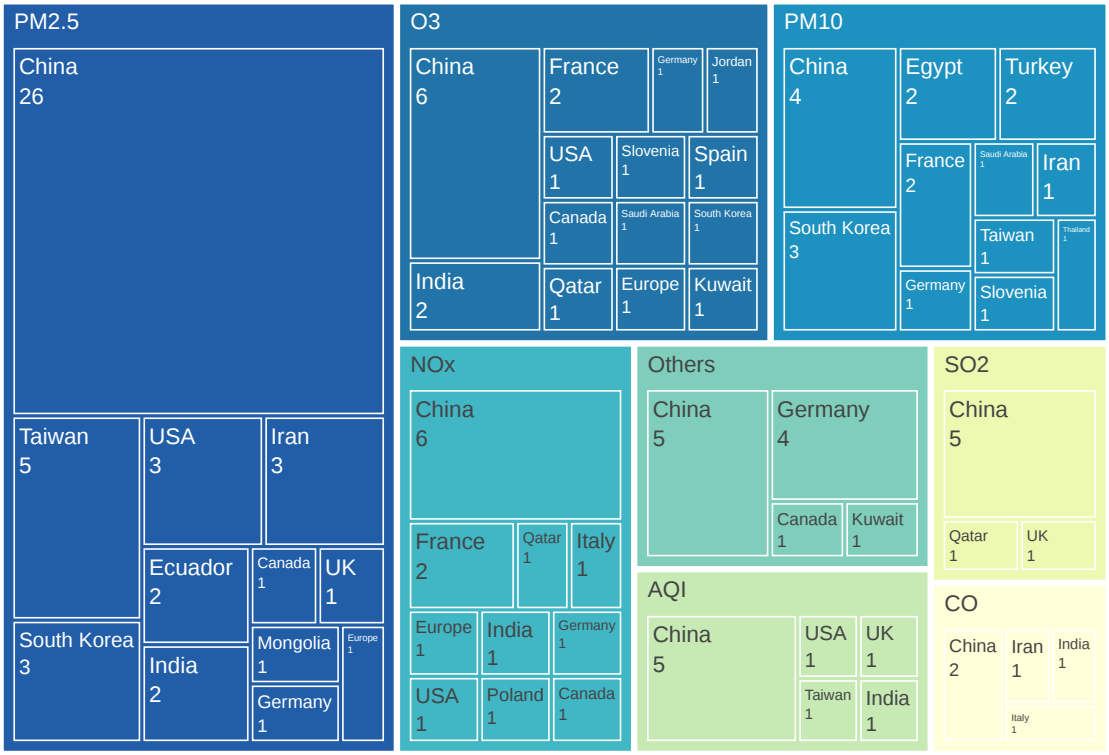


Figure 2.6: The distribution of study areas in terms of prediction target.

refers to code availability [22]. No paper provided code scripts, although the algorithms were available and were explained in the papers. Figure 2.8 illustrates the distribution of dataset combinations in terms of data availability. There are three categories: *Yes* (fifty-nine papers), *No* (thirty papers), and *Partially* (four papers). The first two indicate whether or not the authors provided the data used in the studies, while the papers with *Partially* relate to studies where the authors only contributed a part of the data. Regarding data accessibility across time, the authors began using open data in their research in 2012 (Figure 2.9), which coincides with the emergence of open data portals [23, 24] and smart cities movement [25]. Figure 2.10 displays the data availability per study area. It can be observed that publications with study area China include all three categories.

It would also be interesting to observe the relationship between the affiliation of the authors and the study area of particular research. The results demonstrate that in the majority of the papers (fifty-five), the affiliations of all the co-authors are located in the corresponding study areas. The affiliations of the authors in eleven

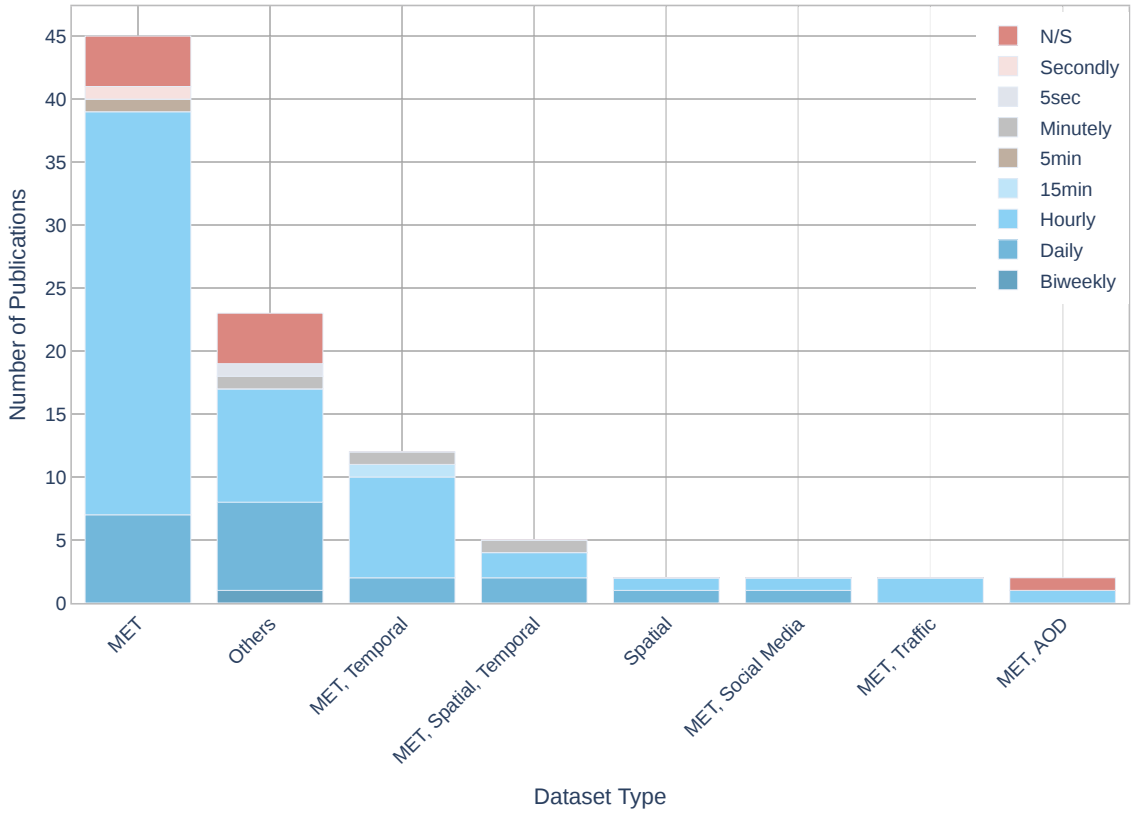


Figure 2.7: The number of publications of dataset combinations in terms of data rate.

papers are in countries other than the study areas. For example, the author's affiliations in the following work [26] are in China, and the study area is in the USA. In twenty-seven papers, the co-authors' affiliations partially correspond to the study area. For instance, in this paper [13] the study area is Canada and the author's affiliations belong to China and Canada.

Algorithm: is the ML algorithm that the applied methods are based on. Figure 2.11 shows the distribution of dataset combinations in terms of ML algorithms. The ML algorithms used in the studies are *NN* (forty-five papers), *Ensemble* (thirty-three papers), *Regression* (twenty-one papers), *Hybrid* (twelve papers) and *Other Algorithms* (four papers). It can be seen, that the NN outnumbers other algorithms. The following are the most common approaches used in each category: *NN*—LSTM, Multilayer Perceptron (MLP), GRU; *Regression*—Support Vector Machine (SVM); *Ensemble*—Random Forest (RF), Extreme Gradient Boosting (XGBoost), Light Gradient Boosted Machine (LightGBM)); *Hybrid* —the major-

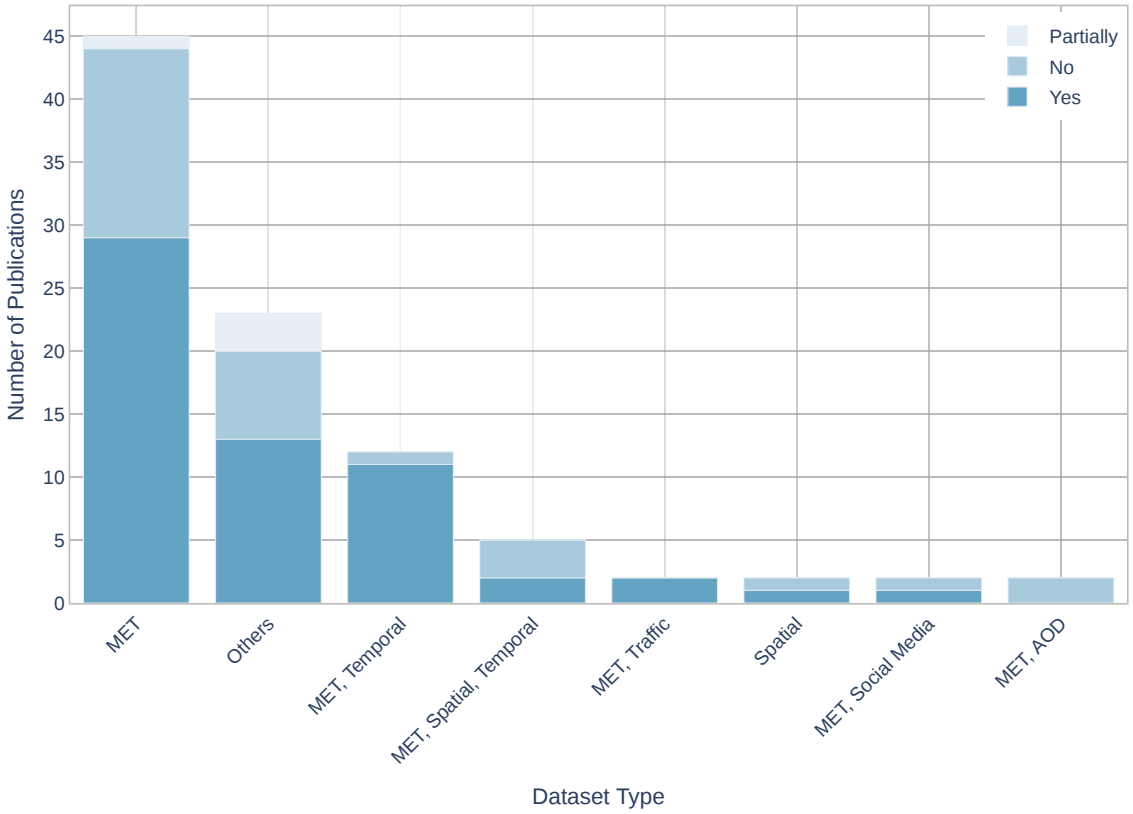


Figure 2.8: The number of publications of dataset combinations in terms of data availability.

ity of the methods of this category are based on SVM, for example, Partial Least Square-SVM, Multi-output SVM and Multi-Task Learning SVM (MM-SVM); *Other Algorithms*—includes the works applied Decision Tree Algorithm (C4.8), Reinforcement Learning, Bayesian Model, Regularisation and Optimisation. Regarding dataset combinations, in contrast to other combinations, '*MET*' and '*Others*' include all categories of the algorithms.

Combining prediction targets and applied methods can help to reveal any correlation between them in order to determine which methods are used to predict a particular target. According to the results of the study, the following connection was detected (main prediction targets and corresponding methods): *Particulate Matters* - LSTM, SVM, RF; O_3 - MLP, Recurrent Neural Network (RNN); NO_x -SVM, RF, RNN; SO_2 - SVM; *CO* - LSTM; *AQI* - SVM.

It would be interesting to know how the use of the algorithms varied over

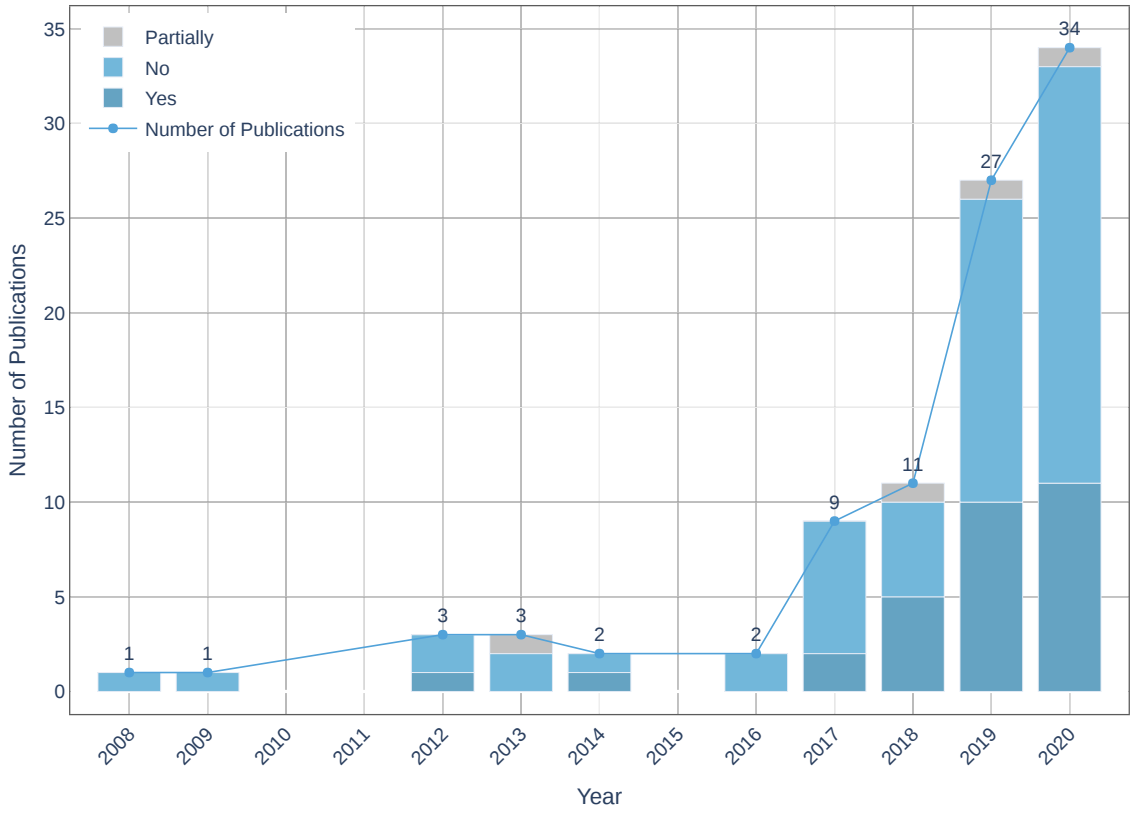


Figure 2.9: Data availability over the years.

time. The publications for each ML algorithm over the years are shown in Figure 2.12. The number of papers has been increasing in recent years, particularly, *NN*, *Ensemble* and *Hybrid* models. Regarding *Regression* methods, the latter approach has been used very consistently since 2008, and in subsequent publications, the *Regression* approach has been used primarily in conjunction with other methods for comparison purposes.

Time Granularity: refers to the time interval, over which the prediction was applied. Figure 2.13 shows the distribution of dataset combinations in terms of time resolution. The used time resolutions are 1 h, 2 h, 3 h, 4 h, 6 h, 8 h, 10 h, 12 h, 24 h, 48 h, 72 h, five days, one week, 15 days and one month (these retrieved intervals are the maximum intervals applied in each article). It is detectable that 24 h is the most used time resolution regarding the number of publications and different dataset combinations. Furthermore, the most extended prediction time resolution, one month, is applied in publication with '*Others*' combination, and considering that the longer resolution reduces accuracy, only one paper uses the

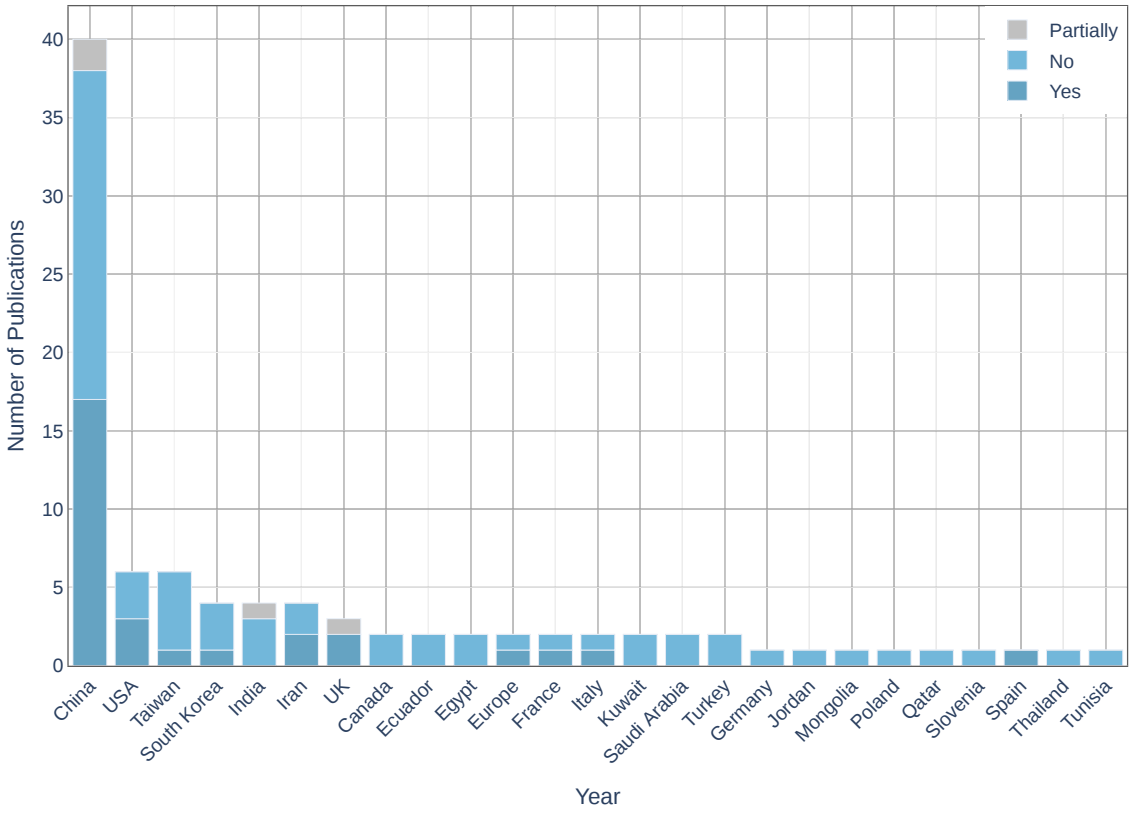


Figure 2.10: Data availability per study area.

one-month prediction.

Evaluation Metric: is the measure that is used to assess the effectiveness of the applied method. In total, sixty-nine metrics were used to evaluate the methods, from which the most used metrics are Root Mean Square Error (RMSE) in seventy-seven papers, Mean Absolute Error (MAE) in forty-two papers, Coefficient of Determination (R^2) in thirty-six papers, and Pearson Correlation Coefficient (R) in twenty-one papers. Figure 2.14 demonstrates the distribution of dataset combinations in terms of evaluation metric (each database combination is marked with a different colour). Compared to other dataset types, '*MET*', '*MET, Temporal*' and '*Others*' were combined with more metrics, particularly, RMSE with '*MET*' (forty-one papers) and MAE with '*MET*' (twenty-four papers) are the most used combinations. Additionally, taking into consideration the most used prediction target ($PM_{2.5}$) and the most used time resolution (24 h), the results show that $PM_{2.5}$ was a prediction target in eighteen papers with the combination of RMSE and '*MET*', and in ten papers with the combination of MAE with '*MET*', and 24 h

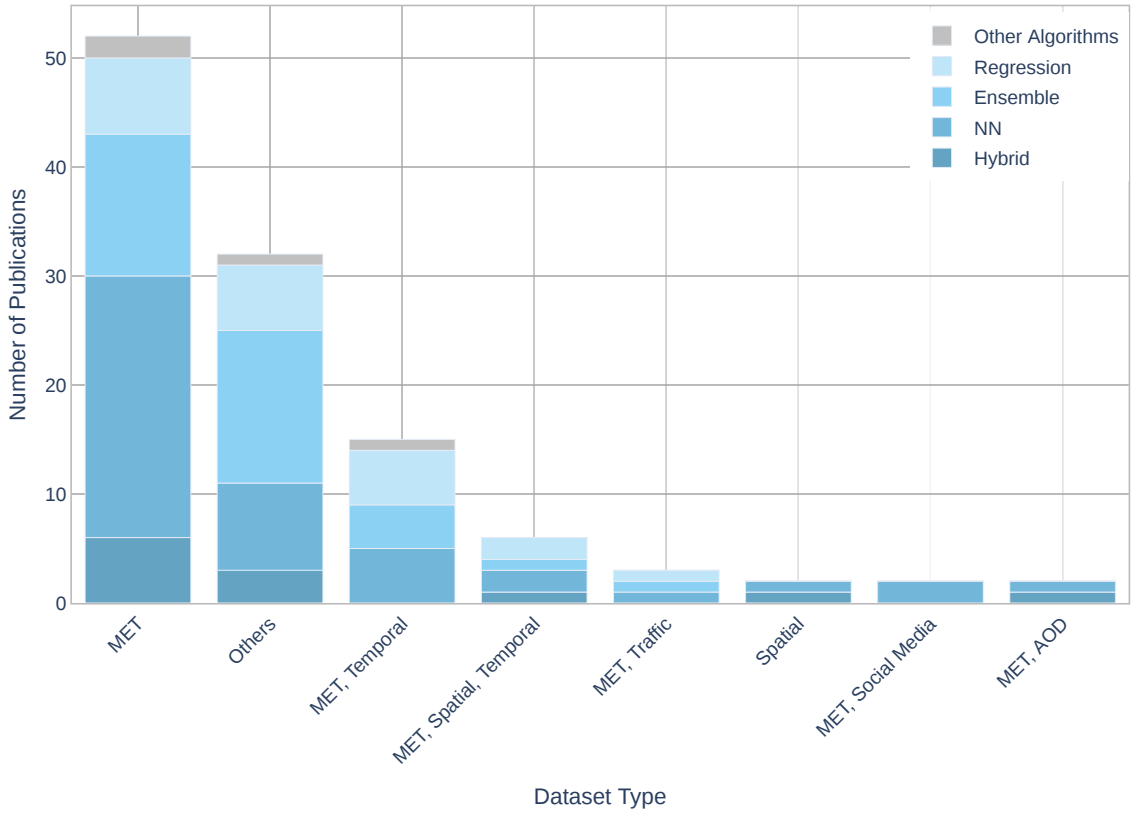


Figure 2.11: The number of publications of dataset combinations in terms of Machine Learning algorithms.

was a predicted time resolution in ten papers with RMSE and '*MET*' combination and in six papers with MAE and '*MET*' combination. Furthermore, the metrics that have been used in more than six publications with corresponding equations and descriptions are extracted and displayed in Table 2.3. The metrics are RMSE, MAE, R^2, R, Mean Absolute Percentage Error (MAPE), Index of Agreement (IA), Mean Square Error (MSE), Normalised Root Mean Square Error (NRMSE) [27–34].

Many aspects influence model performance accuracy, including ML techniques, spatial characteristics, prediction targets, and temporal resolution. Several authors have mentioned the structural limitations of algorithms, including the tendency to overfit, complexity, difficulties with interpretation, and time-consuming [35–37]. Regarding the prediction target, depending on which pollutant is the prediction target the accuracy may vary since the chemical structure of the pollutants is different. For example, Li et al. [38] found out that the proposed model predicts better $PM_{2.5}$ than NO_x, as NO_x is highly reactive and has larger temporal variability.

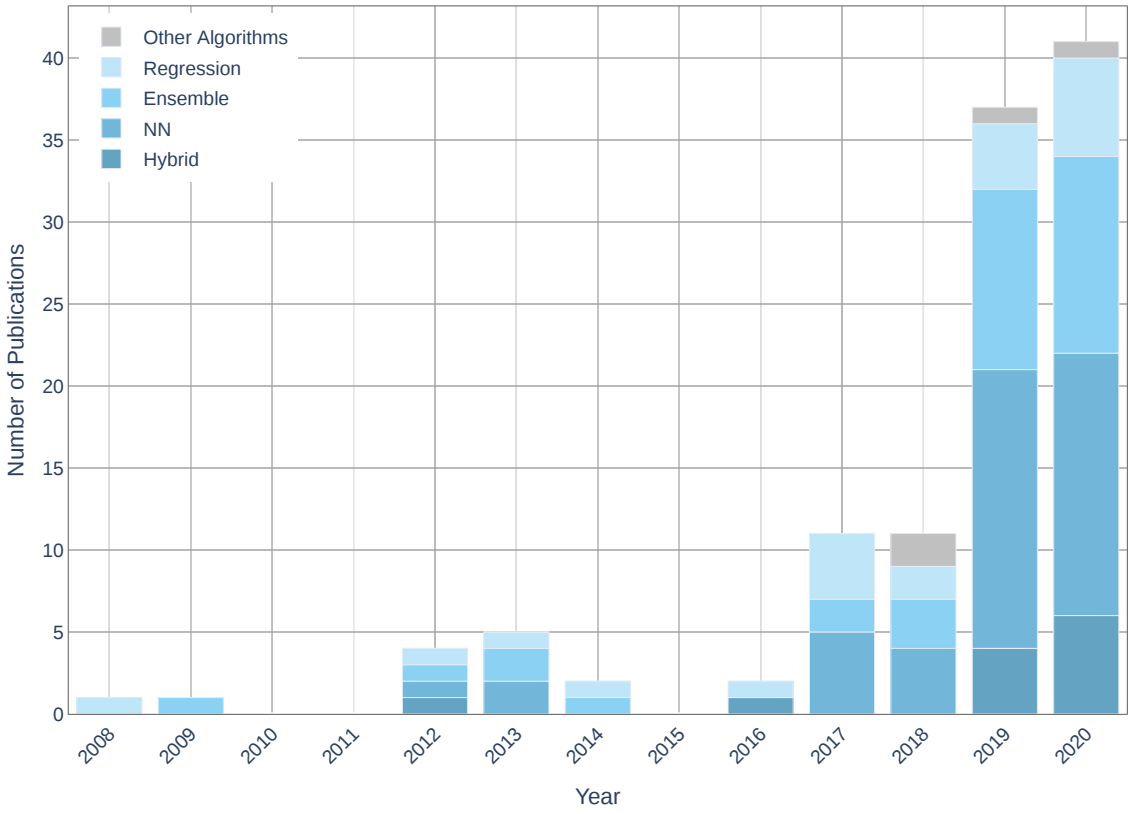


Figure 2.12: The number of publications per Machine Learning algorithms throughout the years.

Therefore, many studies mentioned the implementation of the proposed model for predicting other pollutants as future work [26, 39]. Another constraint is the lack of data in spatiotemporal resolution [40, 41]. Missing values can also be included in this scope, and depending on their quantity, the performance can be drastically reduced [42, 43]. An important factor is the presence of sudden changes. One solution could be to collect more data, as the training dataset will include more sudden changes, resulting in higher performance in the event of abrupt changes [41]. The inclusion of additional datasets closely related to air quality, such as aerosol optical depth and meteorological data, can help address this issue [44]. It might also be useful to apply techniques for handling imbalanced datasets [39]. Another limitation that we have already highlighted is a prediction with a long temporal resolution since due to the accumulated error, the accuracy decreases as the temporal resolution increases [45, 46].

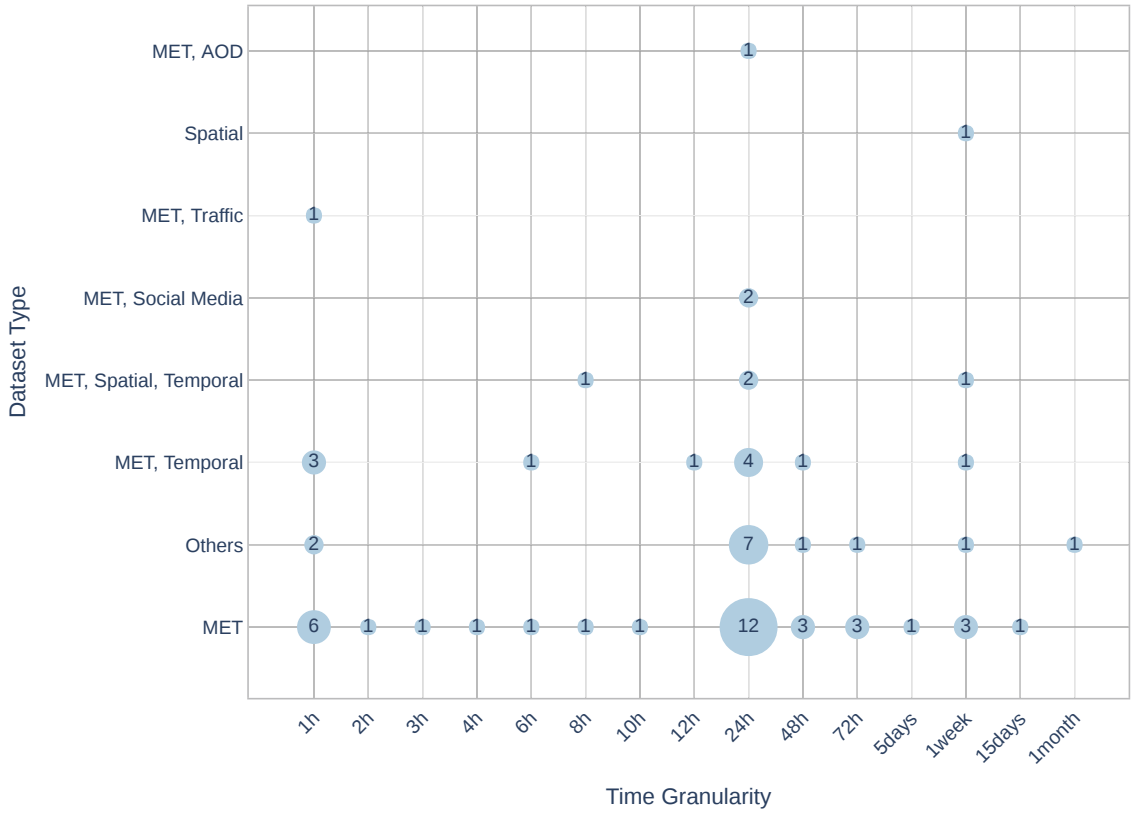


Figure 2.13: The number of publications of dataset combinations in terms of time granularity.

2.2 Graph Neural Network for Air Quality Prediction

By observing the most recent developments in air quality forecasting, a lot of attention has been found on GNN models. Since the works devoted to GNN were not included in the initial stage of the review, this section briefly describes the works that implement GNN to forecast air quality.

Han et al. [47] proposed the Self-Supervised Hierarchical Graph Neural Network (SSH-GNN) based on cities→functional zones→regions 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. [48] proposed Dual Graph Convolutional Network (DGCN) and LSTM network combined with Wireless Sensor Network (WSN) and IoT to perform AQI predictions; especially, DGCN was responsible to process the data from the sensors that were later

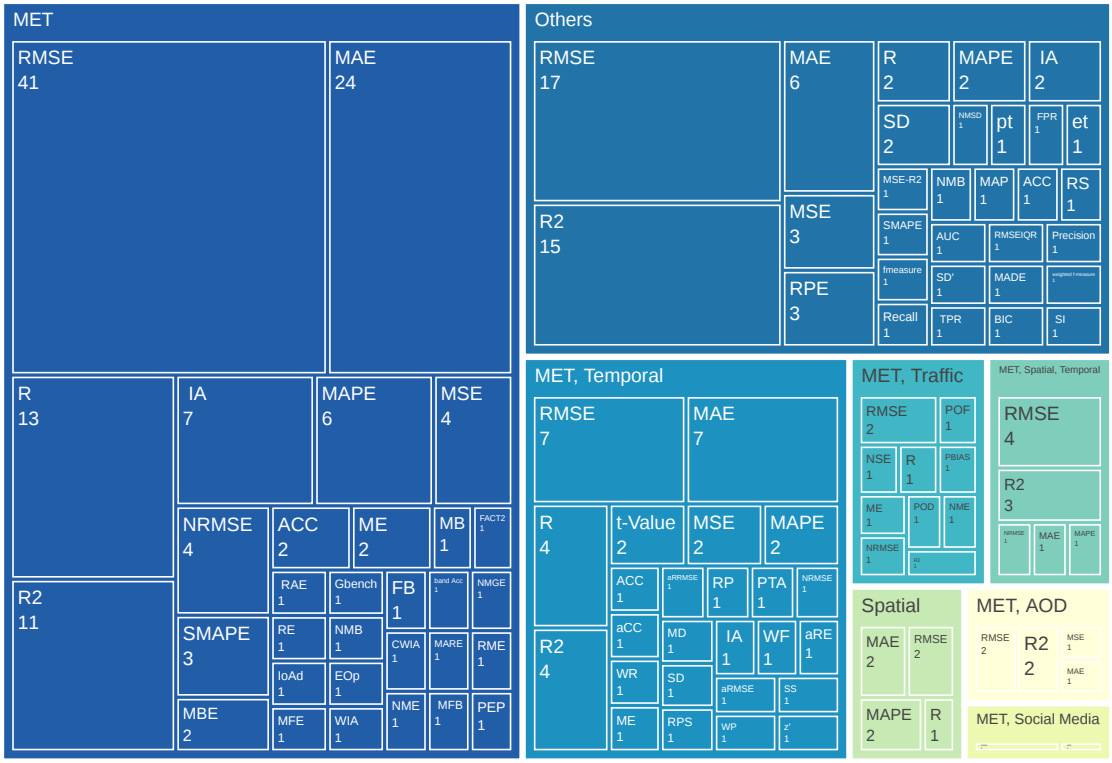


Figure 2.14: The number of publications of dataset combinations in terms of evaluation metrics.

learned by the graph LSTM. Xiao et al. [49] offered a Dual-Path Dynamic Directed Graph Convolutional Network (DP-DDGCN) based on the combination of dual-path dynamic directed graph blocks and GRU. Ouyang et al. [50] proposed a Spatiotemporal Dynamic Graph Convolutional Network (ST-DGCN) based on a time-varying dynamic adjacency matrix to predict $PM_{2.5}$. Ge et al. [51] offered a Multi-scale Spatiotemporal Graph Convolutional Network (MST-GCN) including a multi-scale block, several spatiotemporal blocks and a fusion block to forecast air quality. Wang et al. [52] used Attentive Temporal Graph Convolutional Network to model inter-station relationships (spatial adjacency, functional similarity, and temporal pattern similarity) to predict air quality. Chen et al. [53] proposed the group-aware GNN using the Chinese city air quality dataset to forecast nationwide city air quality. Xu et al. [54] performed air quality forecasting based on a hierarchical GNN; in particular, city-level and station-level graphs were constructed using the Yangtze River Delta city group's dataset. The authors developed two strategies,

Table 2.3: The most used metrics (more than six publications) with corresponding equations and definitions (where N is the number of predicted days, O_i and P_i are observed and predicted values, respectively, and $\overline{O_i}$ is the average of observed data).

Metrics	**Equations**	**Description**
RMSE	$\sqrt{\frac{1}{N}\sum_{i=1}^{N}(O_i - P_i)^2}$	It measures the geometric difference between observed and predicted data.
MAE	$\frac{1}{N}\sum_{i=1}^{N}\lvert O_i - P_i\rvert$	It measures the average magnitude of the errors in a set of predictions, without considering their direction.
R^2	$\frac{\left(\sum_{i=1}^{N}(P_i-\overline{P_i})(O_i-\overline{O_i})\right)^2}{\sum_{i=1}^{N}(P_i-\overline{P_i})^2\sum_{i=1}^{N}(O_i-\overline{O_i})^2}$	It shows how differences in one variable can be explained by a difference in a second variable.
R	$\frac{\sum_{i=1}^{N}(P_i-\overline{P_i})(O_i-\overline{O_i})}{\sqrt{\sum_{i=1}^{N}(P_i-\overline{P_i})^2\sum_{i=1}^{N}(O_i-\overline{O_i})^2}}$	It measures the strength and the direction of a linear relationship between two variables.
MAPE	$\frac{1}{N}\sum_{i=1}^{N}\left\lvert\frac{O_i-P_i}{O_i}\right\rvert$	It measures the size of the error in percentage terms.
IA	$1-\frac{\sum_{i=1}^{N}(O_i-P_i)^2}{\sum_{i=1}^{N}(\lvert O_i-\overline{O_i}\rvert+\lvert P_i-\overline{O_i}\rvert)^2}$	It is the ratio of the mean square error and the potential error.
MSE	$\frac{1}{N}\sum_{i=1}^{N}(O_i - P_i)^2$	It measures the average squared difference between the observed and the predict values
NRMSE	$\frac{\sum_{i=1}^{N}(O_i-P_i)^2}{\sum_{i=1}^{N}O_i^2}$	It is the normalised version of RMSE, which makes easier to compare different models with different scales.

upper delivery and lower updating, to implement the inter-level interactions and introduce a message-passing mechanism to implement the intra-level interactions. Another work is devoted to comparing graph-based and non-graph-based models for $PM_{2.5}$ prediction under distribution shift [55]. Le [56] used Spatiotemporal Graph Convolutional Recurrent Neural Network (STGCRNN) to efficiently explore

the spatiotemporal characteristics of air quality values and related factors.

Zhao et al. [57] introduced a novel model based on a combination of air quality spatiotemporal network and Graph Convolutional Network (GCN) for $PM_{2.5}$ prediction. Gao and Li [58] proposed graph-based LSTM model to perform spatiotemporal prediction of $PM_{2.5}$ concentration. Zhang et al. [59] used a temporal attention network with domain-specific graph regularisation for improving $PM_{2.5}$ prediction. Wang et al. [60] developed a new model called $PM_{2.5}$-GNN to capture fine-grained and long-term influences in the $PM_{2.5}$ process. Zhao and Zettsu [61] proposed multi-attention spatiotemporal graph networks to predict the concentration of $PM_{2.5}$, O_3, and PM_{10}. Qi et al. [62] implemented spectral GCN combined with LSTM using historical data for the last 24 h to forecast the $PM_{2.5}$ concentration for the next 1 h, 2 h, 4 h, 8 h, 12 h, 24 h, 48 h and 72 h. Huang et al. [63] implemented a Spatio-Attention embedded Recurrent Neural Network (SpAttRNN) to predict $PM_{2.5}$, PM_{10} and NO_2 using Beijing's air quality, meteorological and point of interest (POI) datasets; to capture spatial patterns, a self-loop-normalised adjacency matrix was used. Lin et al. [64] proposed the Geo-context based Diffusion Convolutional Recurrent Neural Network (GC-DCRNN) to predict $PM_{2.5}$. 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 (DCRNN) was responsible for collecting information in the temporal dimension.

The overall picture of the publications related to the implementation of GNN for air quality prediction can be seen in Table B.2 in Appendix B. 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 work. 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 for performing the prediction. As shown, most of the works involve GCN combined with 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 2.15 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).

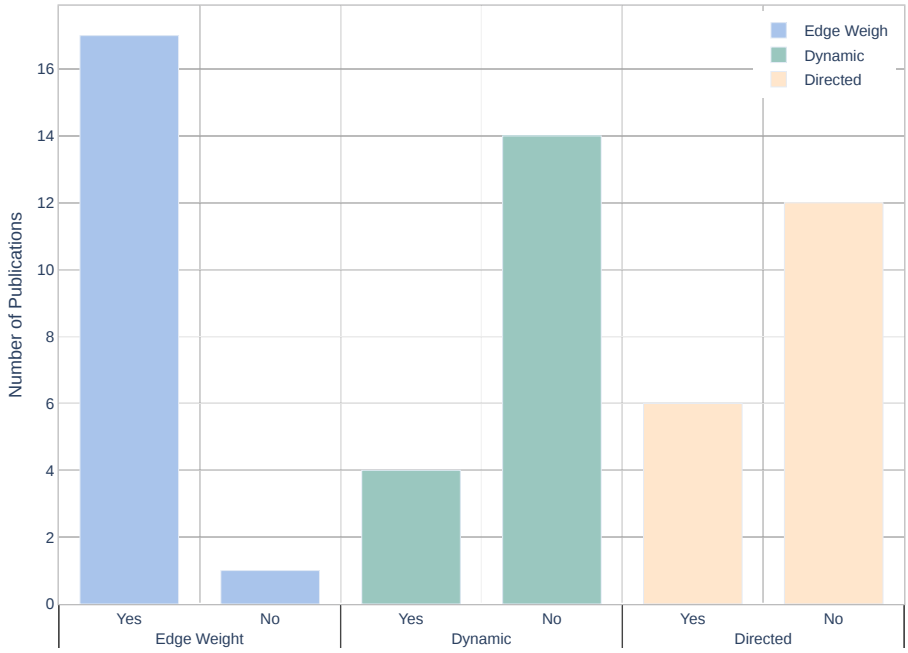


Figure 2.15: The number of publications in terms of edge weights (Yes-with weights, No-without weights), dynamic (Yes-dynamic, No-static) and direction (Yes-directed, No-undirected).

Target: are predictable pollutants. The following pollutants were taken into account: $PM_{2.5}$ (fourteen papers), PM_{10} (four papers), AQI (three papers), NO_2 (two papers), O_3 (two papers), and CO (one paper). The most used pollutant was $PM_{2.5}$.

Figure 2.16 shows the distribution of the prediction target over time. It can be seen that $PM_{2.5}$ has been in use since 2018. From 2020, additional targets are included, and in 2022, studies attempt to predict all the aforementioned prediction targets.

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), POI (five papers), traffic (two papers), road network (two papers) and geographic data (land uses, roads,

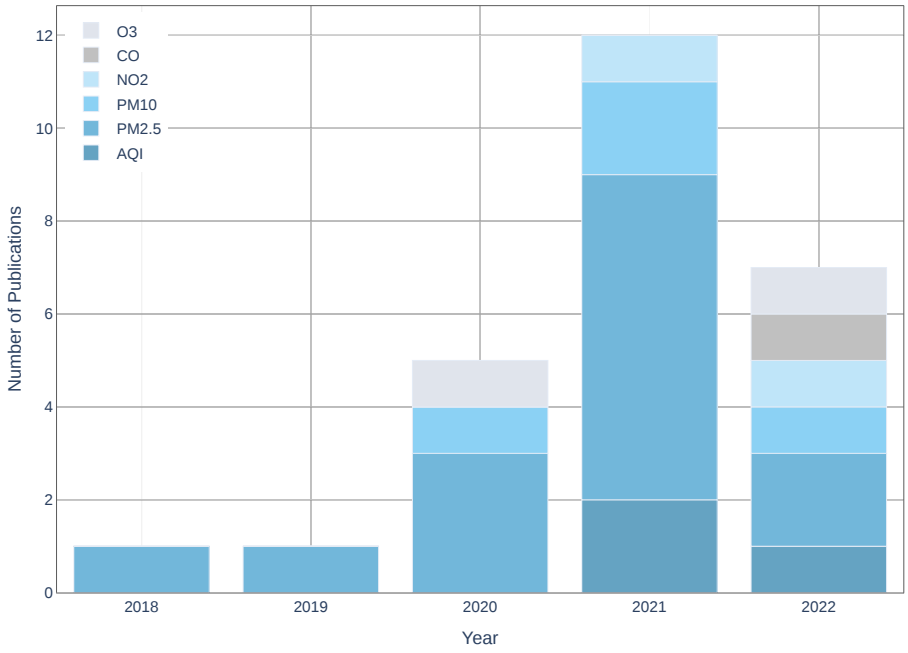


Figure 2.16: The number of publications per prediction target throughout the years.

water areas, buildings) (one paper). Air quality datasets and spatial datasets were most commonly used, which is logical, since the main task was air quality forecasting, and since the main focus of the work was on the implementation of GNN, 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 2.17). In recent years, in particular 2021 and 2022, the analysis began to include also POI, traffic and road network data.

Another interesting observation is related to dataset combinations. The following three combinations were formed: 'AQ, MET, Spatial' (ten papers), 'AQ, MET, Spatial, POI' (three papers), and the rest five combinations, each of which only appears in one publication, have been grouped as 'Others'. Figure 2.18 shows the dataset combinations in chronological order.

Evaluation Metric: is a metric to measure model performance. The following metrics were used, including RMSE (seventeen papers), MAE (sixteen papers), R^2 (three papers), False Alarm Rate (three papers), Accuracy (two papers), MAPE

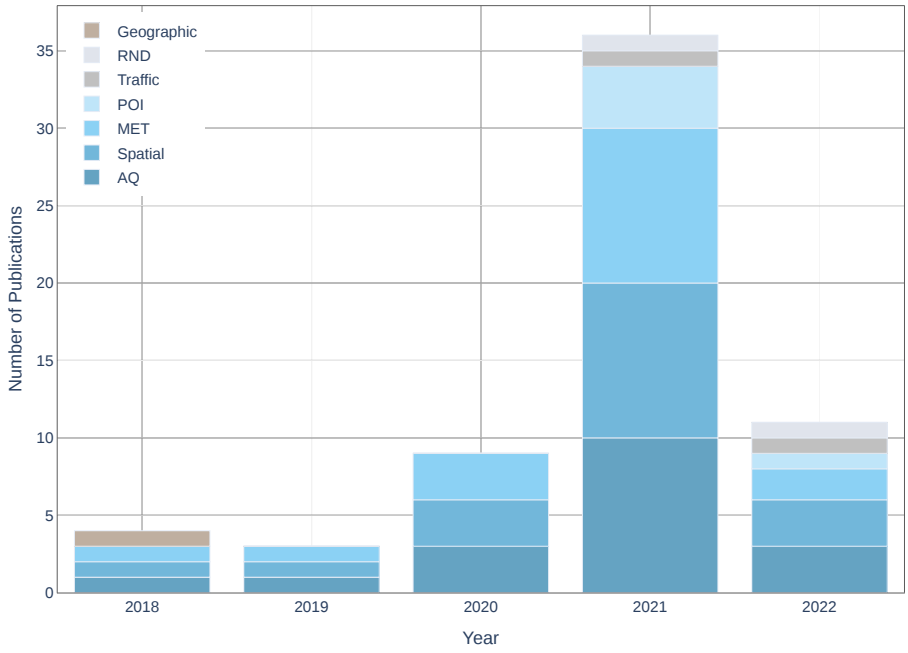


Figure 2.17: The number of publications per dataset throughout the years.

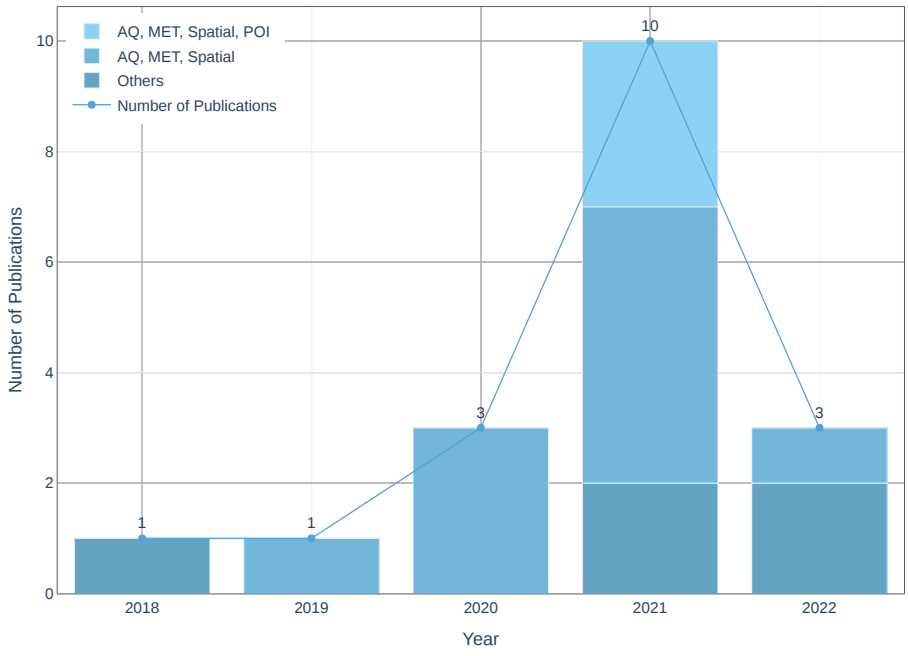


Figure 2.18: The distribution of the dataset combinations throughout the years.

(two papers), IA (two papers), Critical Success Index (two papers), Probability of Detection (two papers), Symmetric MAPE (two papers), train loss (two papers),

test loss (two papers), validation loss (two papers), Spatiotemporal RMSE (one paper), MSE (one paper), and Recall Rate (one paper).

2.3 Summary

Predicting air quality with higher accuracy is becoming increasingly important and necessary. Therefore, it is imperative to explore a variety of aspects of the research field. The principal goal of this chapter is to extract relevant studies on air quality prediction applying ML and GNN models and identify and analyse the key components that the researchers included in their analysis to predict air quality. Below are the main findings of the studies on the use of ML and GNN for air quality forecasting.

Air quality prediction applying ML models: a set of the most relevant papers in this field were selected using ACM, IEEE Xplore, Scopus and WoS databases. Overall, ninety-three papers were selected, and reviewed and, afterwards, the essential features were extracted and synthesised (*Year*, *Study Area*, *Prediction Target*, *Dataset Type*, *Data Rate*, *Period [Days]*, *Open Data*, *Algorithm* and *Time Granularity*). The findings demonstrate that twenty-six datasets are used to supplement data collected by air quality sensors, such as '*MET*', '*Temporal*', '*Spatial*' and '*Social Media*'. The results show a significant difference in the use of '*MET*', which is the main dataset used in 94.6% of the studies, and 48.4% of the studies combined with only air quality data.

Regarding data availability, it was shown that a new stage has begun since 2012, which is associated with the use of open data portals [65], which is crucial for science and contributes to the improvement and development of various research fields, as well as encouraging the emergence of new exciting results, resulting in an increase in the number of publications. Furthermore, open data impacts many areas, including increased transparency, increased efficiency and effectiveness of government services, citizen empowerment, and citizen involvement and participation in governance [66, 67].

A very important finding is to explore and understand which methods are most widely used and dominant in the field to predict a specific target. For example, to predict particulate matter, the most commonly used methods were discovered to be LSTM, SVM, and RF.

In general, it may be inferred that extra datasets can have significant importance,

and involving them in the analysis could improve air quality prediction and yield more accurate results. However, determining which datasets are more relevant is challenging, and it should also be highlighted that including numerous datasets is not always ideal, as having a large dataset might be an issue because it requires more training time and also may contain redundant data.

Air quality prediction applying GNN models: at this stage, eighteen studies were selected from Google Scholar, a brief description of which was presented in the previous section. After reviewing the papers, the key features were extracted, *Year*, *Method*, *Edge Weight*, *Dynamic/Static*, *Directed/Undirected*, *Target*, *Dataset*, and *Evaluation Metric*.

Chronological observation shows that the introduction of GNN in this area is a recent phenomenon, specifically since 2018.

Regarding methods, considering that the main purpose of the studies is to determine and compute the spatiotemporal dependencies of air quality, a graph mainly was combined with GRU, LSTM, and attention-based networks (GNN is responsible to capture spatial dependencies, and GRU, LSTM, and attention-based networks are responsible to capture temporal dependencies).

Regarding the graphs' architecture, they were predominantly static (77.77%) and undirected (66.66%), constructed with weighted edges (94.44%).

Regarding the prediction target, the dominant target is $PM_{2.5}$ (77.77%), and in terms of datasets, in addition to air quality, the following datasets were used (listed in most frequently used order): spatial, meteorological, POI, traffic, road network and geographic datasets.

The final component considered in this part is the evaluation metric. The following metrics were used in the studies (listed in most frequently used order): RMSE, MAE, R^2, False Alarm Rate, Accuracy, MAPE, IA, Critical Success Index, Probability of Detection, Symmetric MAPE, train loss, test loss, validation loss, Spatiotemporal RMSE, MSE, and Recall Rate.

Chapter 3

Methodology and Materials[1,2]

After examining the relevant works and becoming familiar with the most recent developments in the field, the next step is based on the findings and gaps to propose a new innovative methodology that allows to achieve the objectives of this dissertation and fill the identified gaps.

This chapter provides a detailed explanation of the proposed methodology and the extensive observation and examination of the used datasets. Below are listed the main contributions of the current chapter:

- We provided a description of the study area, i.e., the city of Madrid, and the

[2]The part of this chapter previously appeared as articles in the Journals IJCIA, PloS one, IEEE Access and Data in Brief, and as articles in the Conferences of AGILE and EnviroInfo. The original citations are as follows: Iskandaryan, Ditsuhi, Francisco Ramos, and Sergio Trilles. "Comparison of Nitrogen Dioxide Predictions During a Pandemic and Non-pandemic Scenario in the City of Madrid using a Convolutional LSTM Network." *International Journal of Computational Intelligence and Applications* 21, no. 02 (2022): 2250014; Iskandaryan, Ditsuhi, Francisco Ramos, and Sergio Trilles. "Bidirectional convolutional LSTM for the prediction of nitrogen dioxide in the city of Madrid." *PloS one* 17, no. 6 (2022): e0269295; Iskandaryan, Ditsuhi, Francisco Ramos, and Sergio Trilles. "Graph Neural Network for Air Quality Prediction: A Case Study in Madrid." *IEEE Access* 11 (2023): 2729-2742; Iskandaryan, Ditsuhi, Francisco Ramos, and Sergio Trilles. "Reconstructing Secondary Data based on Air Quality, Meteorological and Traffic Data Considering Spatiotemporal Components." *Data in Brief*, 2023; Iskandaryan, Ditsuhi, Silvana Di Sabatino, Francisco Ramos, and Sergio Trilles. "Exploratory Analysis and Feature Selection for the Prediction of Nitrogen Dioxide." *AGILE: GIScience Series* 3 (2022): 1-11; and Iskandaryan, Ditsuhi, Francisco Ramos, and Sergio Trilles. "Spatiotemporal Prediction of Nitrogen Dioxide Based on Graph Neural Networks." *Environmental Informatics*, pp. 111-128. Springer, Cham, 2023.

[2]The tools used in the scope of this dissertation are listed in Appendix D.

prediction target, i.e., NO_2;

- We integrated various data sources in spatiotemporal dimensions, including air quality, meteorological and traffic data from the period of January-June 2019 and January-June 2020, and the location of the monitoring stations and measurement points of the city of Madrid;
- We applied exploratory data analysis to detect existing patterns and relationships between various features;
- We preprocessed the entire dataset by implementing feature engineering techniques;
- We introduced a thorough description of the proposed algorithms by providing the necessary background for a complete understanding of the models' architecture.

The following sections provide a description of the study area and the predicted pollutant, and a comprehensive explanation of the workflow of the proposed methodology, consisting of the following steps: 1) *Data Preparation*, 2) *Exploratory Data Analysis*, 3) *Feature Engineering*, and 4) *ML Model Generation* (Figure 3.1).

3.1 Description of Study Area and Prediction Target

This section introduces the study area, the geographical description, and the chemical properties of the prediction target.

The study area of this work is the city of Madrid (Figure 3.2). It is the EU's second largest city in terms of population (3,305,408[3]), with a total area of approximately 604.31 km^2 and 18 neighbourhoods (the Autonomous Community of Madrid occupies about 8000 km^2 and it includes 178 municipalities). Madrid is located in the centre of the Iberian peninsula in southern Meseta Central. The average altitude is 650 metres (ranging between 570 and 740 meters)[4].

Madrid has a Mediterranean climate (Csa type: C-temperate, s-dry summer, a-hot summer)[5]. During the hot season (June -September) the average daily high

[3]Madrid Population: `https://bit.ly/3CkvR7Z`

[4]Elevation of Madrid: `https://bit.ly/3SyCUQ1`. [Online; accessed 15-February-2023]

[5]Climate Maps of Spain: `https://bit.ly/3C3vYU3`. [Online; accessed 15-February-2023]

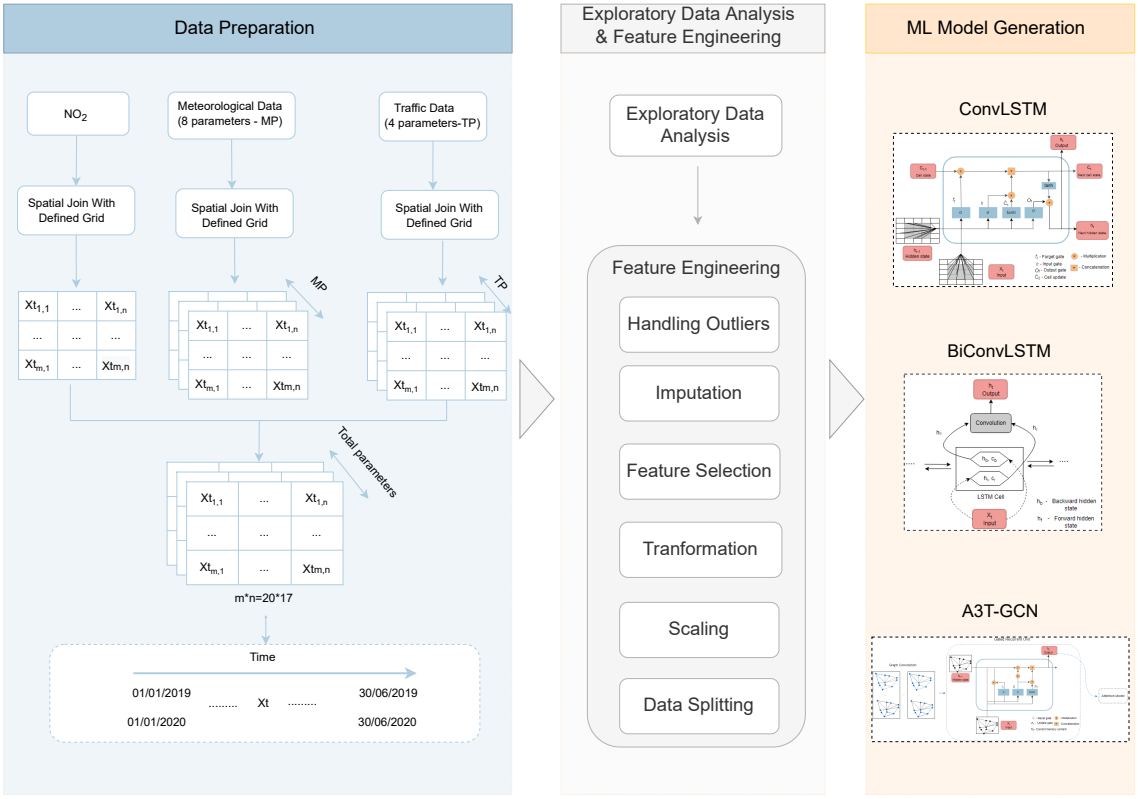


Figure 3.1: The overall workflow of the proposed methodology.

temperature is above 29ºC; in July (the hottest month) the average daily high temperature is 33ºC and the average daily low temperature is 17ºC. During the cool season (November-March) an average daily high temperature is below 15ºC; in January (the coldest month) an average daily low temperature is 1ºC and an average daily high temperature is 10ºC. The highest recorded temperature was 42.2ºC (24 July 1995), and the lowest recorded temperature was -15.3ºC (16 January 1945). Both records were registered at Barajas airport. Regarding precipitation, it is about 455 millimeters per year. During the rainy period (September - July) a sliding 31-day rainfall is 13 millimeters. The average rainfall is 8 millimeters during July (the rainless month). Regarding wind direction, it is most often from the west (April 4 - April 18 and April 26 -October 31), and from the north (April 18 - April 26 and October 31 - April 4). During the windier part of the year (January 27-May 7) the average wind speed is about 3.5 m/s. During the calmer time of year

(May 7-January 27) the average hourly wind speed is 3.2 m/s[6].

According to the study by Khomenko et al. [68] connected to premature mortality due to air pollution in European cities, which examined the pollutants $PM_{2.5}$ and NO_2, Madrid was found to have the highest NO_2 mortality burden. Because of the significance of NO_2 for Madrid, it was selected as an air pollutant for predictive analysis.

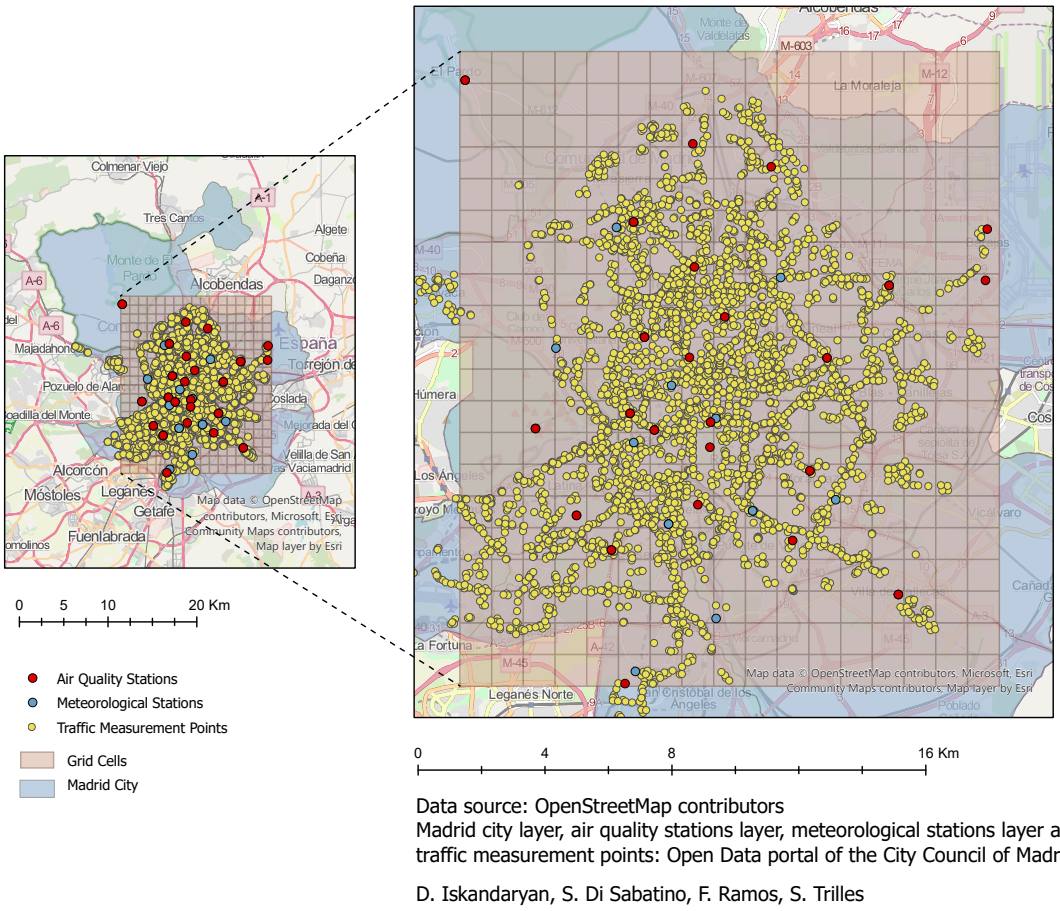


Figure 3.2: Air quality stations, meteorological stations, traffic measurement points (January 2019) and grid cells segments on the defined area of the city of Madrid.

5-10% of NO_2 is produced from direct emissions, and the rest from nitrogen oxide (NO) + oxidants in the atmosphere [69]. Due to combustion reaction (which

[6]Climate and Average Weather Year Round in Madrid: `https://bit.ly/3fDWT16`. [Online; accessed 15-February-2023]

is exothermic, temperatures up to 1500°C are reached) molecular nitrogen (N_2) and molecular oxygen (O_2) in the air can react producing NO which is quickly converted into NO_2 (Eq. 3.1). Particularly as part of the photochemical activity responsible for ozone formation (Eq. 3.4):

$$\mathrm{N_2 + O_2 \longrightarrow 2\,NO} \tag{3.1}$$

$$\mathrm{NO + O_3 \longrightarrow NO_2 + O_2} \tag{3.2}$$

$$\mathrm{NO_2 + h\nu(\lambda < 420}nm\mathrm{) \longrightarrow NO + O} \tag{3.3}$$

$$\mathrm{O + O_2 + M \longrightarrow O_3 + M} \tag{3.4}$$

NO_2 absorbs visible and Ultraviolet (UV) wavelength: 90% of NO_2 +[300-370nm] = NO + oxygen (O); small % of NO_2 +[370-420nm] = NO + O. Table 3.1 shows the source of NO_x. Fossil fuel combustion and biomass burning account for 72.39 % of the global source. The sinks are reactions with ozone, reactions NO_2 with photons, reactions with volatile organic compounds (VOC) to form ozone, reactions with hydroxyl radical (OH) to form a secondary aerosol, reactions of NO with peroxy radical to fast form NO_2. The principal sink of NO_x is oxidation to nitric acid (HNO_3).

- Daytime

$$\mathrm{NO_2 + OH + M \longrightarrow HNO_3 + M} \tag{3.5}$$

- Nighttime

$$\mathrm{NO_2 + O_3 \longrightarrow NO_3 + O_2} \tag{3.6}$$

$$\mathrm{NO_3 + NO_2 + M \longrightarrow N_2O_5 + M} \tag{3.7}$$

$$\mathrm{N_2O_5 + H_2O \xrightarrow{aerosol} 2\,HNO_3} \tag{3.8}$$

Much research has been conducted on the impact of NO_2, particularly on the rise in mortality from cardiovascular and respiratory diseases. For example, Faustini et al. [70] discovered that an increase in yearly concentration of NO_2 by 10 µg/m^3 had a Relative Risk (RR) 1.13 (95% Confidence Interval (CI) 1.09–1.18) on cardiovascular mortality and RR 1.03 (95% CI 1.02–1.03) on respiratory mortality.

Table 3.1: Global budget of nitrogen oxides.

Estimated present-day sources of tropospheric NO_x	**Source, Tg N yr^{-1}** (teragrams N per year)
Fossil fuel combustion	21
Biomass burning	12
Soils (biogenic, denitrification)	6
Lightning	3
NH_3 Oxidation (biogenic and atmospheric)	3
Aircraft + ships (fuel burning)	0.5
Transport from stratosphere	0.1

According to Hoek et al. [71], long-term exposure to NO_2 increases the risk of death by 5% for every 10 µg/m^3 NO_2. Hamra et al. [72] showed that the change in lung cancer incidence or mortality per 10 µg/m^3 increase in exposure is 4% [95% CI 1%-8%]. The authors of the following study [73] identified a link between NO_2 and COPD. The pooled effect of a 10 g/m^3 increase in NO_2 concentration on hospital admissions and on mortality was 1.3% and 2.6%, respectively. Long-term and short-term NO_2 exposure on COPD cases had an RR 2.5 and 1.4%, respectively. The COPD effect associated with a 10 µg/m^3 increase in exposure to outdoor-sourced NO_2 and to an exclusively traffic-sourced NO_2 was 1.7 and 17.8%, respectively. According to Brønnum-Hansen et al. [74], decreasing NO_2 exposure to rural levels (6 µg/m^3) might increase life expectancy by one year in 2040, and a 20% reduction in NO_2 would result in 1.3–1.6 years of disease-free life and 0.3-0.5 years of total life expectancy.

The largest source of NO_x emissions in the EU is the road transport sector (39 % in 2019)[7]. Different measures were implemented to reduce emissions (e.g., providing combustion modification technologies, and flue gas abatement techniques), thanks to which NO_x emissions in the EU decreased by 58.8% between 1990 and 2019. The decline in Spain between 2005 and 2010 is due to the closure of the main brown coal mine (2007), as well as the upgrades of a

[7]European Union emission inventory report 1990-2019: `https://bit.ly/3fE5G3d`. [Online; accessed 15-February-2023]

nearby thermal power plant.

In a study carried out by Cuevas et al. [75], the authors observed the temporal evolution of NO_2 in five Spanish cities, including Madrid, over the period 1996-2012. Applying the shift trend model to NO_2 data, they found that NO_2 levels in the Madrid area had decreased by about 53%. A comparison of average annual values recorded from air quality monitoring stations shows that Madrid has experienced a 37% decline. This decline is linked to the implementation of environmental legislation and technologies and the implications of the global economic crisis. According to the study, the annual decline was 1.1% prior to the recession, and 7.8% during the economic recession. Therefore, it is observable that economic and industrial factors have a considerable impact on NO_2.

Despite the fact that the deployment of control policies and strategies favourably influences air pollution reduction, the problem continues to be a source of concern. New technology can assist in making more informed and efficient decisions.

3.2 Data Preparation

This section introduces the employed datasets and the procedure of data preparation and combination in the spatiotemporal dimensions.

The dataset used in this study consists of air quality, meteorological and traffic data from the period of January-June 2019 and January-June 2020, and the location of air quality and meteorological monitoring stations and traffic measurement points of the city of Madrid. The reason for selecting data only for the first six months of 2019 and 2020 is due to computational limitations, therefore we decided to downsize the data to be able to execute the predictive analysis. The data was acquired from the Open Data portal of the Madrid City Council[8]. There are twenty-four air quality and twenty-six meteorological control stations, and more than 4,000 traffic measurement points. The following variables are included in the dataset:

- Air Quality Data - NO_2 ($\mu g/m^3$).
- Meteorological Data - UV irradiance (mW/m^2), wind speed (m/s), wind direction, temperature (^oC), relative humidity (%), barometric pressure (mb), solar

[8]Open Data Portal of the Madrid City Council: `https://bit.ly/3FFRiQM`. [Online; accessed 15-February-2023]

irradiance (W/m^2), precipitation (l/m^2).

- Traffic Data - Since the attributes of the traffic data can be specific to a certain area, below are the selected traffic attributes with their definition for the city of Madrid.

  - Intensity - the intensity of the measurement point in a period of 15 minutes (vehicles/hour). A negative value implies the absence of data.
  - 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. A negative value implies the absence of data.
  - 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). A negative value implies the absence of data.
  - Average traffic speed - an average speed of the vehicles in a period of 15 minutes (km/h). Only for M30 intercity measuring points. A negative value implies the absence of data.

Although the traffic data is captured every 15 minutes, however, since the NO_2 and meteorological data are at hourly rates, the traffic data was filtered. 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).

After accessing the raw data, the next important step is data preparation and integration. Since the location of the air quality stations, meteorological stations and traffic measurement points are different, it is essential to combine them spatially and temporally. The initial step was to create a grid in a specified area, which was defined as a section of the city of Madrid with a width and height of 1,000 metres within the following boundaries: 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)[9]. The grid was created with the help of ArcPy package[10], specifically with the $CreateFishnet$ function[11]. Within the required extent, the output generated a grid with 340 cells (20 by 17) covering 340 km^2 or 56.27% of the total area of the city of Madrid. The rationale for selecting this area was to have a minimum extent to encompass all air quality control stations. The value of each cell consists of the values of NO_2, meteorological and traffic attributes obtained from assigned stations covered by that cell at a certain time. The value of the cell that does not contain any station was set to zero and in the case of several stations, an average value was calculated and assigned to the cell. The above procedure was repeated for each hour of the selected period. The following functions were used to execute the aforementioned process, including $arcpy.management.AddField$[12], $arcpy.analysis.SpatialJoin$[13], $arcpy.da.SearchCursor$[14], $arcpy.da.UpdateCursor$ [15]. The output was exported as $.csv$ files, which were later used as input in further stages of the analysis. Overall, 4,344 and 4,368 $.csv$ files were generated corresponding to every hour during January-June 2019 and January-June 2020, respectively. The input data: X can be defined as follows:

$$X = X_{no_2} + X_{uv} + X_{ws} + X_{wd} + X_{temp} + X_{hum} + X_{press} + X_{sr} + X_{prec} + X_{intens} + X_{ocup} + X_{load} + X_{ats} \tag{3.9}$$

where $+$ is a vector concatenation operator, $X_{no_2} \in R_{no_2}^{s \times m \times n}$ is the NO_2 input data, R_{no_2} is the NO_2 domain; $X_{uv} \in R_{uv}^{s \times m \times n}$ is the UV input data, R_{uv} is the UV domain; $X_{ws} \in R_{ws}^{s \times m \times n}$ is the wind speed input data, R_{ws} is the wind speed domain; $X_{wd} \in R_{wd}^{s \times m \times n}$ is the wind direction input data, R_{wd} is the wind direction domain; $X_{temp} \in R_{temp}^{s \times m \times n}$ is the temperature input data, R_{temp} is the temperature

[9]Projected coordinate system: `https://epsg.io/25830`. [Online; accessed 15-February-2023]

[10]ArcPy package: `https://bit.ly/3UPYKjy`. [Online; accessed 15-February-2023]

[11]Create Fishnet (Data Management): `https://bit.ly/3Rn62Zj`. [Online; accessed 15-February-2023]

[12]Add Field (Data Management): `https://bit.ly/3LPo1GE`. [Online; accessed 15-February-2023]

[13]Spatial Join (Analysis): `https://bit.ly/3M6SC2J`. [Online; accessed 15-February-2023]

[14]SearchCursor: `https://bit.ly/3y3tcNz`. [Online; accessed 15-February-2023]

[15]UpdateCursor: `https://bit.ly/3y0txjU`. [Online; accessed 15-February-2023]

domain; $X_{hum} \in R_{hum}^{s \times m \times n}$ is the relative humidity input data, R_{hum} is the relative humidity domain; $X_{press} \in R_{press}^{s \times m \times n}$ is the barometric pressure input data, R_{press} is the barometric pressure domain; $X_{sr} \in R_{sr}^{s \times m \times n}$ is the solar irradiance input data, R_{sr} is the solar irradiance domain; $X_{prec} \in R_{prec}^{s \times m \times n}$ is the precipitation input data, R_{prec} is the precipitation domain; $X_{intens} \in R_{intens}^{s \times m \times n}$ is the intensity input data, R_{intens} is the intensity domain; $X_{ocup} \in R_{ocup}^{s \times m \times n}$ is the occupancy time input data, R_{ocup} is the occupancy time domain; $X_{load} \in R_{load}^{s \times m \times n}$ is the load input data, R_{load} is the load domain; $X_{ats} \in R_{ats}^{s \times m \times n}$ is the average traffic speed input data, R_{ats} is the average traffic speed domain, s is the number of samples: 4,344 and 4,368 for January-June 2019 and January-June 2020, respectively, m is equal 20, and n is equal 17. The final input $X \in R^{s \times 340 \times f}$, where s is the number of samples: 4,344 and 4,368 for January-June 2019 and January-June 2020, respectively, 340 is the multiplication of m and n (20*17), and f is the number of features equal to 13 ($X \in R^{4,344 \times 340 \times 13}$ for January-June 2019 and $X \in R^{4,368 \times 340 \times 13}$ for January-June 2020).

A formal description of the data preparation process is given by Algorithm 1.

Algorithm 1 Data preparation

Input: Data - [Hourly NO_2, Meteorological and Traffic data]; Period - [01.01.2019-30.06.2019; 01.01.2020-30.06.2020]

1: **for** each hour ∈ Period **do**
2: Create grid with Fishnet tool (ArcPy library)
3: Add field to the Fishnet
4: **for** each item $i \in$ Data **do**
5: i spatial join with grid: $arcpy.management.AddField$, $arcpy.analysis.SpatialJoin$, $arcpy.da.SearchCursor$, $arcpy.da.UpdateCursor$
6: input the mean of the values of each corresponding cell to the field
7: **end for**
8: **end for**

Output: $.csv$ files for each hour including NO_2, Meteorological and Traffic data

Table 3.2 displays summary statistics for each type of data for the periods studied for the defined area.

Table 3.2: Summary statistics of the periods January-June 2019 and January-June 2020 for each data type.

Phenomena	Descriptors	January-June 2019	January-June 2020
NO_2 ($\mu g/m^3$)	Mean (SD)	36.69 (30.85)	26.03 (25.35)
	Median [Min,Max]	27.0 [0.0, 328]	17.0 [0.0, 326]
UV (mW/m^2)	Mean (SD)	15.83 (30.27)	-
	Median [Min,Max]	1.0 [0.0, 199]	-
Wind speed (m/s)	Mean (SD)	1.41 (1.11)	1.31 (1.05)
	Median [Min,Max]	1.14 [0.0, 8.75]	1.05 [0.0, 8.97]
Wind direction	Mean (SD)	167.80 (105.72)	140.82 (98.35)
	Median [Min,Max]	182.0 [0.0, 359]	135.0 [0.0, 359]
Temperature (^oC)	Mean (SD)	13.38 (8.09)	13.63 (7.6)
	Median [Min,Max]	12.5 [-55.0, 47.3]	12.6 [-55.0, 44.6]
Humidity (%)	Mean (SD)	48.73 (21.60)	60.76 (22.77)
	Median [Min,Max]	47.0 [-25, 100]	62.0 [-25, 100]
Pressure (mb)	Mean (SD)	943.3 (34.91)	940.62 (63.28)
	Median [Min,Max]	945.0 [0.0, 962.0]	945.0 [0.0, 1073.0]
Solar irradiance (W/m^2)	Mean (SD)	220.73 (301.06)	191.95 (279.83)
	Median [Min,Max]	11.0 [0.0, 1103.0]	9.0 [0.0, 1113.0]
Precipitation (l/m^2)	Mean (SD)	0.03 (0.41)	0.03 (0.27)
	Median [Min,Max]	0.0 [0.0, 30.4]	0.0 [0.0, 13.5]
Intensity (vehicles/hour)	Count_non_zero	885863 (59.98%)	892197 (60.09%)
	Mean (SD)	245.69 (402.73)	161.45 (313.33)
	Median [Min,Max]	63.0 [0.0, 6348.0]	34.19 [0.0, 6588.0]
Occupancy time (%)	Count_non_zero	845031 (57.21%)	822652 (55.41%)
	Mean (SD)	3.96 (6.36)	2.57 (4.9)
	Median [Min,Max]	0.95 [0.0, 100.0]	0.42 [0.0, 99.0]
Load	Count_non_zero	881500 (59.68%)	884950 (59.60%)
	Mean (SD)	11.65 (14.91)	7.85 (11.75)
	Median [Min,Max]	4.0 [0.0, 100.0]	2.2 [0.0, 100.0]
Average traffic speed (km/h)	Count_non_zero	233415 (15.8%)	223052 (15.0%)
	Mean (SD)	4.39 (13.28)	4.04 (12.96)
	Median [Min,Max]	0.0 [0.0, 96.5]	0.0 [-127.0, 127.0]

3.3 Exploratory Data Analysis

Following data acquisition and preparation, the next stage is to perform exploratory data analysis, which is the process of conducting a thorough examination to identify patterns and anomalies and test hypotheses, to determine the relationship between various features, spatial correlation between stations, and temporal correlation between items.

According to Figure 3.3, the time series of NO_2 during January-June 2019 and January-June 2020 decreases over time, which might be attributed to domestic heating use throughout the winter. Moreover, the overall concentration during 2020 is lower than that for the same period of 2019, which can be explained by the constraints enforced to control the spread of coronavirus disease 2019 (COVID-19). From Figure 3.4 it can be seen that the maximum values during January-June 2019 are detected around 300 $\mu g/m^3$ and the highest concentration was detected at the station with id 72 (328 $\mu g/m^3$ at the following time: 2019-01-14 19:00:00;

Figure 3.5 shows air quality stations with identified values); and during the 2020 period, the greatest value was identified in the station with id 181 (326 µg/m^3 at the following time: 2020-02-10 09:00:00). It should be noted that according to WHO guideline, the annual mean of NO_2 is 40 µg/m^3, and 1-h mean is 200 µg/m^3.

Figure 3.3: The time series of the concentration of nitrogen dioxide at all the stations during January-June 2019 (top) and January-June 2020 (bottom) in the city of Madrid.

Regarding the spatial correlation, Figure 3.6 displays the heatmaps to detect the correlation between time series in the stations. During the period of 2019, the stations are correlated, except 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 2020, in addition to the station with id 323, the station with id 141 was also found to be less correlated. The time series of NO_2 concentration at station 141 has 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 (Figure 3.7).

Respecting temporal correlation, Figure 3.8 illustrates autocorrelation (or the correlogram, the correlation between values of the same series at different time steps) and partial autocorrelation plots of NO_2 concentration; the daily interval is chosen as a lag length and the plots show the results of 80 lags. The plots were

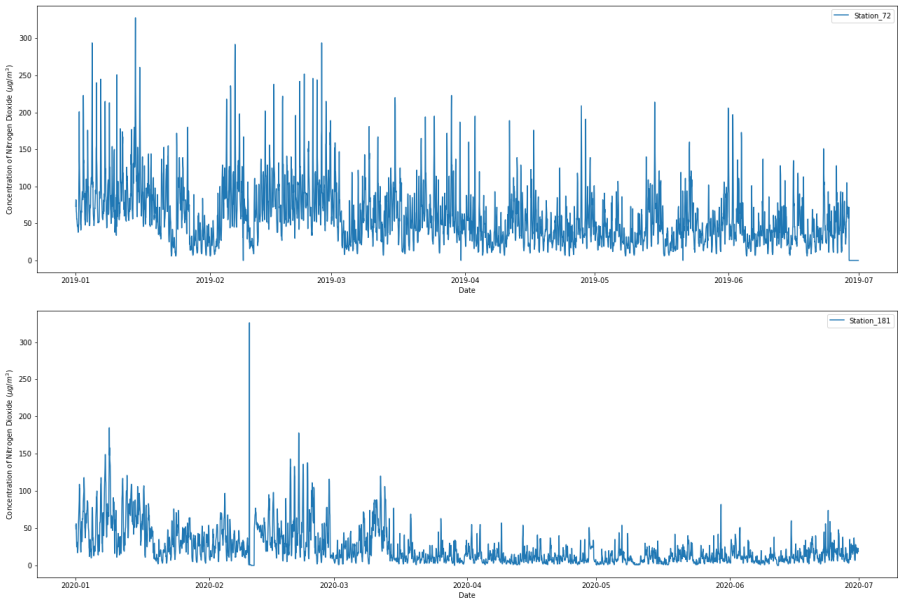


Figure 3.4: The time series of the concentration of nitrogen dioxide at stations with maximum values for each period in the city of Madrid (top: the station with id 72 during January-June 2019; bottom: the station with id 181 during January-June 2020).

generated using $plot\_acf()$[16] and $plot\_pacf()$[17] functions from the statsmodels library[18]. The difference between autocorrelation and partial autocorrelation is that the former calculates the correlation between two lags while considering the influence of previous observations (direct and indirect effects), whereas the latter is simply a real correlation between two lags without intervening observations (only direct effects). These functions assist in determining the optimal lags, which can be selected for effective forecasting. More than 25 lags have a significant positive correlation in the autocorrelation plot, while in the partial autocorrelation plot, there is a statistically significant correlation for lag 1 and 2 periods.

Afterwards, the next step was to identify the relationship between the features. This procedure started by constructing a wind rose for each station to reveal the interconnection between wind speed and wind direction. The wind roses were

[16]Autocorrelation function: `https://bit.ly/3CnYgci`. [Online; accessed 15-February-2023]

[17]Partial autocorrelation function: `https://bit.ly/3fEuMPp`. [Online; accessed 15-February-2023]

[18]Statsmodels: `https://bit.ly/2Th8jMi`. [Online; accessed 15-February-2023]

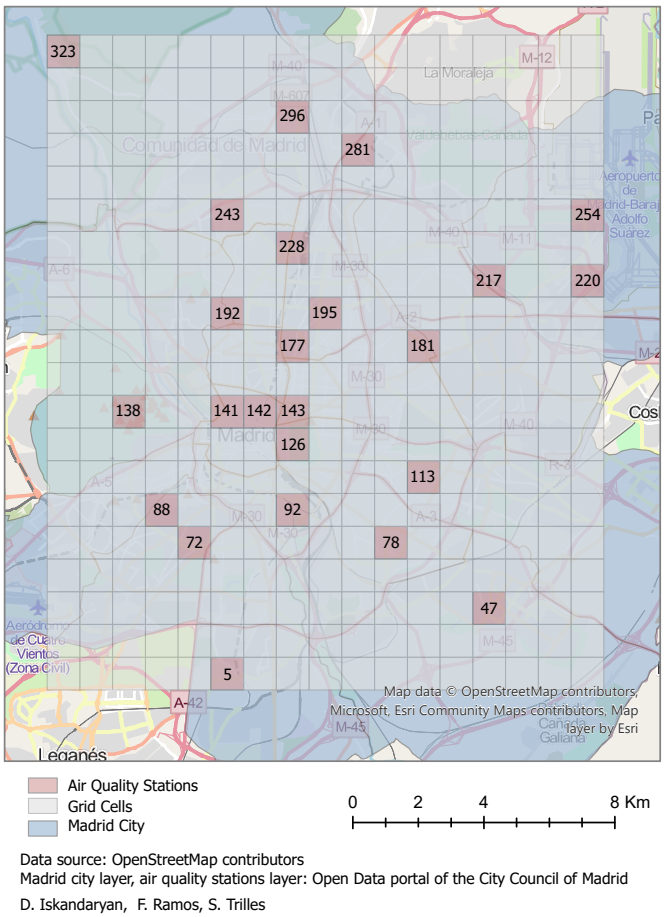


Figure 3.5: Air quality stations with identified values in the city of Madrid.

generated using the WRPLOT VIEW platform[19]. It turned out that out of twenty-six meteorological stations, only ten stations provide data on wind speed and direction. Then, based on the generated wind roses, a map was created showing the dominant wind directions at each station, marked with different colours (Figure 3.9). The output showed that in January the following dominant directions were highlighted (with station id, respectively)- North: 173, 217; North-East: 214, 96; East: 72; Southwest: 138; South: 42; West: 242, 47, 5; and in June- South: 42, 214; South-West: 72, 138, 173, 217, 242; West: 5, 47, 96. Wind speed was classified based on the Beaufort scale [76, 77].

Following the generation of a wind rose for each station, it was found that higher

[19]WRPLOT VIEW: `https://bit.ly/3SPucxf`. [Online; accessed 15-February-2023]

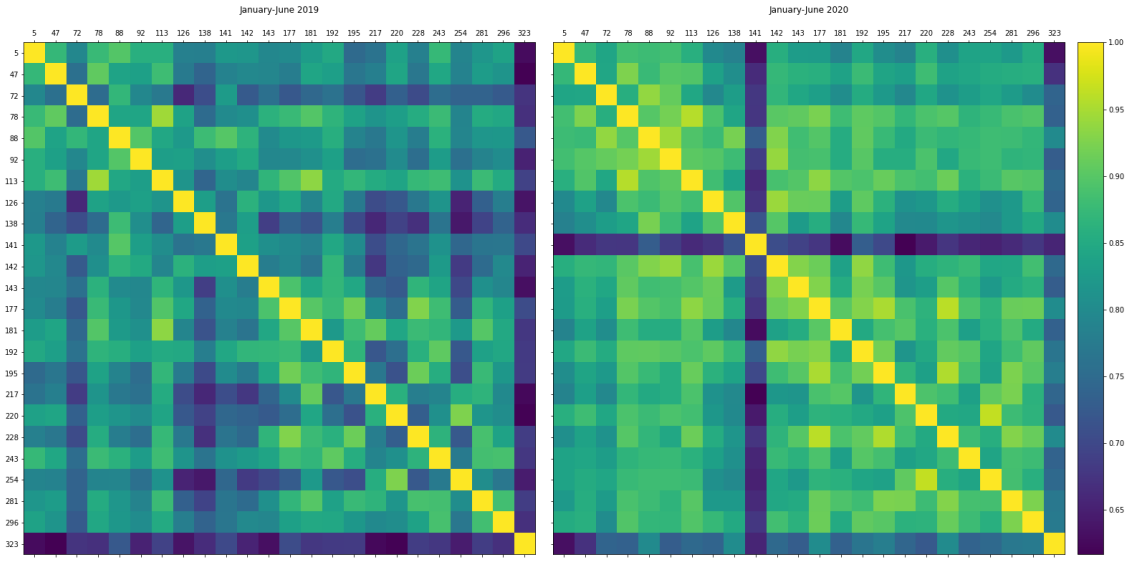


Figure 3.6: The correlation between the time series of nitrogen dioxide at the stations during January-June 2019 (left) and January-June 2020 (right) in the city of Madrid.

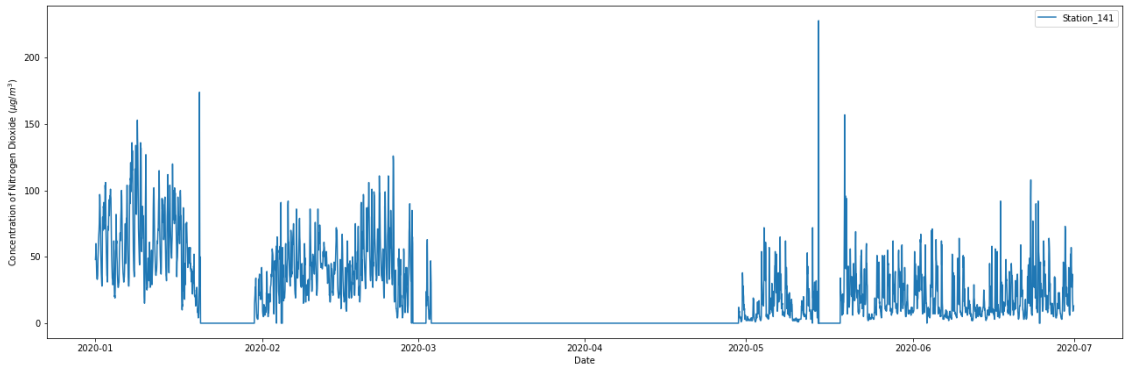


Figure 3.7: The time series of the concentration of nitrogen dioxide at the station with id 141 during January-June 2020.

wind speed does not always correspond to the dominating direction. For example, Figure 3.10 shows that at a station with id=96 during January (calms wind 3.63%) the predominant direction is northeast, but a higher wind speed was recorded in the westerly direction.

To reveal a relationship between concentration and wind speed, these variables' time series were plotted to see the changes over time. For example, Figure 3.11a

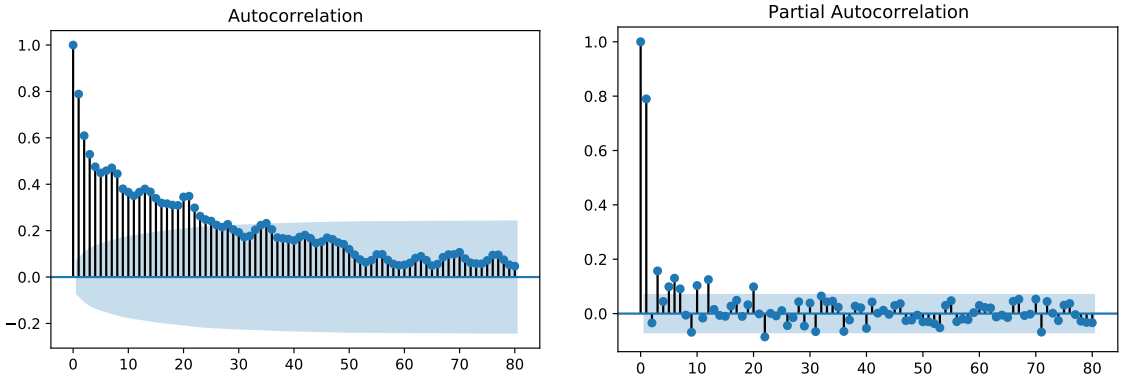


Figure 3.8: Autocorrelation and partial autocorrelation plots with 80 lags from the nitrogen dioxide dataset.

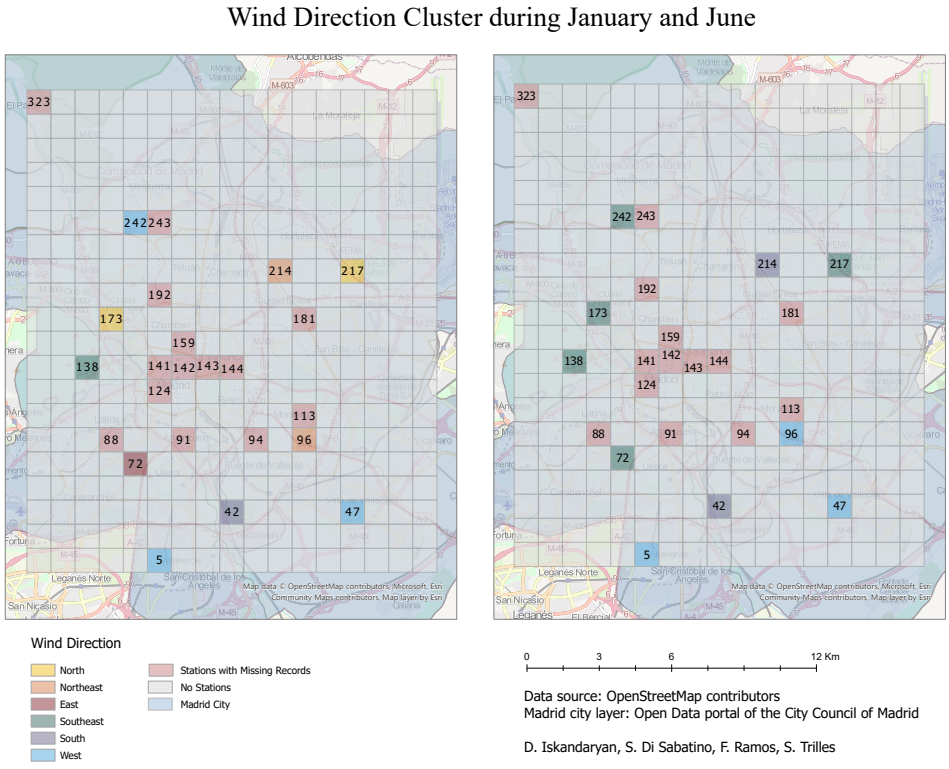


Figure 3.9: Wind direction cluster during January (left) and June (right) 2019 in the city of Madrid.

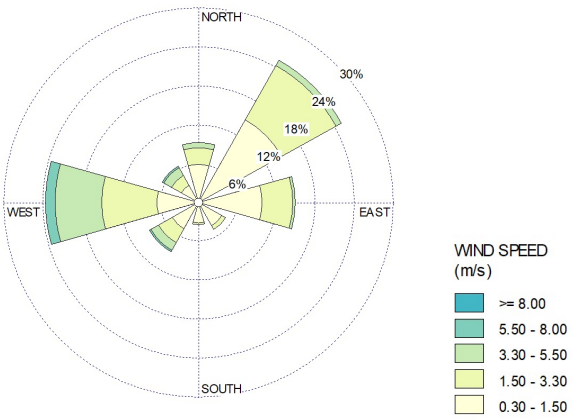
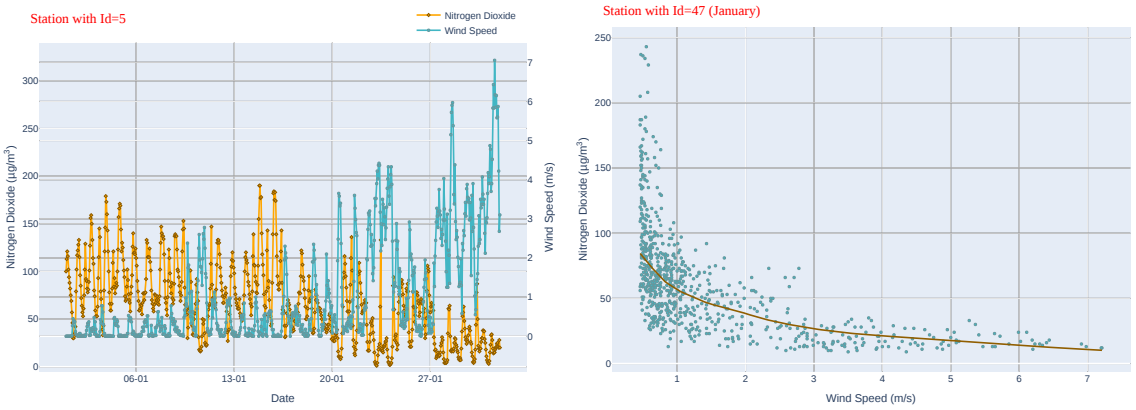


Figure 3.10: Wind rose at the station with id=96 during January

displays a time series of NO_2 and wind speed for the station with id=5 during January. Note that these two variables are inversely proportional; particularly, higher wind speed assumes lower concentration due to increased dilution through advection and increased mechanical turbulence. The scatter plot, which was generated using the y-axis for NO_2 and the x-axis for wind speed, reflects this finding (Figure 3.11b; the trendline is based on locally weighted scatterplot smoothing [78]).



(a) Time series (id=5).

(b) Scatter plot(id=47).

Figure 3.11: Time series of nitrogen dioxide and wind speed at the station with id=5 (a) and scatter plot of nitrogen dioxide and wind speed at the station with id=47 (b) during January 2019 in the city of Madrid.

Another analysis was performed to generate polar plots using openair R pack-

age[20] with the aim to detect the relationship between concentration, wind speed, and wind direction. Figure 3.12 shows polar plots at the station with id=47 during January and June. In the central part with a lower wind speed, the concentration is higher, and in the edges with a higher wind speed - with a lower concentration. In the polar plots obtained using the average NO_2, the concentration is lower in June than in January, which can be explained by domestic heating during winter.

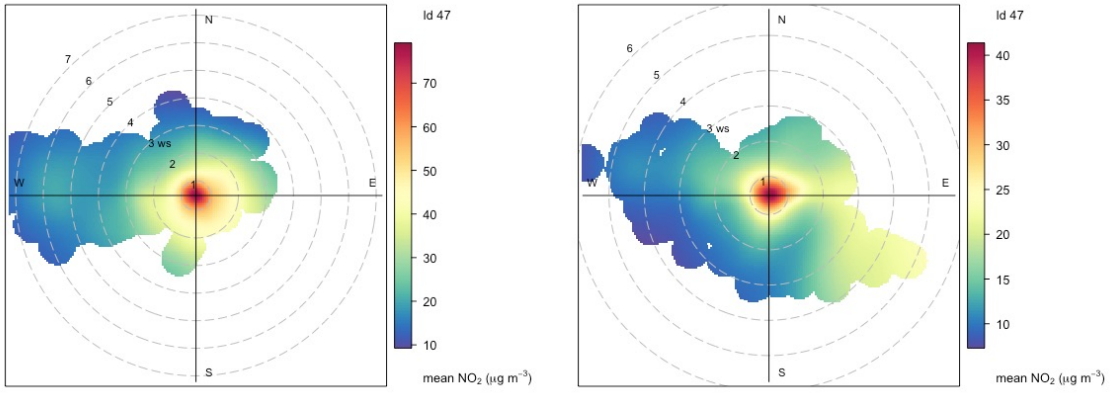


Figure 3.12: Polar plot of wind speed, wind direction and mean concentration of nitrogen dioxide during January 2019 (left) and June 2019 (right) at the station with id=47 in the city of Madrid.

Additionally, analysis was executed to determine the relationship between non-dimensional concentration and non-dimensional wind speed. To calculate the non-dimensional concentration (Eq. 3.10) and non-dimensional wind speed (Eq. 3.11) are illustrated below [79].

$$C\_ADIM = C * U * L * H/EMISSIONS \tag{3.10}$$

$$U\_ADIM = U/U\_arp \tag{3.11}$$

where, C_ADIM is non-dimensional concentration, C is the concentration (µg/m3), U is wind speed (m/s), L is the road length (km) in a certain cell (it was calculated using ArcGIS Pro software), H is the planetary boundary layer height (m) in the Adolfo Suárez Madrid–Barajas Airport (it was generated by the ERA5

[20]Openair R package: `https://bit.ly/3LQUP1Q`. [Online; accessed 15-February-2023]

model: European Centre for Medium-Range Weather Forecasts[21]), $EMISSIONS$ is NO_x, U_ADIM is non-dimensional wind speed, U_arp is the wind speed [10m] in the Adolfo Suárez Madrid–Barajas Airport (m/s) (it was obtained from the ERA5 model).

Figure 3.13 shows the scatter plots of the non-dimensional concentration and non-dimensional wind speed. The plot is ambiguous on how these two features relate to one another.

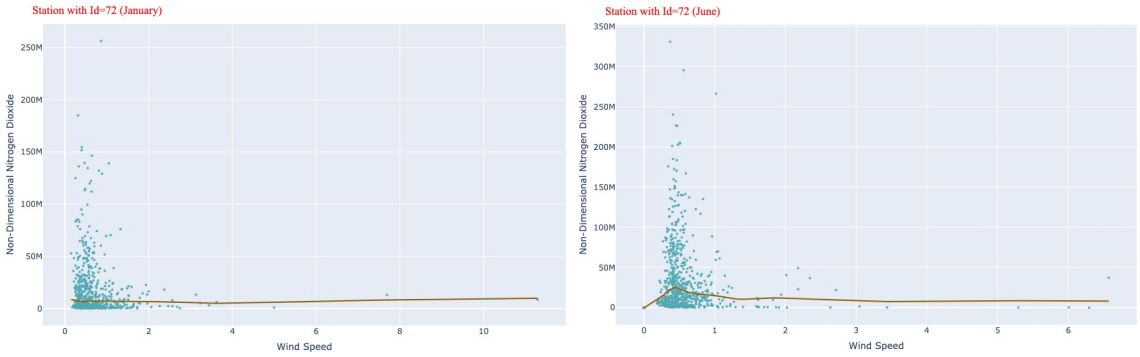


Figure 3.13: Scatter plot of the non-dimensional nitrogen dioxide concentration and non-dimensional wind speed during January 2019 (left) and June 2019 (right) at the station with id=72 in the city of Madrid.

Examining the plots of concentration and wind speed, it was detected that in January and June, the concentration in the station with id=72 is higher and the wind speed is lower; in January, the concentration in the station with id=138 is the lowest; concentration and wind speed are more correlated during winter compared to summer.

The above-mentioned analyses were carried out between NO_2 and other variables, however, it was challenging to reveal any correlation from the plots.

Several factors must be considered for further predictive analysis, in particular, UV and precipitation should be excluded. Regarding UV, it was observed that in January it was only recorded in three stations with NO_2 records (station: id=47, id=38, id=217) and that there were no UV records in June; moreover, there were no UV records for the period of January-June 2020. Regarding precipitation, it was found out that around 99% of the data was 0. Another feature that should be also mentioned in this context is average traffic speed, even though it was included

[21] ECMWF: `https://www.ecmwf.int/en/about`. [Online; accessed 15-February-2023]

in further analyses. This is because the average traffic speed is available only for the M30 road, which is 15.8% of the study area (Figure 3.14 shows average traffic speed for a period of one week).

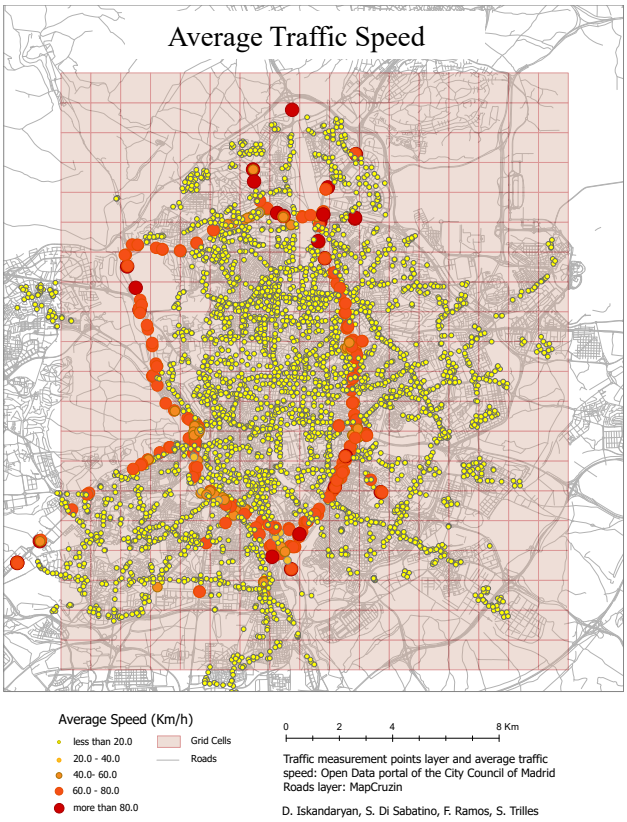


Figure 3.14: Average traffic speed for the period 1-7 January 2019 in the city of Madrid.

3.4 Feature Engineering

The next fundamental step of the workflow is feature engineering. Feature engineering is a preprocessing step of machine learning that is used to extract and organise essential features from raw data, to transform them into meaningful features with the aim to improve the accuracy of the predictive model. It consists of various data engineering techniques, including *Handling Outliers*, *Imputation*, *Feature Selection*, *Transformation*, *Scaling* and *Data Splitting* (see in Figure 3.1).

Handling Outliers: there are the samples or observations that are far from the

rest of the observations (outliers are categorised into three categories, including point, contextual and collective outliers [80]). They can affect the model's accuracy, therefore, it is critical to process them. Three different approaches were used to detect outliers: 1) overview of summary statistics, 2) Isolation Forest (iForest) [81], and 3) Local Outlier Factor (LOF) [82].

Overview of summary statistics: Table 6.2 can serve as a guide for outliers detection. The minimum values of humidity and temperature data indicate that they are outliers. Temperatures below -3° for 2019 and -2° for 2020[22] and humidity with negative values were considered outliers.

iForest: is an unsupervised decision-tree-based algorithm. It randomly selects a feature and then randomly selects a split value between the maximum and minimum values of that feature. Random splitting will create a shorter path for outliers since they usually require fewer partitions to be isolated. The following equation calculates the anomaly score given a data point x and a sample size of n:

$$s(x, n) = 2^{-\frac{E(h(x))}{c(n)}} \tag{3.12}$$

where $h(x)$ is the path length of x, $c(n)$ is the average path length of an unsuccessful search in a Binary Search Tree, n is the number of external nodes and $E(h(x))$ is the average of $h(x)$ from a collection of isolation trees.

Based on the anomaly score the following decisions are possible: 1) s close to 1 are anomalies, 2) s smaller than 0.5 can be considered as normal instances, and 3) if for all instances $s \approx 0.5$, the entire dataset does not include outliers.

LOF: is an unsupervised method which calculates the local density deviation of a certain observation from its neighbours. It is calculated with the following equation:

$$LOF_{\mathsf{k}}(A) = \frac{\sum_{X_{\mathsf{j}} \in N_{\mathsf{k}}(A)} LRD_{\mathsf{k}}(X_{\mathsf{j}}))}{||N_{\mathsf{k}}(A)||} \times \frac{1}{LRD_{\mathsf{k}}(A)} \tag{3.13}$$

where $N_{\mathsf{k}}(A)$ is K-neighbors, which contains the samples that are placed within the circle of radius K-distance; LRD is local reachability density, and it computes with the equation displayed below:

[22] Past Weather in Madrid, Madrid, Spain: `https://bit.ly/3LPZRMf`. [Online; accessed 15-February-2023]

$$LRD_{\mathsf{k}}(A) = \frac{1}{\sum_{X_{\mathsf{j}} \in N_{\mathsf{k}}(A)} \frac{RD(A, X_{\mathsf{j}})}{||N_{\mathsf{k}}(A)||}} \tag{3.14}$$

where RD is reachability density and equal:

$$RD(X_{\mathsf{i}}, X_{\mathsf{j}}) = max(K - distance(X_{\mathsf{j}}), distance(X_{\mathsf{i}}, X_{\mathsf{j}})) \tag{3.15}$$

Therefore, LOF is the ratio of the average LRD of the K neighbors of A to the LRD of A. If LOF is bigger than 1, it indicates that the sample is an outlier.

Imputation: this technique was applied to the detected outliers, as well as to the missing values. The detected outliers were replaced by the average of the previous and next non-outliers. Regarding missing values, only meteorological data were taken into account, given that traffic data has fewer missing values. The Inverse Distance Weighting (IDW) and Nearest Neighbour Interpolation (NNI) methods have been implemented to handle missing meteorological data values [83]. The idea of IDW is to predict the values for unknown points based on the values of the known points. The closer known points have greater influence compared to the farthest points. Closer points are given a higher weight, and as the time interval increases, the weight diminishes. The weighted mean of near observations is used to calculate the estimated value of z at position x with the formula illustrated below[23].

$$\hat{z}(x) = \frac{\sum_i^n w_i z_i}{\sum_i^n w_i} \tag{3.16}$$

where:

$$w_i = |x - x_i|^{-\beta} \tag{3.17}$$

and $\beta \geq 0$, and it is the inverse distance power (degree to which closer points are selected over more further points), $|\ .\ |$ is the euclidean distance.

NNI attempts to estimate the value z at point x based on observations $z_1, z_2, ..., z_n$ at locations $x_1, x_2, ..., x_n$. NNI uses the value z_i that is closest to x[24].

[23]Inverse Distance Weighting (IDW): `https://bit.ly/3CiNULG`. [Online; accessed 15-February-2023]

[24]Nearest Neighbor Interpolation: `https://bit.ly/3dVOJAH`. [Online; accessed 15-February-2023]

Feature Selection: numerous issues, such as the curse of dimensionality [84, 85] and runtime execution, are related to the presence of many features. This, in turn, can make it difficult for a model to generalise data effectively. Hence, feature selection must be implemented to select the optimum combination of datasets, allowing the model to generalise the data effectively. Aside from the reasons stated above, another reason for selecting the most optimal features is to avoid a data shortage; for example, a feature that is recorded and available for the city of Madrid, may not be available for another study area. Thus, the ability to execute analysis with a minimum number of features allows us to generalise the model, broaden the geographical dimension of the application, and reduce the execution time. Therefore, selecting the most relevant features is essential.

Many authors implement feature selection techniques in order to obtain better outcomes. For example, for predicting $PM_{2.5}$, Just et al. [35] applied recursive feature selection based on least mean absolute SHapley Additive exPlanations (SHAP) values, Shah and Mishra [41] used correlation, Xu and Ren [46] employed maximum relevance-minimum redundancy, Zheng et al. [86] used recursive feature elimination with cross-validation for air quality health index prediction, Masmoudi et al. [36] used Ensemble of Regressor Chains-guided Feature Ranking, Liu and Chen [87] applied three-stage feature selection, including Pearson's test, Mutual Information (MI) and binary grey wolf optimisation for predicting AQI. These works confirmed the advantage and importance of implementing feature selection methods.

The feature selection techniques implemented in this work are MI (Eq. 3.18) and Maximum Relevance — Minimum Redundancy (mRMR) [88, 89]. These are introduced below:

Mutual Information: it calculates the mutuality between additional datasets and the target dataset (NO_2). The formula to calculate MI is:

$$\begin{aligned} MI(x;y) = \iint P(x_i,y) log \frac{P(x_i,y)}{P(x_i)P(y)}\, dx_i\, dy \\ = H(x) - H(x|y) \end{aligned} \tag{3.18}$$

where $P(x_i, y)$ is the joint probability distribution of two variables, $P(x_i)$ and $P(y)$ are marginal distributions, $H(x)$ is the entropy for x, and $H(x|y)$ is the conditional entropy.

Maximum Relevance — Minimum Redundancy: mRMR selects the most rel-

evant features to the target also considering minimum redundancy concerning the features that have already been selected. The equation of the mRMR is the following (Eq. 3.19).

$$score_i(f) = \frac{F(f, target)}{\sum_{s \in features\ selected\ until\ i-1} |corr(f, s)|/(i-1)} \tag{3.19}$$

where i is the i-th iteration, f is the feature that is being evaluated, F is F-statistic and $corr$ is Pearson correlation.

Transformation: this technique was involved to convert the wind direction in the following ways: 1) converting it to categorical data with the following categories: north, east, south, west, southwest, northeast, southeast, northwest, and later passing through One Hot Encoder[25], or 2) converting it to u and v components using the following equations (Eq. (3.20))[26].

$$\begin{aligned} u &= ws * \cos(\theta) \\ v &= ws * \sin(\theta) \end{aligned} \tag{3.20}$$

where ws is the wind speed, θ is the wind direction using mathematical direction (mathematical direction = 270-meteorological direction of wind direction).

Another transformation was the conversion of the input data into the supervised learning dataset. Independent and dependent datasets were generated based on the defined time granularity.

Scaling: is a highly effective technique for handling differences between ranges of features. Normalisation and standardisation were implemented at different stages.

Normalisation: to normalise the input data Min-Max (0-1) normalisation was applied (Eq. 3.21).

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{3.21}$$

Standardisation: it is also called Z-score, and it was implemented with the following equations (Eq. 3.22).

[25]One Hot Encoder: `https://bit.ly/2I7wbNu`. [Online; accessed 15-February-2023]

[26]Wind: u and v Components: `https://bit.ly/2CwAUzY`. [Online; accessed 15-February-2023]

$$X' = \frac{X - \mu}{\sigma} \tag{3.22}$$

where μ is the mean and σ is the standard deviation.

Data Splitting: after preprocessing the data with the above methods, the next step is to split the dataset into the train, validation, and test sets. The division of data into the above steps will be provided for each model in the following chapters, which varies depending on the application accompanying model development.

3.5 Machine Learning Methods

This section introduces the proposed methods along with the fundamental concepts needed to understand their architecture. The main goals of this section can be generalised as follows:

- To introduce the concept of ML and the main types;
- To introduce NN, how it works. To describe the main NN approaches by focusing RNN, Convolutional Neural Network (CNN) and GCN;
- To describe with detail the architecture of the proposed methods of the current work, including ConvLSTM, BiConvLSTM, A3T-GCN.

The following subsections are focused to address the aforementioned goals.

3.5.1 Machine Learning Concept

There is no single definition of ML. Several authors have tried to define the concept of ML. According to Samuel [90][91], it is the learning process of computers based on their experience without any explicit programming. Tom Mitchell defined the concept of learning as a composition of three elements: task, performance measures, and learning experience, saying that a computer learns if its performance on a given task improves with experience[27].

Artificial intelligence and DL are frequently used interchangeably with the term ML. To clarify these terms and avoid confusion, Figure 3.15 depicts the relationship between them.

[27]Machine Learning: `https://bit.ly/3C1u1Yo`. [Online; accessed 15-February-2023]

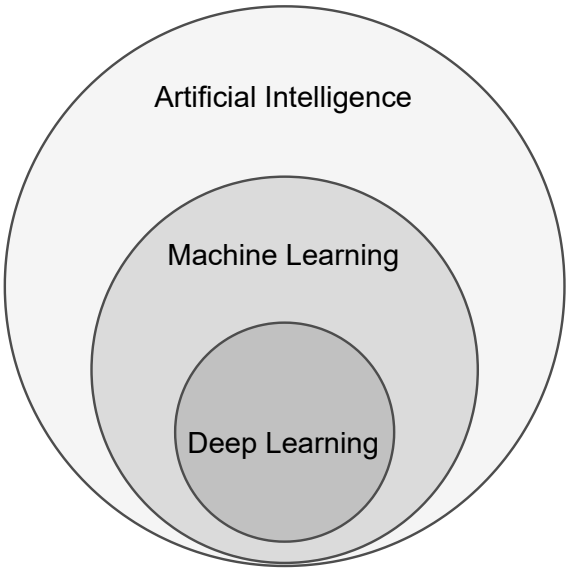


Figure 3.15: Artificial Intelligence, Machine Learning and Deep Learning.

Artificial Intelligence: refers to a computer program's capacity to behave similarly to a human brain, i.e., to make intelligent machines, which is realised through the study, interpretation and adaptation of data. It encompasses ML and DL.

Deep Learning: is a subset of ML which requires minimal manual human intervention. DL is a complex, multi-layered neural network, which employs a huge amount of structured as well as unstructured data.

Returning to the concept of ML, based on human supervision of the learning process, ML applications are classified into four categories: supervised, unsupervised, semisupervised, and reinforcement learning [15] (Figure 3.16).

Supervised Learning: the learners receive training data with labelled samples aiming to identify unknown labels in the testing data. Depending on the learning task, two types of tasks are differentiated: regression and classification. Regression refers to the task when the label is real numbers, and classification refers to the task when the label is a finite set of classes. The following are the supervised learning approaches: K-Nearest Neighbor (KNN), SVM, NN.

Unsupervised Learning: includes unlabeled data. Unsupervised learning algorithms are clustering, anomaly detection and novelty detection, visualisation and dimensionality detection, and association rule learning.

Semisupervised Learning: is the approach that deals with partially known labels,

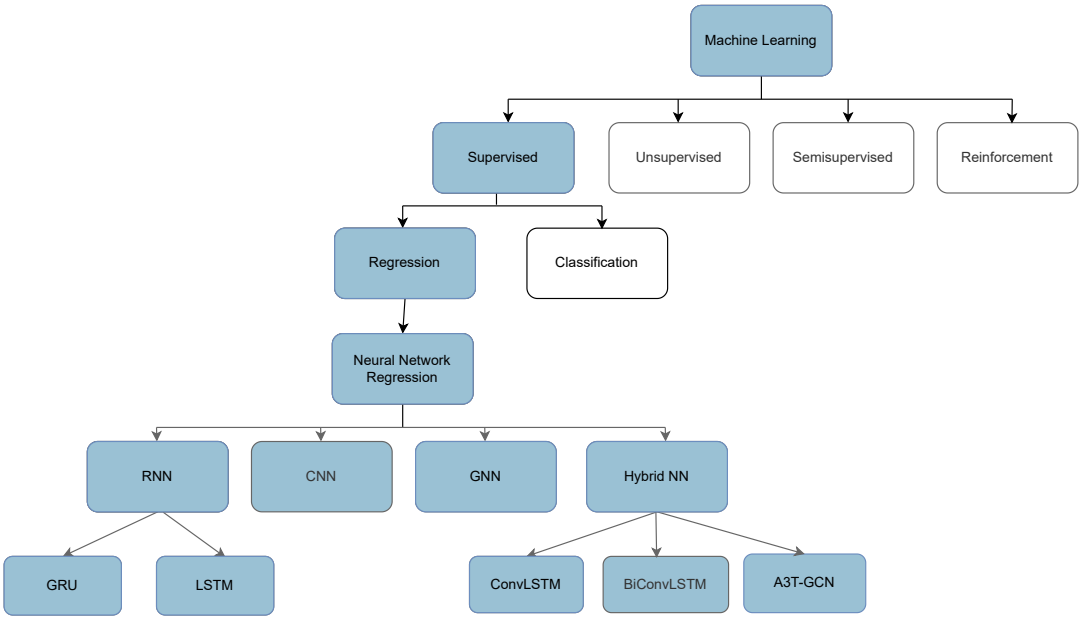


Figure 3.16: The types of Machine Learning.

i.e., it is a combination of supervised and unsupervised learning.

Reinforcement Learning: the logic behind this learning is based on the actions of the agent which gets rewards or penalties during the learning process. Based on these actions the policy or strategy is being defined.

3.5.2 Artificial Neural Network

The development of Artificial Neural Network (ANN) is connected with these key stages: the demonstration of how biological neurons work, the invention of the perceptron, and the discovery of backpropagation.

The first record about ANN was mentioned by McCulloch and Pitts [92] in 1943. Their aim was to present a simplified version of the work of biological neurones, and how they are connected. This is considered the design of the first ANN.

Afterwards, Rosenblatt [93] created the first perceptron. He was inspired by Hebb's rule (when a cell activates another cell, the connection between these two cells becomes stronger). The perceptron was programmed with two layers, the input layer and the output layer, each input connection has a weight. The results are the weighted sum of the inputs passed through the step function. Based on the error the network was improved by reinforcing the connection weights. However,

the perceptron had weaknesses, i.e., not being able to learn the non-linear pattern, which was solved later by stacking multiple perceptrons, known as MLP. It is composed of input, hidden and output layers.

The next important invention was the backpropagation training algorithm [94, 95] based on which it was possible to train MLP. The backpropagation learning algorithm calculates the gradient error by passing through forward and backward networks, i.e., by tweaking the weights and bias, it reduces the network error.

With the rise of DL, the number of ANN applications has grown tremendously. Below are presented the most common DL networks: CNN, RNN and GNN.

CNN: implementing MLP has limitations. With a parallel network increase, the parameters grow faster, which causes difficulties with model optimisation. The invention of CNN solved the above problem by reducing the number of parameters without losing too much information that affects the quality of the model [96]. CNN requires grid-based input aiming to learn spatial feature hierarchies, from low-level to high-level patterns. The neurons are only connected to a small region of the previous layer. The essential component of CNN is a convolution layer, which computes the output of neurons computing a dot product between their weights and a small region (receptive field) they are connected. The result goes through an activation function and then generally follows the pooling and fully connected layers. The convolutional layer's parameters consist of a set of learnable filters (convolution kernels). During the forward pass, by sliding (convolving) each filter across the width and height of the input and computing dot products between the filter's entries and the input at any position, a 2D activation map (feature map) is generated that gives the responses of that filter at every spatial position. Stacking all activation maps along the depth dimension an output is created.

RNN: works with sequential data: to label, classify or generate sequences. It uses previous outputs as inputs, i.e., the input consists of two elements: the present and the recent past. It has a short-term memory. Figure 3.17 shows the architecture of the RNN.

The standard RNN has weaknesses: vanishing and exploding gradient[28] [97]. Sometimes the gradients get smaller and smaller, closer to zero, and as a result, the weights barely update, causing the training data to never converge. An

[28]Why are deep neural networks hard to train?: `https://bit.ly/3HjQ4NY`. [Online; accessed 15-February-2023]

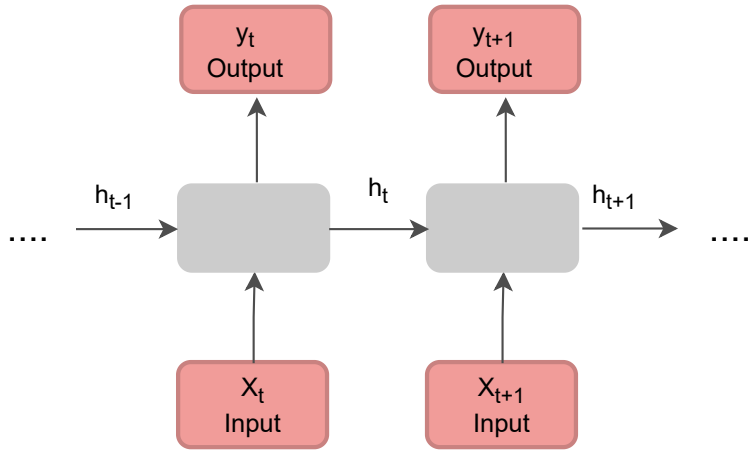


Figure 3.17: The architecture of Recurrent Neural Network.

explosive gradient is a reverse situation where the gradient gets bigger and bigger and the algorithm diverges, making the network unstable. To overcome these issues the following RNN methods were constructed: GRU and LSTM.

GRU: is a type of RNN introduced by Cho et al. [98]. It consists of two gates: reset and update gates, which determine what information is stored. Figure 3.18 presents the architecture of the GRU. It can be defined with the following equations:

$$z_t = \sigma(W^{(z)}x_t + U^{(z)}h_{t-1}) \tag{3.23}$$

$$r_t = \sigma(W^{(r)}x_t + U^{(r)}h_{t-1}) \tag{3.24}$$

$$h'_t = \tanh\left(Wx_t + r_t \circ Uh_{t-1}\right) \tag{3.25}$$

$$h_t = z_t \circ h_{t-1} + (1 - z_t) \circ h'_t \tag{3.26}$$

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 $\circ$ is the Hadamard product.

LSTM: extends the memory of RNN enabling the network to remember inputs over a long period of time. It was invented by Hochreiter and Schmidhuber [99].

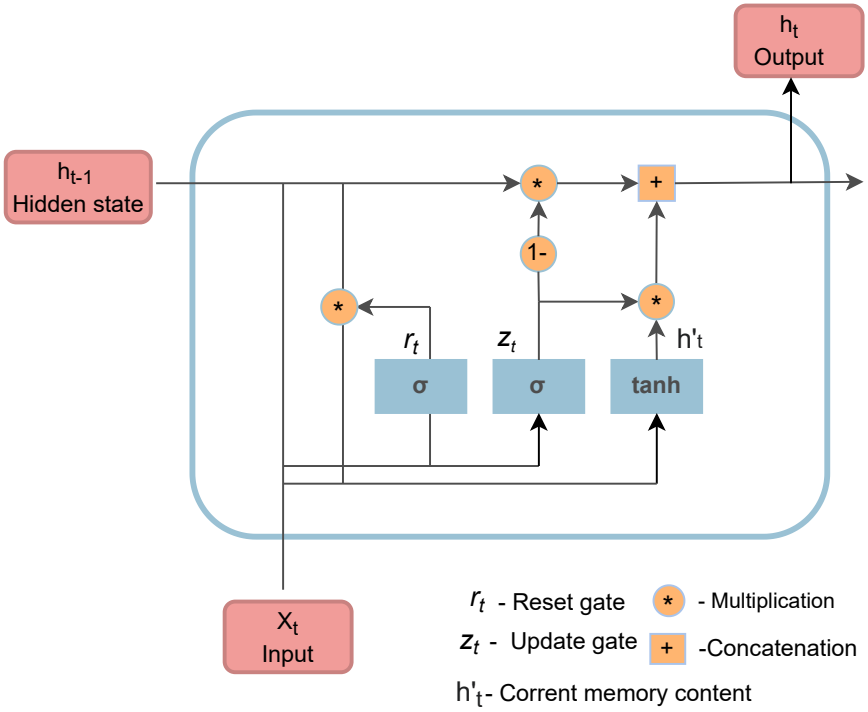


Figure 3.18: The architecture of Gated Recurrent Unit.

LSTM contains information in memory and can control that information. It consists of three gates: input, forget and output gate. These gates help to decide which part of the information to keep, what to remove and what to let affect the current output. Figure 3.19 presents the architecture of the LSTM, and it is defined by the equations below.

$$i_t = \sigma(W_i^X X_t + W_i^h h_{t-1} + b_i) \tag{3.27}$$

$$f_t = \sigma(W_f^X X_t + W_{f^h} h_{t-1} + b_f) \tag{3.28}$$

$$o_t = \sigma(W_o^X X_t + W_o^h h_{t-1} + b_o) \tag{3.29}$$

$$C_t = f_t \circ C_{t-1} + i_t \circ tanh(W_c^X X_t + W_c^h h_{t-1} + b_C) \tag{3.30}$$

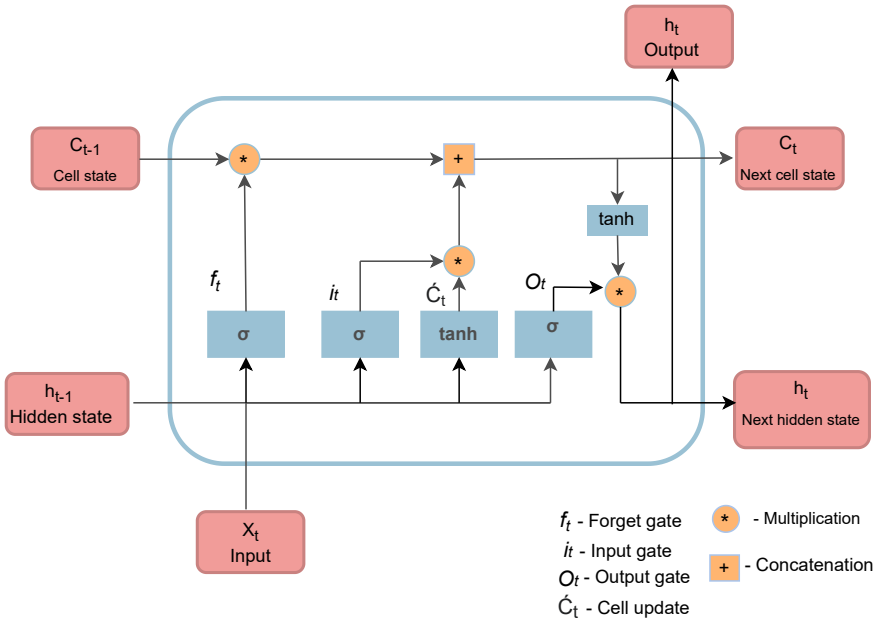


Figure 3.19: The architecture of Long Short-Term Memory.

$$h_t = o_t \circ tanh(C_t) \tag{3.31}$$

where i_t is the input gate, f_t is the forget gate, and o_t is the output gate, W is the weight matrix, X_t is the current input data, h_{t-1} is previous hidden output, C_t is the cell state, $\circ$ is Hadamard product.

GNN: it is a type of NN which works on graph-structured data. In the case of CNN, convolutions depend on the positions of the instances (positions of the pixels in the case of images), while there are many fields where the dataset does not have a grid-based structure with a fixed order. These datasets can be processed by GNN which can work with unfixed node ordering.

The graph can be designated as $G = (V, E, A)$, where V is the set of nodes, E is the set of edges and $A \in R^{N \times N}$ (N is the number of nodes) is the adjacency matrix which presents the connectivity of the edge of two nodes [100].

Regarding directionality to edges, the graphs are categorised into directed and undirected. In the case of the directed edges, the edge has a source node and a destination node, i.e., the information flows from the source to the destination node. In the case of the undirected edges, there is no concept of source or destination nodes, and information flows in both directions.

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 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).

3.5.3 Proposed Methods

This section gives a comprehensive description of the implemented ML methods. The following approaches, including ConvLSTM, BiConvLSTM and A3T-GCN, were proposed and developed to conduct spatiotemporal prediction.

ConvLSTM: is composed of a CNN and LSTM network (Figure 3.20). It consists of two components: the encoding network and the forecasting network. The encoding LSTM compresses the entire input sequence into a hidden state tensor, which the forecasting LSTM subsequently unfolds to generate the final prediction [101].

$$\begin{aligned}\tilde{X}_{t+1}, ...\tilde{X}_{t+K} = \arg\max_{X_{t+1},...X_{t+K}} p(X_{t+1}, ...X_{t+K}|\hat{X}_{t-J+1}, \hat{X}_{t-J+2}, ...\hat{X}_t) \approx \\ \arg\max_{X_{t+1},...X_{t+K}} p(X_{t+1}, ...X_{t+K}|f_{encoding}(\hat{X}_{t-J+1}, \hat{X}_{t-J+2}, ...\hat{X}_t) \approx \\ g_{forecasting}(f_{encoding}(\hat{X}_{t-J+1}, \hat{X}_{t-J+2}, ...\hat{X}_t))\end{aligned} \tag{3.32}$$

Its architecture allows for capturing spatiotemporal information. The convolution structures are used at both the input-to-state and the state-to-state transitions. A ConvLSTM with a larger transitional kernel captures faster motions while one with a smaller kernel captures slower motions. It is worth mentioning that ConvLSTM differs from CNN+LSTM in that the latter uses CNN first, followed by a separate LSTM unit, whereas in ConvLSTM, the LSTM's internal matrix multiplication is converted to convolution operations. The architecture of ConvLSTM is defined by the equations below [101, 102].

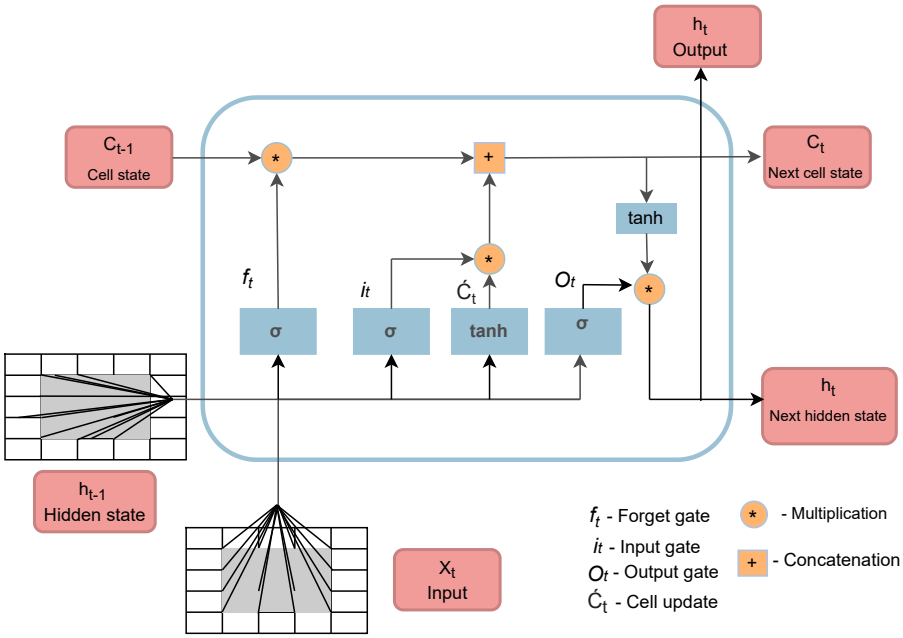


Figure 3.20: The architecture of Convolutional Long Short-Term Memory.

$$i_t = \sigma(W_i^X * X_t + W_i^h * h_{t-1} + b_i) \tag{3.33}$$

$$f_t = \sigma(W_f^X * X_t + W_{f^h} * h_{t-1} + b_f) \tag{3.34}$$

$$o_t = \sigma(W_o^X * X_t + W_o^h * h_{t-1} + b_o) \tag{3.35}$$

$$C_t = f_t \circ C_{t-1} + i_t \circ tanh(W_c^X * X_t + W_c^h * h_{t-1} + b_C) \tag{3.36}$$

$$h_t = o_t \circ tanh(C_t) \tag{3.37}$$

where i_t is the input gate, f_t is the forget gate, and o_t is the output gate (these gates control the flow of information through the cell), W is the weight matrix in the forward ConvLSTM cell, X_t is the current input data, h_{t-1} is the previous hidden output, C_t is the cell state, "$*$" represents the convolution operation and "$\circ$" represents the Hadamard product. The input of a ConvLSTM is a 5D tensor with shape (samples, time steps, channels, rows, columns). The parameters of

ConvLSTM can be found with the following link[29], including $filters$, $kernel\_size$, $padding$, $data\_format$, and $recurrent\_activation$.

BiConvLSTM: is an upgraded ConvLSTM with two sets of hidden and cell states for forward and backward time sequences. As a result, BiConvLSTM can obtain a deeper understanding by accessing long-range context in both directions. Figure 3.21 shows the architecture of BiConvLSTM unit. The h_f, c_f and h_b, c_b are the sets for forward and backward passes, respectively. Two sets are stacked and sent through the convolution layer, with the output being passed as input to the next BiConvLSTM cell. There are numerous ways to combine the two sets before delivering them to the convolution layer, including summing, computing the average, multiplying, or concatenating. The aforementioned are only a few parameters that must be defined during the tuning phase in order to complete the model's architecture. Below is the mathematical expression of BiConvLSTM network [102].

$$Y_t = \tanh(W_y^{Hf} * H_t^f + W_y^{Hb} * H_{t-1}^b) \tag{3.38}$$

where H^f is hidden state from forward ConvLSTM unit, H^b is hidden state from backward ConvLSTM unit, and Y_t is the final output.

A3T-GCN: the next proposed method is A3T-GCN, which architecture is based on graph theory. The graph considered in the scope of the current work is an undirected weighted graph, and the learning task is a node regression since the main objective of the current work is to predict the concentration of NO_2 in each station in a given time interval.

A3T-GCN model is the combination of GCN, GRU (stated in Section 3.5.2) and the attention methods (Figure 3.22) [103]. The GRU and attention mechanisms are responsible for temporal aggregation, and GCN deals with spatial aggregation.

GCN: there are two types of GCN: Spatial GCN and Spectral GCN [104]. 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

[29] tf.keras.layers.ConvLSTM1D: `https://bit.ly/3SG7uH7`. [Online; accessed 15-February-2023]

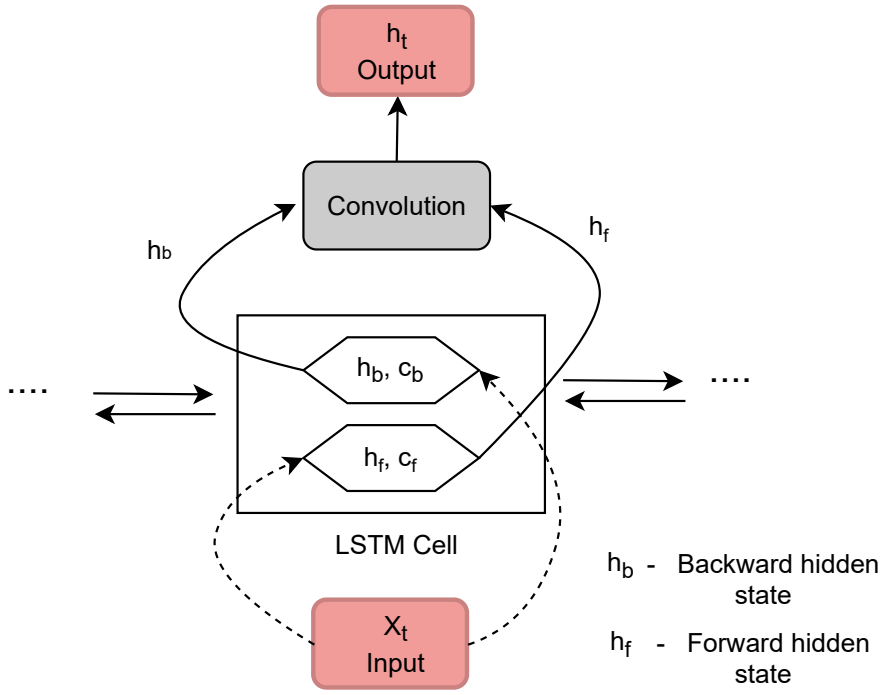


Figure 3.21: The architecture of Bidirectional Convolutional Long Short-Term Memory.

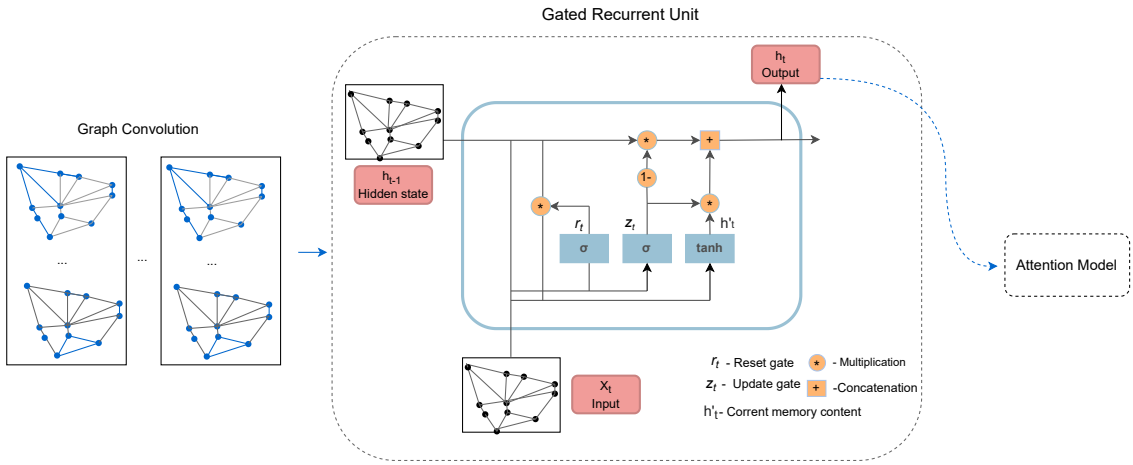


Figure 3.22: The architecture of Attention Temporal Graph Convolutional Network.

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_θ with signal x in the Fourier domain [105].

$$g_\theta * x = U g_\theta U^T x \tag{3.39}$$

where θ is a model parameter, U is the eigenvector of the normalised Laplacian matrix L (Eq. 3.40).

$$L = I_N - D^{-1/2} A D^{-1/2} = U \lambda U^T \tag{3.40}$$

where $I_N \in R^{N \times N}$ is the identity matrix, $D \in R^{N \times N}$ is the diagonal degree matrix and λ is the diagonal matrix of the eigenvalues of the Laplacian matrix, and $U^T x$ is the graph Fourier transform of x. These operations require intense computations, i.e., multiplication with the eigenvector matrix U can be expensive for large graphs. To overcome this problem, Chebyshev polynomials $T_k(x)$ with K order were employed (Eq. 3.45).

$$g_{\theta'}(\lambda) \approx \sum_{k=0}^{K} {\theta'}_k T_k(\hat{\lambda}) \tag{3.41}$$

where $\hat{\lambda} = \frac{2\lambda}{\lambda_{max}} - I_n$. $\theta' \in R^K$ is a vector of Chebyshev coefficients.

$$g_{\theta'} * x \approx \sum_{k=0}^{K} {\theta'}_k T_k(\hat{L}) x \tag{3.42}$$

where $\hat{L} = \frac{2L}{\lambda_{max}} - I_n$. Since $(U \lambda U^T)^k = U \lambda^k U^T$, the equation depends only on nodes that are at maximum K steps away from the central node (K^{th}-order neighbourhood). Furthermore, focusing only first order, i.e., K = 1, and by approximating $\lambda_{max} \approx 2$, it can be formulated as follows:

$$g_{\theta'} * x \approx {\theta'}_0 x + {\theta'}_1 (L - I_N) x = {\theta'}_0 x + {\theta'}_1 D^{-1/2} A D^{-1/2} x \tag{3.43}$$

To reduce the number of free parameters and to avoid overfitting, GCN assumes $\theta = \theta_0 = -\theta_1$, and the equation becomes

$$g_\theta * x \approx \theta_1 (I_N + D^{-1/2} A D^{-1/2}) x \tag{3.44}$$

Then it was introduced the renormalisation trick: $I_n + D^{-1/2}AD^{-1/2} \rightarrow D^{-1/2}AD^{-1/2}$, with $\hat{A} = A + I_n$ and $\hat{D}_{ii} = \sum_j \hat{A}_{ij}$. Then the final convolved signal can be defined as follows:

$$Z = \hat{D}^{-1/2}\hat{A}\hat{D}^{-1/2}X\Theta \tag{3.45}$$

where $\Theta \in R^{C \times F}$ is a matrix of filter parameters and $Z \in R^{N \times F}$ is the convolved signal matrix.

Attention: the attention model focuses on a few relevant things in the complex input while ignoring others in networks. Bahdanau et al. [106] 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.

$$c_i = \sum_{j=1}^{T_x} a_{ij} h_j \tag{3.46}$$

where c_i is the context vector, a_{ij} is the weights and h_j is the hidden state. The weights, a_{ij}, can be calculated with the following equations, by applying softmax to normalise alignment scores. Alignment scores show how well the elements of the input sequence and the current output match each other.

$$a_{ij} = \frac{exp(e_{ij})}{\sum_{k=1}^{T_x} exp(e_{ik})} \tag{3.47}$$

$$e_{ij} = a(s_{i-1}, h_j) \tag{3.48}$$

where e_{ij} is alignment scores (the output score of a feedforward neural network), and s_{i-1} is the previous decoder output.

There are several categories of the attention mechanism. Soft attention is the type used in this study (in the case of soft attention, the context vector is the weighted sum of the encoder hidden states, while in the case of hard attention, instead of a weighted average of all hidden states, a single hidden state is chosen based on attention scores).

3.6 Summary

This chapter is devoted to the introduction and description of the proposed methodology and employed materials. First of all, the study area and the prediction target were defined and introduced, on the basis of which the datasets were introduced. The datasets employed in this work are air quality (NO_2), meteorological data (UV irradiance, wind speed, wind direction, temperature, relative humidity, barometric pressure, solar irradiance, and precipitation) and traffic data (intensity, occupancy time, load and average traffic speed) from the period of January-June 2019 and January-June 2020, and the location of air quality and meteorological monitoring stations and traffic measurement points of the city of Madrid. Afterwards, the steps of the proposed methodology were implemented, including *Data Preparation*, *Exploratory Data Analysis*, *Feature Engineering*, and *ML Model Generation*.

Data Preparation: refers to the employed datasets integration in spatiotemporal dimensions using ArcGIS Pro with the ArcPy package. The result of integration is the input data $X \in R^{s \times 340 \times f}$, where s is the number of samples: 4,344 and 4,368 for January-June 2019 and January-June 2020, respectively, 340 is the number of the cells that make up the entire grid covering the defined area of the city of Madrid, and f is the number of features equal to 13 ($X \in R^{4,344 \times 340 \times 13}$ for January-June 2019 and $X \in R^{4,368 \times 340 \times 13}$ for January-June 2020).

Exploratory Data Analysis: refers to the detection of relationships and correlations that exist between features, to achieve which different analyses have been used. The result showed, from all the observations and analyses, the most correlated feature with NO_2 is wind speed. The features that must be excluded in the further analysis are UV and precipitation.

Feature Engineering: includes techniques implemented to the datasets in order to preprocess raw data. The following techniques were applied: *Handling Outliers* (overview of summary statistics, iForest, and LOF), *Imputation* (IDW and NNI), *Feature Selection* (MI and mRMR), *Transformation* (conversion of wind direction and conversion of the input data into the supervised learning dataset), *Scaling* (normalisation and standardisation) and *Data Splitting* (train, validation and test sets).

ML Model Generation: refers to the detailed description of the proposed meth-

ods, including ConvLSTM, BiConvLSTM and A3T-GCN.

Chapter 4

Convolutional Long Short-Term Memory Network[1]

To predict NO_2 within the specified region is challenging due to the complexity of NO_2 formation. In order to consider the most relevant factors in the spatiotemporal dimensions and carry out the prediction at a continuous location within the defined zone, the ConvLSTM algorithm was developed and implemented.

Furthermore, in addition to selecting a model, it is also critical to consider factors that can directly or indirectly affect air quality. The selection and consideration of those factors can improve prediction accuracy. One of these factors is the lockdowns imposed due to the COVID-19 pandemic. To combat the COVID-19 epidemic, all countries adopted rigorous traffic restrictions and self-quarantine measures [107], resulting in a reduction in air pollution [108]. This was especially evident in Madrid, where, due to COVID-19 restrictions, the concentration of NO_2 dropped to 62% [109].

Based on the aforementioned, the main contributions of this chapter are highlighted as follows:

- We conducted spatiotemporal prediction of NO_2 using a grid-based approach: ConvLSTM;

[1]The part of this chapter previously appeared as an article in the Journal of IJCIA. The original citation is as follows: Iskandaryan, Ditsuhi, Francisco Ramos, and Sergio Trilles. ”Comparison of Nitrogen Dioxide Predictions During a Pandemic and Non-pandemic Scenario in the City of Madrid using a Convolutional LSTM Network.” *International Journal of Computational Intelligence and Applications* 21, no. 02 (2022): 2250014.

- We performed the predictive analysis for pandemic (January-June 2020) and non-pandemic (January-June 2019) periods to observe and identify the effects of the restrictions implemented to suppress the advance of COVID-19;
- We implemented the feature selection technique: MI to select the most relevant combination of the features and performed the predictive analysis for the selected combination;
- We performed the predictive analysis in different time intervals, including 1-hour, 12-hour, 24-hour and 48-hour;
- We compared the proposed method with the reference method (LSTM).

The sections below explain in detail the procedure that leads to the achievement of allocated contributions. They focus on the experimental analysis workflow (*Experimental Analysis*), and the results obtained with subsequent discussions (*Results and Discussion*).

4.1 Experimental Analysis

This section presents a detailed description of the experimental analysis. The workflow of the analysis includes data preprocessing and model development. In particular, it consists of the following steps: *Data Preparation*, *Feature Engineering*, and *Modeling*, as shown in Figure 4.1.

Data Preparation: the process of data preparation was previously described in Section 3.2. We used NO_2 and meteorological data to build and evaluate ConvLSTM.

Feature Engineering: this process with all substeps (*Handling Outliers*, *Imputation*, *Feature Selection*, *Transformation*, *Scaling* and *Data Splitting*) was also considered in this analysis (described in Section 3.4). Below is a description of each technique.

Handling Outliers: the outliers were detected based on the overview of summary statistics of the datasets (stated Section 3.4).

Imputation: as previously stated, there are twenty-four air pollution control stations and twenty-six meteorological stations, implying that around 8% of the 340 cells have data. IDW method was used to fill in missing meteorological data

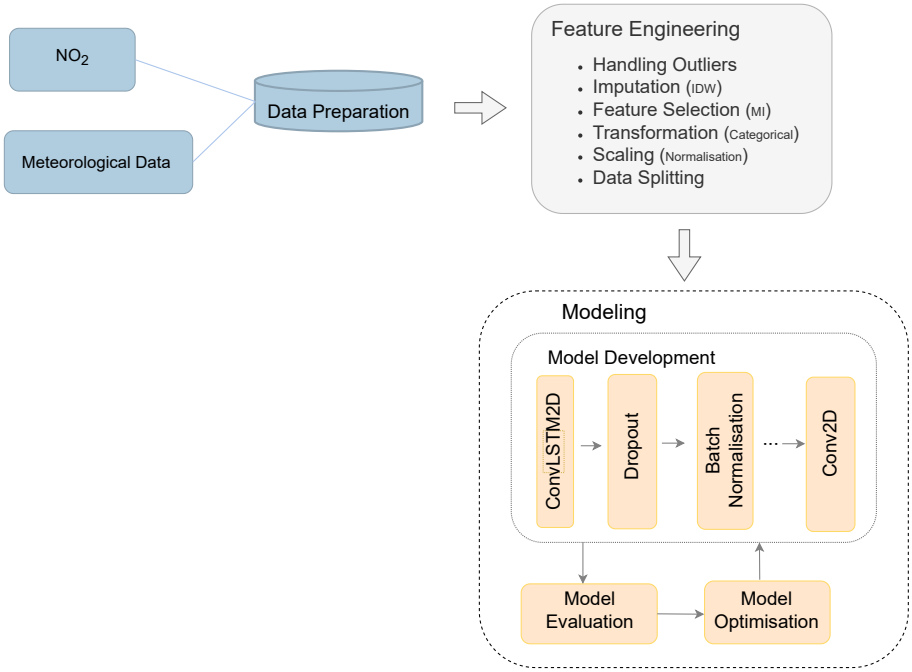


Figure 4.1: The workflow of the Convolutional Long Short-Term Memory-based nitrogen dioxide predictive analysis.

based on the fact that meteorological data do not change dramatically within space [83].

Feature Selection: this analysis comprised only nine features, including NO_2 and meteorological data. UV and precipitation were excluded from further analysis due to the lack of a UV record during the pandemic period and nearly all of the precipitation data being zero. Additionally, MI was chosen and implemented on the remaining features as a feature selection approach [88].

Figure 4.2 shows the scores of five additional datasets based on MI. Wind direction is not taken into account in the MI calculation, considering the fact that it is circular data and based on their properties must be analysed differently. There are many studies devoted to the analysis of circular data [110–112]. In our work, we transformed them before being used in further analysis.

Since the wind speed has a higher score compared to other variables (Figure 4.2), the wind direction was chosen for further analysis alongside the wind speed, considering their strong connectivity.

Transformation: this phase involves converting wind direction into categorical data (north, east, south, west, southwest, northeast, southeast, and northwest),

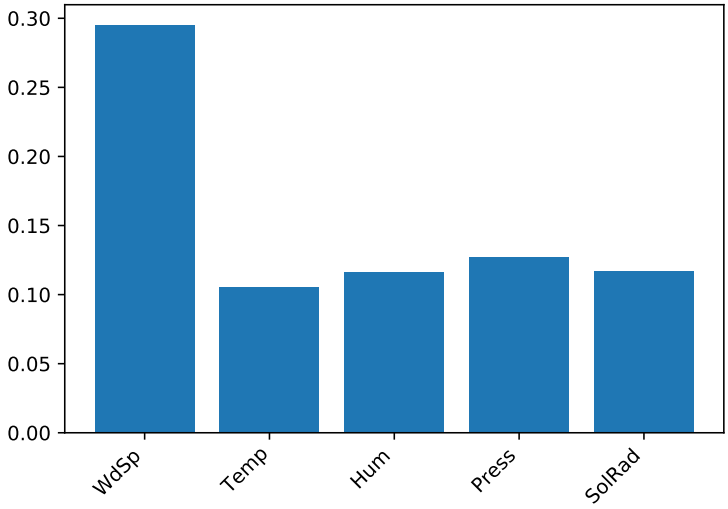


Figure 4.2: The feature importance scores based on Mutual Information.

and implementing One Hot Encoder. Another transformation refers to the generation of independent and dependent datasets based on the time granularity (to predict NO_2 in t' hours based on the data for the previous 24 hours, where $t' \in \{1, 12, 24, 48\}$).

Scaling: in this analysis, the input data went through to Min-Max (0-1) normalisation (Eq. 3.21).

Data splitting: splitting procedure involves dividing the dataset of each period: pandemic (January-June 2020) and non-pandemic (January-June 2019), into training (60%), validation (20%) and testing (20%) sets.

Modeling: it consists of three substeps: *Model Development*, *Model Evaluation* and *Model Optimisation*.

Model Development: the model architecture consists of the following layers, including $ConvLSTM2D$[2], $Dropout$[3], and $Batch\ Normalisation$[4], which is finalised with a $Conv2D$ layer[5].

$ConvLSTM2D$: or 2D Convolutional LSTM. It combines LSTM with 2D convolutions (stated in Section 3.5.3) [101].

[2]ConvLSTM2D layer: `https://bit.ly/3VXDZTC`. [Online; accessed 15-February-2023]

[3]Dropout layer: `https://bit.ly/3f1nk0W`. [Online; accessed 15-February-2023]

[4]BatchNormalisation layer: `https://bit.ly/3TtkFvS`. [Online; accessed 15-February-2023]

[5]Conv2D layer: `https://bit.ly/3N1VwG4`. [Online; accessed 15-February-2023]

$Dropout$: works as a regularisation technique, which randomly and temporarily modifies the network by excluding or dropping out a defined percentage of the neurons. Afterwards, the modified network goes through forward and backward propagation. Repeating this procedure many times helps prevent overfitting and creates a more robust model [113, 114]. Below is the mathematical description of the feed-forward operation including dropout:

$$r_j^{(l)} \sim Bernoulli(p) \tag{4.1}$$

$$\tilde{y}^{(l)} = r^{(l)} * y^{(l)} \tag{4.2}$$

$$z_i^{(l+1)} = w_i^{(l+1)} \tilde{y}^{(l)} + b_i^{(l+1)} \tag{4.3}$$

$$y_i^{(l+1)} = f(z_i^{(l+1)}) \tag{4.4}$$

where, $r^{(l)}$ is a vector of Bernoulli random variables, $\tilde{y}^{(l)}$ is a thinned output, $z^{(l)}$ is the vector of inputs into layer l, $y^{(l)}$ is the vector of outputs from layer l, $w^{(l)}$ are the weights, and $b^{(l)}$ are the biases. Essentially, the Eq. 4.1 generates a dropout mask, and then that mask was used to disconnect some neurons (Eq. 4.2), after which the multiplication of the weights and adding the bias were implemented (Eq. 4.3) and finally, using the activation function the output was calculated (Eq. 4.4).

$Batch\ Normalisation$: normalises the layer inputs leading to stabilisation and acceleration of the training procedure of Deep Neural Network (DNN). This technique helps to combat the problem of internal covariate shift, which occurs due to the change in the distribution of the input of each layer, causing a slow training process. The following equations present the essence of batch normalisation [115, 116]:

$$\mu_B = \frac{1}{m} \sum_{i=1}^{m} x_i \tag{4.5}$$

$$\sigma_B^2 = \frac{1}{m} \sum_{i=1}^{m} (x_i - \mu_B)^2 \tag{4.6}$$

$$\hat{x} = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} \tag{4.7}$$

$$y_i = \gamma \hat{x}_i + \beta \equiv BN_{\gamma,\beta}(x_i) \tag{4.8}$$

where μ_B is the mini-batch mean, σ_B^2 is the mini-batch variance,$\hat{x}$ is the normalised values, γ and β are learnable parameters that scale and shift the normalised values, ϵ is the smoothing term to prevent a division by a zero value.

$Conv2D$: or 2D convolution layer. This creates a convolution kernel that is convolved with the layer's input to create an output tensor (stated in Section 3.5.2) [96, 117].

Model Evaluation: the final model was evaluated using RMSE. The results are presented in the next section.

Model Optimisation: is one of the essential processes and a core component of ML. It enables choosing the ideal model architecture for a particular database. By tweaking and tuning model configurations or hyperparameters, ML optimisation seeks to increase the accuracy of a model and minimise its loss function. At this point, it is important to mention the difference between hyperparameters and parameters.

$Parameters$: are internal configuration variables that are learned from the data during the training. Typically, model training starts with parameter initialisation to some values. These values are then updated using an optimisation algorithm. The parameters determined through training are used to construct the final model (the examples of parameters are weights and bias).

$Hyperparameters$: are explicitly defined by ML engineer to control the learning process. They are external to the model because the model cannot change their values during training. The algorithm uses hyperparameters when it is learning, but they are not included in the final model (examples of hyperparameters are learning rate and kernel size).

Returning back to optimisation, Figure 4.3 shows the common ML optimisation techniques, including Grid Search, Random Search, Bayesian, Gradient-based and Evolutionary Optimisation [118].

The technique we used in this work is Grid Search. It exhaustively considers all hyperparameter combinations specified manually with the aim of finding the optimal combination. It was carried out using $GridSearchCV$[6], which tests all combinations of the values passed in the dictionary and evaluates the model for each

[6]GridSearchCV: `https://bit.ly/2QBg3Hy`. [Online; accessed 15-February-2023]

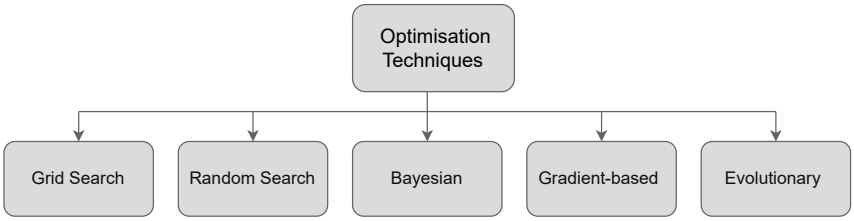


Figure 4.3: Machine Learning optimisation techniques.

combination by applying the Cross-Validation method. Based on the accuracy obtained for each combination of hyperparameters, it enables selecting the combination with the best performance. To improve and accelerate the Grid Search process, the following techniques were included: $Blocking\ Time\ Series\ Split$ and $Early\ Stopping\ callback$.

$Blocking\ Time\ Series\ Split$: was chosen instead of cross-validation because it considers the time series aspect and avoids leakage from one set to another. It works by adding margins at two positions. The first is between the training and validation folds, the second is between the folds used in each iteration. This structure helps prevent the lag value model from being reused (as an estimate/response) and from memorising patterns from one iteration to the next.

$Early\ Stopping\ callback$[7]: is a type of callback (callback is a set of functions applied at specified stages of the training procedure to help control the learning process), which monitors model performance for each epoch on the validation set during training and stops training once the validation error stops decreasing to prevent overfitting.

$GridSearchCV$ was applied to the sampled dataset, which was generated by sampling the data every six hours, to reduce computing time for parameter optimisation. The hyperparameters that went through the optimisation process are the $number\ of\ filters$, $kernel\ size$, $dropout\ rate$, $optimiser$ and $kernel\ initialiser$. The results are presented in Table 4.1, along with the options that were tested, and the option that was finally selected is in bold. It is worth mentioning that the testing options were chosen considering common approaches from different works in the domain [119], [120], [121]. Below is a brief description of each hyperparameter included in this work.

$Number\ of\ filters$: filters are introduced in Section 3.5.2. We tuned the hyper-

[7]Early Stopping callback: `https://bit.ly/3TQtyQc`. [Online; accessed 15-February-2023]

Table 4.1: Hyperparameter optimisation with GridSearchCV.

Hyperparameters	**Options**
Number of Filters	8, **16**, 32
Kernel Size	(3,3), (5,5), (7,7), **(9, 9)**
Dropout Rate	**0.2**, 0.3, 0.5
Optimiser	RMSprop, **Adam**
Kernel Initialiser	uniform, normal, glorot_normal, **glorot_uniform**

parameter by testing the model with 8, 16 and 32 filters. It turned out that the performance of the model with 16 filters is superior to that of the other filters.

$Kernel\ size$: in this work, we tested the model with 3×3, 5×5, 7×7, and 9×9 kernels. The model performance with 9×9 kernels was found to be superior to that of the other kernels.

$Dropout\ rate$: varies from 0.0 to 1.0, where 0.0 means no outputs from the layer and 1.0 means no dropout. The model was evaluated by trying 0.2, 0.3 and 0.5 dropout rates. The results with a 0.2 rate outperformed the other two options.

$Optimiser$: helps improve accuracy and reduce overall loss by adapting the attributes of a neural network (e.g., weights, learning rate). Examples of optimisers are Adagrad, Adadelta, Adam, Momentum, and RMSProp. We implemented the model using Adam and RMSProp. The result showed that the implementation with Adam is superior to the one with RMSProp. Below is a detailed explanation of Adam optimiser.

$Adam$ (Adaptive Moment Estimation) is a stochastic gradient descent method introduced by Kingma and Ba, which computes adaptive learning rates for each parameter. It stores the decaying average of the past gradients and of the past squared gradients. Below are the mathematical definition of Adam optimiser (Eq. 4.9 - 4.13):

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \tag{4.9}$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \tag{4.10}$$

$$\hat{m}_t = \frac{m_t}{(1-\beta_1^t)} \tag{4.11}$$

$$\hat{v}_t = \frac{v_t}{(1-\beta_2^t)} \tag{4.12}$$

$$\theta_t = \theta_{t-1} - \alpha \frac{\hat{m}_t}{(\sqrt{\hat{v}_t} + \epsilon)} \tag{4.13}$$

where g_t is a gradient, which is equal to $\delta_\theta f_t(\theta_{t-1})$, α is stepsize or learning rate, $\beta_1, \beta_2 \in [0, 1)$ are exponential decay rates, θ_0 is initial parameter vector, $m_0 = 0$ is initialised first moment vector, $v_0 = 0$ is initialised second moment vector, m_t is the estimate of the first moment of the gradients, v_t is the estimate of the second moment of the gradients, $t = 0$ is initialised timestep, ϵ is the smoothing term to prevent a division by a zero value.

$Kernel\ initialiser$: is a strategy to assign the weights to small random values as a starting point for model optimisation. Weight initialisation is intended to prevent layer activation outputs from exploding or vanishing gradients, which, in turn, prevents the network from converging too slowly. There are different weight initialisation techniques, such as Zeros, Orthogonal, Uniform and Normal. The techniques applied in this work are Uniform, Normal, Glorot Normal and Glorot Uniform. Glorot Uniform is the one which implementation yields better model performance. Below is a detailed description of Glorot Uniform.

$Glorot\ Uniform$ (Xavier uniform initialiser) suggested by Glorot and Bengio. In this case, the biases are initialised as zero and the weights are initialised from the following distribution:

$$W_{ij} \sim U\left[-\frac{1}{\sqrt{n}}, \frac{1}{\sqrt{n}}\right] \tag{4.14}$$

where $U[-a, a]$ is the uniform distribution with the interval $(-a, a)$ and n is the size of the previous layer (the number of columns of W).

4.2 Results and Discussion

Following parameter optimisation, the finalised model was implemented in two scenarios: a) including all features; and b) including only selected features (NO_2,

wind speed, and wind direction). The executed results are shown in Table 4.2 (the best performances are in bold).

Table 4.2: Root Mean Square Error (μg/m^3) of Convolutional Long Short-Term Memory and Long Short-Term Memory for the periods January-June 2019 (non-pandemic) and January-June 2020 (pandemic) in terms of features combination and time granularities.

		All Features (First Scenario)		Selected Features (Second Scenario)	
Method	**Hours**	**Non-pandemic period (Jan-Jun 2019)**	**Pandemic period (Jan-Jun 2020)**	**Non-pandemic period (Jan-Jun 2019)**	**Pandemic period (Jan-Jun 2020)**
ConvLSTM	1	**13.46**	**11.55**	**1.46**	**1.22**
	12	21.05	25.11	2.09	1.63
	24	26.02	20.17	2.21	1.58
	48	25.23	26.15	2.12	1.62
LSTM	1	**27.94**	32.16	**1.51**	**1.46**
	12	34.01	32.12	2.89	2.52
	24	33.69	**32.0**	2.57	2.00
	48	33.8	32.16	2.52	2.29

First of all, it can be seen that the feature selection significantly improved the results. In the case of ConvLSTM, RMSE was decreased by 89.15% during the non-pandemic period (detected best performance with corresponding hours: 13.46 μg/m^3-1 hour; 1.46 μg/m^3-1 hour), and by 89.44% during the pandemic period (detected best performance with corresponding hours: 11.55 μg/m^3-1 hour; 1.22 μg/m^3-1 hour). In the case of LSTM, RMSE was decreased by 94.60% during the non-pandemic period (detected best performance with corresponding hours: 27.94 μg/m^3-1 hour; 1.51 μg/m^3-1 hour), and by 95.44% during the pandemic period (detected best performance with corresponding hours: 32.0 μg/m^3- 24 hours; 1.46 μg/m^3-1 hour).

Regarding ML algorithms, ConvLSTM outperformed LSTM, particularly, in the

first scenario compared to the second scenario the differences between the two models are significant. In terms of the first scenario, ConvLSTM outperformed LSTM by 51.83% during the non-pandemic period (ML models with detected best performance and corresponding hours: ConvLSTM - 13.46 μg/m^3- 1 hour; LSTM - 27.94 μg/m^3-1 hour), and by 63.91% during the pandemic period (ML models with detected best performance and corresponding hours: ConvLSTM - 11.55 μg/m^3- 1 hour; LSTM - 32.0 μg/m^3-24 hours). In terms of the second scenario, ConvLSTM outperformed LSTM by 3.31% during the non-pandemic period (ML models with detected best performance and corresponding hours: ConvLSTM - 1.46 μg/m^3- 1 hour; LSTM - 1.51 μg/m^3-1 hour), and by 16.44% during the pandemic period (ML models with detected best performance and corresponding hours: ConvLSTM - 1.22 μg/m^3- 1 hour; LSTM - 1.46 μg/m^3-1 hour).

Regarding the two different periods, the pandemic period exceeds the non-pandemic period in the second scenario for all time intervals, in particular, for the best performance detected in the 1-hour time interval, the pandemic period outperformed the non-pandemic period in terms of ConvLSTM by 16.44% (pandemic period-1.22 μg/m^3, non-pandemic period - 1.46 μg/m^3), and in terms of LSTM by 3.31% (pandemic period-1.46 μg/m^3, non-pandemic period - 1.51 μg/m^3). However, the difference in the first scenario is not significant. Although the variance of the pandemic year is lower than for the non-pandemic year, the algorithms are trained and tested separately for each period, which means that the models will most likely learn and generalise all existing patterns for both periods during training. In terms of time granularity, 1-hour granularity outperformed other granularities in all sub-scenarios, but this trend does not maintain for other time granularities, which could be related to the selection of the historical time lags [124]. Based on the above findings, it can be concluded that analysis involving feature selection delivers higher accuracy. ConvLSTM being able to convey spatial information in addition to temporal information has a clear advantage over LSTM, which can also be noted from the final results.

4.3 Summary

This chapter introduced ConvLSTM to predict NO_2 by recording spatiotemporal interconnections and pollutant concentration-controlling parameters. A detailed explanation of the elements that make up the ConvLSTM architecture was provided,

along with all steps in the development procedure. One of the main objectives of this chapter has been to address the impact of COVID-19 on the formation of pollution. The comparison between pandemic and non-pandemic periods by applying ConvLSTM was provided. The analysis was carried out for different time resolutions with different feature combinations. The final results showed that the proposed model outperformed the LSTM, which can be explained by the ability of the ConvLSTM to generalise and transfer the spatiotemporal information. In terms of datasets, the analyses performed with selected features surpassed the results performed with all features due to the drawback of high dimensionality.

Chapter 5

Bidirectional Convolutional Long Short-Term Memory Network[1]

Conducting a spatiotemporal analysis to capture the spatiotemporal dependencies controlling air quality leads to the next stage, which is the introduction of the next grid-based approach called BiConvLSTM. The following are the main highlighted contributions:

- We implemented and developed BiConvLSTM to forecast NO_2;
- We compared BiConvLSTM to reference models (fully connected LSTM (LSTM-FC), ConvLSTM) in terms of accuracy and runtime;
- We compared two feature selection techniques MI and mRMR to find out which technique has the highest impact on the accuracy of the predictive analysis;
- We extracted the optimum feature combination that leads to the best model performance;

[1]The part of this chapter previously appeared as an article in the Journal of PloS one and as an article in the Conference of AGILE. The original citations are as follows: Iskandaryan, Ditsuhi, Francisco Ramos, and Sergio Trilles. "Bidirectional convolutional LSTM for the prediction of nitrogen dioxide in the city of Madrid." *PloS one* 17, no. 6 (2022): e0269295; and Iskandaryan, Ditsuhi, Silvana Di Sabatino, Francisco Ramos, and Sergio Trilles. "Exploratory Analysis and Feature Selection for the Prediction of Nitrogen Dioxide." *AGILE: GIScience Series* 3 (2022): 1-11.

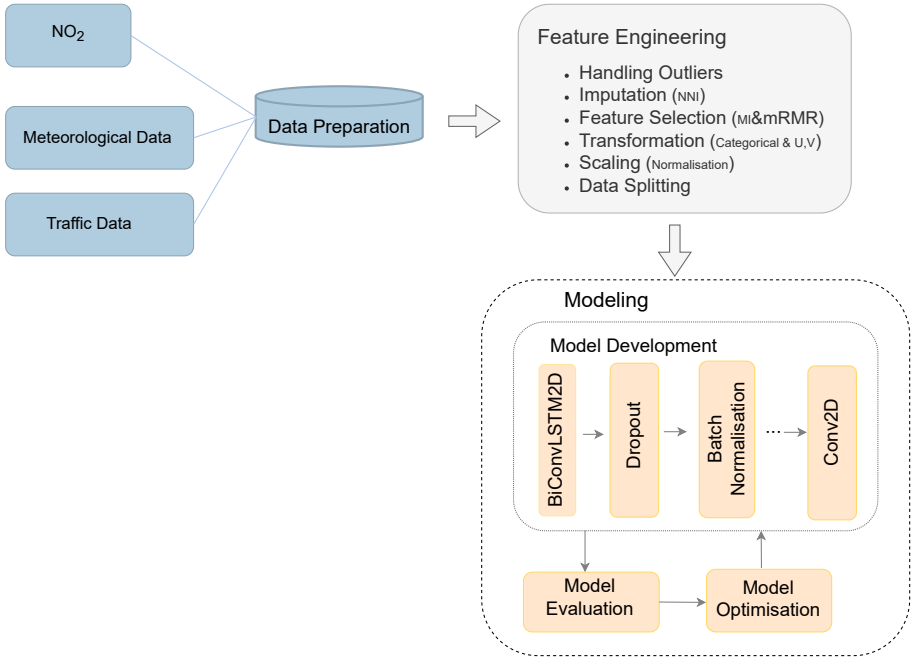


Figure 5.1: The workflow of the Bidirectional Convolutional Long Short-Term Memory-based nitrogen dioxide predictive analysis.

- We examined and compared two transformation approaches applied to wind direction in terms of model performance.

Below are presented the sections devoted to the thorough description of the experimental analysis (*Experimental Analysis*), and the results acquired from this analysis (*Results and Discussion*).

5.1 Experimental Analysis

A thorough explanation of the experimental analysis is provided in this section. Figure 5.1 depicts the elements of the workflow, including *Data Preparation*, *Feature Engineering*, and *Modeling* components.

Data Preparation: as already mentioned, the process of data preparation is described in Section 3.2. For the analysis of this chapter, we used NO_2, meteorological and traffic data.

Feature Engineering: this process with all substeps (*Handling Outliers*, *Imputation*, *Feature Selection*, *Transformation*, *Scaling* and *Data Splitting*) was also

described previously (Section 3.4). Below is a description of each technique.

Handling Outliers: the outliers were detected using summary statistics of the datasets (stated Section 3.4).

Imputation: considering the fact that meteorological data do not change dramatically within space, the NNI was implemented [83].

Feature Selection: in this stage MI and mRMR were implemented with two scenarios: 1) highlighting the advantage of the feature extraction technique by implementing MI, and 2) comparing MI and mRMR.

First scenario: before implementing MI, the following variables were eliminated from the future predictive analysis: average traffic speed, traffic load, UV, precipitation. Average traffic speed was removed because it is only known for M30 road, which is 15.8% of the study area (Table 3.2). Traffic load was also excluded considering that it is correlated with other variables: intensity and occupancy time (according to the definition of traffic load, it is calculated using intensity and occupancy time). In terms of UV, it was observed that there are no UV records for June 2019 and the entire period of January-June 2020. Regarding precipitation, it was found out that nearly all of the data was zero, therefore this component was also removed.

The feature relevance scores of seven additional datasets based on MI are shown in Figure 5.2. For further analysis in the second scenario, features with a score greater than 0.005 were selected, including wind speed, barometric pressure, intensity and occupancy time. It should be mentioned that the wind direction was also selected in consideration of the interconnection with wind speed. The reason for not including wind direction in the MI computation procedure is due to the fact that wind direction is circular data that must be converted before use (details below).

Second scenario: based on the transformation mechanism, the experiments were carried out with the following subscenarios:

First subscenario: wind direction was converted to the following categories: north, east, south, west, southwest, northeast, southeast, northwest, and later it was included in the analysis by implementing One Hot Encoder.

Second subscenario: wind direction was converted to u and v components (Eq. 3.20).

Feature selection techniques were implemented for each subscenario. Figure

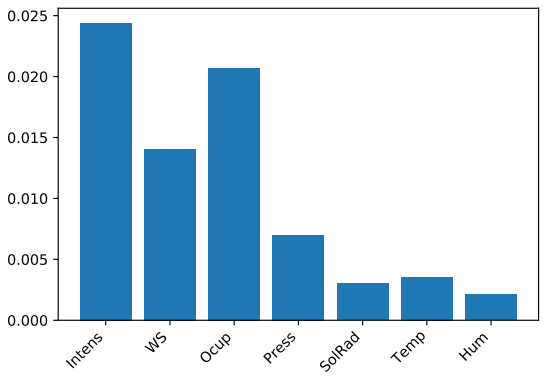


Figure 5.2: The feature importance scores based on Mutual Information.

5.3 and Figure 5.4 show the results of both scenarios based on the MI technique. The features selected were those with a score higher than 0.005. In Figure 5.3 it can be observed that among seventeen features, the following six were selected: intensity, occupancy time, wind speed, pressure, load and average traffic speed. Figure 5.4 shows eleven features, the following eight were selected: intensity, occupancy time, wind speed, pressure, load, average traffic speed, u component and v component.

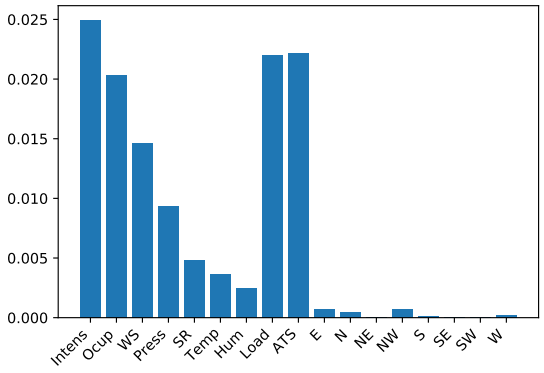


Figure 5.3: Feature selection using the Mutual Information technique (Wind direction with One Hot Encoder).

Transformation: this phase involves converting wind direction into categorical data, and passing through One Hot Encoder or into u and v components. Another transformation was the conversion of the input data into the supervised learning dataset. Independent and dependent datasets were generated based on the

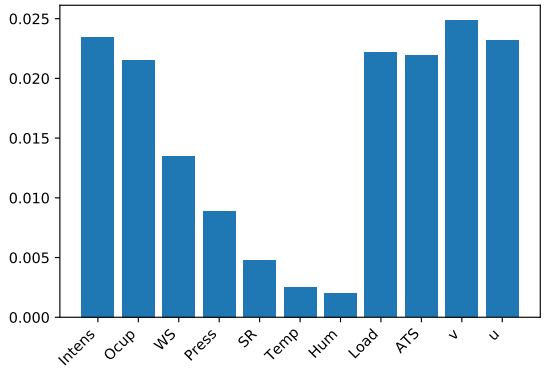


Figure 5.4: Feature selection using the Mutual Information technique (Wind direction with u and v components).

defined time granularity (to predict NO_2 in t' hours based on the data for the previous t' hours, where t' was equal to 6).

Scaling: in this analysis, the input data went through to Min-Max (0-1) normalisation (Eq. 3.21).

Data splitting: the splitting procedure was to divide the whole dataset (January-June 2019 and January-June 2020) into training (60%), validation (20%) and testing (20%) sets. The dimension of each set is illustrated in Table 5.1.

Modeling: it consists of three substeps: *Model Development*, *Model Evaluation* and *Model Optimisation*.

Model Development: the model architecture consists of three layers, including bidirectional ConvLSTM2D (stated in Section 3.5.3), Dropout (stated in Section 4.1), and Batch Normalisation (stated in Section 4.1), which is finalised with Conv2D layer (stated in Section 3.5.2).

Model Evaluation: the final model was evaluated using RMSE and MAE. The results are presented in the next section.

Model Optimisation: this stage presents the procedure required for model construction. The parameter optimisation of the proposed model was performed by applying one of the common optimisation techniques, named Grid Search using $GridSearchCV$[2]. Considering the essence of time series data $Blocking\ Time\ Series\ Split$ was implemented instead of cross-validation because it considers

[2]GridSearchCV: `https://bit.ly/2QBg3Hy`. [Online; accessed 15-February-2023]

Table 5.1: The dimension of each set.

Set	**Dimension ($x \times y \times z_1/z_2$)(*)**
Training Set	$4,344 \times 340 \times 16/13$
Validation Set	$2,184 \times 340 \times 16/13$
Testing Set	$2,184 \times 340 \times 16/13$

* x – Number of samples; y – Number of grid cells ($340 = 20 \times 17$); z_1 – Number of all features (NO_2, wind speed, temperature, humidity, barometric pressure, solar irradiance, intensity, occupancy time, north, east, south, west, southwest, northeast, southeast, northwest), z_2 – Number of selected features (NO_2, wind speed, barometric pressure, intensity, occupancy time, north, east, south, west, southwest, northeast, southeast, northwest). Note that features include wind directions after the implementation of One Hot Encoder.

the time series aspect and prevents leakage from one set to another. Another technique that was used to improve the performance of Grid Search is $Early\ Stopping\ callback$[3]. All these techniques are described in detail in Section 4.1.

To reduce the computation time for parameter optimisation, $GridSearchCV$ was applied to one-month data. Table 5.2 shows optimised hyperparameters: $Number\ of\ filters$, $Kernel\ size$, $Optimiser$, $Merge\ Mode$ and $Number\ of\ Layers$ (the description of these hyperparameters is stated in Section 4.1) with the options that were tried (the testing options were chosen considering common approaches from different works in the domain, which is also stated in Section 4.1), and the one that was finally selected is indicated in bold. Below is a brief description of each hyperparameter included in this work.

$Number\ of\ filters$: we tuned the hyperparameter by testing the model with 8, 16 and 32 filters. It turned out that the performance of the model with 16 filters is superior to that of the other filters.

$Kernel\ size$: we tested the model with 3×3, 5×5, 7×7, and 9×9 kernels. The

[3]Early Stopping callback: `https://bit.ly/3TQtyQc`. [Online; accessed 15-February-2023]

Table 5.2: Hyperparameter optimisation with GridSearchCV.

Hyperparameters	**Options**
Number of Filters	8, **16**, 32
Kernel Size	**(3,3)**, (5,5), (7,7), (9, 9)
Optimiser	RMSprop, **Adam**
Merge Mode	**'concat'**, 'mul', 'sum', 'ave'
Number of Layers	2, **3**, 4

model performance with 3×3 kernels was found to be superior to that of the other kernels.

$Optimiser$: we tested the model with Adam and RMSProp optimisers. The result showed that the implementation with Adam is superior to the one with RMSProp.

$Merge\ Mode$: we tested the model with concatenation, multiplication, sum and average merge modes. The performance with concatenation was superior to the rest of the options.

$Number\ of\ Layers$: we tested the model with 2, 3 and 4 layers. The results showed that the model performance with 3 layers was superior to that of the other two options.

Overall, the architecture of the model was built based on the chosen parameters by stacking three BiConvLSTM layers with a kernel size of 3×3, 16 filters and with an Adam optimiser. Concatenation was selected as the merge mode, which means that the forward and backward ConvLSTM units were concatenated before passing information to the next unit.

Regarding the baseline models, LSTM-FC had the following structure: two LSTM layers with 2048 units followed by the Dropout layer and the model was finalised by adding a Dense layer; ConvLSTM had 5×5 kernel size with filters equal to 32, followed by Batch Normalisation and Dropout layers and it was finalised with 1×1 convolution layer.

Algorithm 2 provides a pseudo code of NO_2 prediction procedure.

Algorithm 2 Nitrogen dioxide prediction

Input: Comma Separated Values (CSV) files for each hour including NO_2, Meteorological and Traffic data

function CALCULATE NEAREST NEIGHBOUR INTERPOLATION(Meteorological data)

return *zero values of meteorological data impute by NNI*

end function

function HANDLING OUTLIERS(data)

return *outliers converted to the average of the previous and the next non-outliers*

end function

function TRANSFORMATION(data)

return *independent and dependent data generation based on time resolution*

end function

function DATA SPLITTING(data)

Split data on training, validation and testing sets with the following order: January-March 2020 - validation sets; April - June 2020 - testing set

end function

Normalise input set

Reshape data based on selected model architecture

function CREATE MODEL(model parameters by default)

return *model architecture*

end function

function GRIDSEARCHCV(parameters to tune)

return *best parameters*

end function

function EVALUATE MODEL(model with best parameters)

return *error estimated with evaluation metrics*

end function

Output: RMSE, MAE

5.2 Results and Discussion

As earlier stated, the analysis was conducted in two scenarios. The outcomes for each of them are listed below.

First scenario: the experiments in this scenario were conducted in two steps, including all features and including only features extracted after MI implementation in order to address the following questions: 1) Is the proposed model capable of outperforming the reference models? and 2) Can the implementation of feature selection improve model performance?

All features component includes nine features (NO_2, wind speed, wind direction, temperature, relative humidity, barometric pressure, solar irradiance, intensity, and occupancy time). Selected features component includes wind speed, wind direction, barometric pressure, intensity and occupancy time. The results obtained and the runtime of the models over the following 6-hour lag are presented in Table 5.3.

All Features: in this case, BiConvLSTM outperforms ConvLSTM and LSTM-FC in terms of RMSE and MAE, with values of 19.14 and 13.06, respectively. In particular, in terms of RMSE, BiConvLSTM improves results compared to ConvLSTM by 41.9%, and to LSTM-FC by 50.8%. In terms of MAE, BiConvLSTM improves results compared to ConvLSTM by 59.24%, and to LSTM-FC by 59.4%. Regarding runtime, due to the complexity of the BiConvLSTM architecture, the model takes a comparable amount of time to converge.

Table 5.3: Prediction errors (Root Mean Square Error, Mean Absolute Error) and runtime of the models for the next 6 hours prediction implemented on all features.

	Models	**RMSE** (μ**g/m^3**)	**MAE** (μ**g/m^3**)	**Time**
All Features	LSTM-FC	38.89	32.17	4m 15s
	ConvLSTM	32.95	32.04	33m 15s
	BiConvLSTM	19.14	13.06	36m 57s
Selected Features	LSTM-FC	15.68	13.54	3m 58s
	ConvLSTM	15.11	11.9	27m 53s
	BiConvLSTM	12.65	9.72	34m 33s

Selected Features: as in the first case, in this case also BiConvLSTM surpassed other models. Especially, in terms of RMSE, BiConvLSTM improves results compared to ConvLSTM by 16.28%, and to LSTM-FC by 19.32% in terms of MAE, BiConvLSTM improves results compared to ConvLSTM by 18.32%, and to LSTM-FC by 28.21%. Regarding runtime, BiConvLSTM converges slower than ConvLSTM and LSTM-FC.

The difference between the two cases is a significant reduction in the values in terms of runtime and error, which is associated with the peculiarities of the implementation of the feature selection methodology. It is essential to understand why only a few features (wind speed, wind direction, barometric pressure, intensity, and occupancy time) were chosen out of all the possibilities, as well as the relationship between NO_2 and features with a higher MI index, the inclusion of which improved the model's performance. In terms of wind speed and direction, the correlation exists since increasing wind speed implies a lower concentration due to increased dilution through advection and increased mechanical turbulence. In terms of traffic data, the transportation industry is one of the main generators of NO_x (nitrogen oxide and NO_2). For example, NO_x accounted for nearly 46% of total emissions in the EU in 2013 [125].

Overall, BiConvLSTM outperforms other reference models; nevertheless, regarding the execution time, it takes comparatively longer. MAE is defined in the same unit as the target variable; therefore, in the current work, it corresponds to the unit of NO_2 ($\mu g/m^3$). Note that MAE is 9.72 $\mu g/m^3$, which can be considered sufficient compared with mean values of NO_2 (36.69 and 26.03 for 2019 and 2020, respectively).

Second Scenario: this scenario by implementing MI and mRMR techniques in two subscenarios (based on wind direction conversion) tends to answer the following questions: 1) Which feature extraction technique is better: MI or mRMR? 2) What is the optimum feature combination that leads to the best model performance? and 3) Which wind direction transformation affects getting the best model performance?

The results can be shown in Table 5.4. The outcomes of the first subscenario outperformed the results of the second subscenario by including all of the features. However, the results of MI do not follow the same pattern. Especially, MI deteriorated the results of the first subscenario, but it boosted the second subscenario's overall performance. An additional finding is that, with all features included, the

conversion of wind direction into categories and the subsequent implementation of One Hot Encoder outperformed the conversion to u and v components.

Table 5.4: Root Mean Square Error and Mean Absolute Error of Subscenarios I and II using Bidirectional Convolutional Long Short-Term Memory (units in μg/m^3).

	All Features		Selected Features (MI)	
	RMSE	**MAE**	**RMSE**	**MAE**
Subscenario I	18.99	12.89	26.92	20.00
Subscenario II	24.87	16.49	22.32	16.89

Regarding mRMR, the results are illustrated in Table 5.5 (first subscenario) and Table 5.6 (second subscenario). It is detectable that the errors are significantly reduced. In the case of the first subscenario, the best combination of the features is obtained when K=7 (RMSE–3.44, MAE–2.87). The selected features are load, northwest direction, pressure, wind speed, average traffic speed, occupancy time and north direction. In the case of the second subscenario, the best result was obtained when K=5 (RMSE–4.20, MAE–3.65). The selected features are load, pressure, wind speed, average traffic speed and occupancy time.

Table 5.5: Root Mean Square Error and Mean Absolute Error of extracted features based on Maximum Relevance — Minimum Redundancy (K is the number of features) using Bidirectional Convolutional Long Short-Term Memory (subscenario I).

	RMSE (μg/m^3)	**MAE (μg/m^3)**
K=3	6.81	5.97
K=4	5.61	5.18
K=5	3.55	3.07
K=6	4.90	4.37
K=7	**3.44**	**2.87**
K=8	19.91	15.51

Following the outcome, it can be concluded that mRMR outperformed MI since the latter selects the most relevant features. In contrast, mRMR selects the

Table 5.6: Root Mean Square Error and Mean Absolute Error of extracted features based on Maximum Relevance — Minimum Redundancy (K is the number of features) using Bidirectional Convolutional Long Short-Term Memory (subscenario II).

	RMSE (μg/m^3)	**MAE (μg/m^3)**
K=3	5.60	4.84
K=4	5.26	4.69
K=5	**4.20**	**3.65**
K=6	23.51	14.05
K=7	33.48	21.29
K=8	31.80	21.77

relevant features with minimal redundancy. In addition, it is important to see what features were chosen and what caused this choice. After implementing mRMR, the load was selected in both cases. Given the importance of traffic data for NO_2 production and the definition of load, the choice of this feature is obvious. The other features that yield better results are pressure, wind speed, average traffic speed and occupancy time. The last two features, as already mentioned, are chosen because of the importance of traffic data for NO_2 production. Regarding wind speed, as mentioned in the exploratory analysis, there is a strong correlation between wind speed and NO_2. Regarding the wind direction transformation, the u and v components were not included in the selected subsets after applying mRMR, although the northwest and north directions were included. The best subsets of the first subscenario outperformed the second subscenario, improving RMSE by 18.1% and MAE by 21.37%. Therefore, also in the case of implementing mRMR, the wind direction conversion to categories surpassed the u and v conversion.

Regarding the overall results, the proposed model outperforms the reference models, and the feature selection strategy improves overall accuracy significantly. Especially mRMR yields better results compared to MI, given the fact that mRMR, in addition to selecting relevant features, tries to select the next relevant feature that has a minimum correlation with already selected features. In terms of runtime, the BiConvLSTM took longer to converge, which can be explained by the model's complexity.

5.3 Summary

This chapter introduced BiConvLSTM to predict NO_2 using air quality, meteorological and traffic data from the period of January-June 2019 and January-June 2020 in the city of Madrid. A detailed description of the components and the development procedure of the proposed model was presented.

The chapter was constructed in two scenarios based on the subsets of features used in the analyses: 1) highlighting the advantage of the feature extraction technique by implementing MI, and 2) comparing MI and mRMR.

First Scenario: the comparison between the proposed model and ConvLSTM and LSTM-FC was produced. The outcome demonstrated that BiConvLSTM outperformed the reference models. Additionally, feature selection implemented with the technique MI improved the final results by 33.9% and 25.27% in terms of RMSE and MAE, respectively. However, the model architecture makes BiConvLSTM slower at runtime, and data convergence takes longer. It is important to note that by examining the results of the MAE and comparing them with the average concentration values, the proposed model can be considered a reliable and robust model.

Second Scenario: this part concentrated on applying MI and mRMR, obtaining the most relevant features related to NO_2, and comparing the results of both methods. Another direction was the preprocessing of wind direction data applying the following conversion methods: converting the wind direction into u and v components or into categorical data. The results show that the conversion of the wind direction in One Hot Encoder is superior to the conversion to the u and v components. Regarding feature selection methods, it was found that the implementation of mRMR yields better results compared to MI, given the fact that mRMR selects the next relevant feature that has a minimum correlation with already selected features.

It is essential to consider the impact of the COVID-19 during 2020 to combat some measures, such as traffic restrictions and self-isolation. As a result, these events have affected the air pollution concentration. In the case of Madrid, due to COVID-19 restrictions, the concentration of NO_2 dropped to 62% [109]. These sudden changes may impact the model's performance, and it would be ideal for comparing the results to a different period in the future to uncover these effects.

Chapter 6

Attention Temporal Graph Convolutional Network[1]

The distribution of air quality stations in the city of Madrid (Figure 3.5) does not have any specific pattern, they are spread without any significant order. Most of the time, the forecast of the concentration of pollutants in the atmospheric air is required to be performed at the stations where they were registered. To conduct predictive analysis in the air quality monitoring stations, taking into account their spatiotemporal relationships, a GNN can be implemented that is able to process non-Euclidean structured data. The following are the significant contributions addressed within the scope of this chapter:

- We conducted spatiotemporal prediction of NO_2 using a GNN, namely A3T-GCN;
- We performed the predictive analysis in different time intervals, including 1-12 h, 12-24 h, 24-36 h and 36-48 h;
- We compared the proposed method with reference methods (LSTM, GRU) in terms of determined evaluation metrics (RMSE, MAE, R);

[1]The part of this chapter previously appeared as an article in the Journal of IEEE Access and as an article in the Conference of EnviroInfo. The original citation is as follows: Iskandaryan, Ditsuhi, Francisco Ramos, and Sergio Trilles. "Graph Neural Network for Air Quality Prediction: A Case Study in Madrid." *IEEE Access* 11 (2023): 2729-2742; and Iskandaryan, Ditsuhi, Francisco Ramos, and Sergio Trilles. "Spatiotemporal Prediction of Nitrogen Dioxide Based on Graph Neural Networks." *Environmental Informatics*, pp. 111-128. Springer, Cham, 2023.

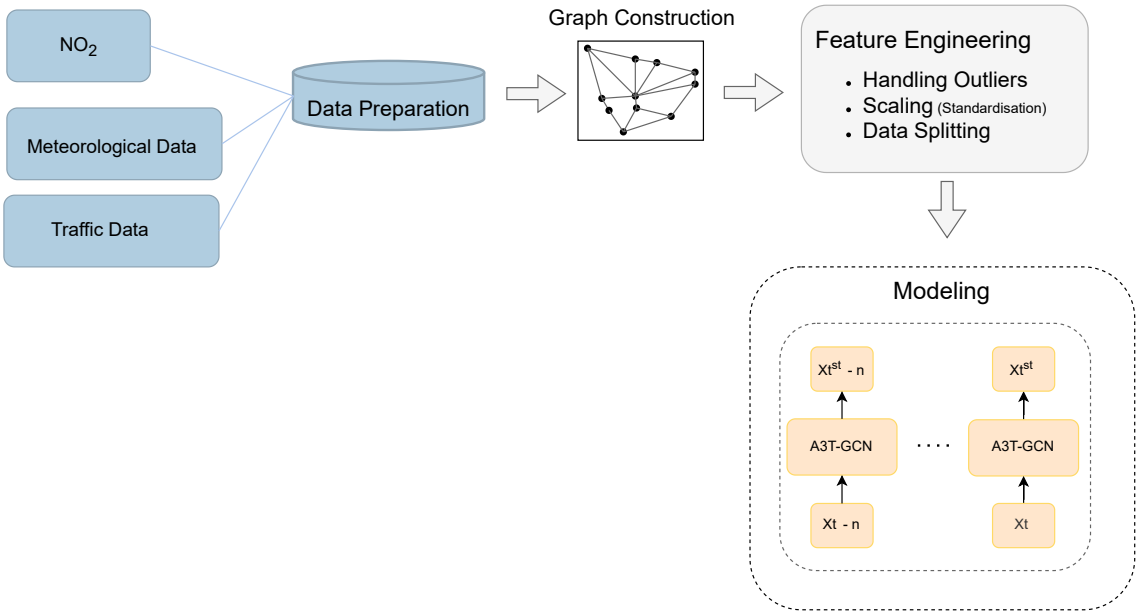


Figure 6.1: The workflow of the Attention Temporal Graph Convolutional Network-based nitrogen dioxide predictive analysis.

- We implemented outliers detection techniques (iForest, LOF) and compared the results obtained before and after outliers detection.

The following sections describe the stages of the experimental analysis (*Experimental Analysis*), the output of this analysis and discussions that follow from this output (*Results and Discussion*).

6.1 Experimental Analysis

This section presents a detailed explanation of the experimental analysis. The workflow is shown in Figure 6.1, which consists of the following steps: *Data Preparation*, *Graph Construction*, *Feature Engineering*, and *Modeling*.

Data Preparation: the data used in the scope of this chapter are NO_2, meteorological and traffic data, which were integrated with the spatiotemporal dimensions (described in Section 3.2). After combining all features, twenty-four cells that include air quality monitoring stations were selected for further analysis, in particular, to be used as input to the proposed method working with non-Euclidean distances. As a result, the input data before graph generation has the dimensions mentioned in Table 6.1.

Table 6.1: The dimension of each set.

Set	**Dimension ($x \times y \times z$)(*)**
Training Set	$4,344 \times 24 \times 18$
Testing Set	$4,367 \times 24 \times 18$

* x – Number of samples; y – Number of stations; z – Number of the features (NO_2, 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.

Table 6.2 shows summary statistics of each data type for the periods used in the analyses, including data that exists only in the selected twenty-four cells:

Graph Construction: The next block after data preparation is graph construction. Following the graph structure's definition, air quality stations will be considered graph nodes in this work. All stations are interconnected, forming graph edges, and the distances between them will be considered edge weights. The distance between nodes was calculated using $arcpy.analysis.GenerateNearTable$[2] function. It should be mentioned that to create the adjacency matrix, the original distance between two nodes was converted to 1/distance (Eq.6.1), 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.

$$A_{ij} = \begin{cases} \frac{1}{d_{ij}}, & \text{i}\neq\text{j} \\ 0, & \text{otherwise} \end{cases} \tag{6.1}$$

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 this study, for each time t, the node features can be

[2]Generate Near Table (Analysis): `https://pro.arcgis.com/en/pro-app/latest/tool-reference/analysis/generate-near-table.htm`. [Online; accessed 15-February-2023]

Table 6.2: Summary statistics of the periods January-June 2019 and January-June 2020 for each data type that exists in the selected 24 cells.

Phenomena	Descriptors	January-June 2019	January-June 2020
Nitrogen dioxide (µg/m^3)	Mean (SD)	36.63 (30.86)	25.62 (25.36)
	Median [Min,Max]	27.0 [0.0, 328.0]	16.0 [0.0, 326.0]
Wind speed (m/s)	Mean (SD)	1.33 (1.04)	1.25 (0.98)
	Median [Min,Max]	1.09 [0.0, 8.75]	1.02 [0.0, 8.97]
Wind direction	Mean (SD)	167.80 (105.72)	140.82 (98.35)
	Median [Min,Max]	182.0 [0.0, 359]	135.0 [0.0, 359]
Temperature (°C)	Mean (SD)	13.21 (7.81)	13.45 (7.26)
	Median [Min,Max]	12.3 [-3.0, 40.6]	12.3 [-2.0, 38.1]
Humidity (%)	Mean (SD)	49.57 (20.86)	62.49 (21.46)
	Median [Min,Max]	48.0 [0.0, 100]	63.0 [0.0, 100]
Barometric pressure (mb)	Mean (SD)	943.81 (17.89)	943.67 (20.23)
	Median [Min,Max]	944.0 [0.0, 962.0]	945.0 [0.0, 1073.0]
Solar irradiance (W/m^2)	Mean (SD)	223.96 (302.56)	193.21 (279.86)
	Median [Min,Max]	14.0 [0.0, 1103.0]	10.0 [0.0, 1113.0]
Intensity (vehicles/hour)	Mean (SD)	315.18 (303.05)	200.38 (240.88)
	Median [Min,Max]	252.62 [0.0, 3712.87]	109.32 [0.0, 2436.5]
Occupancy time (%)	Mean (SD)	5.04 (5.26)	3.28 (4.23)
	Median [Min,Max]	3.68 [0.0, 55.47]	1.75 [0.0, 51.6]
Load	Mean (SD)	17.51 (14.15)	11.53 (12.05)
	Median [Min,Max]	16.0 [0.0, 93.36]	7.08 [0.0, 68.57]
Average traffic speed (km/h)	Mean (SD)	0.37 (1.30)	0.42 (1.58)
	Median [Min,Max]	0.0 [0.0, 35.29]	0.0[-3.74, 73.33]

assigned as $X_t \in R^{N \times M}$, where N is the number of nodes and M is the features. Figure 6.2 shows the graph constructed based on air quality stations located in the city of Madrid. It consists of 24 nodes and 276 edges (connecting each pair of nodes). The numbers on the nodes in Figure 6.2 are the identifier of each cell of the grid that was initially given, which contains a certain station. Algorithm 3 shows the procedure of creating a graph network on the map.

The prediction of NO_2 was performed based on 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 aforementioned procedure can be defined as a function of the air quality stations network G and the feature matrix X (Eq. 6.2).

[3]Find the centroid of polygons in ArcGIS Pro: `https://bit.ly/3rjMWst`. [Online; accessed 15-February-2023]

[4]XY To Line: `https://bit.ly/3y13sBl`. [Online; accessed 15-February-2023]

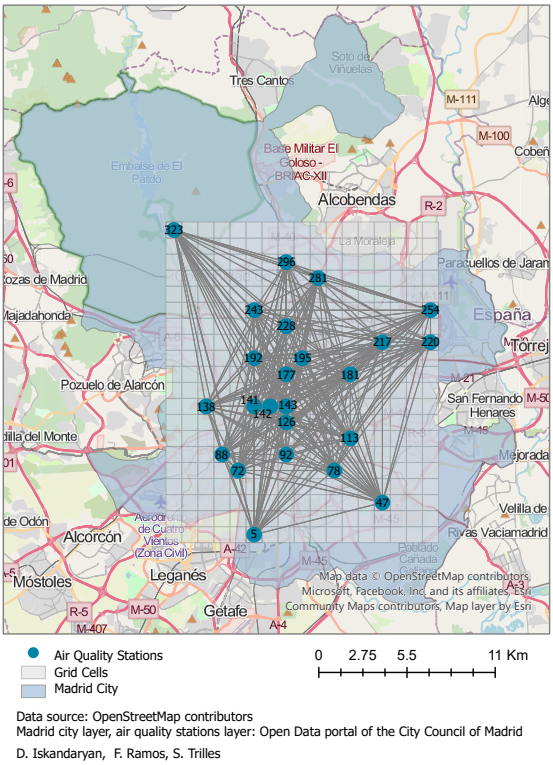


Figure 6.2: Graph network of the air quality stations placed in the city of Madrid.

$$[X^a_{t+1},,X^a_{t+T}] = f(G;(X_{t-n},,X_{t-1},X_t)) \tag{6.2}$$

where T is the next hours, n is the previous hours, X^a_t is the concentration of NO_2 at time t and X_t is a combination of NO_2, meteorological and traffic data. Each sample of the input has the following structure:

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

where 24 is the number of nodes, 18 is the number of node's features, T is equal to 12, [2, 552] from $edge\_index$ refers to the fact that every edge was considered two times (276*2=552). Algorithm 4 shows the procedure of input data preparation for GNN.

Feature Engineering: this step includes the following substeps: *Handling Outliers*, *Transformation*, *Scaling* and *Data Splitting*.

Handling Outliers: in this step iForest and LOF were implemented in order to detect outliers. The following parameters were defined with respect to each technique:

Algorithm 3 Creating graph network on the city of Madrid

Input: Grid with 340 Cells (20*17)

1: **function** EXTRACT CELLS HAVING AIR QUALITY STATIONS(data)
2: **return** *24 cells including air quality stations*
3: **end function**
4: **function** CALCULATE CENTROIDS OF THE EXTRACTED CELLS AND CREATE SEPARATE FEATURE CLASS [3](24 extracted cells)
5: **return** *feature class of centroids*
6: **end function**
7: **function** DRAW NETWORK BETWEEN EACH PAIR OF POINTS(feature class of centroids)
8: **return** *Draw a Network between each pair of points with all combinations using arcpy.management.XYToLine function* [4]
9: **end function**

Output: Figure 6.2

iForest– $n\_estimators$=100, $max\_samples$=all the samples, $contamination$=float(0.05), $max\_features$=1.0; LOF– $n\_neighbors$=all the samples, $metric$ = "manhattan", $contamination$ = 0.05 ($contamination$ is the proportion of outliers in the dataset, and it was set to 0.05, meaning that 5% of the dataset was considered to be outliers; $max\_features$ was set to 1.0, which means that only the given feature was considered in the detection process).

Transformation: this phase involves converting wind direction into categorical data (north, east, south, west, southwest, northeast, southeast, and northwest), and implementing One Hot Encoder. Another transformation is to generate independent and dependent datasets based on the defined time granularity (to predict NO_2 in t' hours based on the data for the previous 12 hours, where $t' \in \{1-12, 12-24, 24-36, 36-48\}$).

Scaling: before converting the data into a graph construction, the input data were standardised (Eq. 3.22).

Data splitting: this step includes the procedure of splitting the dataset into training (January-June 2019) and testing (January-June 2020) sets.

Modeling: this block refers to the construction of the architecture of the proposed model. It consists of three graph convolutional layers (the output of the layers is

Algorithm 4 Data preparation for Graph Neural Network

Input: Data - [Hourly NO_2, Meteorological and Traffic data]; Period -[01.01.2019-30.06.2019; 01.01.2020-30.06.2020]

function MERGE THE DATA SPATIALLY AND TEMPORALLY(data)
  for each hour $\in$ Period **do**
    Create grid with $CreateFishnet$ function (ArcPy library)
    Add field to the Fishnet
    for each item $i \in$ Data **do**
      i spatial join with grid
      input the mean of the values of each corresponding cell to the field
    end for
  end for
  return *$.csv$ files for each hour including NO_2, Meteorological and Traffic data [4,344 and 4,367 $.csv$ files for January-June 2019 and January-June 2022, respectively with the following dimension: $4,344 \times 340 \times 18$: (January-June 2019); $4,367 \times 340 \times 18$ (January-June 2020)]*
end function
function EXTRACT CELLS HAVING AIR QUALITY STATIONS(data)
  return *extract cells or rows including NO_2, Meteorological and Traffic data, where air quality monitoring stations exist [$4,344 \times 24 \times 18$: (January-June 2019); $4,367 \times 24 \times 18$ (January-June 2020)]*
end function
function CREATE ADJACENCY MATRIX(location of Air Quality stations (nodes))
  Calculate distance between each pair of air quality stations using $arcpy.analysis.GenerateNearTable$
  return *adjacency matrix filled with inverse distance (1/distance) between each pair of nodes*
end function
function NORMALISE DATA(data)
  return *normalised data using Z-Score method*
end function
function GENERATING DATASET FOR GNN BASED ON THE DEFINED TEMPORAL INTERVAL: [1-12, 12-24, 24-36 AND 36-48] (T=12)(data)
  return *Dataset with 8711 samples (Training Set–4,344, Testing Set–4367; each sample–Data(x=[24, 18, T], $edge\_index$=[2, 552], $edge\_attr$=[552], y=[24, T], $batch$=[64]))*
end function

Output:Dataset with 8711 samples (each sample: Data(x=[24, 18, T], $edge\_index$=[2, 552], $edge\_attr$=[552], y=[24, T], $batch$=[64]))

used in GRU: update gate, reset gate and hidden state) followed by learnable transformations. Table 6.3 summarises the parameters and settings applied in the analysis.

Table 6.3: Details of the experimental settings.

Parameter	**Value**
Number of records	8,711
Time interval (h)	1
Training set	January-June 2019: 4,344
Testing set	January-June 2020: 4,367
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	[32, 64,128, 256]
Optimiser	Adam
Loss function	MSE

Regarding the reference models, they consist of the fully connected layer with 432 units (24*18), followed by three stacked LSTM layers (in the case of GRU model, it consists of three stacked GRU layers) with 512, 1,024 and 512 units, and the models were finalised with another fully connected layer with 24 units (representing NO_2 for all stations). It should be mentioned that the analysis was performed in the Google Colab cloud service using the PyTorch Geometric Temporal library [126].

6.2 Results and Discussion

This part illustrates the output of the analysis. The analysis was carried out under two scenarios: a) before outliers detection and b) after outliers detection. Below are the results for each of them.

[5]Google Colab: `https://bit.ly/3refm74`. [Online; accessed 15-February-2023]

First Scenario: in this scenario, the analysis was performed without implementing outlier detection. The results of the analysis are shown in Table 6.4 (the best results are indicated in bold). It should be mentioned that the averaged value of NO_2 from all stations was calculated for the testing period. Algorithm 5 provides a pseudo code of NO_2 prediction procedure.

Algorithm 5 Nitrogen dioxide prediction

Input: Dataset with 8711 samples (Training Set–4,344, Testing Set–4,367; each sample–Data(x=[24, 18, T], $edge\_index$=[2, 552], $edge\_attr$=[552], y=[24, T], $batch$=[64]))

function CREATE MODEL
2: **return** *A3T-GCN architecture based on the settings from Table 6.3*
end function
4: **function** EVALUATE MODEL(model with best parameters)
return *error estimated with evaluation metric*
6: **end function**
function EVALUATE REFERENCE MODELS (LSTM AND GRU)(models with defined parameters)
8: **return** *error estimated with evaluation metric (RMSE, MAE, R)*
end function

Output: RMSE, MAE, R for A3T-GCN, LSTM and GRU (Table 6.4)

The experiments were carried out for different numbers of units of the proposed model (A3T-GCN-32, A3T-GCN-64, A3T-GCN-128, A3T-GCN-256), and for reference models (LSTM, GRU). In terms of RMSE, the lowest value found for the 1-12 hours time interval implemented by A3T-GCN-128 is 16.34 μg/m^3, which outperforms the best performance of LSTM (18.77 μg/m^3) and the best performance of GRU (19.11 μg/m^3) found for the 12-24 hours time interval by 12.95% and 14.50%, respectively. Regarding MAE, the lowest value found for the 12-24 hours time interval implemented by A3T-GCN-128 is 13.25 μg/m^3, which outperforms the best performance of LSTM (13.77 μg/m^3) and the best performance of GRU (13.44 μg/m^3) found in the same interval by 3.78% and 1.41%, respectively. Regarding R, the highest value found for the 1-12 hours time interval implemented by A3T-GCN-256 is 0.72, which outperforms the best performance of LSTM and the best performance of GRU (in both cases is 0.68, regarding time interval, for LSTM it was found for 1-12 hours and 12-24 hours; and for GRU for 12-24 hours interval)

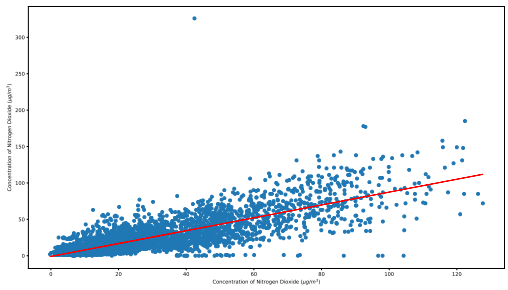
Table 6.4: Performance evaluation metrics of Attention Temporal Graph Convolutional Network, Long Short-Term Memory and Gated Recurrent Unit in terms of time granularities before outliers detection.

Method	**Hours**	**RMSE** (μg/m^3)	**MAE** (μg/m^3)	**R**
A3T-GCN-32	1-12	16.96	14.18	0.67
	12-24	17.45	14.72	0.67
	24-36	18.05	15.23	0.63
	36-48	18.25	15.41	0.62
A3T-GCN-64	1-12	17.19	14.48	0.69
	12-24	16.85	14.07	0.67
	24-36	17.40	14.49	0.64
	36-48	17.93	15.13	0.63
A3T-GCN-128	1-12	**16.34**	13.57	0.63
	12-24	17.05	**13.25**	0.67
	24-36	18.29	15.53	0.63
	36-48	18.32	15.58	0.64
A3T-GCN-256	1-12	16.60	13.62	**0.72**
	12-24	16.77	13.77	0.67
	24-36	17.52	14.64	0.65
	36-48	17.77	14.77	0.63
LSTM	1-12	19.67	15.01	**0.68**
	12-24	**18.77**	**13.77**	**0.68**
	24-36	20.49	15.25	0.63
	36-48	20.50	14.42	0.63
GRU	1-12	19.71	14.47	0.67
	12-24	**19.11**	**13.44**	**0.68**
	24-36	20.80	15.02	0.61
	36-48	20.11	14.25	0.65

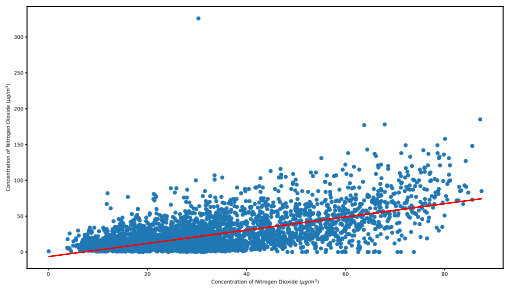
by 5.56%.

Regarding the time interval pattern, the results at closer time intervals exceeded the outcomes at more distant intervals in the case of hidden units with 256. Figure

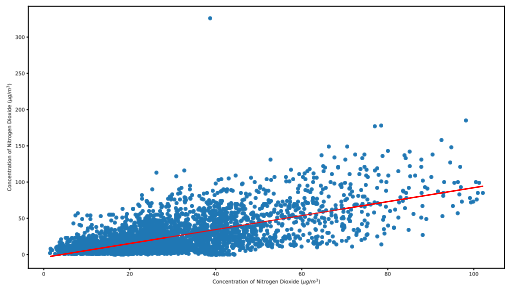
6.3 shows the scatter plot of actual (y axis) and predicted values (x axis) of NO_2 for the next defined hours (1-12 h, 12-24 h, 24-36 h, 36-48 h) when the hidden unit is 256.



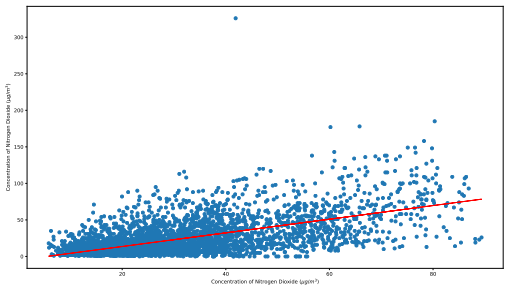
(a) Actual and predicted values for the next 1-12 h.



(b) Actual and predicted values for the next 12-24 h.



(c) Actual and predicted values for the next 24-36 h.



(d) Actual and predicted values for the next 36-48 h.

Figure 6.3: Scatter plot of actual and predicted values of nitrogen dioxide at the station with id 181 during January-June 2020 in the city of Madrid.

Second Scenario: this scenario introduces the results after implementing iForest (Table 6.5) and LOF (Table 6.6).

iForest: in terms of RMSE, the lowest value found for the 12-24 hours time interval implemented by A3T-GCN-256 is 14.59 μg/m^3, which outperforms the best performance of LSTM (17.55 μg/m^3) and the best performance of GRU (17.02 μg/m^3) found in the same interval by 16.87% and 14.28%, respectively. Regarding MAE, the lowest value found for the 1-12 hours time interval implemented by A3T-GCN-256 is 12.14 μg/m^3, which outperforms the best performance of LSTM (12.68 μg/m^3) and the best performance of GRU (12.20 μg/m^3) found for the 12-24 hours time interval by 4.26% and 0.49%, respectively. Regarding R, the highest value is 0.70 found for the 1-12 hours time interval implemented by A3T-GCN-64

and for the 1-12 and 12-24 hours time interval implemented by A3T-GCN-256, which outperforms the best performance of LSTM (0.65) and the best performance of GRU (0.66) found for the 12-24 hours time interval by 7.14% and 5.71%, respectively.

LOF: In terms of RMSE, the lowest value found for the 1-12 hours time interval implemented by A3T-GCN-128 is 14.98 μg/m^3, which outperforms the best performance of LSTM (18.26 μg/m^3) found for the 12-24 hours time interval and the best performance of GRU (18.10 μg/m^3) found for the 12-24 hours time interval by 17.96% and 17.24%, respectively. Regarding MAE, the lowest value found for the 1-12 hours time interval implemented by A3T-GCN-128 is 12.48 μg/m^3, which outperforms the best performance of LSTM (13.20 μg/m^3) and the best performance of GRU (12.92 μg/m^3) found for the 12-24 hours time interval by 5.45% and 3.41%, respectively. Regarding R, the highest value is 0.72 found for the 1-12 hours time interval implemented by A3T-GCN-128, which outperforms the best performance of LSTM (0.65) found for the 1-12 hours time interval and the best performance of GRU (0.66) found for the 1-12 and 12-24 hours time interval by 9.72% and 8.33%, respectively.

The results clearly show the advantage of implementing outlier detection techniques. Comparing the results of Table 6.4 and Table 6.5, the outlier detection technique, iForest, improves A3T-GCN in terms of RMSE by 10.71%, in terms of MAE by 8.38%; LSTM in terms of RMSE by 6.50%, in terms of MAE by 7.92%; GRU in terms of RMSE by 10.94%, in terms of MAE by 9.23%. Regarding Table 6.4 and Table 6.6, the outlier detection technique, LOF, improves A3T-GCN in terms of RMSE by 8.32%, in terms of MAE by 5.81%; LSTM in terms of RMSE by 2.72%, in terms of MAE by 4.14%; GRU in terms of RMSE by 5.29%, in terms of MAE by 3.87%. Regarding R the results were not changed a lot, in particular in the case of iForest, A3T-GCN was reduced by 2.78%, and in the case of LOF it was equal (0.72); LSTM was reduced by 4.41%, and GRU was reduced by 2.94% after implementing both outlier detection techniques.

Overall, it can be seen that the proposed approach, A3T-GCN, outperformed the reference methods in all scenarios. It is important to point out shortcomings in the period of the data sets used in this analysis, in particular with respect to the test set selected from January to June 2020, which was significantly affected by the restrictions implemented to suppress the advance of COVID-19. The data were selected for this period due to the availability of data at the time of the experiments.

Table 6.5: Performance evaluation metrics of Attention Temporal Graph Convolutional Network, Long Short-Term Memory and Gated Recurrent Unit in terms of time granularities after outliers detection with Isolation Forest.

Method	**Hours**	**RMSE** (μg/m^3)	**MAE** (μg/m^3)	**R**
A3T-GCN-32	1-12	14.74	12.36	0.69
	12-24	15.59	13.21	0.68
	24-36	16.10	13.57	0.62
	36-48	15.98	13.61	0.64
A3T-GCN-64	1-12	14.68	12.29	**0.70**
	12-24	15.25	12.89	0.67
	24-36	16.45	14.14	0.64
	36-48	16.09	13.72	0.64
A3T-GCN-128	1-12	14.90	12.51	0.69
	12-24	15.21	12.89	0.67
	24-36	15.94	13.47	0.64
	36-48	16.30	14.01	0.63
A3T-GCN-256	1-12	14.60	**12.14**	**0.70**
	12-24	**14.59**	13.48	**0.70**
	24-36	16.05	13.63	0.63
	36-48	16.41	14.12	0.63
LSTM	1-12	17.92	13.58	0.64
	12-24	**17.55**	**12.68**	**0.65**
	24-36	19.32	14.12	0.62
	36-48	18.50	13.73	0.63
GRU	1-12	18.57	13.85	0.61
	12-24	**17.02**	**12.20**	**0.66**
	24-36	18.34	13.79	0.61
	36-48	19.13	13.94	0.61

Additionally, in our recent paper on the A3T-GCN implementation for NO_2 prediction [127], the analysis was carried out using January-June 2019 and January-June 2022. The proposed approach was compared to Temporal Graph Convolutional

Table 6.6: Performance evaluation metrics of Attention Temporal Graph Convolutional Network, Long Short-Term Memory and Gated Recurrent Unit in terms of time granularities after outliers detection with Local Outlier Factor.

Method	**Hours**	**RMSE** (μ**g/m^3**)	**MAE** (μ**g/m^3**)	**R**
A3T-GCN-32	1-12	15.60	13.16	0.70
	12-24	15.57	13.12	0.68
	24-36	16.37	13.81	0.65
	36-48	16.43	13.88	0.64
A3T-GCN-64	1-12	15.32	12.81	0.70
	12-24	15.39	12.91	0.68
	24-36	17.04	14.51	0.63
	36-48	16.86	14.33	0.63
A3T-GCN-128	1-12	**14.98**	**12.48**	**0.72**
	12-24	15.89	13.39	0.67
	24-36	16.67	14.14	0.64
	36-48	16.89	14.39	0.64
A3T-GCN-256	1-12	15.48	12.93	0.71
	12-24	16.04	13.48	0.67
	24-36	16.43	13.89	0.65
	36-48	16.60	14.13	0.65
LSTM	1-12	18.47	13.84	**0.65**
	12-24	**18.26**	**13.20**	0.64
	24-36	19.73	14.61	0.62
	36-48	18.57	13.46	0.64
GRU	1-12	**18.10**	13.23	**0.66**
	12-24	18.24	**12.92**	**0.66**
	24-36	20.20	14.81	0.59
	36-48	18.98	13.60	0.62

Network (TGCN), LSTM and GRU. The comparison results also emphasised the superiority of A3T-GCN over the defined reference methods (Table 6.7).

First of all, it can be seen that for the A3T-GCN, TGCN and LSTM models, the

Table 6.7: Performance evaluation metrics of Attention Temporal Graph Convolutional Network, Temporal Graph Convolutional Network, Long Short-Term Memory and Gated Recurrent Unit in terms of time granularities.

Method	**Hours**	**RMSE (μg/m^3)**	**MAE (μg/m^3)**	**R**
A3T-GCN	1-12	**19.14**	**15.33**	**0.59**
	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	**21.48**	**16.24**	**0.49**
	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	**22.33**	**16.70**	**0.57**
	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	**0.56**
	12-24	**22.29**	**16.97**	0.54
	24-36	25.38	19.41	0.47
	36-48	23.45	17.45	0.31

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 μg/m^3) outperformed TGCN (21.48 μg/m^3) by 10.89%, LSTM (22.33 μg/m^3) by 14.29%, and GRU (22.29 μg/m^3) by 14.13%. In terms of MAE, the A3T-GCN (15.33 μg/m^3) outperformed TGCN (16.24 μg/m^3) by 5.6%, LSTM (16.70 μg/m^3) by 8.2%, and GRU (16.97 μg/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 with the unit of the target variable (NO_2: µg/m^3). Therefore, based on the results obtained (RMSE-19.14 µg/m^3, MAE-15.33 µg/m^3), the proposed method can be considered sufficient compared with the mean values of 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 μg/m^3) outperformed LSTM(22.33 μg/m^3) by 3.81%, and GRU (22.29 μg/m^3) by 3.63%. In terms of MAE, TGCN (16.24 μg/m^3) outperformed LSTM (16.70 μg/m^3) by 2.75%, and GRU (16.97 μg/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.

6.3 Summary

The main goal of this chapter is to predict NO_2 by implementing A3T-GCN on the data from Madrid air quality monitoring stations combined with meteorological and traffic data from the period of January-June 2019 (training set) and January-June 2020 (testing set). The proposed method was implemented with four different hidden units: 32, 64, 128, and 256, and it was compared to the following reference methods: LSTM and GRU. Another important contribution was the implementation of outliers detection techniques (iForest, LOF) and comparing the final results before detecting outliers and after detecting and handling outliers.

The results highlighted the superiority of the proposed method over the reference methods. In particular, before outlier detection, in terms of RMSE, the best results were obtained when experiments were carried out with hidden units equal to 128 (16.34 μg/m^3 in the 1-12 h time interval), and it outperforms LSTM by 12.95% and GRU by 14.50%. In terms of MAE, the best results were obtained when experiments were carried out with hidden units equal to 128 (13.25 μg/m^3 in the 12-24 h time interval), and it outperforms LSTM by 3.78% and GRU by 1.41%. In terms of R, the best results were obtained when experiments were carried out with hidden units equal to 256 (0.72 in the 1-12 h time interval), and it outperforms

LSTM and GRU by 5.56%.

After implementing iForest, in terms of RMSE, the best results were obtained when experiments were carried out with hidden units equal to 256 (14.59 μg/m^3 in the 12-24 h time interval), and it outperforms LSTM by 16.87% and GRU by 14.28%. In terms of MAE, the best results were obtained when experiments were carried out with hidden units equal to 256 (12.14 μg/m^3 in the 1-12 h time interval), and it outperforms LSTM by 4.26% and GRU by 0.49%. In terms of R, the best results were obtained when experiments were carried out with hidden units equal to 64 (0.70 in the 1-12 h time interval) and 256 (0.70 in the 1-12 h and 12-24 h time intervals), and it outperforms LSTM by 7.14% and GRU by 5.71%.

After implementing LOF, in terms of RMSE, the best results were obtained when experiments were carried out with hidden units equal to 128 (14.98 μg/m^3 in the 1-12 h time interval), and it outperforms LSTM by 17.96% and GRU by 17.24%. In terms of MAE, the best results were obtained when experiments were carried out with hidden units equal to 128 (12.48 μg/m^3 in the 1-12 h time interval), and it outperforms LSTM by 5.45% and GRU by 3.41%. In terms of R, the best results were obtained when experiments were carried out with hidden units equal to 128 (0.72 in the 1-12 h time interval), and it outperforms LSTM by 9.72% and GRU by 8.33%.

In the case of the implementation of A3T-GCN with the hidden units equal to 256 (hidden units were fixed) to the data of the periods of January-June 2019 and January-June 2022, in terms of RMSE, the best result of A3T-GCN (19.14 μg/m^3 in the 1-12 h time interval) outperforms TGCN by 10.89%, LSTM by 14.29% and GRU by 14.13%. In terms of MAE, the best result of A3T-GCN (15.33 μg/m^3 in the 1-12 h time interval) outperforms TGCN by 5.6%, LSTM by 8.2% and GRU by 9.7%. In terms of R, the best result of A3T-GCN (0.59 in the 1-12 h time interval) outperforms TGCN by 16.95%, LSTM by 3.39% and GRU by 5.08%.

Chapter 7

Conclusions and Future work

Considering the impact of air quality on people's health and the environment, its control and improvement have become essential tasks. One way to achieve these stated goals is to predict air quality more accurately by applying ML techniques. The important characteristic to consider when selecting and implementing specific ML technique is the ability to capture and compute the multidimensional information and interconnections producing air pollution, which, in particular, exist in the spatiotemporal dimensions. The study, development, validation, and evaluation of models focused on the prediction of air quality through processing the spatiotemporal dependencies are the primary target of this dissertation.

Steps towards achieving the main objectives of the work began with the exploration and study of the state-of-the-art, identifying current trends, as well as the existing gaps in the subject area which were reported in Chapter 2. The final selected papers underwent the review process by extracting and comparing the following features: *Year*, *Study Area*, *Prediction Target*, *Dataset Type*, *Data Rate*, *Period (Days)*, *Open Data*, *Algorithm* and *Time Granularity*. One of the main observations is that the most important datasets, apart from air quality used for air quality forecasting, are meteorological, temporal, spatial, and traffic datasets (Figure 2.2), which can be explained by the strong relationships and impact of these datasets on air quality formation. Another interesting observation is the presence of spatial and temporal datasets that highlights the spatiotemporal dependencies and behaviour of air quality. Moreover, it can be seen, that the number of publications has increased recently, which can be explained by several factors, in particular, due to the new stage since 2012 associated with the use of open

data portals [128]. Also, with the help of new advancement methods, it becomes possible to conduct more accurate predictive analysis and, in parallel with the development of technology, it becomes possible to observe finer particles that are more hazardous to health.

Afterwards, the information and knowledge obtained were applied to the study area, which in this work is the city of Madrid. Based on the findings, air quality, meteorological, and traffic data, and the location of air quality and meteorological monitoring stations and traffic measurement points from January to June 2019 and from January to June 2020 were used in this work. Regarding the temporal dataset, only the chronological behaviour or temporal dependency of the selected datasets was taken into account. Considering the research questions, the data were incorporated in spatiotemporal dimensions which were followed by exploratory data analysis with the aim of disclosing the existing relationships between different features. It was revealed that the study area's air quality stations have both spatial and temporal correlations in terms of NO_2 concentrations. Regarding the relationships between the NO_2 and the rest of the features, the most relevant feature turned out to be wind speed. Feature engineering techniques and the proposed predictive methods, including ConvLSTM, BiConvLSTM and A3T-GCN, were introduced in Chapter 3.

The development and implementation procedure of ConvLSTM is placed in Chapter 4. This method was implemented during different periods: pandemic and non-pandemic periods, in order to predict NO_2. Additionally, different temporal granularities (1-hour, 12-hour, 24-hour and 48-hour) were provided using the historical NO_2 and meteorological data. The results highlighted the superiority of ConvLSTM over the reference method in terms of periods, time granularity and selected features (best results in terms of scenarios: [First Scenario-non-pandemic: ConvLSTM-13.46 μg/m^3-1 hour; LSTM-27.94 μg/m^3-1 hour; First Scenario-pandemic: ConvLSTM-11.55 μg/m^3-1 hour; LSTM-32.0 μg/m^3-24 hours; Second Scenario-non-pandemic: ConvLSTM - 1.46 μg/m^3-1 hour; LSTM-1.51 μg/m^3-1 hour; Second Scenario-pandemic: ConvLSTM-1.22 μg/m^3-1 hour; LSTM-1.46 μg/m^3-1 hour]).

Chapter 5 proposed and developed another advanced method named BiConvLSTM. An additional focus of this chapter is the implementation of feature selection techniques in order to select the minimal relevant features with the most optimal combination. The results showed that the proposed method outperformed

the reference models both before selecting features and after selecting relevant features. Regarding feature selection methods, the implementation of mRMR yields better results compared to MI, given the fact that mRMR, in addition to selecting relevant features, selects the next relevant feature that has a minimum correlation with previously selected features. With respect to runtime, BiConvLSTM is slower compared to reference methods due to the model architecture, and it takes longer to converge the data.

The A3T-GCN method was developed and described in the next chapter, Chapter 6. Additionally, given that outliers can negatively impact model performance, this chapter proposes outlier detection methods, including iForest and LOF, and carries out experiments with and without outliers. The results highlighted the superiority of A3T-GCN over the reference methods, and the importance of the implementation of outlier detection techniques to improve the models' performance (best results in terms of evaluation metrics: [RMSE: 16.34 μg/m^3 (A3T-GCN with 128 units before outliers detection), 14.59 μg/m^3 (A3T-GCN with 256 units after iForest), 14.98 μg/m^3 (A3T-GCN with 128 units after LOF); MAE: 13.25 μg/m^3 (A3T-GCN with 128 units before outliers detection), 12.14 μg/m^3 (A3T-GCN with 256 units after iForest), 12.48 μg/m^3 (A3T-GCN with 128 units after LOF); R: 0.72 (A3T-GCN with 256 units before outliers detection), 0.70 (A3T-GCN with 64 and 256 units after iForest), 0.72 (A3T-GCN with 128 units after LOF)].

To highlight the contribution and the improvement of our work to the domain is not straightforward, as it is challenging to compare our proposed methodology with other works, as each work's proposed methodology is implemented in different datasets with different scenarios. However, several improvements can be mentioned. First, it refers to the prediction target, which in our case is NO_2. From Table B.1, Table B.2 and Figure 2.5, it can be seen that the dominant prediction target is $PM_{2.5}$ (ML-based methods: $PM_{2.5}$ in forty-eight papers, NO_x in sixteen papers, GNN-based methods: $PM_{2.5}$ in fourteen papers, NO_2 in two papers). Therefore, there are many studies dedicated to the prediction of $PM_{2.5}$ rather than NO_2. Moreover, out of these studies, only four works are focused on the spatiotemporal prediction of NO_2([48, 63, 129, 130]). To forecast the concentration of NO_2 might be more important and concerning due to finer spatial heterogeneity compared to $PM_{2.5}$, in particular, it is more concerning for our study area [68]. Compared to these four works, the improvement of our work is the involvement and incorporation of meteorological and traffic data in addition to NO_2. Particularly, in

the following work [48] the authors applied GNN-based method on data consisting of air quality, location of the stations and temporal data, another difference is that the authors constructed an unweighted graph, i.e., the edges were not weighted. The study by Huang et al. [63] also performed spatiotemporal analysis without considering traffic data, although it should be noted that they used POI data, which may be considered in future work. The work done by Chen et al. [130] did not include meteorological data and as traffic data was mentioned calculated road lengths within each grid cell. The fourth work [129] focused on the spatiotemporal prediction of NO_2 used air quality, meteorological, traffic, and additional geographic features. However, traffic data features were different from our traffic datasets, they included traffic-related NO_x, traffic density, and distance to major roadways. Regarding methodology, the authors proposed a clustering-enhanced ensemble machine learning approach, however, the implementation had several limitations, such as the difference in the size of the samples, and the uneven distribution of sampling locations.

Another important factor is the consideration of COVID-19 for predictive analysis. Due to traffic restrictions and self-quarantine measures to control and curb the COVID-19, air pollution decreased dramatically, for instance, the concentration of NO_2 in the city of Madrid dropped to 62% [109]. Existing forecasting models can be significantly impacted by these abrupt changes in air quality levels. As the systematic review includes the paper published before 28 September 2020, by that time there was no paper focusing on NO_2 prediction using machine learning methods and considering COVID-19. The next factor refers to the proposed methods, the current research is the first to propose the BiConvLSTM and A3T-GCN implementation for air quality prediction, and the ConvLSTM implementation for NO_2 prediction, making this research a groundbreaking contribution to the domain of air quality prediction. An additional contribution of this work is the provision of the code and data implemented in the scope of this dissertation, given the importance of reproducibility in science, which is presented in Appendix C.

Applying these methods to the control components specified for the current analysis (air quality, meteorological, and traffic data), the results demonstrate that ConvLSTM and BiConvLSTM outperformed the reference methods defined for each experiment. Regarding A3T-GCN, which is able to capture non-Euclidean dependencies, outperformed defined reference methods (LSTM, GRU). It is difficult to compare A3T-GCN with ConvLSTM and BiConvLSTM due to differences

in models' architecture. For example, if we compare the results of the first sub-scenario with all features included in Table 5.4 (applied BiConvLSTM) with Table 6.4 (applied A3T-GCN), it can be seen that the results are close. In terms of RMSE, BiConvLSTM is higher (18.99 μg/m^3), and in terms of MAE it is lower (12.89 μg/m^3). However, such a comparison is not sufficiently justified, since for the calculation of BiConvLSTM we used data from the entire grid, while in the case of A3T-GCN only data from the cells containing air quality stations were included.

The choice of one or another method can be determined by the defined task: if the main idea is to predict air quality over the entire grid, then grid-based methods can be used, and if the main goal is to predict only for air quality stations, the graph-based approach is more efficient. Regarding feature engineering techniques, this stage is very essential before performing predictive analysis. Moreover, the results highlighted the benefits of feature engineering techniques, in particular the importance of selecting the most relevant features and the data cleansing procedure in terms of outlier detection.

Regarding future work, the proposed methodology could be applied to another study area to evaluate how performance varies depending on location peculiarities. The final performance is likely to be affected by the spatial aspects of different regions, the distance between stations, the number of stations, and the available features. Another extension could be the integration of other datasets, such as aerosol optical depth, land use, population density, and street networks, as well as including these features over a longer period.

Additionally, the proposed procedure can be applied to other pollutants other than NO_2. The accuracy of the predictive analysis may vary depending on the selected pollutant based on the chemical structure of the pollutants. For example, Li et al. [38] showed that the proposed model predicts better $PM_{2.5}$ than NO_x, due to the high reactivity and greater temporal variability of NO_x.

Regarding the architecture of the proposed models, further modifications can be made, for example, in the case of A3T-GCN, several layers can be stacked. Since the complexity of the architecture causes a relatively long execution time, in parallel with improving the accuracy, the execution time must also be taken into account.

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_{i} on V_{j} is different from that of V_{j} on V_{i}. 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.

Also, it should be noted that 2020 was a year with certain peculiarities, namely the COVID-19 pandemic and its consequences, including traffic restrictions and self-isolation. Therefore, it would be ideal to choose a period other than 2020 in order to avoid the impact of COVID-19 on the analyses. It is important to mention that the reason for choosing the data for the periods January-June 2019 and January-June 2020 is conditioned by the fact that the meteorological data at the start of the experiments were available only for those years[1]. Furthermore, in our recent paper, considering the impact of COVID-19, the data to use in the experiments were acquired for the periods of January-June 2019 and January-June 2022. The paper is devoted to the implementation of the A3T-GCN to predict NO_2 [127]. The comparison between the proposed approach with the reference methods (TGCN, LSTM and GRU) also emphasised the superiority of A3T-GCN over the defined reference methods.

[1]Meteorological data. Hourly data from 2019: `https://bit.ly/3hz4nn6`. [Online; accessed 15-February-2023]

Appendix A

Publications

A.1 Related thesis topic

- **Journals with impact:**

    1. **Ditsuhi Iskandaryan**, Francisco Ramos, Sergio Trilles. *Air quality prediction in smart cities using machine learning technologies based on sensor data: a review*. Applied Sciences 10.7 (2020): 2401. JCR: Impact factor (**Q2**). Related chapter 2.
    2. **Ditsuhi Iskandaryan**, Francisco Ramos, Sergio Trilles. *Features Exploration from Datasets Vision in Air Quality Prediction Domain*. Atmosphere 12.3 (2021): 312. JCR: Impact factor (**Q2**). Related chapter 2.
    3. **Ditsuhi Iskandaryan**, Francisco Ramos, Sergio Trilles. *Comparison of Nitrogen Dioxide Predictions During a Pandemic and Non-pandemic Scenario in the City of Madrid using a Convolutional LSTM Network*. International Journal of Computational Intelligence and Applications (2022): 2250014. JCR: Impact factor (**Q3**). Related chapters 3 and 4.
    4. **Ditsuhi Iskandaryan**, Francisco Ramos, Sergio Trilles. *Bidirectional convolutional LSTM for the prediction of nitrogen dioxide in the city of Madrid*. PloS one 17.6 (2022): e0269295. JCR: Impact factor (**Q1**). Related chapters 3 and 5.
    5. **Ditsuhi Iskandaryan**, Francisco Ramos, Sergio Trilles. *Graph Neural Network for Air Quality Prediction: A Case Study in Madrid*. IEEE

Access 11 (2023): 2729-2742. JCR: Impact factor (**Q1**). Related chapters 2, 3 and 6.

6. **Ditsuhi Iskandaryan**, Francisco Ramos, Sergio Trilles. *Reconstructing Secondary Data based on Air Quality, Meteorological and Traffic Data Considering Spatiotemporal Components*. Data in Brief (2023). SJR: Impact factor (**Q4**). Related chapter 3.

- **Conferences:**

1. **Ditsuhi Iskandaryan**, Silvana Di Sabatino, Francisco Ramos, Sergio Trilles. *Exploratory Analysis and Feature Selection for the Prediction of Nitrogen Dioxide*. AGILE: GIScience Series 3 (2022): 6. Related chapters 3 and 5.
2. **Ditsuhi Iskandaryan**, Francisco Ramos, Sergio Trilles. *Spatiotemporal Prediction of Nitrogen Dioxide Based on Graph Neural Networks*. Advances and New Trends in Environmental Informatics: Environmental Informatics and the UN Sustainable Development Goals. Cham: Springer International Publishing (2022): 111-128. Related chapters 2, 3 and 6.

- **Book chapter:**

1. **Ditsuhi Iskandaryan**, Francisco Ramos, Sergio Trilles. *Application of deep learning and machine learning in air quality modeling*. Current Trends and Advances in Computer-Aided Intelligent Environmental Data Engineering. Academic Press, 2022. 11-23. Related chapter 1.

A.2 Non-related thesis topic

- **Journals with impact:**

1. **Ditsuhi Iskandaryan**, Francisco Ramos, Denny Asarias Palinggi, Sergio Trilles. *The effect of weather in soccer results: an approach using machine learning techniques*. Applied Sciences 10.19 (2020): 6750. JCR: Impact factor (**Q2**).

- **Conferences:**

    1. Francisco Ramos, **Ditsuhi Iskandaryan**, Iva Koribska. *DATA VISUALISATION FOR TEACHERS: HOW TO READ, INTERPRET AND SHOW DATA CORRECTLY*, EDULEARN22 Proceedings. IATED (2022): 8022-8022.
    2. Francisco Ramos, **Ditsuhi Iskandaryan**, Águeda Gómez-Cambronero. *IMPROVING TEACHERS VISUAL PRESENTATIONS WITH SIMPLICITY, CLARITY AND BREVITY*, EDULEARN19 Proceedings. IATED (2019): 6218-6218.

Appendix B

Features of the selected papers

Table B.1: Features of the papers dedicated to the implementation of Machine Learning for air quality prediction. N/S: Not Specified. Published in Zenodo [131].

Work	Year	Study Area	Prediction Target	Dataset Type	Data Rate	Period (Days)	Open Data	Algorithm	Time Granularity	Evaluation Metric
[35]	2020	USA	$PM_{2.5}$	Spatial, Temporal, AOD, PBL Height	Daily	5779	No	Hybrid	24 h	RMSE, SD, R^2
[132]	2020	Canada	UFP	MET, Traffic, Land Use, BEV	N/S	120	No	Ensemble		RMSE, R^2
[133]	2020	Taiwan	$PM_{2.5}$, PM_{10}	MET	N/S	2192	No	Hybrid	8 h	RMSE, MAE
[38]	2020	China	$PM_{2.5}$, NO_x	MET, Traffic	Hourly	731	No	Regression, Ensemble	1 h	RMSE, ME, NRMSE, NME, POD, POF, R^2
[26]	2020	USA	$PM_{2.5}$	MET, Temporal	Hourly	730	No	NN		RMSE, MAE, MAPE
[41]	2020	India	$PM_{2.5}$	MET	Hourly	1230	No	NN		RMSE, R^2
[134]	2020	USA	AQI	MET	Hourly	851	Yes	Regression	1 h	RMSE, MAE, NRMSE, R
[135]	2020	Turkey	PM_{10}	Spatial, Land Use	N/S	3652	No	Regression, Ensemble, NN		RMSE, MAE, R^2
[136]	2020	China	$PM_{2.5}$	MET	Hourly	31	Yes	NN	1 h	RMSE, R
[86]	2020	China	AQHI, IAQL	MET, Temporal	Hourly	730/1826	Yes	Ensemble	12 h	Acc, MSE, WP, WR, WF
[137]	2020	China	PM_{10}	MET	Daily	1096	No	NN	24 h	RMSE, ME, R, EOp
[36]	2020	Tunisia, Italy		MET, Temporal	Hourly	1461/366	No	Ensemble	1 week	aRRMSE, aRMSE, R^2, aCC, MSE, aRE, RP
[37]	2020	China	$PM_{2.5}$	MET	N/S	46	Yes	Ensemble	24 h	RMSE, MAE, SMAPE
[40]	2020	China	$PM_{2.5}$	MET	Hourly	1825	No	NN	1 week	RMSE
[138]	2020	China	$PM_{2.5}$	MET	N/S	1096	Yes	NN	24 h	RMSE, MAE, MAPE
[45]	2020	China	O_3	MET, UV Index	Daily	1491	Yes	Hybrid	1 week	RMSE, MAE, MAPE, IA
[139]	2020	South Korea	$PM_{2.5}$, PM_{10}	MET	Hourly	1461	Yes	Hybrid	15days	RMSE, MAE

Table B.1: (cont)

Work	Year	Study Area	Prediction Target	Dataset Type	Data Rate	Period (Days)	Open Data	Algorithm	Time Granularity	Evaluation Metric
[39]	2020	China	$PM_{2.5}$, PM_{10}, NO_2, NO, CO	MET	Daily	4656	No	NN	24 h	MSE
[140]	2020	Taiwan	$PM_{2.5}$	MET, Spatial, Temporal	Hourly	365	Yes	Ensemble	24 h	RMSE, NRMSE, R^2
[141]	2020	UK	$PM_{2.5}$	MET, Spatial, Temporal, AOD, Land Use	Daily	3287	Partially	Ensemble	24 h	RMSE, MSE, R^2
[142]	2020	Ecuador	$PM_{2.5}$	MET, Spatial, Temporal, Traffic	5 s	4	No	Other Algorithms		Acc
[143]	2020	China	$PM_{2.5}$	MET	Hourly	365	No	Ensemble	48 h	MSE, IA, NMGE, R^2
[144]	2020	China	$PM_{2.5}$	MET	Hourly	1461	No	Ensemble	24 h	RMSE, MB, ME, R
[145]	2020	China	AQI	MET	Hourly	2192	No	NN	48 h	RMSE, Acc
[146]	2020	China	AQI	MET	Hourly	730	Yes	NN	24 h	RMSE, MAE, R^2, FB
[147]	2020	South Korea	$PM_{2.5}$, PM_{10}	MET, Temporal, Spatial	Minutely	7	No	Hybrid		RMSE
[148]	2020	China	$PM_{2.5}$, PM_{10}, O_3, NO_2, SO_2, CO	MET, Social Media	Daily	731	Yes	NN	24 h	RMSE, MAE
[149]	2020	Thailand	PM_{10}	MET	Secondly	59	No	NN	1 h	RMSE, MAE, MAPE, R
[87]	2020	China	AQI	Spatial	Daily	1086	Yes	Hybrid	5 days	RMSE, MAE, MAPE, R
[150]	2020	Germany	CO_2, NH_3, NO, NO_2, NO_x, O_3, PM_1, $PM_{2.5}$, PM_{10}, PN_{10}	MET, Temporal, Traffic, SP	Hourly	62	No	NN	1 h	RMSE, R, NMB, NMSD, RS, SD, SD′
[42]	2020	Mongolia	$PM_{2.5}$	MET, Temporal, Land Use, PD	Hourly	2922	No	Regression, Ensemble	24 h	RMSE, R^2
[43]	2020	Taiwan	$PM_{2.5}$	MET, Temporal, Spatial	Hourly	2192	No	NN	8 h	RMSE, MAE, MAPE
[151]	2020	Turkey	PM_{10}	MET	Daily	766	No	Regression, NN		RMSE, MAE, R^2
[152]	2020	Jordan	O_3	MET, Temporal	Daily	1496	No	NN, Regression, Ensemble	24 h	RMSE, MAE, R^2

Table B.1: (cont)

Work	Year	Study Area	Prediction Target	Dataset Type	Data Rate	Period (Days)	Open Data	Algorithm	Time Granularity	Evaluation Metric
[153]	2019	South Korea	PM_{10}, $PM_{2.5}$	MET, Spatial, Human Movements	Hourly	115	No	NN, Regression	1 h	RMSE, R^2
[154]	2019	China/ Taiwan	$PM_{2.5}$	MET	Hourly	3693	No	NN, Other Algorithms	5 days	RMSE
[155]	2019	South Korea	O_3	MET	Hourly	1096	No	Ensemble	24 h	IA
[129]	2019	USA	NO_2, NO_x	MET, Spatial, Traffic	biweekly	8023	No	Ensemble		RMSE, R^2, RMSEIQR
[130]	2019	Europe	NO_2, $PM_{2.5}$	AOD, Traffic, Land Use, Altitude	N/S	365	Yes	Regression, Ensemble, NN		RMSE, R^2, MSE-R^2
[156]	2019	China	$PM_{2.5}$	MET, AOD	Hourly	1096	Yes	Hybrid	24 h	RMSE, R^2
[157]	2019	China	SO_2	MET, Temporal, Land Use, OMI-SO_2, PPS, TS	Daily	365	Partially	Hybrid	24 h	RMSE, R^2, RPE
[158]	2019	China	$PM_{2.5}$	MET	Hourly	731	No	NN	3 h	RMSE
[159]	2019	China	$PM_{2.5}$	MET, WFD, Spatial	N/S	61	No	Ensemble	24 h	MAE, SMAPE, MSE
[160]	2019	China	$PM_{2.5}$	MET	Hourly	1826	Yes	NN	2 h	RMSE, MAE, SMAPE
[161]	2019	China	$PM_{2.5}$	MET	N/S	2191	Yes	Ensemble	1 week	RMSE, MAE
[162]	2019	Italy	CO(GT), NO_2(GT)	MET	Hourly	183	Yes	NN	1 h	RMSE, MAE, MAPE
[163]	2019	China	$PM_{2.5}$	Spatial	Hourly	365	No	NN	1 week	RMSE, MAE, MAPE
[164]	2019	China	AQI	MET, WFD, Traffic, POI Distribution, FAPE, RND	Hourly	366	Yes	NN	48 h	MAE, MAP
[165]	2019	Taiwan	$PM_{2.5}$	MET	Hourly	2557	No	Hybrid	4 h	RMSE, G_{bench}
[166]	2019	Iran	$PM_{2.5}$	MET	Hourly	1826	No	Ensemble, NN, Hybrid	48 h	RMSE, MAE, R^2
[167]	2019	Poland	NO_2	MET, Temporal, Traffic	Hourly	731	No	Ensemble		MAPE, MADE, BIC, R^2
[168]	2019	India	O_3, $PM_{2.5}$, NO_x, CO	MET, Traffic	Hourly	730	No	NN		RMSE, NSE, PBIAS, R
[169]	2019	China	$PM_{2.5}$	MET	Hourly	1826	No	NN	72 h	RMSE, IA, MAE, R
[46]	2019	China	$PM_{2.5}$	MET	Hourly	366	No	NN	10 h	RMSE, NRMSE, MAE, SMAPE, R

Table B.1: (cont)

Work	Year	Study Area	Prediction Target	Dataset Type	Data Rate	Period (Days)	Open Data	Algorithm	Time Granularity	Evaluation Metric
[170]	2019	China	$PM_{2.5}$	MET, AOD	N/S	730	Yes	NN		RMSE, MAE, MSE, R^2
[171]	2019	Iran	$PM_{2.5}$	MET, Temporal, Spatial, AOD, Altitude	Daily	1460	Yes	Ensemble, NN		RMSE, MAE, R^2
[172]	2019	India	O_3	MET	Hourly	92	No	Ensemble		IoAd, R^2, PEP
[173]	2019	China	O_3	MET	Hourly	365	No	Ensemble, NN		RMSE, R, NMB, NME, MFB, MFE
[174]	2019	UK	SO_2	MET	Hourly	120	Yes	Ensemble		RMSE, MAE, R^2, RAE
[175]	2019	Taiwan	AQI	MET, Temporal	Hourly	851	No	Regression, NN	6 h	RMSE, MAE, R^2
[176]	2019	Iran	PM_{10}, $PM_{2.5}$	MET, Temporal, Spatial	Daily	3652	Yes	Regression, NN	1 week	RMSE, R^2
[177]	2018	China	$PM_{2.5}$	MET, Temporal, AOD	Hourly	731	Partially	NN	72 h	RMSE, MAE, MSE, IA, TPR, FPR, SI
[178]	2018	Slovenia	PM_{10}, O_3	MET, Temporal	Hourly	1461	No	Other Algorithms	24 h	MAE, RPS
[179]	2018	China	O_3	MET, Land Use, Elevation, AEI, NDVI, RND, PD	Hourly	365	Yes	Ensemble		RMSE, R^2, RPE
[180]	2018	China	$PM_{2.5}$	MET, AOD, Elevation, PD, RND, NDVI	Daily	1095	Yes	Ensemble	1 month	RMSE, R^2, RPE
[181]	2018	China	$PM_{2.5}$	MET, Spatial	Hourly	61	No	Regression	24 h	total accuracy index (pt), a total absolute error index (et)
[182]	2018	UK	AQI	MET	Hourly	605	Yes	NN		RMSE, MAPE, band Acc
[183]	2018	Kuwait	O_3	MET	Hourly	669	No	NN	72 h	RMSE, MAE
[184]	2018	Spain	O_3	MET	Hourly	730	Yes	Ensemble	24 h	RMSE, MAE, R^2
[185]	2018	Egypt	PM_{10}	MET, Temporal	Hourly	276	No	Regression	1 h	RMSE, R, t-Value

Table B.1: (cont)

Work	Year	Study Area	Prediction Target	Dataset Type	Data Rate	Period (Days)	Open Data	Algorithm	Time Granularity	Evaluation Metric
[186]	2018	China	$PM_{2.5}$	MET	Hourly	1826	No	NN	1 h	RMSE, MAE, IA, R
[187]	2018	USA	O_3, $PM_{2.5}$, SO_2	MET	Hourly	3652	Yes	Other Algorithms	24 h	RMSE
[188]	2017	USA	BC	MET, Spatial, Temporal	Daily	4383	Yes	Regression	24 h	R^2
[13]	2017	Canada	O_3, $PM_{2.5}$, NO_2	MET, Temporal	Hourly	1826	No	NN	48 h	MAE, R, ME, SS
[189]	2017	China	$PM_{2.5}$	MET, Social Media	Hourly	365	No	NN	24 h	RMSE
[190]	2017	Ecuador	$PM_{2.5}$	MET	Daily	1827	No	Ensemble, Regression, NN		MSE, MAPE
[191]	2017	China	$PM_{2.5}$	MET, Temporal, Spatial, AOD	Daily	365	Yes	Ensemble		RMSE, R^2
[192]	2017	Kuwait	PNCs	MET	5min	30	No	NN		RMSE, NRMSE, IA, R^2
[193]	2017	Egypt	PM_{10}	MET, Temporal	Hourly	368	No	Regression	1 h	RMSE, R, z', t-value
[194]	2017	China	NO_2, NO_x, O_3, $PM_{2.5}$, SO_2	MET, Temporal	Daily	2191	No	NN	24 h	RMSE, MAE, IA, R^2
[195]	2017	China	AQI	MET	Daily	851	No	Regression		RMSE, MAE, MAPE, MSE
[196]	2016	Qatar	O_3, NO_2, SO_2	MET, Temporal	15min	92	No	Regression	24 h	RMSE, NRMSE, PTA
[197]	2016	France	O_3, NO_2, PM_{10}	MET	Hourly	1733	No	Hybrid	24 h	RMSE, MAE, NRMSE, MBE, IA, R
[198]	2014	Saudi Arabia	PM_{10}	MET	Hourly	366	No	Regression	1 h	RMSE, MAE, MBE, FACT2, R, IA
[199]	2014	France	O_3, NO_2, PM_{10}	MET	Hourly	731	Yes	Ensemble	72 h	RMSE
[20]	2013	China	$PM_{1.0}$, UFP	MET, Traffic, Temporal	Minutely	3	No	Regression, Ensemble, NN		AUC, R, R^2, Precision, Recall, f measure, weighted f-measure
[200]	2013	Greece	O_3	MET	Hourly	7305	No	NN	6 h	RMSE, R^2, R
[201]	2013	India	AQI	MET	Daily	1825	Partially	Ensemble		RMSE, MAE, R

Table B.1: (cont)

Work	Year	Study Area	Prediction Target	Dataset Type	Data Rate	Period (Days)	Open Data	Algorithm	Time Granularity	Evaluation Metric
[202]	2012	China	SPM, SO_2, NO_2, O_3	MET	Daily	1095	Yes	Regression	24 h	RMSE, MAE, CWIA, RE
[203]	2012	Iran	CO	MET	Hourly	1492	No	Hybrid	24 h	RMSE, RME, MARE, R^2
[204]	2012	Saudi Arabia	O_3	MET, Temporal	Minutely	183	No	NN, Ensemble	1 h	MAE, MAPE, SD, MD, R
[205]	2009	Europe	O_3	MET, Land Data, Chemical, Emission	Hourly	120	No	Ensemble	24 h	RMSE
[206]	2008	China	RSP(PM_{10}), NO_x, SO_2	MET	Hourly	61	No	Regression	1 week	RMSE, MAE, WIA

Table B.2: Features of the papers dedicated to the implementation of Graph Neural Network for air quality prediction (*).

Work	**Year**	**Method**	**Edge Weight**	**Dynamic/ Static**	**Directed/ Undirected**	**Target**	**Dataset**	**Evaluation Metric**
[47]	2022	SSH-GNN	Yes	Static	Undirected	AQI	AQ, MET, Spatial, Traffic, POI, RND	MAE, RMSE
[48]	2022	DGCN with Graph LSTM	No	Static	Directed	$PM_{2.5}$, PM_{10}, NO_2, CO, O_3	AQ, Spatial	MAE, RMSE, ACC
[49]	2022	DP-DDGCN	Yes	Dynamic	Directed	$PM_{2.5}$	AQ, MET, Spatial	MAE, RMSE
[50]	2021	ST-DGCN	Yes	Dynamic	Undirected	$PM_{2.5}$	AQ, MET, Spatial	MAE, RMSE
[51]	2021	MST-GCN	Yes	Static	Undirected	$PM_{2.5}$	AQ, MET, Spatial, POI, RND	ACC, MAE, RMSE
[52]	2021	ATGCN	Yes	Static	Undirected	Not Specified	AQ, MET, Spatial, POI	MAE, RMSE
[53]	2021	GAGNN	Yes	Static	Undirected	AQI	AQ, MET, Spatial	MAE, RMSE

* **Method**: *SSH-GNN*–Self-Supervised Hierarchical Graph Neural Network, *DGCN*–Dual Graph Convolution Network, *DP-DDGCN*–Dual-path Dynamic Directed Graph Convolutional Network, *ST-DGCN*–Spatial-Temporal 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; **Target**: *AQI*–Air Quality Index, $PM_{2.5}$–Particulate Matter with a diameter of less than 2.5 micrometres, PM_{10}–Particulate Matter with a diameter of less than 10 micrometres, NO_2–Nitrogen Dioxide, *CO*–Carbon Monoxide, O_3–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.

Table B.2: (cont)

Work	Year	Method	Edge Weight	Dynamic/ Static	Directed/ Undirected	Target	Dataset	Evaluation Metric
[54]	2021	HGNN	Yes	Dynamic	Directed	AQI	AQI, MET, Spatial, POI	MAE, RMSE
[55]	2021	Naive, LR, ARIMA, MLP, GCN, STGCN, Temporal, ASTGCN	Yes	Static	Undirected	$PM_{2.5}$	AQ, MET, Spatial	RMSE
[56]	2021	Spatiotemporal GCRNN	Yes	Static	Undirected	$PM_{2.5}$, PM_{10}	AQ, MET, Spatial, Traffic	RMSE, R^2, spRMSE
[57]	2021	AQSTN-GCN	Yes	Static	Undirected	$PM_{2.5}$	AQ, MET, Spatial	MAE, RMSE, MAPE, MSE, R^2
[58]	2021	GLSTM	Yes	Static	Directed	$PM_{2.5}$	AQ, MET, Spatial	RMSE, MAE, MAPE, IA
[63]	2021	SpAttRNN	Yes	Static	Undirected	$PM_{2.5}$, PM_{10}, NO_2	AQ, MET, Spatial, POI	RMSE, MAE, SMAPE

* **Method**: *HGNN*–Hierarchical Graph Neural Networks, *LR*–Linear Regression, *ARIMA*–Autoregressive Integrated Moving Average, *MLP*–Multilayer Perceptron, *GCN*–Graph Convolutional Neural Network, *STGCN*–Spatial-Temporal Graph Convolutional Network, *ASTGCN*–Attention based Spatial-Temporal Graph Convolution Network, *GCRNN*–Graph Convolutional Recurrent Neural Network, *AQSTN*–Air Quality Spatial-Temporal Network, *SpAttRNN*– Spatio-Attention embedded Recurrent Neural Network; **Evaluation Metric**: R^2– Coefficient of Determination, *spRMSE*–Spatiatemporal RMSE, *MAPE*–Mean Absolute Percentage Error, *MSE*–Mean Square Error, *IA*–Index of Agreement, *SMAPE*–Symmetric Mean Absolute Percentage Error.

Table B.2: (cont)

Work	Year	Method	Edge Weight	Dynamic/ Static	Directed/ Undirected	Target	Dataset	Evaluation Metric
[59]	2020	TA-GCN	Yes	Static	Directed	$PM_{2.5}$	AQ, MET, Spatial	Train_loss, Test_loss, Validation_loss, RMSE, MAE, CSI, POD, FAR
[60]	2020	PM2.5-GNN	Yes	Dynamic	Directed	$PM_{2.5}$	AQ, MET, Spatial	Train_loss, Test_loss, Validation_loss, RMSE, MAE, CSI, POD, FAR
[61]	2020	MASTGN	Yes	Static	Undirected	$PM_{2.5}$, PM_{10}, O_3	AQ, MET, Spatial	MAE, SMAPE
[62]	2019	GC-LSTM	Yes	Static	Undirected	$PM_{2.5}$	AQ, MET, Spatial	MAE, RMSE, IA, Recall Rate, FAR, R^2
[64]	2018	GC-DCRNN	Yes	Static	Undirected	$PM_{2.5}$	AQ, MET, Spatial, Geographic (land uses, roads, water areas, buildings)	MAE, RMSE

* **Method**: *TA-GCN*–Temporal Attention-GCN, *MASTGN*–Multi-attention Spatio-Temporal Graph Networks, *GC-LSTM*–Graph Convolution Network embedded Long Short-term Memory, *GC-DCRNN*–Geo-context based Diffusion Convolutional Recurrent Neural Network; **Evaluation Metric**: , *CSI*–Critical Success Index, *POD*–Probability of Detection, *FAR*–False Alarm Rate.

Appendix C

Reproducibility

One of the main objectives of this work is to support reproducibility, given the importance of this phenomenon. Considering the role of reproducibility in research, this section provides the instructions to reproduce the data and the code displayed in the directory tree format and file structure format (Figure C.1). The root directory named `Air_Quality_Prediction` is composed of two main sub-directories `Data` and `Code`, which is available at the Zenodo repository [207].

`Data` sub-directory consists of `Raw-Data` and `Processed-Data`. `Raw-Data` includes `AirQuality-Data`, `Meteorological-Data`, and `Traffic-Data`; `Processed-Data` includes `AirMetTraffic_2019_2020_firstSixMonths .zip`, `Madrid_wind_yyyy.csv`, `Madrid_Stations_yyyy.csv`, `Madrid_Exploration.zip`, and `distanceNodes.txt`.

`AirQuality-Data`: consists of `Anio201912.zip`, `Anio202012.zip` and `informacion_estaciones_red_calidad_aire.geo`. The first two $.zip$ files[1] contain hourly air quality data for 2019 and 2020, respectively. The data from January to June 2019 and from January to June 2020 were used in the current work. Each $.zip$ file contains data in three format: $.txt$, $.csv$, $.xml$. For our analysis, we used $.csv$ files. Each record is structured as follows (Table C.1; in the brackets English version of the columns):

The **POINT_SAMPLE** field includes the complete station code (province, municipality, station, magnitude and technique of sampling); **H01** corresponds to the data of 1 a.m. of that day; **V01** is the validation code; **H02** at 2 a.m.; **V02** and so on. **Magnitude** refers to the pollutants that were recorded by the stations, of which

[1] Air quality. Hourly data since 2001: `https://bit.ly/2IeGcrs`

Table C.1: Air Quality Data.

PROVINCIA (PROVINCE)	MUNICIPIO (MUNICIPALITY)	ESTACION (STATION)	MAGNITUD (MAGNITUDE)	PUNTO_MUESTREO POINT_SAMPLE	ANO YEAR	MES MONTH	DIA DAY	H01 H01	V01 V01	H02 H02	V02 V02
28	79	4	1	28079004_1_38	2019	1	1	23	V	17	V

we only focused on nitrogen dioxide, which is mentioned under magnitude 8[2].

The location of the air quality monitoring stations is available in $.csv$, $.xlsx$, and $.geo$ format[3]. This work used the $.geo$ format: `informacion_estaciones_red_calidad_aire.geo`.

`Meteorological-Data`: consists of `mmm_meteo20.csv`, `mmm_meteo19.csv` and `Estaciones_control_datos_meteorologicos.geo`. The `mmm` of the names of `mmm_meteo20.csv` and `mmm_meteo19.csv` refers to the name of the corresponding month[4]. Each record of these $.csv$ files is structured as follows (Table C.2; in the brackets English version of the columns):

Table C.2: Meteorological Data.

PROVINCIA (PROVINCE)	MUNICIPIO (MUNICIPALITY)	ESTACION (STATION)	MAGNITUD (MAGNITUDE)	PUNTO_MUESTREO POINT_SAMPLE	ANO YEAR	MES MONTH	DIA DAY	H01 H01	V01 V01	H02 H02	V02 V02
28	79	104	82	28079004_82_98	2019	1	1	23	V	17	V

The **POINT_SAMPLE** field includes the complete station code (province, municipality, station, magnitude and technique of sampling); **H01** corresponds to the data of 1 a.m. of that day; **V01** is the validation code; **H02** at 2 a.m.; **V02** and so on. **Magnitude** refers to the codes of the meteorological features (features with corresponding codes: UV (mW/m^2)-80, wind speed (m/s)-81, wind direction-82, temperature (ºC)-83, relative humidity (%)-86, barometric pressure (mb)-87, solar irradiance (W/m^2)-88, precipitation (l/m^2)-89)[5].

The location of the meteorological monitoring stations is available in $.csv$, $.xlsx$, and $.geo$ format[6]. This work used the $.geo$ format: `Estaciones_control_datos_meteorologicos.geo`.

`Traffic-Data`: consists of `mm-yyyy.zip` and `pmed_ubicacion_mm-yyyy`

[2]Interpreter of air quality data files: `https://bit.ly/3Utz9g5`
[3]Air quality. Control stations: `https://bit.ly/2Kp8TlV`
[4]Meteorological data. Hourly data from 2019: `https://bit.ly/3DIkLLk`
[5]Interpreter of meteorological data files: `https://bit.ly/3LzX8qb`
[6]Meteorological data. Control stations: `https://bit.ly/3S3ZP5x`

`.zip`. `mm-yyyy.zip` is available for each month, which contains *.csv* file[7]. The name of each *.csv* file contains the name of the corresponding month with the corresponding year. Each record is structured as follows (Table C.3; in the brackets English version of the columns):

Table C.3: Traffic Data.

id (id)	fecha (date)	tipo_elem (element_type)	intensidad (intensity)	ocupacion (occupancy_time)	carga (load)	vmed average traffic speed	error error	periodo_integracion (integration_period)
1001	01/01/2019 00:00	M30	2340	11	0	63	N	5

The SICTRAM database records and integrates all the vehicle detectors' data of the control measurement points over periods of 15 minutes. This current work used the following data: date (it was used to create hourly *.csv* files), intensity, occupancy time, load and average traffic data[8].

- **Intensity** - Intensity of the measurement point in a period of 15 minutes (vehicles/hour). A negative value implies the absence of data.
- **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. A negative value implies the absence of data.
- **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). A negative value implies the absence of data.
- **Average traffic speed** - Average speed of the vehicles in a period of 15 minutes (km/h). Only for M30 intercity measuring points. A negative value implies the absence of data.

The location of the traffic measurement points is available for every month in *.csv*, *.xlsx*, and *.zip* format[9]. This work used *.zip* file: `pmed_ubicacion_mm-yyyy.zip`, each of them contains *.dbf*, *.prj*, *.shp*, and *.shx* files.

[7]Traffic. Historical traffic data since 2013: `https://bit.ly/3BBUxHs`

[8]Description of traffic dataset: `https://bit.ly/3qTUJwZ`

[9]Traffic. Location of traffic measurement points: `https://bit.ly/2rOkHCX`

AirMetTraffic_2019_2020_firstSixMonths.zip: contains *.csv* files generated for each hour from January to June 2019 and from January to June 2020. Each *.csv* file name has the following structure:
fishnetAirMetyyyy_m_dd_h.csv. There are 4344 and 4368 *.csv* files corresponding to every hour during January-June 2019 and January-June 2020, respectively. Each *.csv* file consists of 340 rows and 14 columns (*#FID, NO2, UV, windSpeed, windDir, Temp, Humidity, Pressure, SolarRad, Prec, intensidad, ocupacion, carga, vmed*).

Madrid_wind_yyyy.csv: is the modified data of the content of AirMetTraffic_2019_2020_firstSixMonths.zip. The modification was applied to the wind direction. It was transformed in two ways: 1) converting wind direction into categorical data (north, east, south, west, southwest, northeast, southeast, and northwest), and passing through One Hot Encoder; 2) converting wind direction into u and v components. The Madrid_wind_yyyy.csv contains records for every hour during January-June 2019 and January-June 2020, and for every cell of the defined area of the city of Madrid. The columns are *NO2, windSpeed, Temp, Humidity, Pressure, SolarRad, intensidad, ocupacion, carga, vmed, v_comp, u_comp, windDir_Categ_east, windDir_Categ_north, windDir_Categ_northeast, windDir_Categ_northwest, windDir_Categ_south, windDir_Categ_southeast, windDir_Categ_southwest, windDir_Categ_west*.

Madrid_Stations_yyyy.csv: is part of Madrid_wind_yyyy.csv, which includes only data from cells containing air quality monitoring stations. The data are appended in one column with the following order: *NO2, intensidad, ocupacion, windSpeed, Pressure, SolarRad, Temp, Humidity, carga, vmed, v_comp, u_comp, windDir_Categ_east, windDir_Categ_north, windDir_Categ_northeast, windDir_Categ_northwest, windDir_Categ_south, windDir_Categ_southeast, windDir_Categ_southwest, windDir_Categ_west*.

Madrid_Exploration.zip: contains the result of an exploratory analysis that identifies the relationship between nitrogen dioxide and additional features (meteorological and traffic data).

distanceNodes.txt: includes the distance between the air quality monitoring stations placed in the city of Madrid (24 stations, 276 edges each edge is placed 2 times depending on the node order: origin, destination).

Code sub-directory consists of Process_Raw_Data,

Chapter4-ConvLSTM, Chapter5-BiConvLSTM, and Chapter6-A3T_GCN.

Process_Raw_Data: is composed of Process_Air_Quality_Data.ipynb, Process_Meteorological_Data.ipynb, Process_Traffic_Data.ipynb, Air_Met_Data_Generation.ipynb, Traffic_Data_Generation.ipynb, Combine_Generated_AirQuality_Met_Traf.ipynb, and Extract_Stations_Data.ipynb. The first three files are dedicated to processing the raw data for each dataset, respectively. Air_Met_Data_Generation.ipynb combines processed air quality and meteorological data in a spatiotemporal dimension. Traffic_Data_Generation.ipynb combines processed traffic data in a spatiotemporal dimension. Combine_Generated_AirQuality_Met_Traf.ipynb combines generated air quality, meteorological and traffic data for each hour in a separate $.csv$ file. Extract_Stations_Data.ipynb contains the procedure to extract cells or rows including NO_2, Meteorological and Traffic data, where air quality monitoring stations exist.

Chapter4-ConvLSTM: includes ConvLSTM.ipynb, which develops and tests the ConvLSTM method by implementing it in two different periods: pandemic and non-pandemic, described in the Chapter 4.

Chapter5-BiConvLSTM: includes BiConvLSTM.ipynb, Data_Preprocessing.ipynb, GridSearchCV.ipynb, mRMR.ipynb, and Mutual_Information.ipynb, executing the procedure described in Chapter 5. BiConvLSTM.ipynb develops and tests the BiConvLSTM method. Data_Preprocessing.ipynb refers to the data pre-processing step, including implementation of NN, outlier detection based on the statistical summary of the dataset, and the conversion of the wind direction (converting it to categorical data (north, east, south, west, southwest, northeast, southeast, northwest) and passing through One Hot Encoder). GridSearchCV.ipynb refers to parameter optimisation of the proposed model performed by applying GridSearchCV with Blocking Time Series Split. mRMR.ipynb and Mutual_Information.ipynb execute two feature selection techniques: MI and mRMR, respectively.

Chapter6-A3T_GCN: includes Madrid_Graph_Network.ipynb, distanceNodes.ipynb, A3T_GCN.ipynb, IsolationForest.ipynb, and LOF.ipynb, executing the procedure described in Chapter 6. Madrid_Graph_Network.ipynb contains the procedure for constructing a graph network of the air quality stations placed in the city of Madrid. distanceNodes

`.ipynb` includes the procedure for calculating the distance between the air quality stations placed in the city of Madrid (24 stations).
`A3T_GCN.ipynb` develops and tests the A3T-GCN method.
`IsolationForest.ipynb`, and `LOF.ipynb` execute two outlier detection techniques: iForest and LOF, respectively.

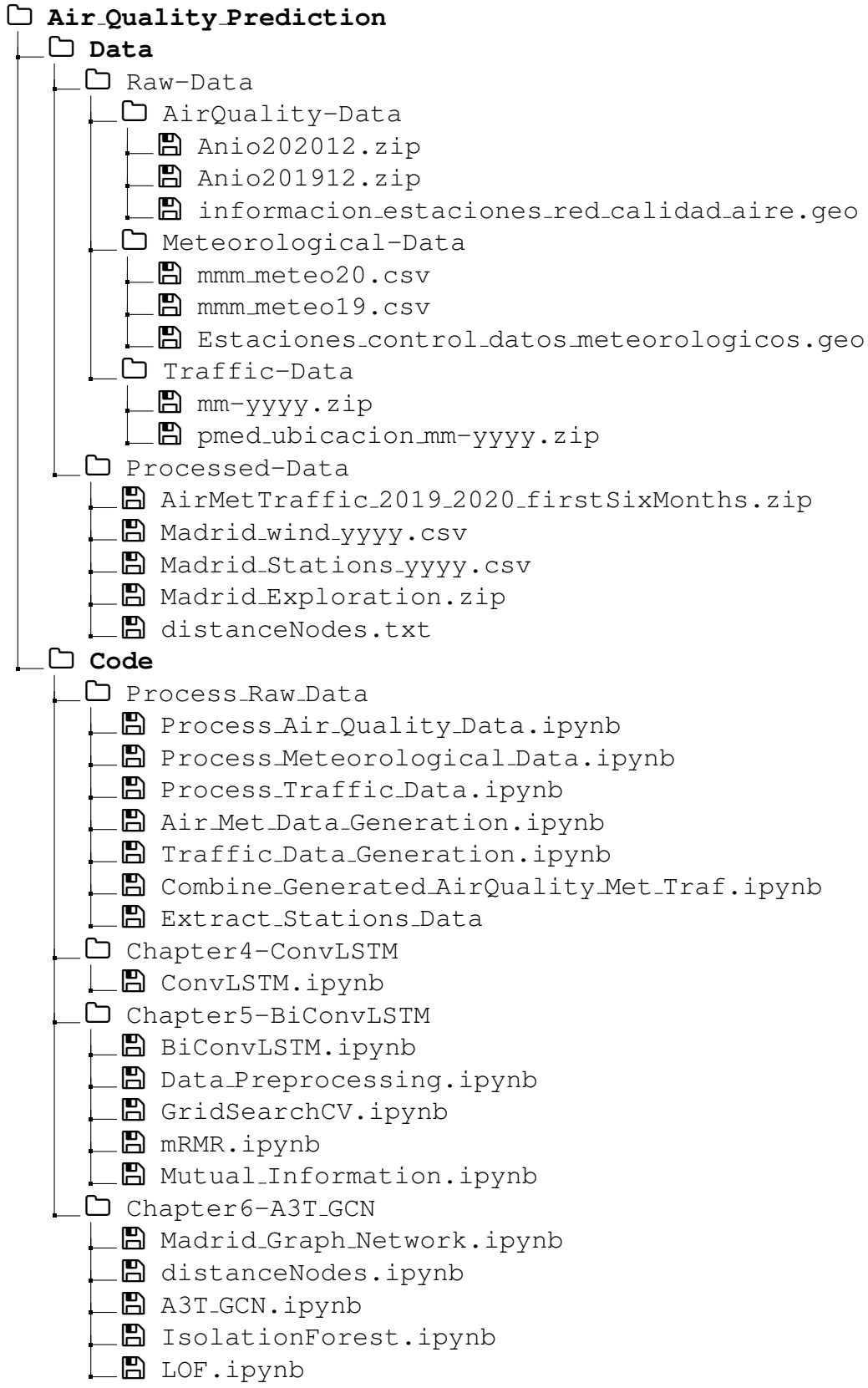


Figure C.1: Directory tree illustrating the data and implemented code.

Appendix D

The Tools Used

This section introduced the tools, software, and libraries with their versions and description used in this work.

- Google Colab Pro: cloud service to execute python code[1].
- Python[2]: is a high-level and object-oriented programming language.
- Pandas 1.3.5[3]: open source Python package to analyse and manipulate data.
- NumPy 1.21.6[4] (Numerical Python): to perform numerical computations. It works with multi-dimensional array objects.
- Scikit-learn 1.0.2 (sklearn)[5]: to build ML models.
- Statsmodels 0.12.2 [6]: to estimate different statistical models and explore statistical data.
- TensorFlow 2.8.2[7]: an end-to-end open source platform for ML, developed by the Google Brain Team. It supports GPUs and CPUs.

[1]Google Colab: `https://bit.ly/3e4YsVE`
[2]Python: `https://www.python.org/`
[3]Pandas: `https://pandas.pydata.org/`
[4]NumPy: `https://numpy.org/`
[5]Scikit-learn: `https://scikit-learn.org/stable/`
[6]Statsmodels: `https://www.statsmodels.org/stable/index.html`
[7]Tensorflow: `https://www.tensorflow.org/`

- Keras 2.8.0[8]: runs on top of TensorFlow to develop DL models.
- PyTorch 1.12.1[9]: created by Meta AI. It provides tensor computation and NN based on a tape-based autograd system. We used the PyTorch Geometric Temporal library to process spatiotemporal signals [126].
- Plotly 4.13.0 (Plotly Python Graphing Library)[10]: to generate graphs.
- Matplotlib 3.2.2[11]: multi-platform data visualization library.
- ArcGIS Pro[12] with ArcPy package[13]: to explore, visualise, and analyse data. The process of combining air quality data, meteorological data and traffic data in both spatial and temporal terms, as well as the creation of maps, was realised in this environment.
- WRPLOT VIEW 8.0.2 (Wind Rose plot for Meteorological Data)[14]: provides wind rose plots which illustrate the distribution of wind speed and wind direction at a particular location.
- Openair R package[15]: contains a set of tools to analyse and understand air pollution data. We used it to generate polar plots[16].

[8]Keras: `https://keras.io/`
[9]PyTorch: `https://pytorch.org/`
[10]Plotly Python Graphing Library: `https://plotly.com/python/`
[11]Matplotlib: `https://matplotlib.org/`
[12]ArcGIS Pro: `https://bit.ly/3SBHfSy`
[13]ArcPy package: `https://bit.ly/3USwDAi`
[14]WRPLOT VIEW: `https://bit.ly/3SPucxf`
[15]Openair R package: `https://bit.ly/3ROdjlf`
[16]Polar plots: `https://bit.ly/3Syh13j`

Bibliography

[1] Peter Rafaj, Gregor Kiesewetter, Timur Gül, Wolfgang Schöpp, Janusz Cofala, Zbigniew Klimont, Pallav Purohit, Chris Heyes, Markus Amann, Jens Borken-Kleefeld, et al. Outlook for clean air in the context of sustainable development goals. *Global Environmental Change*, 53:1–11, 2018.

[2] Philip J Landrigan, Richard Fuller, Nereus JR Acosta, Olusoji Adeyi, Robert Arnold, Abdoulaye Bibi Baldé, Roberto Bertollini, Stephan Bose-O'Reilly, Jo Ivey Boufford, Patrick N Breysse, et al. The lancet commission on pollution and health. *The lancet*, 391(10119):462–512, 2018.

[3] Robert A Rohde and Richard A Muller. Air pollution in china: mapping of concentrations and sources. *PloS one*, 10(8):e0135749, 2015.

[4] Health Effects Institute. State of global air 2019, 2019.

[5] Ioannis Manisalidis, Elisavet Stavropoulou, Agathangelos Stavropoulos, and Eugenia Bezirtzoglou. Environmental and health impacts of air pollution: a review. *Frontiers in public health*, page 14, 2020.

[6] James Longhurst, Jo Barnes, Tim Chatterton, Laura De Vito, Mark Everard, ENDA Hayes, Emily Prestwood, and Ben Williams. Analysing air pollution and its management through the lens of the un sustainable development goals: A review and assessment. *WIT Trans. Ecol. Environ*, 230:3–14, 2018.

[7] Yuan Zhao, Ya Tan, and Shilan Feng. Does reducing air pollution improve the progress of sustainable development in china? *Journal of Cleaner Production*, 272:122759, 2020.

[8] Qiaoqiao Wang, Jianwei Gu, and Xurong Wang. The impact of sahara

dust on air quality and public health in european countries. *Atmospheric Environment*, 241:117771, 2020.

[9] Daniel A Vallero. *Fundamentals of air pollution*. Academic press, 2014.

[10] Carlos Granell, Andreas Kamilaris, Alexander Kotsev, Frank O Ostermann, and Sergio Trilles. Internet of things. In *Manual of digital earth*, pages 387–423. Springer, Singapore, 2020.

[11] Sergio Trilles, Andrea Calia, Óscar Belmonte, Joaquín Torres-Sospedra, Raúl Montoliu, and Joaquín Huerta. Deployment of an open sensorized platform in a smart city context. *Future Generation Computer Systems*, 76: 221–233, 2017.

[12] Antonella Plaia and Mariantonietta Ruggieri. Air quality indices: a review. *Reviews in Environmental Science and Bio/Technology*, 10(2):165–179, 2011.

[13] Huiping Peng, Aranildo R Lima, Andrew Teakles, Jian Jin, Alex J Cannon, and William W Hsieh. Evaluating hourly air quality forecasting in canada with nonlinear updatable machine learning methods. *Air Quality, Atmosphere & Health*, 10(2):195–211, 2017.

[14] Paulo SG De Mattos Neto, Paulo Renato A Firmino, Hugo Siqueira, Yara De Souza Tadano, Thiago Antonini Alves, Joao Fausto L De Oliveira, Manoel Henrique Da Nóbrega Marinho, and Francisco Madeiro. Neural-based ensembles for particulate matter forecasting. *IEEE Access*, 9:14470–14490, 2021.

[15] Aurélien Géron. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. " O'Reilly Media, Inc.", 2019.

[16] Yue-Hua Dai and Wei-Xing Zhou. Temporal and spatial correlation patterns of air pollutants in chinese cities. *PloS one*, 12(8):e0182724, 2017.

[17] Chunsheng Fang, Xiaodong Tan, Yue Zhong, and Ju Wang. Research on the temporal and spatial characteristics of air pollutants in sichuan basin. *Atmosphere*, 12(11):1504, 2021.

[18] Alessandro Liberati, Douglas G Altman, Jennifer Tetzlaff, Cynthia Mulrow, Peter C Gøtzsche, John PA Ioannidis, Mike Clarke, Philip J Devereaux, Jos Kleijnen, and David Moher. The prisma statement for reporting systematic reviews and meta-analyses of studies that evaluate health care interventions: explanation and elaboration. *Journal of clinical epidemiology*, 62(10):e1–e34, 2009.

[19] Auriol Degbelo, Carlos Granell, Sergio Trilles, Devanjan Bhattacharya, Sven Casteleyn, and Christian Kray. Opening up smart cities: citizen-centric challenges and opportunities from giscience. *ISPRS International Journal of Geo-Information*, 5(2):16, 2016.

[20] Gaurav Pandey, Bin Zhang, and Le Jian. Predicting submicron air pollution indicators: a machine learning approach. *Environmental Science: Processes & Impacts*, 15(5):996–1005, 2013.

[21] Barouch Giechaskiel, Tero Lähde, Sawan Gandi, Stefan Keller, Philipp Kreutziger, and Athanasios Mamakos. Assessment of 10-nm particle number (pn) portable emissions measurement systems (pems) for future regulations. *International Journal of Environmental Research and Public Health*, 17(11): 3878, 2020.

[22] Benito M Zaragozí, Sergio Trilles, and José T Navarro-Carrión. Leveraging container technologies in a giscience project: A perspective from open reproducible research. *ISPRS International Journal of Geo-Information*, 9 (3):138, 2020.

[23] Judie Attard, Fabrizio Orlandi, Simon Scerri, and Sören Auer. A systematic review of open government data initiatives. *Government Information Quarterly*, 32(4):399–418, 2015.

[24] Renata Máchová and Martin Lnenicka. Evaluating the quality of open data portals on the national level. *Journal of theoretical and applied electronic commerce research*, 12(1):21–41, 2017.

[25] Vito Albino, Umberto Berardi, and Rosa Maria Dangelico. Smart cities: Definitions, dimensions, performance, and initiatives. *Journal of urban technology*, 22(1):3–21, 2015.

[26] Jun Ma, Yuexiong Ding, Jack CP Cheng, Feifeng Jiang, Vincent JL Gan, and Zherui Xu. A lag-flstm deep learning network based on bayesian optimization for multi-sequential-variant pm2. 5 prediction. *Sustainable Cities and Society*, 60:102237, 2020.

[27] Alexei Botchkarev. Performance metrics (error measures) in machine learning regression, forecasting and prognostics: Properties and typology. *arXiv preprint arXiv:1809.03006*, 2018.

[28] Mohammad Hossin and MN Sulaiman. A review on evaluation metrics for data classification evaluations. *International Journal of Data Mining & Knowledge Management Process*, 5(2):1, 2015.

[29] Diane Ivy, James A Mulholland, and Armistead G Russell. Development of ambient air quality population-weighted metrics for use in time-series health studies. *Journal of the Air & Waste Management Association*, 58(5): 711–720, 2008.

[30] Yudong Tian, Grey S Nearing, Christa D Peters-Lidard, Kenneth W Harrison, and Ling Tang. Performance metrics, error modeling, and uncertainty quantification. *Monthly Weather Review*, 144(2):607–613, 2016.

[31] Sungil Kim and Heeyoung Kim. A new metric of absolute percentage error for intermittent demand forecasts. *International Journal of Forecasting*, 32 (3):669–679, 2016.

[32] Shaocai Yu, Brian Eder, Robin Dennis, Shao-Hang Chu, and Stephen E Schwartz. New unbiased symmetric metrics for evaluation of air quality models. *Atmospheric Science Letters*, 7(1):26–34, 2006.

[33] Cort J Willmott and Donald E Wicks. An empirical method for the spatial interpolation of monthly precipitation within california. *Physical Geography*, 1(1):59–73, 1980.

[34] Nico JD Nagelkerke et al. A note on a general definition of the coefficient of determination. *Biometrika*, 78(3):691–692, 1991.

[35] Allan C Just, Kodi B Arfer, Johnathan Rush, Michael Dorman, Alexandra Shtein, Alexei Lyapustin, and Itai Kloog. Advancing methodologies for

applying machine learning and evaluating spatiotemporal models of fine particulate matter (pm2. 5) using satellite data over large regions. *Atmospheric Environment*, 239:117649, 2020.

[36] Sahar Masmoudi, Haytham Elghazel, Dalila Taieb, Orhan Yazar, and Amjad Kallel. A machine-learning framework for predicting multiple air pollutants' concentrations via multi-target regression and feature selection. *Science of The Total Environment*, 715:136991, 2020.

[37] Ying Zhang, Rongrong Zhang, Qunfei Ma, Yanhao Wang, Qingqing Wang, Zihao Huang, and Linyan Huang. A feature selection and multi-model fusion-based approach of predicting air quality. *ISA transactions*, 100:210–220, 2020.

[38] Zhiyuan Li, Steve Hung Lam Yim, and Kin-Fai Ho. High temporal resolution prediction of street-level pm2. 5 and nox concentrations using machine learning approach. *Journal of Cleaner Production*, page 121975, 2020.

[39] Iat Hang Fong, Tengyue Li, Simon Fong, Raymond K Wong, and Antonio J Tallón-Ballesteros. Predicting concentration levels of air pollutants by transfer learning and recurrent neural network. *Knowledge-Based Systems*, 192: 105622, 2020.

[40] Bo Zhang, Hanwen Zhang, Gengming Zhao, and Jie Lian. Constructing a pm2. 5 concentration prediction model by combining auto-encoder with bi-lstm neural networks. *Environmental Modelling & Software*, 124:104600, 2020.

[41] Jalpa Shah and Biswajit Mishra. Analytical equations based prediction approach for pm2. 5 using artificial neural network. *arXiv preprint arXiv:2002.11416*, 2020.

[42] Temuulen Enebish, Khang Chau, Batbayar Jadamba, and Meredith Franklin. Predicting ambient pm 2.5 concentrations in ulaanbaatar, mongolia with machine learning approaches. *Journal of Exposure Science & Environmental Epidemiology*, pages 1–10, 2020.

[43] Yue-Shan Chang, Hsin-Ta Chiao, Satheesh Abimannan, Yo-Ping Huang, Yi-Ting Tsai, and Kuan-Ming Lin. An lstm-based aggregated model for air pollution forecasting. *Atmospheric Pollution Research*, 2020.

[44] Congcong Wen, Shufu Liu, Xiaojing Yao, Ling Peng, Xiang Li, Yuan Hu, and Tianhe Chi. A novel spatiotemporal convolutional long short-term neural network for air pollution prediction. *Science of the total environment*, 654: 1091–1099, 2019.

[45] Yuqin Mo, Qi Li, Hamed Karimian, Shuwei Fang, Boyuan Tang, Gong Chen, and Sonali Sachdeva. A novel framework for daily forecasting of ozone mass concentrations based on cycle reservoir with regular jumps neural networks. *Atmospheric Environment*, 220:117072, 2020.

[46] Xinghan Xu and Weijie Ren. Prediction of air pollution concentration based on mrmr and echo state network. *Applied Sciences*, 9(9):1811, 2019.

[47] Jindong Han, Hao Liu, Haoyi Xiong, and Jing Yang. Semi-supervised air quality forecasting via self-supervised hierarchical graph neural network. *IEEE Transactions on Knowledge and Data Engineering*, 2022.

[48] R Saravana Ram, K Venkatachalam, Mehedi Masud, and Mohamed Abouhawwash. Air pollution prediction using dual graph convolution lstm technique. *INTELLIGENT AUTOMATION AND SOFT COMPUTING*, 33(3): 1639–1652, 2022.

[49] Xiao Xiao, Zhiling Jin, Shuo Wang, Jing Xu, Ziyan Peng, Rui Wang, Wei Shao, and Yilong Hui. A dual-path dynamic directed graph convolutional network for air quality prediction. *Science of The Total Environment*, 827: 154298, 2022.

[50] Xiaocao Ouyang, Yan Yang, Yiling Zhang, and Wei Zhou. Spatial-temporal dynamic graph convolution neural network for air quality prediction. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2021.

[51] Liang Ge, Kunyan Wu, Yi Zeng, Feng Chang, Yaqian Wang, and Siyu Li. Multi-scale spatiotemporal graph convolution network for air quality prediction. *Applied Intelligence*, 51(6):3491–3505, 2021.

[52] Chunyang Wang, Yanmin Zhu, Tianzi Zang, Haobing Liu, and Jiadi Yu. Modeling inter-station relationships with attentive temporal graph convolutional network for air quality prediction. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, pages 616–634, 2021.

[53] Ling Chen, Jiahui Xu, Binqing Wu, Yuntao Qian, Zhenhong Du, Yansheng Li, and Yongjun Zhang. Group-aware graph neural network for nationwide city air quality forecasting. *arXiv preprint arXiv:2108.12238*, 2021.

[54] Jiahui Xu, Ling Chen, Mingqi Lv, Chaoqun Zhan, Sanjian Chen, and Jian Chang. Highair: A hierarchical graph neural network-based air quality forecasting method. *arXiv preprint arXiv:2101.04264*, 2021.

[55] Yachuan Liu, Jiaqi Ma, Paramveer Dhillon, and Qiaozhu Mei. A new benchmark of graph learning for pm2. 5 forecasting under distribution shift. 2021.

[56] Van-Duc Le. Spatiotemporal graph convolutional recurrent neural network model for citywide air pollution forecasting. 2021.

[57] Guyu Zhao, Hongdou He, Yifang Huang, and Jiadong Ren. Near-surface pm2. 5 prediction combining the complex network characterization and graph convolution neural network. *Neural Computing and Applications*, 33 (24):17081–17101, 2021.

[58] Xi Gao and Weide Li. A graph-based lstm model for pm2. 5 forecasting. *Atmospheric Pollution Research*, 12(9):101150, 2021.

[59] Jiafeng Zhang, Zumin Wang, Yan Liu, Rongli Gai, Lingyan Hu, and Qijie Zou. Temporal attention with domain-specific graph regularization for pm 2.5 forecasting. In *2020 International Conference on Artificial Intelligence and Computer Engineering (ICAICE)*, pages 510–517. IEEE, 2020.

[60] Shuo Wang, Yanran Li, Jiang Zhang, Qingye Meng, Lingwei Meng, and Fei Gao. Pm2. 5-gnn: A domain knowledge enhanced graph neural network for pm2. 5 forecasting. In *Proceedings of the 28th International Conference on Advances in Geographic Information Systems*, pages 163–166, 2020.

[61] Peijiang Zhao and Koji Zettsu. Mastgn: Multi-attention spatio-temporal graph networks for air pollution prediction. In *2020 IEEE International Conference on Big Data (Big Data)*, pages 1442–1448. IEEE, 2020.

[62] Yanlin Qi, Qi Li, Hamed Karimian, and Di Liu. A hybrid model for spatiotemporal forecasting of pm2. 5 based on graph convolutional neural network and long short-term memory. *Science of the Total Environment*, 664:1–10, 2019.

[63] Yu Huang, Josh Jia-Ching Ying, and Vincent S Tseng. Spatio-attention embedded recurrent neural network for air quality prediction. *Knowledge-Based Systems*, 233:107416, 2021.

[64] Yijun Lin, Nikhit Mago, Yu Gao, Yaguang Li, Yao-Yi Chiang, Cyrus Shahabi, and José Luis Ambite. Exploiting spatiotemporal patterns for accurate air quality forecasting using deep learning. In *Proceedings of the 26th ACM SIGSPATIAL international conference on advances in geographic information systems*, pages 359–368, 2018.

[65] Fernando Benitez-Paez, Auriol Degbelo, Sergio Trilles, and Joaquin Huerta. Roadblocks hindering the reuse of open geodata in colombia and spain: A data user's perspective. *ISPRS International Journal of Geo-Information*, 7 (1):6, 2018.

[66] Auriol Degbelo, Carlos Granell, Sergio Trilles, Devanjan Bhattacharya, and Jonas Wissing. Tell me how my open data is re-used: increasing transparency through the open city toolkit. In *Open Cities— Open Data*, pages 311–330. Springer, 2020.

[67] Fernando Benitez-Paez, Alexis Comber, Sergio Trilles, and Joaquin Huerta. Creating a conceptual framework to improve the re-usability of open geographic data in cities. *Transactions in GIS*, 22(3):806–822, 2018.

[68] Sasha Khomenko, Marta Cirach, Evelise Pereira-Barboza, Natalie Mueller, Jose Barrera-Gómez, David Rojas-Rueda, Kees de Hoogh, Gerard Hoek, and Mark Nieuwenhuijsen. Premature mortality due to air pollution in european cities: A health impact assessment. *The Lancet Planetary Health*, 2021.

[69] John Seinfeld and Spyros Pandis. Atmospheric chemistry and physics. 1997. *New York*, 2008.

[70] Annunziata Faustini, Regula Rapp, and Francesco Forastiere. Nitrogen dioxide and mortality: review and meta-analysis of long-term studies. *European Respiratory Journal*, 44(3):744–753, 2014.

[71] Gerard Hoek, Ranjini M Krishnan, Rob Beelen, Annette Peters, Bart Ostro, Bert Brunekreef, and Joel D Kaufman. Long-term air pollution exposure and cardio-respiratory mortality: a review. *Environmental health*, 12(1):1–16, 2013.

[72] Ghassan B Hamra, Francine Laden, Aaron J Cohen, Ole Raaschou-Nielsen, Michael Brauer, and Dana Loomis. Lung cancer and exposure to nitrogen dioxide and traffic: a systematic review and meta-analysis. *Environmental health perspectives*, 123(11):1107–1112, 2015.

[73] Zili Zhang, Jian Wang, and Wenju Lu. Exposure to nitrogen dioxide and chronic obstructive pulmonary disease (copd) in adults: a systematic review and meta-analysis. *Environmental Science and Pollution Research*, 25(15): 15133–15145, 2018.

[74] Henrik Brønnum-Hansen, Anne Mette Bender, Zorana Jovanovic Andersen, Jan Sørensen, Jakob Hjort Bønløkke, Hendriek Boshuizen, Thomas Becker, Finn Diderichsen, and Steffen Loft. Assessment of impact of traffic-related air pollution on morbidity and mortality in copenhagen municipality and the health gain of reduced exposure. *Environment International*, 121:973–980, 2018.

[75] Carlos A Cuevas, Alberto Notario, José Antonio Adame, Andreas Hilboll, Andreas Richter, John P Burrows, and Alfonso Saiz-Lopez. Evolution of no 2 levels in spain from 1996 to 2012. *Scientific Reports*, 4(1):1–8, 2014.

[76] EL Delmar-Morgan. The beaufort scale. *The Journal of Navigation*, 12(1): 100–102, 1959.

[77] Scott Huler. *Defining the wind: the Beaufort scale and how a 19th-century admiral turned science into poetry*. Crown, 2007.

[78] William S Cleveland. Robust locally weighted regression and smoothing scatterplots. *Journal of the American statistical association*, 74(368):829–836, 1979.

[79] Roland B Stull. Practical meteorology: an algebra-based survey of atmospheric science. 2015.

[80] Kwei-Herng Lai, Daochen Zha, Junjie Xu, Yue Zhao, Guanchu Wang, and Xia Hu. Revisiting time series outlier detection: Definitions and benchmarks. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*, 2021.

[81] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. In *2008 eighth ieee international conference on data mining*, pages 413–422. IEEE, 2008.

[82] Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. Lof: identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pages 93–104, 2000.

[83] EG Beek. Spatial interpolation of daily meteorological data. *Theoretical evaluation of available techniques. Report*, 53:43, 1991.

[84] Naomi Altman and Martin Krzywinski. The curse (s) of dimensionality. *Nat Methods*, 15(6):399–400, 2018.

[85] Michel Verleysen and Damien François. The curse of dimensionality in data mining and time series prediction. In *International work-conference on artificial neural networks*, pages 758–770. Springer, 2005.

[86] Hong Zheng, Yunhui Cheng, and Haibin Li. Investigation of model ensemble for fine-grained air quality prediction. *China Communications*, 17(7):207–223, 2020.

[87] Hui Liu and Chao Chen. Spatial air quality index prediction model based on decomposition, adaptive boosting, and three-stage feature selection: A case study in china. *Journal of Cleaner Production*, page 121777, 2020.

[88] Hanchuan Peng, Fuhui Long, and Chris Ding. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on pattern analysis and machine intelligence*, 27(8):1226–1238, 2005.

[89] Zhenyu Zhao, Radhika Anand, and Mallory Wang. Maximum relevance and minimum redundancy feature selection methods for a marketing machine learning platform. In *2019 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pages 442–452. IEEE, 2019.

[90] Arthur L Samuel. Some studies in machine learning using the game of checkers. ii—recent progress. *Computer Games I*, pages 366–400, 1988.

[91] Arthur L Samuel. Some studies in machine learning using the game of checkers. ii—recent progress. *IBM Journal of research and development*, 11(6):601–617, 1967.

[92] Warren S McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4): 115–133, 1943.

[93] Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.

[94] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

[95] Paul John Werbos. *The roots of backpropagation: from ordered derivatives to neural networks and political forecasting*, volume 1. John Wiley & Sons, 1994.

[96] Saad Albawi, Tareq Abed Mohammed, and Saad Al-Zawi. Understanding of a convolutional neural network. In *2017 international conference on engineering and technology (ICET)*, pages 1–6. Ieee, 2017.

[97] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166, 1994.

[98] Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*, 2014.

[99] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[100] Jonathan L Gross and Jay Yellen. *Handbook of graph theory*. CRC press, 2003.

[101] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. *arXiv preprint arXiv:1506.04214*, 2015.

[102] Hongmei Song, Wenguan Wang, Sanyuan Zhao, Jianbing Shen, and Kin-Man Lam. Pyramid dilated deeper convlstm for video salient object detection. In *Proceedings of the European conference on computer vision (ECCV)*, pages 715–731, 2018.

[103] Jiandong Bai, Jiawei Zhu, Yujiao Song, Ling Zhao, Zhixiang Hou, Ronghua Du, and Haifeng Li. A3t-gcn: Attention temporal graph convolutional network for traffic forecasting. *ISPRS International Journal of Geo-Information*, 10 (7):485, 2021.

[104] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. Spectral networks and locally connected networks on graphs. *arXiv preprint arXiv:1312.6203*, 2013.

[105] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[106] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[107] Wei Liu, Xiao-Guang Yue, and Paul B Tchounwou. Response to the covid-19 epidemic: the chinese experience and implications for other countries, 2020.

[108] Zander S Venter, Kristin Aunan, Sourangsu Chowdhury, and Jos Lelieveld. Covid-19 lockdowns cause global air pollution declines. *Proceedings of the National Academy of Sciences*, 117(32):18984–18990, 2020.

[109] José M Baldasano. Covid-19 lockdown effects on air quality by no2 in the cities of barcelona and madrid (spain). *Science of the Total Environment*, 741:140353, 2020.

[110] S Rao Jammalamadaka and Ambar Sengupta. *Topics in circular statistics*, volume 5. world scientific, 2001.

[111] Ignacio Leguey, Pedro Larrañaga, Concha Bielza, and Shogo Kato. A circular-linear dependence measure under johnson–wehrly distributions and its application in bayesian networks. *Information Sciences*, 486:240–253, 2019.

[112] Nicholas I Fisher. *Statistical analysis of circular data*. cambridge university press, 1995.

[113] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

[114] Pierre Baldi and Peter J Sadowski. Understanding dropout. *Advances in neural information processing systems*, 26, 2013.

[115] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015.

[116] Shibani Santurkar, Dimitris Tsipras, Andrew Ilyas, and Aleksander Madry. How does batch normalization help optimization? *Advances in neural information processing systems*, 31, 2018.

[117] Keiron O'Shea and Ryan Nash. An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*, 2015.

[118] Li Yang and Abdallah Shami. On hyperparameter optimization of machine learning algorithms: Theory and practice. *Neurocomputing*, 415:295–316, 2020.

[119] Stefan Tsokov, Milena Lazarova, and Adelina Aleksieva-Petrova. A hybrid spatiotemporal deep model based on cnn and lstm for air pollution prediction. *Sustainability*, 14(9):5104, 2022.

[120] Ghufran Isam Drewil and Riyadh Jabbar Al-Bahadili. Air pollution prediction using lstm deep learning and metaheuristics algorithms. *Measurement: Sensors*, 24:100546, 2022.

[121] Aysenur Gilik, Arif Selcuk Ogrenci, and Atilla Ozmen. Air quality prediction using cnn+ lstm-based hybrid deep learning architecture. *Environmental science and pollution research*, pages 1–19, 2022.

[122] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[123] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256. JMLR Workshop and Conference Proceedings, 2010.

[124] Hong Zheng, Haibin Li, Xingjian Lu, and Tong Ruan. A multiple kernel learning approach for air quality prediction. *Advances in Meteorology*, 2018, 2018.

[125] Abhishek Tiwary and Jeremy Colls. *Air pollution: measurement, modelling and mitigation*. CRC Press, 2017.

[126] Benedek Rozemberczki, Paul Scherer, Yixuan He, George Panagopoulos, Alexander Riedel, Maria Astefanoaei, Oliver Kiss, Ferenc Beres, Guzman Lopez, Nicolas Collignon, and Rik Sarkar. PyTorch Geometric Temporal: Spatiotemporal Signal Processing with Neural Machine Learning Models. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management*, page 4564–4573, 2021.

[127] Ditsuhi Iskandaryan, Francisco Ramos, and Sergio Trilles. Graph neural network for air quality prediction: A case study in madrid. *IEEE Access*, 11: 2729–2742, 2023.

[128] Fernando Benitez-Paez, Auriol Degbelo, Sergio Trilles, and Joaquin Huerta. Roadblocks hindering the reuse of open geodata in colombia and spain: A data user's perspective. *ISPRS International Journal of Geo-Information*, 7 (1):6, 2017.

[129] Lianfa Li, Mariam Girguis, Frederick Lurmann, Jun Wu, Robert Urman, Edward Rappaport, Beate Ritz, Meredith Franklin, Carrie Breton, Frank Gilliland, et al. Cluster-based bagging of constrained mixed-effects models for high spatiotemporal resolution nitrogen oxides prediction over large regions. *Environment international*, 128:310–323, 2019.

[130] Jie Chen, Kees de Hoogh, John Gulliver, Barbara Hoffmann, Ole Hertel, Matthias Ketzel, Mariska Bauwelinck, Aaron van Donkelaar, Ulla A Hvidtfeldt, Klea Katsouyanni, et al. A comparison of linear regression, regularization, and machine learning algorithms to develop europe-wide spatial models of fine particles and nitrogen dioxide. *Environment international*, 130:104934, 2019.

[131] Ditsuhi Iskandaryan, Francisco Ramos, and Sergio Trilles. The features of the selected papers in the field of air quality prediction, December 2020. URL `https://doi.org/10.5281/zenodo.4302469`.

[132] Junshi Xu, An Wang, Nicole Schmidt, Matthew Adams, and Marianne Hatzopoulou. A gradient boost approach for predicting near-road ultrafine particle concentrations using detailed traffic characterization. *Environmental Pollution*, page 114777, 2020.

[133] Yue-Shan Chang, Satheesh Abimannan, Hsin-Ta Chiao, Chi-Yeh Lin, and Yo-Ping Huang. An ensemble learning based hybrid model and framework for air pollution forecasting. *Environmental Science and Pollution Research*, 27(30):38155–38168, 2020.

[134] Mauro Castelli, Fabiana Martins Clemente, Aleš Popovič, Sara Silva, and Leonardo Vanneschi. A machine learning approach to predict air quality in california. *Complexity*, 2020, 2020.

[135] Aslı Bozdağ, Yeşim Dokuz, and Öznur Begüm Gökçek. Spatial prediction of pm10 concentration using machine learning algorithms in ankara, turkey. *Environmental Pollution*, page 114635, 2020.

[136] Rui Feng, Han Gao, Kun Luo, and Jian-ren Fan. Analysis and accurate prediction of ambient pm2. 5 in china using multi-layer perceptron. *Atmospheric Environment*, page 117534, 2020.

[137] Qingchun Guo, Zhenfang He, Shanshan Li, Xinzhou Li, Jingjing Meng, Zhanfang Hou, Jiazhen Liu, Yongjin Chen, et al. Air pollution forecasting using artificial and wavelet neural networks with meteorological conditions. *Aerosol and Air Quality Research*, 20(6):1429–1439, 2020.

[138] Unjin Pak, Jun Ma, Unsok Ryu, Kwangchol Ryom, U Juhyok, Kyongsok Pak, and Chanil Pak. Deep learning-based pm2. 5 prediction considering the spatiotemporal correlations: A case study of beijing, china. *Science of The Total Environment*, 699:133561, 2020.

[139] Guang Yang, HwaMin Lee, and Giyeol Lee. A hybrid deep learning model to forecast particulate matter concentration levels in seoul, south korea. *Atmosphere*, 11(4):348, 2020.

[140] Mike Lee, Larry Lin, Chih-Yuan Chen, Yu Tsao, Ting-Hsuan Yao, Min-Han Fei, and Shih-Hau Fang. forecasting air quality in taiwan by using machine learning. *Scientific Reports*, 10(1):1–13, 2020.

[141] Mahdieh Danesh Yazdi, Zheng Kuang, Konstantina Dimakopoulou, Benjamin Barratt, Esra Suel, Heresh Amini, Alexei Lyapustin, Klea Katsouyanni, and Joel Schwartz. Predicting fine particulate matter (pm2. 5) in the greater london area: An ensemble approach using machine learning methods. *Remote Sensing*, 12(6):914, 2020.

[142] Rasa Zalakeviciute, Marco Bastidas, Adrian Buenaño, and Yves Rybarczyk. A traffic-based method to predict and map urban air quality. *Applied Sciences*, 10(6):2035, 2020.

[143] Ke Gu, Zhifang Xia, and Junfei Qiao. Stacked selective ensemble for pm 2.5 forecast. *IEEE Transactions on Instrumentation and Measurement*, 69(3): 660–671, 2019.

[144] Jinghui Ma, Zhongqi Yu, Yuanhao Qu, Jianming Xu, Yu Cao, et al. Application of the xgboost machine learning method in pm2. 5 prediction: A case study of shanghai. *Aerosol and Air Quality Research*, 20(1):128–138, 2020.

[145] Lei Zhang, Dong Li, and Quansheng Guo. Deep learning from spatio-temporal data using orthogonal regularizaion residual cnn for air prediction. *IEEE Access*, 8:66037–66047, 2020.

[146] Kefei Zhang, Jesse Thé, Guangyuan Xie, and Hesheng Yu. Multi-step ahead forecasting of regional air quality using spatial-temporal deep neural networks: A case study of huaihai economic zone. *Journal of Cleaner Production*, 277:123231, 2020.

[147] Dan Zhang and Simon S Woo. Real time localized air quality monitoring and prediction through mobile and fixed iot sensing network. *IEEE Access*, 8:89584–89594, 2020.

[148] Weixin Zhai and Chengqi Cheng. A long short-term memory approach to predicting air quality based on social media data. *Atmospheric Environment*, page 117411, 2020.

[149] Chadaphim Photphanloet and Rajalida Lipikorn. Pm10 concentration forecast using modified depth-first search and supervised learning neural network. *Science of The Total Environment*, page 138507, 2020.

[150] Laura Goulier, Bastian Paas, Laura Ehrnsperger, and Otto Klemm. Modelling of urban air pollutant concentrations with artificial neural networks using novel input variables. *International Journal of Environmental Research and Public Health*, 17(6):2025, 2020.

[151] A Altikat. Modeling air pollution levels in volcanic geological regional properties and microclimatic conditions. *International Journal of Environmental Science and Technology*, pages 1–8, 2020.

[152] Maryam Aljanabi Mohammad Shkoukani Mohammad Hijjawi. Ground-level ozone prediction using machine learning techniques: A case study in amman, jordan.

[153] Sung-Hyun Kim, Dae-Sung Son, Min-Ho Park, and Hyun-Seok Hwang. Developing a big data analytic model and a platform for particulate matter prediction: A case study. *International Journal of Fuzzy Logic and Intelligent Systems*, 19(4):242–249, 2019.

[154] Shuo-Wen Chang, Chung-Ling Chang, Long-Tin Li, and Shih-Wei Liao. Reinforcement learning for improving the accuracy of pm2. 5 pollution forecast under the neural network framework. *IEEE Access*, 8:9864–9874, 2019.

[155] Ebrahim Eslami, Ahmed Khan Salman, Yunsoo Choi, Alqamah Sayeed, and Yannic Lops. A data ensemble approach for real-time air quality forecasting using extremely randomized trees and deep neural networks. *NEURAL COMPUT APPL*, pages 1–17, 2019.

[156] Xintong Li and Xiaodong Zhang. Predicting ground-level pm2. 5 concentrations in the beijing-tianjin-hebei region: A hybrid remote sensing and machine learning approach. *Environmental Pollution*, 249:735–749, 2019.

[157] Rui Li, Lulu Cui, Ya Meng, Yilong Zhao, and Hongbo Fu. Satellite-based prediction of daily so2 exposure across china using a high-quality random forest-spatiotemporal kriging (rf-stk) model for health risk assessment. *Atmospheric Environment*, 208:10–19, 2019.

[158] Dongming Qin, Jian Yu, Guojian Zou, Ruihan Yong, Qin Zhao, and Bo Zhang. A novel combined prediction scheme based on cnn and lstm for urban pm 2.5 concentration. *IEEE Access*, 7:20050–20059, 2019.

[159] Ying Zhang, Yanhao Wang, Minghe Gao, Qunfei Ma, Jing Zhao, Rongrong Zhang, Qingqing Wang, and Linyan Huang. A predictive data feature exploration-based air quality prediction approach. *IEEE Access*, 7:30732–30743, 2019.

[160] Qing Tao, Fang Liu, Yong Li, and Denis Sidorov. Air pollution forecasting using a deep learning model based on 1d convnets and bidirectional gru. *IEEE Access*, 7:76690–76698, 2019.

[161] Saba Ameer, Munam Ali Shah, Abid Khan, Houbing Song, Carsten Maple, Saif Ul Islam, and Muhammad Nabeel Asghar. Comparative analysis of

machine learning techniques for predicting air quality in smart cities. *IEEE Access*, 7:128325–128338, 2019.

[162] Lkhagvadorj Munkhdalai, Tsendsuren Munkhdalai, Kwang Ho Park, Tsatsral Amarbayasgalan, Erdenebileg Erdenebaatar, Hyun Woo Park, and Keun Ho Ryu. An end-to-end adaptive input selection with dynamic weights for forecasting multivariate time series. *IEEE Access*, 7:99099–99114, 2019.

[163] Jun Ma, Yuexiong Ding, Vincent JL Gan, Changqing Lin, and Zhiwei Wan. Spatiotemporal prediction of pm2. 5 concentrations at different time granularities using idw-blstm. *IEEE Access*, 7:107897–107907, 2019.

[164] Ling Chen, Yifang Ding, Dandan Lyu, Xiaoze Liu, and Hanyu Long. Deep multi-task learning based urban air quality index modelling. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 3 (1):1–17, 2019.

[165] Yanlai Zhou, Fi-John Chang, Li-Chiu Chang, I-Feng Kao, Yi-Shin Wang, and Che-Chia Kang. Multi-output support vector machine for regional multi-step-ahead pm2. 5 forecasting. *Science of The Total Environment*, 651:230–240, 2019.

[166] Hamed Karimian, Qi Li, Chunlin Wu, Yanlin Qi, Yuqin Mo, Gong Chen, Xianfeng Zhang, Sonali Sachdeva, et al. Evaluation of different machine learning approaches to forecasting pm2. 5 mass concentrations. *Aerosol and Air Quality Research*, 19(6):1400–1410, 2019.

[167] Joanna A Kamińska. A random forest partition model for predicting no2 concentrations from traffic flow and meteorological conditions. *Science of The Total Environment*, 651:475–483, 2019.

[168] Mrigank Krishan, Srinidhi Jha, Jew Das, Avantika Singh, Manish Kumar Goyal, and Chandrra Sekar. Air quality modelling using long short-term memory (lstm) over nct-delhi, india. *Air Quality, Atmosphere & Health*, 12 (8):899–908, 2019.

[169] Mengwei Jia, Xinghong Cheng, Xinghong Zhao, Chongzhi Yin, Xiangzhi Zhang, Xianghua Wu, Liming Wang, Renjian Zhang, et al. Regional air quality forecast using a machine learning method and the wrf model over

the yangtze river delta, east china. *Aerosol and Air Quality Research*, 19(7): 1602–1613, 2019.

[170] Yin Xing, Jianping Yue, Chuang Chen, Yunfei Xiang, Yang Chen, and Manxing Shi. A deep belief network combined with modified grey wolf optimization algorithm for pm2. 5 concentration prediction. *Applied Sciences*, 9(18):3765, 2019.

[171] Mehdi Zamani Joharestani, Chunxiang Cao, Xiliang Ni, Barjeece Bashir, and Somayeh Talebiesfandarani. Pm2. 5 prediction based on random forest, xgboost, and deep learning using multisource remote sensing data. *Atmosphere*, 10(7):373, 2019.

[172] Sankaralingam Mohan and Packiam Saranya. A novel bagging ensemble approach for predicting summertime ground-level ozone concentration. *Journal of the Air & Waste Management Association*, 69(2):220–233, 2019.

[173] Rui Feng, Hui-jun Zheng, An-ran Zhang, Chong Huang, Han Gao, and Yu-cheng Ma. Unveiling tropospheric ozone by the traditional atmospheric model and machine learning, and their comparison: A case study in hangzhou, china. *Environmental pollution*, 252:366–378, 2019.

[174] A Masih. Application of ensemble learning techniques to model the atmospheric concentration of so2. *Global Journal of Environmental Science and Management*, 5(3):309–318, 2019.

[175] Dong-Her Shih, Ting-Wei Wu, Wen-Xuan Liu, and Po-Yuan Shih. An azure aces early warning system for air quality index deteriorating. *International journal of environmental research and public health*, 16(23):4679, 2019.

[176] Mahmoud Reza Delavar, Amin Gholami, Gholam Reza Shiran, Yousef Rashidi, Gholam Reza Nakhaeizadeh, Kurt Fedra, and Smaeil Hatefi Afshar. A novel method for improving air pollution prediction based on machine learning approaches: a case study applied to the capital city of tehran. *ISPRS International Journal of Geo-Information*, 8(2):99, 2019.

[177] Yegang Chen. Prediction algorithm of pm2. 5 mass concentration based on adaptive bp neural network. *Computing*, 100(8):825–838, 2018.

[178] Jana Faganeli Pucer, Gregor Pirš, and Erik Štrumbelj. A bayesian approach to forecasting daily air-pollutant levels. *Knowledge and Information Systems*, 57(3):635–654, 2018.

[179] Yu Zhan, Yuzhou Luo, Xunfei Deng, Michael L Grieneisen, Minghua Zhang, and Baofeng Di. Spatiotemporal prediction of daily ambient ozone levels across china using random forest for human exposure assessment. *Environmental Pollution*, 233:464–473, 2018.

[180] Keyong Huang, Qingyang Xiao, Xia Meng, Guannan Geng, Yujie Wang, Alexei Lyapustin, Dongfeng Gu, and Yang Liu. Predicting monthly high-resolution pm2.5 concentrations with random forest model in the north china plain. *Environmental Pollution*, 242:675–683, 2018.

[181] Wentao Yang, Min Deng, Feng Xu, and Hang Wang. Prediction of hourly pm2. 5 using a space-time support vector regression model. *Atmospheric Environment*, 181:12–19, 2018.

[182] Yuchao Zhou, Suparna De, Gideon Ewa, Charith Perera, and Klaus Moessner. Data-driven air quality characterization for urban environments: A case study. *IEEE Access*, 6:77996–78006, 2018.

[183] Brian S Freeman, Graham Taylor, Bahram Gharabaghi, and Jesse Thé. Forecasting air quality time series using deep learning. *Journal of the Air & Waste Management Association*, 68(8):866–886, 2018.

[184] Raquel Martınez-Espana, Andrés Bueno-Crespo, Isabel Timón, Jesús Soto, Andrés Munoz, and José M Cecilia. Air-pollution prediction in smart cities through machine learning methods: A case of study in murcia, spain. *Journal of Universal Computer Science*, 24(3):261–276, 2018.

[185] Nabil Mohamed Eldakhly, Magdy Aboul-Ela, and Areeg Abdalla. A novel approach of weighted support vector machine with applied chance theory for forecasting air pollution phenomenon in egypt. *International Journal of Computational Intelligence and Applications*, 17(01):1850001, 2018.

[186] Chiou-Jye Huang and Ping-Huan Kuo. A deep cnn-lstm model for particulate matter (pm2. 5) forecasting in smart cities. *Sensors*, 18(7):2220, 2018.

[187] Dixian Zhu, Changjie Cai, Tianbao Yang, and Xun Zhou. A machine learning approach for air quality prediction: Model regularization and optimization. *Big data and cognitive computing*, 2(1):5, 2018.

[188] Yara Abu Awad, Petros Koutrakis, Brent A Coull, and Joel Schwartz. A spatio-temporal prediction model based on support vector machine regression: Ambient black carbon in three new england states. *Environmental research*, 159:427–434, 2017.

[189] XY Ni, Hong Huang, and WP Du. Relevance analysis and short-term prediction of pm2. 5 concentrations in beijing based on multi-source data. *Atmospheric environment*, 150:146–161, 2017.

[190] Jan Kleine Deters, Rasa Zalakeviciute, Mario Gonzalez, and Yves Rybarczyk. Modeling pm2. 5 urban pollution using machine learning and selected meteorological parameters. *Journal of Electrical and Computer Engineering*, 2017, 2017.

[191] Yu Zhan, Yuzhou Luo, Xunfei Deng, Huajin Chen, Michael L Grieneisen, Xueyou Shen, Lizhong Zhu, and Minghua Zhang. Spatiotemporal prediction of continuous daily pm2. 5 concentrations across china using a spatially explicit machine learning algorithm. *Atmospheric environment*, 155:129–139, 2017.

[192] Abdullah N Al-Dabbous, Prashant Kumar, and Abdul Rehman Khan. Prediction of airborne nanoparticles at roadside location using a feed–forward artificial neural network. *Atmospheric Pollution Research*, 8(3):446–454, 2017.

[193] Nabil Mohamed Eldakhly, Magdy Aboul-Ela, and Areeg Abdalla. Air pollution forecasting model based on chance theory and intelligent techniques. *International Journal on Artificial Intelligence Tools*, 26(06):1750024, 2017.

[194] Jiangshe Zhang and Weifu Ding. Prediction of air pollutants concentration based on an extreme learning machine: the case of hong kong. *International journal of environmental research and public health*, 14(2):114, 2017.

[195] Bing-Chun Liu, Arihant Binaykia, Pei-Chann Chang, Manoj Kumar Tiwari, and Cheng-Chin Tsao. Urban air quality forecasting based on multi-

dimensional collaborative support vector regression (svr): A case study of beijing-tianjin-shijiazhuang. *PloS one*, 12(7):e0179763, 2017.

[196] Khaled Bashir Shaban, Abdullah Kadri, and Eman Rezk. Urban air pollution monitoring system with forecasting models. *IEEE Sensors Journal*, 16(8): 2598–2606, 2016.

[197] Wani Tamas, Gilles Notton, Christophe Paoli, Marie-Laure Nivet, and Cyril Voyant. Hybridization of air quality forecasting models using machine learning and clustering: An original approach to detect pollutant peaks. *Aerosol and Air Quality Research*, 16(2):405–416, 2016.

[198] Arwa S Sayegh, Said Munir, Turki M Habeebullah, et al. Comparing the performance of statistical models for predicting pm10 concentrations. *Aerosol and Air Quality Research*, 14(3):653–665, 2014.

[199] Edouard Debry and Vivien Mallet. Ensemble forecasting with machine learning algorithms for ozone, nitrogen dioxide and pm10 on the prev'air platform. *Atmospheric environment*, 91:71–84, 2014.

[200] Antonios Papaleonidas and Lazaros Iliadis. Neurocomputing techniques to dynamically forecast spatiotemporal air pollution data. *Evolving Systems*, 4 (4):221–233, 2013.

[201] Kunwar P Singh, Shikha Gupta, and Premanjali Rai. Identifying pollution sources and predicting urban air quality using ensemble learning methods. *Atmospheric Environment*, 80:426–437, 2013.

[202] Chi-Man Vong, Weng-Fai Ip, Pak-kin Wong, and Jing-yi Yang. Short-term prediction of air pollution in macau using support vector machines. *Journal of Control Science and Engineering*, 2012, 2012.

[203] Bijan Yeganeh, M Shafie Pour Motlagh, Yousef Rashidi, and Hamidreza Kamalan. Prediction of co concentrations based on a hybrid partial least square and support vector machine model. *Atmospheric Environment*, 55: 357–365, 2012.

[204] Syed Masiur Rahman, AN Khondaker, and Radwan Abdel-Aal. Self organizing ozone model for empty quarter of saudi arabia: Group method data

handling based modeling approach. *Atmospheric environment*, 59:398–407, 2012.

[205] Vivien Mallet, Gilles Stoltz, and Boris Mauricette. Ozone ensemble forecast with machine learning algorithms. *Journal of Geophysical Research: Atmospheres*, 114(D5), 2009.

[206] Wenjian Wang, Changqian Men, and Weizhen Lu. Online prediction model based on support vector machine. *Neurocomputing*, 71(4-6):550–558, 2008.

[207] Ditsuhi Iskandaryan, Francisco Ramos, and Sergio Trilles. Supplementary Materials for 'Spatiotemporal Prediction of Air Quality Using Machine Learning Techniques', November 2022. URL `https://doi.org/10.5281/zenodo.7351424`.