

**Universidad Internacional de la Rioja (UNIR)**

**Escuela Superior de Ingeniería y  
Tecnología**

**Máster en Análisis y Visualización de Datos  
Masivos**

# Predicción del tráfico en la ciudad de Madrid usando Redes Neuronales de Grafos

**Trabajo Fin de Máster**

**Presentado por:** Alejandro de la Calle Negro

**Director/a:** Ricardo Serafín Alonso Rincón

**Ciudad:** Salamanca

**Fecha:** Febrero 2022



## Resumen

Uno de los mayores retos de las sociedades modernas es la gestión del tráfico rodado y los problemas que se derivan directamente: la congestión, los accidentes y la contaminación. Para poder mitigar y suprimir estos efectos adversos, es necesario crear descripciones cada vez más precisas y realistas que puedan dar cuenta de un fenómeno tan complejo como es el tráfico. En concreto, gracias al cada vez mayor volumen de datos disponibles en la actualidad, en los últimos años han sido numerosos los trabajos basados en aprendizaje automático que han planteado modelos predictivos por medio del aprendizaje a través de fuentes de datos reales. En concreto, el tipo de redes neuronales basadas en grafos es capaz de captar la correlación espacial y temporal que existe entre los distintos elementos de una red de carreteras entendida como un grafo. Este trabajo tiene como objetivo la evaluación de una pareja de modelos predictivos basados en redes neuronales de grafos, DCRNN y Graph Wavenet (GWN), aplicados al caso de la ciudad de Madrid. Para ello, se ha compilado un dataset a partir de datos abiertos correspondientes a medidas tomadas durante seis meses por una red de sensores en las vías de la *M30* de la ciudad de Madrid. En primer lugar el análisis del conjunto de datos obtenido muestra una fuerte periodicidad con dos marcadas frecuencias: una diaria, con altibajos en torno a las horas de entrada y salida de la jornada laboral y un mínimo de actividad nocturno; y otra semanal, con una clara separación entre los valores de los días laborables y el fin de semana. En segundo lugar, los resultados de las evaluaciones revelan una dependencia nula de la parte espacial. Este hecho puede deberse a que las vías de circulación de la *M30* son de único sentido, no existiendo en esas carreteras flujo en ambos sentidos que pueda ser recogido por un mismo sensor. Además, los resultados del modelo GWN muestran que la inclusión de un término adaptativo que permite obtener la matriz de adyacencias en tiempo de entrenamiento mejora apreciablemente la predicción. Aunque la inclusión solamente de dicho término posee un error claramente superior al resto de casos, los resultados muestran que es posible realizar un pronóstico con este tipo de modelos sin un conocimiento de la estructura del grafo.

**Palabras clave:** Redes Neuronales de grafos, Grafos, Modelos del tráfico, Aprendizaje profundo, Datos Abiertos





## Abstract

One of the greatest challenges facing modern societies is the management of road traffic and the problems that directly result from it: congestion, accidents and pollution. In order to mitigate and suppress these adverse effects, it is necessary to create increasingly accurate and realistic descriptions that can account for such a complex phenomenon as traffic. In particular, thanks to the ever-increasing volume of data currently available, in recent years there have been numerous works based on machine learning that have proposed predictive models by means of learning through real data sources. Specifically, the type of neural networks based on graphs is able to capture the spatial and temporal correlation that exists between the different elements of a road network understood as a graph. This work aims to evaluate a pair of predictive models based on graph neural networks, DCRNN and Graph WaveNet (GWN), applied to the case of the city of Madrid. For this purpose, a dataset has been compiled from open data corresponding to measurements taken during six months by a sensor network on the roads of the city of Madrid. Firstly, the analysis of the obtained dataset shows a strong periodicity with two marked frequencies: a daily one, with ups and downs around the hours of entry and exit of the working day and a minimum of night activity; and a weekly one, with a clear separation between weekday and weekend values. Secondly, the results of the evaluations reveal a null dependence on the spatial dependency. This fact may be due to the fact that the traffic roads of the *M30* are one-way, with no two-way flow on these roads that can be picked up by the same sensor. In addition, the results of the GWN model show that the inclusion of an adaptive term to obtain the adjacency matrix at training time improves the prediction significantly. Although the inclusion of this term alone has a clearly higher error than the other cases, the results show that it is possible to make a forecast with this type of model without knowledge of the network structure.

**Keywords:** Graphs, Graph Neural Networks, Deep Learning, Open Data



Para Pilar, Carmela y Amaia.



## **Agradecimientos**

En primer lugar me gustaría agradecer a mi director de TFM, Ricardo Alonso Rincón, su experiencia, disposición e interés a la hora de guiar este trabajo.

También quiero agradecer a mi madre, Julia, y a mis padres políticos, Concha y Pedro, el apoyo y tiempo que me han prestado al cuidar de las pequeñas mientras yo escribía. Agradecer muy especialmente a Pilar, mi esposa, por su incommensurable amor e insondable paciencia, que me ha permitido realizar este trabajo. Sin ella jamás hubiera podido acabar. Y a Carmela y Amaia, por si algún día leen estas líneas y descubren así que es lo que hacía yo durante muchas de las noches de estos últimos meses mientras ellas dormían.



## ***Índice general***

<b>Resumen</b>	<b>III</b>
<b>Abstract</b>	<b>V</b>
<b>Agradecimientos</b>	<b>IX</b>
<b>Índice General</b>	<b>XI</b>
<b>Índice de Figuras</b>	<b>XIII</b>
<b>Índice de Tablas</b>	<b>XV</b>
<b>1. Introducción</b>	<b>1</b>
1.1. Motivación . . . . .	1
1.2. Estructura . . . . .	4
<b>2. Contexto y estado del arte</b>	<b>7</b>
2.1. Contexto . . . . .	7
2.1.1. Congestión del tráfico . . . . .	7
2.1.2. Accidentes de tráfico . . . . .	8
2.1.3. Calidad del aire . . . . .	11
2.2. Estado del arte . . . . .	14
2.2.1. Modelos basados en técnicas de aprendizaje automático clásico . . . . .	15
2.2.2. Modelos basados en redes neuronales . . . . .	17
2.2.3. Modelos basados en redes neuronales de grafos . . . . .	21
2.3. Conclusiones . . . . .	25
<b>3. Objetivos y metodología</b>	<b>29</b>
3.1. Objetivo general . . . . .	29
3.2. Objetivos específicos . . . . .	29
3.3. Metodología . . . . .	30

3.3.1. Identificación de requisitos . . . . .	30
3.3.2. Descripción del desarrollo . . . . .	35
<b>4. Desarrollo específico de la contribución</b>	<b>39</b>
4.1. Implementación del dataset . . . . .	39
4.1.1. Fuente de datos . . . . .	39
4.1.2. Análisis exploratorio . . . . .	41
4.1.3. Preprocesamiento y creación del dataset . . . . .	47
4.2. Modelos predictivos . . . . .	52
4.2.1. Definición del problema . . . . .	52
4.2.2. Diffusion Convolutional Recurrent Neural Network (DCRNN) . . . . .	53
4.2.3. Graph Wavenet (GWN) . . . . .	53
4.3. Evaluación de resultados . . . . .	55
4.3.1. DCRNN . . . . .	55
4.3.2. Graph Wavenet . . . . .	57
4.3.3. Comparativa . . . . .	63
<b>5. Conclusiones y trabajo futuro</b>	<b>67</b>
5.1. Conclusiones . . . . .	67
5.2. Trabajo futuro . . . . .	70
<b>Bibliografía</b>	<b>73</b>



## Índice de Figuras

1. Accidentes con heridos implicados en España, desde 1993 hasta 2019 . . . . .	2
2. Niveles medios anuales de NO <sub>2</sub> por estación de medida en la ciudad de Madrid durante 2019. . . . .	3
3. Congestión del tráfico semanal en la ciudad de Madrid durante 2019. . . . .	8
4. Horas y dinero perdidos derivados de los atascos de tráfico. . . . .	9
5. Número de muertes causadas por accidentes de tráfico en España. . . . .	10
6. Accidentes con heridos implicados en España, desde 1993 hasta 2019 . . . . .	10
7. Mapa del mundo mostrando la tasa de mortalidad por país causada por la contaminación del aire en 2020. . . . .	12
8. Serie temporal de la emisión de contaminantes en la Unión Europea provenientes del tráfico rodado. . . . .	13
9. Serie temporal del porcentaje de nivel de NO <sub>2</sub> desde 2010 . . . . .	13
10. Niveles medios anuales de NO <sub>2</sub> por estación de medida en la ciudad de Madrid durante 2019. . . . .	14
11. Estructura de una jerarquía en una red de carreteras. . . . .	15
12. Diagrama del flujo del modelo KRR-GPR. . . . .	16
13. Esquema de una red neuronal recurrente. . . . .	17
14. Esquema de una red neuronal recurrente. . . . .	18
15. Diagrama del modelo 2D-LSTM. . . . .	18
16. Diagrama del modelo DVMST. . . . .	20
17. Diagrama del modelo STDN. . . . .	21
18. Comparación de una estructura regular de datos y un grafo. . . . .	22
19. Representación de una secuencia temporal de grafos. . . . .	22
20. Diagrama del modelo DCRNN. . . . .	23
21. Diagrama del modelo STAWnet. . . . .	25
22. Crecimiento de los lenguajes de programación más usados en los países ricos en los últimos años. . . . .	32

23. Librerías y frameworks más populares en 2021. . . . .	33
24. Porcentaje de popularidad de TensorFlow y Pytorch. . . . .	34
25. Diagrama de flujo de las tareas. . . . .	36
26. Mapa de los puntos de medida. . . . .	42
27. Histogramas de las variables del conjunto de datos <i>M30</i> . . . . .	44
28. Señales recogidas durante una semana en seis puntos de medida aleatorios. . .	45
29. Señales recogidas durante una semana en seis puntos de medida aleatorios. . .	46
30. Valores medios diarios del conjunto de datos de la <i>M30</i> . . . . .	47
31. Valores medios semanales del conjunto de datos de la <i>M30</i> . . . . .	48
32. Valores medios mensuales del conjunto de datos de la <i>M30</i> . . . . .	49
33. Valores medios anuales del conjunto de datos de la <i>M30</i> . . . . .	50
34. Arquitectura del modelo DCRNN. . . . .	54
35. Bloque de Graph Wavenet. . . . .	54
36. Valores de la pérdida para el entrenamiento del modelo DCRNN. . . . .	56
37. Error cuadrático medio sobre el conjunto de test en función del horizonte de predicciones para el modelo DCRNN. . . . .	57
38. Pérdidas de los conjuntos de entrenamiento y validación para el entrenamiento del modelo GWN. . . . .	58
39. Error cuadrático medio en función del horizonte de predicciones para el modelo GWN. . . . .	61
40. Pérdidas de validación del modelo GWN para diferentes longitudes de secuencia. .	61
41. Error cuadrático medio para las predicciones del modelo GWN, con diferentes longitudes de secuencia de entrada y salida. . . . .	62

## Índice de Tablas

1. Resultados del error cuadrático medio para la combinación de SVM con técnicas de limpieza de datos. . . . .	16
2. Comparativa de resultados del error cuadrático medio para predicción con redes convolucionales. . . . .	19
3. Comparativa de resultados del error cuadrático medio para predicción con modelos de aprendizaje profundo . . . . .	20
4. Comparativa del error cuadrático medio para técnicas de grafos. . . . .	24
5. Resumen de los modelos repasados en el estado del arte. . . . .	26
6. Comparativa entre los lenguajes Python y R. . . . .	35
7. Esquema del histórico del tráfico . . . . .	40
8. Esquema del conjunto de datos sobre los puntos de medida. . . . .	41
9. Estadística del número de puntos de medida del dataset. . . . .	42
10. Principales estadísticas del dataset <i>M30</i> (nuestro), y de los datasets METR-LA y PEMS-BAY. . . . .	43
11. Estadística de los principales estimadores para el dataset utilizado en los experimentos. . . . .	43
12. Muestra de las cinco primeras filas y seis primeras columnas del dataset <i>M30</i> . . . . .	51
13. Partición del dataset <i>M30</i> . . . . .	52
14. MAE y el RMSE para los entrenamientos del modelo DCRNN. . . . .	57
15. Error cuadrático medio (RMSE) del modelo GWN para diferentes tipos de convolución. . . . .	59
16. Error absoluto medio (MAE) del modelo GWN para diferentes tipos de convolución. . . . .	60
17. Error porcentual absoluto medio (MAPE) del modelo GWN para diferentes tipos de convolución. . . . .	60
18. Error cuadrático medio (RMSE) del modelo GWN para diferentes longitudes de secuencia. . . . .	62
19. Resultados de la comparativa de modelos . . . . .	64

20. Tiempos de ejecución para los experimentos con los modelos DCRNN y GWN. .	64
21. Error absoluto medio (MAE) para los modelos DCRNN y GWN sobre METR-LA, PEMS-BAY y M30. . . . .	65
22. Error cuadrático medio (RMSE) para los modelos DCRNN y GWN sobre METR-LA, PEMS-BAY y M30. . . . .	65

## 1. Introducción

En este Capítulo se realizará una breve introducción al presente trabajo, que comenzará presentando una motivación para abordar la problemática del pronóstico del tráfico rodado en vías urbanas, seguido de una descripción de la propuesta que realiza este estudio ante dicha problemática. También se presentará la estructura de esta memoria, repasando el contenido de las diferentes secciones en las que se divide este estudio. La Sección 1.1 contendrá la motivación, mientras que en la Sección 1.2 se enunciará la estructura de esta memoria.

### 1.1. Motivación

Actualmente la gestión del tráfico es uno de los retos a los que se enfrentan las sociedades modernas. La mayoría de los europeos vive en ciudades, ya que según la Comisión Europea en 2010 el 73 % de los europeos vive en zonas urbanas, e incluso se espera que esta cifra llegue al 80 % en 2050 (Comisión Europea, 2017). Esta concentración de personas hace que la movilidad sea hoy por hoy uno de los grandes problemas a resolver de la humanidad este siglo.

De esta concentración de personas también se derivan los grandes problemas que se derivan del tráfico, que son la congestión producida por el tráfico rodado, los accidentes y, por último, uno de los problemas que la humanidad no puede demorar para este siglo: la contaminación.

El primero de esta problemática es la congestión. Según TomTom (2021b), las ciudades más congestionadas del planeta son Bengaluru en India y Manila en Filipinas, con un nivel de congestión del 71 %, seguida de Bogotá, en Colombia, con un 68 %. La ciudad europea más congestionada es Moscú, con un índice de congestión de un 59 %. Barcelona es la ciudad más congestionada de España en 2019, con un índice del 29 % de congestión, seguida de Granada (25 %), Palma de Mallorca (24 %) y Madrid (23 %). De hecho, en Madrid se estima que el tiempo perdido en trayectos en las horas punta del día asciende a 30 minutos. Los efectos negativos de los atascos también son económicos, ya que según estimaciones de la Comisión Europea los costes derivados se cifran en alrededor de 130 mil millones de euros de 2011 (Comisión

Europea, 2017).

Por otro lado, en la Figura 1 se puede observar la evolución de las muertes en España por accidentes de tráfico desde 1960 hasta 2019, año en el que se produjeron unos 1,755 fallecimientos, según estudios de la Dirección General de Tráfico (DGT, 2020). De ellos, un 53 % de las personas eran peatones, ciclistas o motoristas. En vías urbanas, el número de fallecidos se incrementó un 6 % más que en 2018. Es de destacar que una gran parte de las muertes causadas por accidentes de tráfico se produjeron en ciudades, ya que un 30 % del total de accidentes se concentraron en ellas, el mayor porcentaje desde que la DGT tiene registros. Esto indica la gran importancia que tiene la previsión del tráfico para poder regular la congestión y mitigar si no suprimir la siniestralidad en vías urbanas.

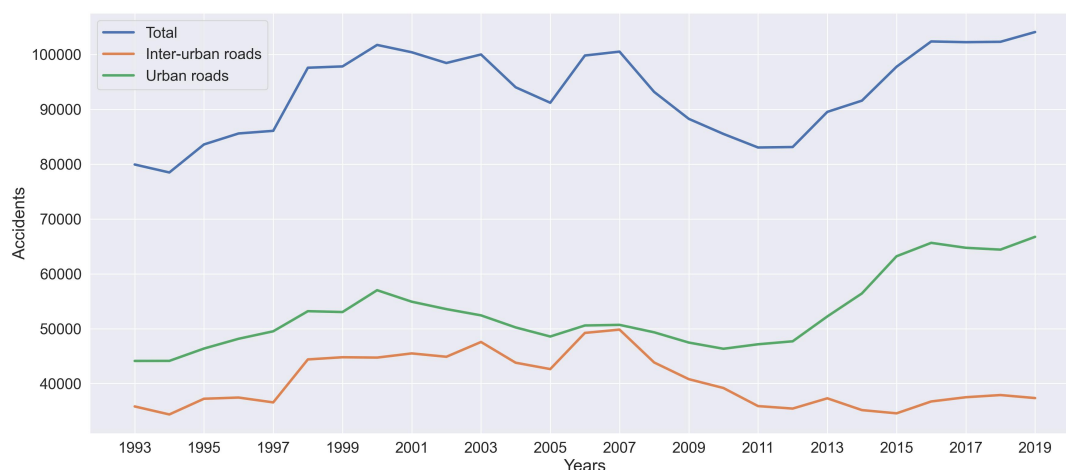


Figura 1: Accidentes con heridos implicados en España, desde 1993 hasta 2019 (DGT, 2021b).

Uno de los inconvenientes asociados al tráfico es el no menos importante problema de la contaminación del aire, ya que según datos de la Organización Mundial de la Salud un 91 % de la población del planeta vive en áreas donde se exceden los límites de contaminación fijados por las instituciones públicas (WHO, 2018b). Esta misma contaminación es la responsable de entre dos y cuatro millones de muertes al año (WHO, 2018a). Entre las causas de la contaminación el tráfico rodado se cuenta entre las que causan polución en áreas urbanas, siendo responsable de emisiones de  $\text{NO}_2$ ,  $\text{PM}_{2.5}$  y  $\text{PM}_{10}$  entre otros contaminantes. En los últimos años se han emprendido muchos esfuerzos por parte de autoridades locales y nacionales para reducir esta contaminación, como son la implantación de protocolos de actuación para la limitación del tráfico urbano. En concreto, sucesivas políticas han conseguido reducir un 20 % la concentración de  $\text{NO}_2$  en la ciudad de Madrid (Ayuntamiento de Madrid, 2019). Sin embargo, actualmente todavía existen dos estaciones de medida donde se recogen niveles de este contaminante que

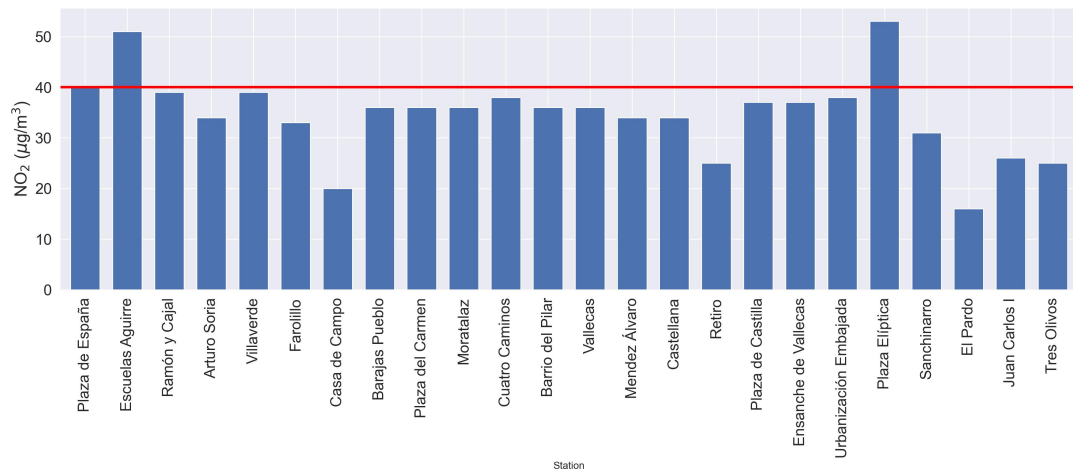


Figura 2: Niveles medios anuales de NO<sub>2</sub> por estación de medida en la ciudad de Madrid durante 2019. La línea gruesa horizontal representa el límite anual de este contaminante por estación, que es de  $40\mu\text{g}/\text{cm}^3$ . Datos recogidos de Ayuntamiento de Madrid (2019)

exceden los límites legales, tal como puede verse en la Figura 2.

Es por ello que, tanto para la congestión producida por el tráfico como para los accidentes o la contaminación, contar con precisos modelos predictivos ayudará a crear sólidas políticas públicas que conduzcan a la gestión y solución de estos problemas (Comisión Europea, 2017).

En los últimos años han sido numerosos los trabajos que han considerado modelos de aprendizaje automático para la tarea del pronóstico del tráfico (Zhang et al. 2019). Entre estos modelos podemos distinguir tres categorías: los modelos basados en técnicas de aprendizaje clásicas (P. Cai et al. 2016; Z. Guo et al. 2019; Tang et al. 2019), los modelos basados de aprendizaje profundo (Ma et al. 2017; S. Sun et al. 2019; Yao et al. 2019; Yao et al. 2018; Zhang et al. 2017; Zhao et al. 2017) y, dentro de estos, aquellos modelos basados más específicamente de aprendizaje profundo basado en grafos (Do et al. 2019; S. Guo et al. 2019; Li et al. 2018; Tian et al. 2021; Wu et al. 2019; Yu et al. 2018).

Aunque ofrecen unos resultados aceptables, los modelos basados en aprendizaje automático clásico cuentan con la desventaja de que necesitan de un ajuste y pre-procesado manual de los datos de entrada, por lo que hace que estos modelos sean inadecuados en grandes volúmenes de datos. En cambio, los modelos basados en aprendizaje profundo cuentan con una mayor eficiencia en el cálculo sobre grandes cantidades de datos, pudiendo capturar correlaciones temporales entre las distintas variables del tráfico (Tian et al. 2021). Sin embargo, estos mismos modelos suelen ser ineficaces captando correlaciones espaciales entre los distintos puntos del tráfico. Son los modelos de aprendizaje profundo basados en grafos los capaces

de describir al mismo tiempo inter-dependencias espaciales y temporales entre los nodos de una red de carreteras por la que fluye el tráfico de vehículos (Li et al. 2018; Wu et al. 2019). Es en este tipo de modelos en los que se centrará el presente trabajo, los cuales de manera habitual han sido entrenados utilizando datasets públicos obtenidos a partir de datos provenientes de ciudades estadounidenses. Sin embargo, hasta donde llega el conocimiento del autor, actualmente no existen estudios donde se muestren resultados sobre la eficiencia de modelos entrenados con datos originarios de ciudades españolas y, más concretamente, con datos abiertos de la ciudad de Madrid.

El presente trabajo tiene por objeto el evaluar la eficiencia y precisión de una selección de modelos de aprendizaje profundo basados en grafos con datos abiertos tomados de fuentes públicas relativos a la ciudad de Madrid. Hasta donde llega el conocimiento del autor, la compilación de este conjunto de datos específico para el entrenamiento de modelos predictivos del trabajo sería el primero realizado relativo a las carreteras de la ciudad de Madrid. Por otra parte, también sería el primer estudio aplicado a Madrid de modelos predictivos de esta naturaleza. El código resultante de este trabajo se puede consultar en el repositorio cuya *url* es: <https://github.com/itsAlexNoir/bovra>

## 1.2. Estructura

Este trabajo está dispuesto de la siguiente manera:

- En el Capítulo 2 se presenta el contexto y el estado del arte. Una revisión de los principales problemas surgidos del tráfico se expone en la Sección 2.1, mientras que en la Sección 2.2 se discute el estado del arte en modelos predictivos del tráfico basados en aprendizaje automático.
- En el Capítulo 3 se expone el objetivo general que persigue este trabajo, así como los objetivos específicos, enumerados en la Sección 3.2. En la Sección 3.3 se presenta la metodología seguida.
- El Capítulo 4 está dedicado al desarrollo específico de la contribución. La discusión de conjunto de datos está expuesto en la Sección 4.1, donde en la Subsección 4.1.1 se indican las fuentes primarias de los datos, en la Subsección 4.1.2 se presenta un pequeño estudio previo sobre los datos que forman el conjunto y, por último, en la Subsección 4.1.3 se discute el pre-procesado de los datos realizado sobre los datos descargados para su posterior uso. En la Sección 4.2 se exponen los modelos predictivos que forman parte



de este estudio, describiendo más en detalle su funcionamiento. Finalmente, en la Sección 4.3 se presenta una evaluación realizada a los modelos con el conjunto de datos compilado en la Sección 4.1, en primer lugar cada modelo por separado (en las Subsecciones 4.3.1 y 4.3.2 respectivamente), para en último lugar discutir una comparativa entre los tres modelos en la Subsección 4.3.3.

- Finalmente, las conclusiones y el trabajo futuro se discuten en el Capítulo 5. Las conclusiones del trabajo se exponen individualmente en la Sección 5.1, de forma que resulten alineadas con los objetivos marcados en la Sección 3.2, mientras que en la Sección 5.2 se apuntan algunas líneas de trabajo futuro.



## **2. Contexto y estado del arte**

En el siguiente Capítulo se tratará de repasar el contexto en el que se enmarca el presente trabajo, recorriendo las principales problemáticas que genera el tráfico en las sociedades actuales: la congestión, los accidentes y la contaminación. También se discutirá una visión general de los trabajos que en los últimos años han abordado el problema de la predicción del tráfico, centrándose en aquellos que hacen uso de las técnicas de aprendizaje automático. En primer lugar, en la Sección [2.1](#) se hace una exposición del contexto. Tras ello, en la Sección [2.2](#) se encuentra un recorrido del estado del arte en modelos predictivos.

### **2.1. Contexto**

En Europa, en el año 2010, el 73 % de su población vivirá en zonas urbanas, y se estima que este porcentaje crecerá a más del 80 % en 2050 (Comisión Europea, [2017](#)). Debido a este hecho, se hace necesario crear una movilidad urbana robusta y eficiente que permita la expansión de las ciudades y su circulación interna. Por otro lado, es también importante reducir el impacto negativo de los problemas asociados al tráfico rodado como la congestión de la circulación en vías urbanas, los accidentes de tráfico y la contaminación del aire (Comisión Europea, [2017](#)). A continuación se repasará brevemente estos problemas. En la Subsección [2.1.1](#) se expone el problema de la congestión del tráfico y su evolución en los últimos años. En la Subsección [2.1.2](#) se presenta el problema de los accidentes de tráfico. Por último, en la Subsección [2.1.3](#) se discute el tema de la contaminación causada por el tráfico rodado, así como sus consecuencias adversas para la salud.

#### **2.1.1. Congestión del tráfico**

De acuerdo con el índice de tráfico de TomTom, Madrid posee un nivel de congestión del 23 %, lo que supone un aumento del 1 % comparado con el año anterior (TomTom, [2021a](#)). Más aún, se estima que el tiempo perdido en horas punta en un viaje de 30 minutos de duración en la ciudad de Madrid es de 17 minutos por las mañanas y 14 por las tardes. Esto suma un total de 117 horas perdidas al año, el equivalente a 4 días y 21 horas.

	Sun	Mon	Tue	Wed	Thu	Fri	Sat
12:00 AM	6%	2%	2%	3%	3%	5%	6%
	6%	1%	3%	2%	2%	3%	5%
02:00 AM	5%	1%	3%	2%	2%	2%	4%
	4%	1%	2%	1%	2%	2%	3%
04:00 AM	3%	0%	1%	1%	1%	1%	2%
	2%	0%	0%	0%	0%	0%	1%
06:00 AM	1%	12%	12%	12%	12%	9%	1%
	0%	44%	45%	45%	42%	36%	1%
08:00 AM	0%	59%	59%	60%	56%	43%	2%
	2%	40%	41%	42%	40%	26%	4%
10:00 AM	4%	17%	19%	20%	19%	15%	6%
	7%	16%	18%	18%	18%	17%	11%
12:00 PM	9%	16%	18%	18%	19%	18%	14%
	10%	15%	16%	17%	18%	22%	15%
02:00 PM	8%	16%	19%	20%	21%	48%	12%
	3%	17%	19%	20%	20%	52%	5%
04:00 PM	4%	15%	18%	19%	19%	30%	5%
	7%	26%	30%	31%	33%	27%	8%
06:00 PM	11%	39%	45%	46%	47%	25%	13%
	12%	31%	37%	38%	40%	21%	16%
08:00 PM	12%	15%	20%	21%	24%	20%	17%
	9%	9%	10%	11%	14%	16%	14%
10:00 PM	5%	5%	6%	6%	7%	10%	10%
	3%	3%	4%	5%	5%	6%	7%

Figura 3: Congestión del tráfico semanal en la ciudad de Madrid durante 2019 (en promedio). En esta Figura se puede apreciar que las horas punta están concentradas entre las 8 am y las 6 pm durante los días laborables. Esta franja horaria corresponde a las horas habituales de entrada y salida del trabajo. Reproducido de TomTom (2021a).

Asimismo, en 2011 en Europa los costes derivados de la congestión del tráfico ascendieron a 130 mil millones de euros, alrededor de un 1 % del PIB de la Unión Europea (Comisión Europea, 2017). En la Figura 4, aparte del tiempo perdido en atascos contados en horas por conductor, es interesante observar el coste de la congestión estimado por persona en tres países diferentes: Estados Unidos, Gran Bretaña y Alemania. Este coste suma anualmente un total de 300 \$ (The Economist, 2021).

### 2.1.2. Accidentes de tráfico

A pesar de la tendencia general a la baja de accidentes de tráfico en España durante las últimas décadas, en vías urbanas e inter-urbanas (ver Figura 5), el número de víctimas ha crecido ligeramente durante los últimos años (DGT, 2021b). A lo largo de 2019 (el último año publicado con datos oficiales), hubo un total de 104,080 accidentes, lo que representa un ascenso de 1,781 con respecto al año anterior (DGT, 2019). Si observamos la Figura 6, casi el doble de accidentes ocurrieron en vías urbanas (66,741) comparado con los que sucedieron en vías



Figura 4: Horas y dinero perdidos derivados de la congestión de tráfico. En la tabla de la izquierda se muestra el tiempo perdido para diferentes ciudades. En la tabla de la derecha aparece el coste promedio por conductor estimado para Estados Unidos, Gran Bretaña y Alemania. Reproducido de The Economist (2021).

inter-urbanas (37,339). De hecho, el número de accidentes en carreteras descendió en 553 durante el último año (2019), mientras que durante el mismo periodo se incrementó en 2,334 en vías urbanas. La misma comparación entre zonas urbanas e interurbanas es incluso más marcada en el caso de Madrid. Durante 2019 hubo 3,216 accidentes en vías inter-urbanas y 13,412 en urbanas, lo que supone una proporción de cuatro veces más en un caso que en el otro (DGT, 2021b). Es importante señalar que el 50 % de bajas causadas en la Unión Europea son peatones o ciclistas y, especialmente, personas mayores están entre las víctimas con más accidentes con resultado de muerte (Comisión Europea, 2017).

Más aún, la relación entre los accidentes y la congestión del tráfico ha sido extensamente es-

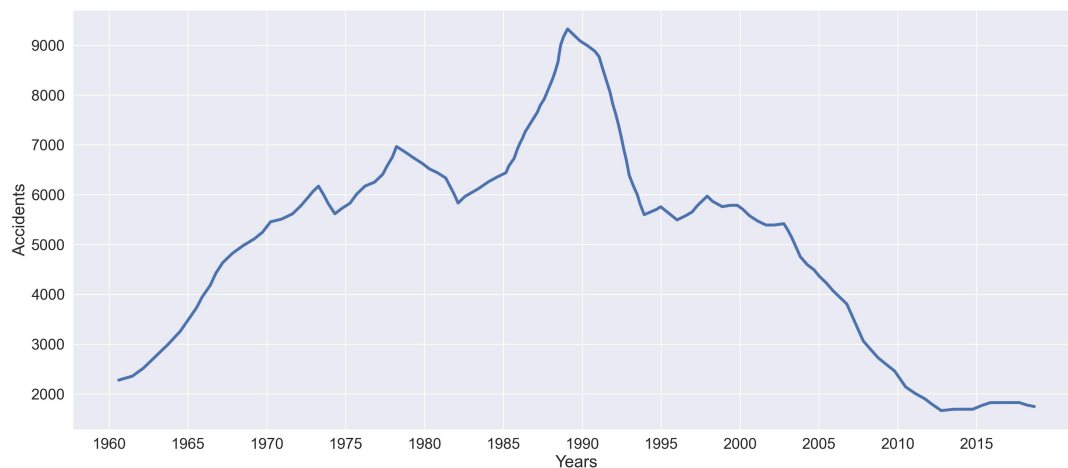


Figura 5: Número de muertes causadas por accidentes de tráfico en España. El gráfico abarca 60 años, desde 1960 hasta 2019 (DGT, [2019](#)).

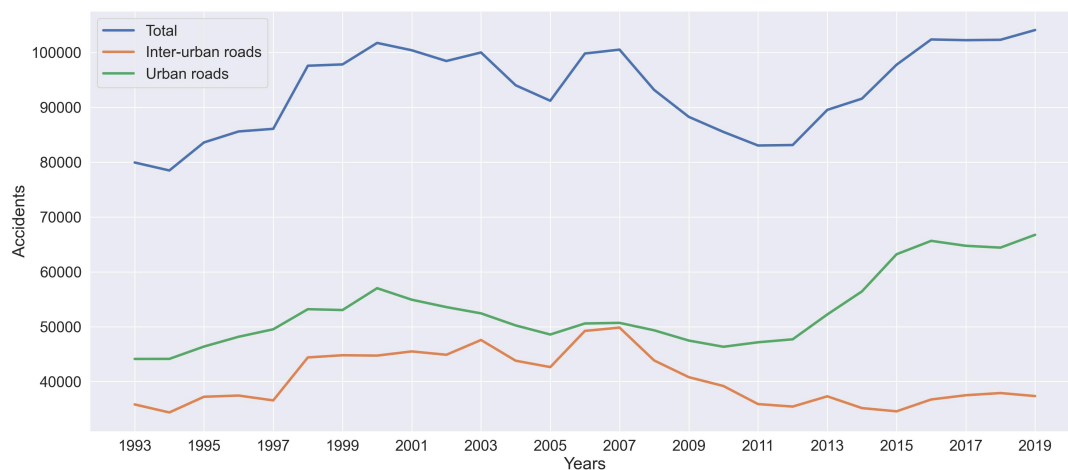


Figura 6: Accidentes con heridos implicados en España, desde 1993 hasta 2019 (DGT, [2021b](#)). En la Figura es posible ver que el número de accidentes en vías urbanas es significativamente superior comparado con el de vías inter-urbanas.

tudiada en el pasado (Retallack et al. [2019](#)). En los trabajos más tempranos existía un acuerdo general en que ambos aspectos, congestión y accidentes, estaban linealmente relacionados, y que, por lo tanto, más congestión provocaba más accidentes (Head A, [1959](#); J. Sun et al. [2016](#); Vitaliano et al. [1991](#); Woo, [1957](#)). Sin embargo, recientes estudios apuntan a una dependencia en forma de *U*, en la que podemos distinguir tres regiones: en la primera bajos niveles de congestión causan menos accidentes debido a un tráfico fluido; en la región central el nivel cada vez más alto de congestión provoca a su vez una tasa cada vez más alta de accidentes hasta alcanzar un umbral por encima el nivel de congestión es tan alto que conduce a una muy baja tasa de accidentes (Cadar et al. [2017](#); Wang et al. [2013](#)). Este comportamiento no lineal es particularmente interesante porque revela la inter-dependencia espacial y temporal del tráfico. Por la tanto, para poder regular y también mitigar los efectos adversos causados por la congestión y los accidentes de tráfico, políticos y legisladores deberán contar con métodos de predicción

del tráfico capaces de describir un fenómeno con toda su complejidad espacio-temporal (DGT, 2021a; TfL, 2021).

### 2.1.3. Calidad del aire

La calidad del aire está considerada uno de los mayores retos para las sociedades modernas. Su problemática está principalmente causada por la contaminación del aire, lo que supone un riesgo medio-ambiental y un grave riesgo para la salud pública. Podemos definir la polución del aire como la presencia de ciertas sustancias en la atmósfera terrestre, como son algunos gases, partículas y biomoléculas. Estas sustancias pueden provocar distintos daños al entorno y a los seres vivos, como humanos, animales y plantas. En humanos, estas pueden causar alergias, enfermedades e incluso la muerte (WHO, 2018a).

De acuerdo con datos de la Organización Mundial de la salud (OMS), se estima que entre 2 y 4 millones de muertes prematuras son causadas cada año como consecuencia de la contaminación del aire (WHO, 2018a, 2018b), y también que el 91 % de la población mundial vive en áreas donde la contaminación excede los límites marcados por la autoridades (WHO, 2018b). La Figura 7 muestra un mapa con la relación entre tasa de mortalidad y polución en 2012. Solo en Europa, la mala calidad del aire provocó alrededor de medio millón de muertes en 2015, 422,000 de las cuales fueron causadas por  $PM_{2.5}$ , 79,000 por dióxido de nitrógeno ( $NO_2$ ) y 17,700 por ozono troposférico ( $O_3$ ). En España se calcula que unas 30,000 personas al año están afectadas por la polución del aire. De hecho, en 2013 la Agencia Internacional para la Investigación del Cáncer (IARC), una agencia especializada dentro de la OMS, incluyó la lista de principales contaminantes atmosféricos como carcinógenos (IARC, 2016).

El origen de la polución tiene diversas fuentes, la mayoría causadas por el efecto de la humanidad (WHO, 2018b). Entre ellas podemos citar:

- Industrias, fábricas y centrales energéticas basadas en combustibles fósiles, refinerías e incineradoras.
- Transporte rodado, marítimo y aéreo.
- Quemas controladas, la mayoría provenientes de la agricultura y el control forestal.

Si nos fijamos en las áreas urbanas, la contaminación es mayormente proveniente de calefacciones (muchas de ellas siguen funcionando por combustión de carbón o combustibles fósiles), tráfico marítimo y aéreo (en ciudades que cuentan con un puerto y aeropuerto respectivamente)

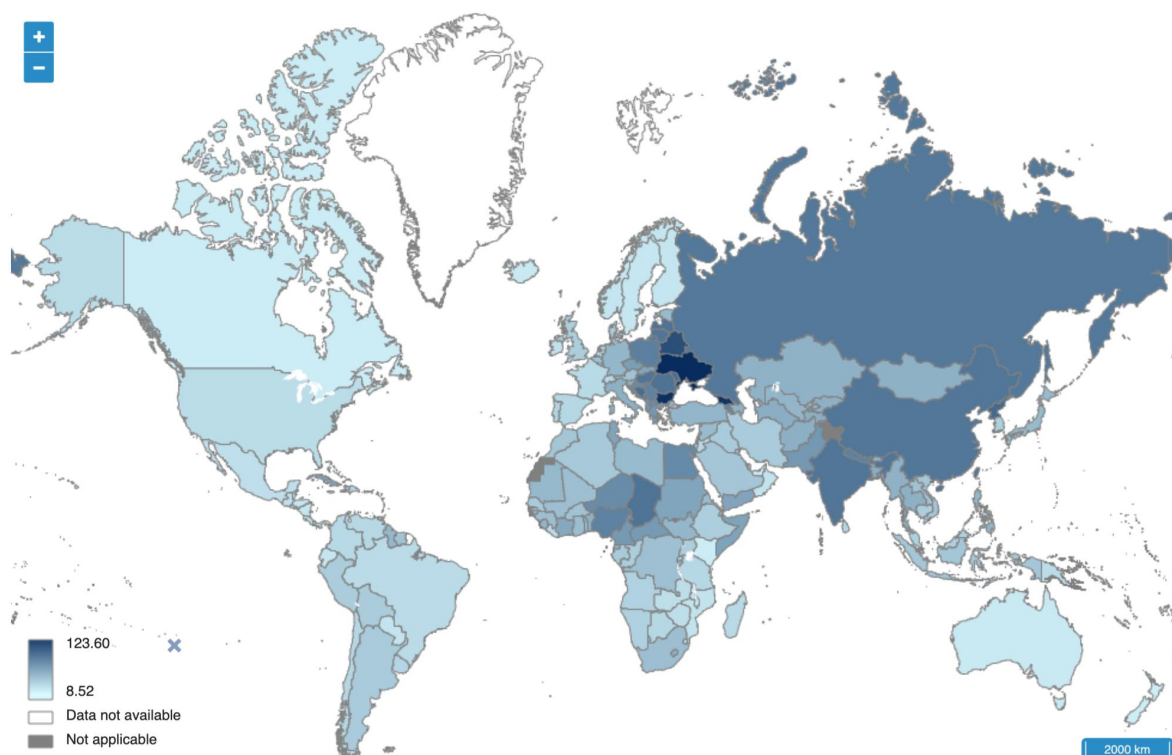


Figura 7: Mapa del mundo mostrando la tasa de mortalidad por cada 100,000 habitantes de cada país, causada por contaminación del aire en 2020. Reproducida de WHO (2021a)

y también tráfico rodado (EEA, 2020). El transporte por carreteras es responsable de la mayor parte de las emisiones de  $\text{NO}_2$  en la Unión Europea, así como también de los niveles de  $\text{PM}_{2.5}$  y  $\text{PM}_{10}$  (EEA, 2020). Es conocido que exposiciones al dióxido de nitrógeno puede causar tos y dificultades respiratorias, y una exposición prolongada a altos niveles pueden incrementar el riesgo de sufrir enfermedades respiratorias (WHO, 2021b).

Durante la última década gobiernos locales, regionales y nacionales a lo largo de diferentes países han creado iniciativas para la prevención y el control de la contaminación del aire, con el principal objetivo de reducir su avance. La idea que sostiene la mayoría de esos planes es limitar las emisiones de gases contaminantes en las ciudades imponiendo restricciones al tráfico rodado. Por ejemplo, como podemos observar en la Figura 8, las emisiones provenientes del tráfico se han reducido gradualmente durante esta última década, parcialmente debido a importantes restricciones y estrictos protocolos llevados a cabo por toda la Unión Europea. Concretamente en el caso de la ciudad de Madrid, los niveles recogidos de  $\text{NO}_2$  han disminuido cerca de un 20 % comparado con 2010 (Ayuntamiento de Madrid, 2019) (ver Figura 9). No obstante, los niveles de este gas sigue excediendo el límite legal permitido en algunas zonas de Madrid, tal como se puede apreciar en la Figura 10.



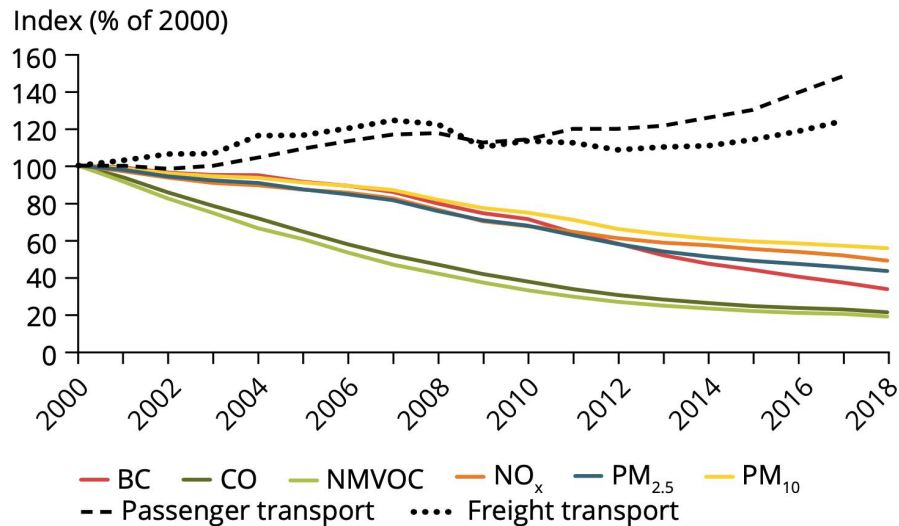


Figura 8: Serie temporal de la emisión de contaminantes en la Unión Europea provenientes del tráfico rodado. Los contaminantes son NO<sub>x</sub>, PM<sub>10</sub>, PM<sub>2.5</sub>, SO<sub>x</sub>, NMVOC, NH<sub>3</sub>, BC, CO y CH<sub>4</sub>. Los niveles están representados como porcentajes respecto a los niveles del año 2000. Por comparativa, en el mismo gráfico se muestra la estadística de la actividad para el sector del transporte en los 28 miembros de la Unión. Reproducido de EEA (2020).

Entre las acciones que se implantaron en Madrid estuvo la definición de un área de bajas emisiones, dentro de la cual el acceso de determinados vehículos (como aquellos con motor diesel o antiguos) estaba restringido. En otros de los protocolos se contemplaba la prohibición de circular dentro de ciertos perímetros de la ciudad cuando determinados niveles de contaminantes excedían los niveles permitidos.

Por todo lo expuesto anteriormente, el pronóstico del tráfico es crucial, no solo para tomar decisiones y acciones como planificación alternativa de rutas por adelantado cuando se impongan restricciones ante niveles altos de contaminación del aire, sino también, e incluso más

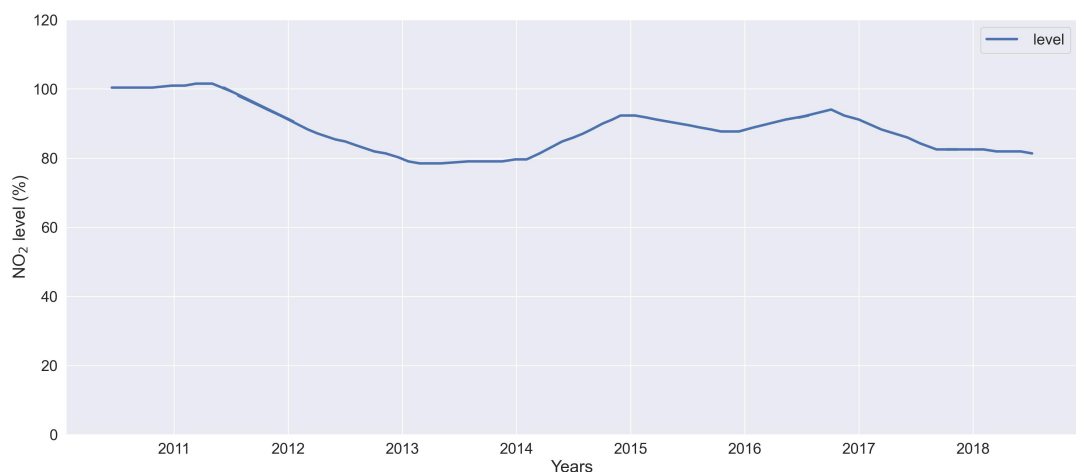


Figura 9: Serie temporal del porcentaje de nivel de NO<sub>2</sub> desde 2010. Durante el periodo que abarca el gráfico, el nivel se ha reducido cerca del 20 %. Datos tomados de Ayuntamiento de Madrid (2019).

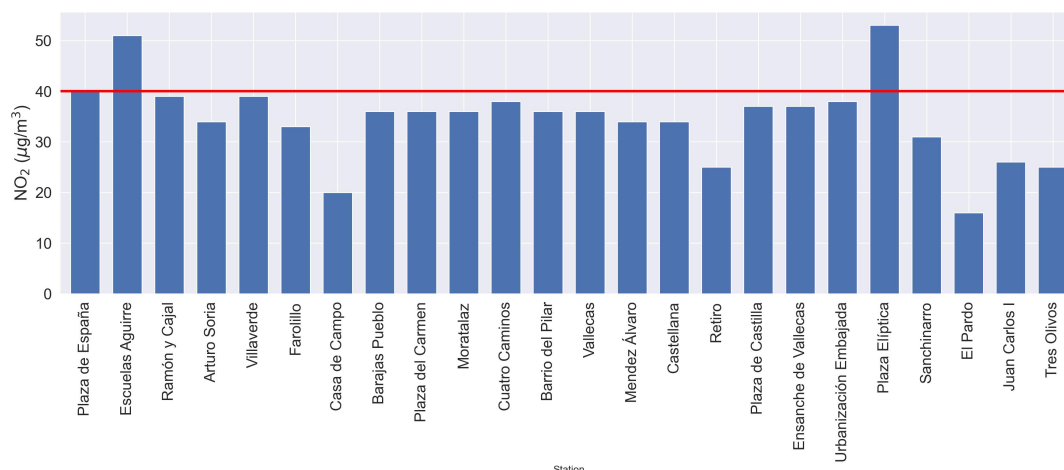


Figura 10: Niveles medios anuales de NO<sub>2</sub> por estación de medida en la ciudad de Madrid durante 2019. La línea gruesa horizontal representa el límite anual de este contaminante por estación, que es de  $40\mu g/cm^3$ . Datos recogidos de Ayuntamiento de Madrid (2019).

importante, para adelantarse a futuros episodios de contaminación debido al tráfico y así poder mitigar y finalmente suprimir dichos episodios de polución.

Por lo tanto, es necesario contar con modelos capaces de describir la naturaleza compleja y cambiante del tráfico. El desarrollo de dichos modelos deben proporcionar pronósticos a cada vez tiempos más largos de una forma precisa. Para ello deben poder captar la dinámica espacial y temporal del tráfico, así como su fuerte inter-dependencia. Las fluctuaciones horarias en el flujo del tráfico asociadas a la propagación de tráfico denso a través de ciertas vías. Es este sentido, no solo es necesario investigar en nuevos y más sofisticados modelos, sino en su evaluación es diferentes escenarios.

## 2.2. Estado del arte

En esta Sección se trazará un repaso al estado del arte en predicción del tráfico, centrándose en aquellos trabajos basados en técnicas de aprendizaje automático para describir y capturar la dinámica del tráfico.

Gracias a la expansión del *Big Data*, las técnicas basadas en el uso masivo de datos de tráfico reales han sufrido un avance notable durante los últimos años. A diferencia de trabajos anteriores, estos modelos basados enteramente en datos históricos del tráfico no asumen ninguna dinámica a priori. En este sentido, Zhang et al. (2019) distinguen tres categorías: modelos estadísticos, modelos basados en aprendizaje automático clásico y aquellos basados en aprendizaje profundo (Deep Learning). Este Capítulo se centrará en las dos últimas categorías. En la

Subsección 2.2.1 se repasarán los modelos basados en aprendizaje automático. En la Subsección 2.2.2 se tratarán los trabajos basados en modelos de aprendizaje profundo. Por último, en la Subsección 2.2.3 se apuntará a los trabajos que abordan el problema del tráfico empleando modelos de aprendizaje profundo basados en grafos.

### 2.2.1. Modelos basados en técnicas de aprendizaje automático clásico

Entre los modelos que hacen uso de aprendizaje automático clásico podemos contar el estudio de P. Cai et al. (2016), que emplea el algoritmo k-NN (las siglas de su nombre en inglés, k-Nearest Neighbours) para describir la correlación espacio-temporal entre distintos segmentos de carreteras. Como se puede apreciar en la Figura 11, dicha correlación se establece mediante una jerarquía entre los distintos segmentos de carretera que unen las intersecciones de una red de carreteras. Por medio de esta estructura, es posible establecer una matriz cuyos elementos son las velocidades a las que se circulan por los distintos segmentos, correspondiendo las columnas a los distintos segmentos, y las filas a los distintos instantes temporales.

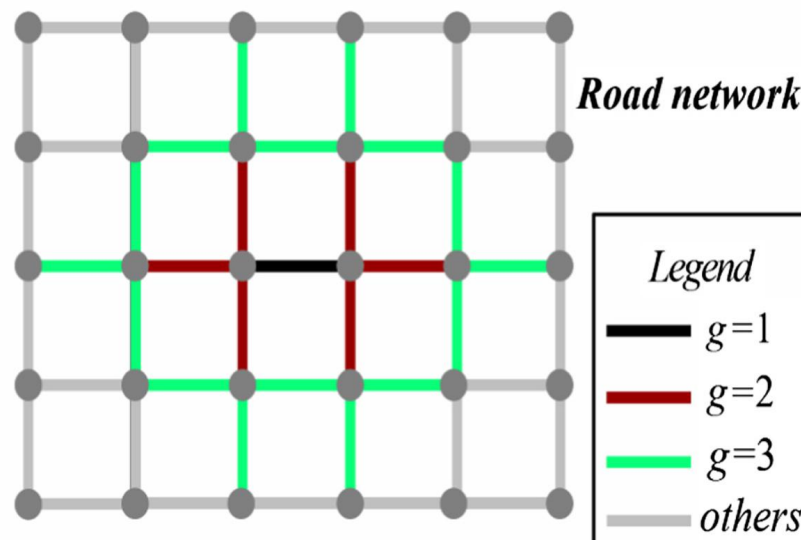


Figura 11: Estructura de una jerarquía en una red de carreteras. El primer grado ( $g=1$ ) representa el segmento primario, mientras que el segundo grado ( $g=2$ ) representa a los segmentos conectados con el segmento de primer grado, y así sucesivamente para grados de orden más alto. Reproducido de P. Cai et al. (2016).

En el trabajo de Tang et al. (2019) se propone el uso de técnicas de limpieza de ruido en datos para la mejora de la precisión del algoritmo Support-Vector Machine (SVM). Las técnicas empleadas en la limpieza son Empirical Decomposition Model (EMD), Ensemble Empirical Decomposition Model (EEDM), Moving Average (MA), filtro Butterworth y cuatro tipos diferentes de señales ondulatorias. En el estudio, todas las técnicas de limpieza contribuyen a la mejora de la predicción, siendo la técnica EEDM la que muestra un mejor rendimiento.

Tabla 1: Resultados del error cuadrático medio para la combinación de SVM con técnicas de limpieza en datos empleada en Tang et al. (2019). Las columnas representan el número de pasos temporales a los que se proyecta a futuro la predicción. Los datos empleados en los modelos han sido medidos con un intervalo de diez minutos. En negrita está marcada la técnica con los mejores resultados. Datos extraídos de la misma fuente.

Method	1	3	6	10
SVM	20.0182	25.6457	32.208	42.5377
<b>SVM + EEMD</b>	<b>10.6307</b>	<b>20.1839</b>	<b>29.9837</b>	<b>39.1415</b>
SVM + EMD	12.7362	23.3944	30.0881	40.1533
SVM + BW	12.4016	22.7495	29.3605	40.1960
SVM + MA	23.2528	27.5552	34.6196	44.8840
SVM + WL (coif)	16.5823	24.5212	30.4244	40.5440
SVM + WL (db)	15.2325	23.7432	29.7694	39.4520
SVM + WL (haar)	16.7455	25.3329	30.7687	40.8019
SVM + WL (sym)	15.8924	24.1056	29.8211	40.3005

Z. Guo et al. (2019) proponen un modelo modular en el que por medio de un extractor de características, y sirviéndose del método k-means se distinguen diferentes tipos de nodos del tráfico. Posteriormente, la predicción es hecha por un modelo híbrido basado en regresión de kernel Ridge y procesos gaussianos (ver Figura 12).

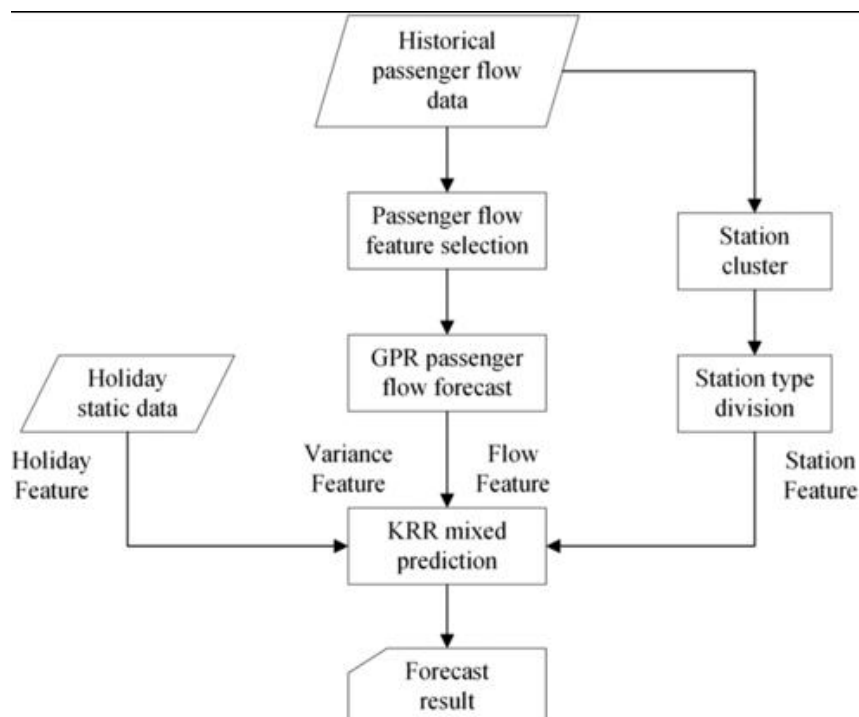


Figura 12: Diagrama de flujo del modelo híbrido presente en Z. Guo et al. (2019), en el que se combinan regresiones de procesos gaussianos con regresiones de kernel Ridge. Figura reproducida de la misma fuente.

Como reflejan una de las conclusiones de este último trabajo, los métodos de aprendizaje

automático clásico necesitan de una ingeniería de características manual para alcanzar unos resultados aceptables, lo que supone un gran esfuerzo e intervención para tratar con volúmenes grandes de datos (Tian et al. 2021).

### 2.2.2. Modelos basados en redes neuronales

Gracias al cada vez mayor número de datos, durante la última década se han podido construir modelos que describieran la complejidad del histórico del tráfico gracias al aprendizaje profundo (L. Cai et al. 2020). Típicamente, el histórico del tráfico es una serie temporal compuesta de las lecturas de ciertas variables relativas al tráfico. Debido a su naturaleza altamente dependiente de diversos factores, esta secuencia puede fluctuar de manera abrupta (Tian et al. 2021).

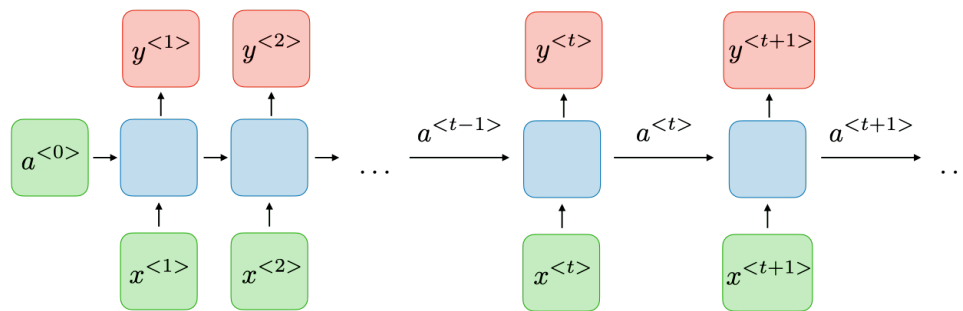


Figura 13: Esquema de la arquitectura de una red neuronal recurrente. Reproducido de Amidi et al. (2018).

Las arquitecturas basadas en Redes Neuronales Recurrentes (RNR, o RNN por sus siglas en inglés), han sido base de la mayor parte de trabajos, ya que estas redes son capaces de procesar una colección de datos de manera secuencial (Yao et al. 2019). Esto hace que puedan retener en memoria una serie de parámetros para describir las dependencias temporales y así predecir elementos futuros de la serie. En la Figura 13 se puede observar cómo, para un tiempo  $t$ , el estado anterior de la red  $a^{t-1}$  es propagado junto con la entrada  $x^t$  hacia arriba para obtener el resultado  $y^t$ . A su vez, el estado resultante de ese instante temporal  $a^t$  se propagará hacia adelante en el tiempo (hacia la derecha en el diagrama) para ser propagado arriba en la red con el input  $x^{t+1}$  en siguiente paso temporal para obtener  $y^{t+1}$ . De esta manera, la red es capaz de computar una predicción para cada instante temporal de forma recursiva sirviéndose de las predicciones pasadas. En el caso de series temporales, las entradas suelen comprender varios instantes temporales para ampliar el contexto de la predicción y así mejorar la precisión del pronóstico (ver Figura 14).

Por ejemplo, Zhao et al. (2017) emplean un tipo de arquitectura de RNR muy popular llamada

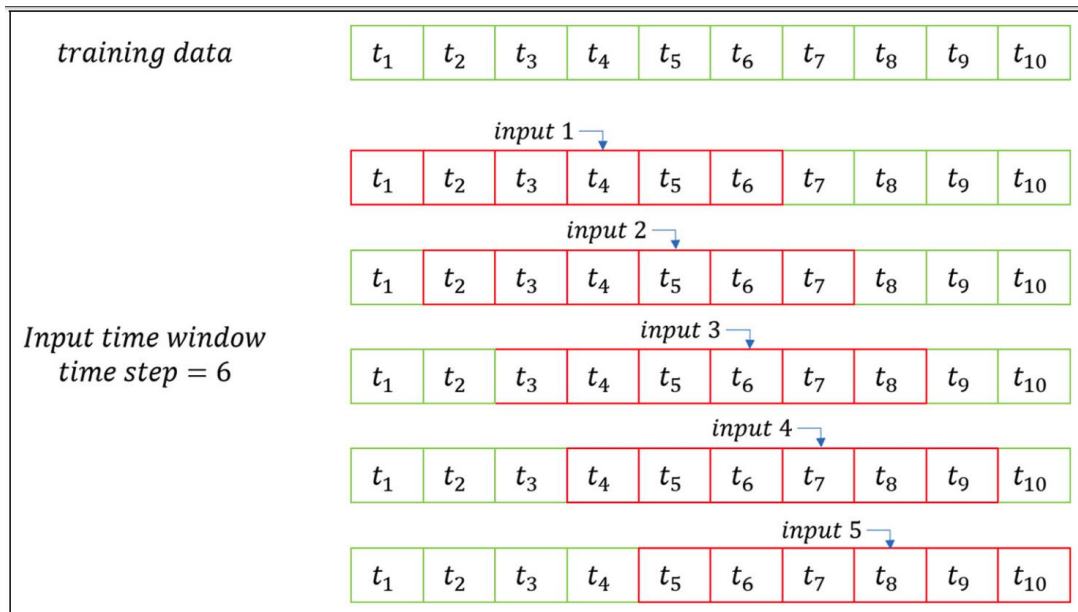


Figura 14: Esquema de la arquitectura de una red neuronal recurrente. Reproducido de S. Sun et al. (2019).

Long-Short Term Memory (LSTM) para capturar la dinámica no-lineal de una serie temporal. El modelo LSTM usado en este trabajo está formado por una red bi-dimensional para incluir la dependencia espacial entre distintos puntos del tráfico (ver Figura 15).

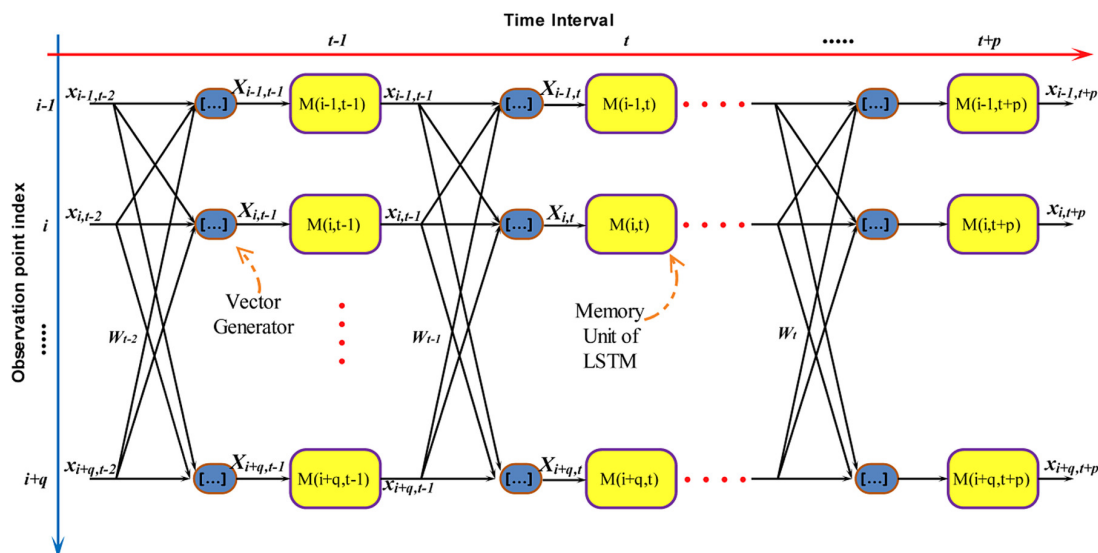


Figura 15: Diagrama del modelo LSTM bi-dimensional. En la Figura, el eje de abscisas representa la dimensión temporal, mientras que el eje de ordenadas representa la dimensión espacial entre distintos puntos de medida. Reproducido de Zhao et al. (2017).

En otro ejemplo de RNR aplicadas al tráfico, Zhang et al. (2017) emplean otra arquitectura llamada Gated-Recurrent Unit (GRU). En este trabajo, los autores combinan datos del histórico del tráfico con datos climáticos para entrenar una red basada en GRU. La inclusión de la influencia del clima en el cálculo hace que la precisión de la red sea superior al caso de no incluir

información sobre el clima.

Sin embargo, los modelos de RNR toman cada serie temporal de forma diferente por lo que no son capaces de correlacionar espacialmente los puntos de medida donde fueron capturadas esas series temporales. Es por ello que posteriormente muchos de los siguientes trabajos se han centrado en buscar una descripción explícitamente espacial y temporal de los datos recogidos del tráfico.

Paralelamente, en los trabajos de Ma et al. (2017) y S. Sun et al. (2019), los autores disponen los datos espaciales y temporales en una malla de dos dimensiones, que forma que la matriz resultante pueda ser interpretada como una imagen. Trabajar el histórico del tráfico en diferentes puntos como imágenes les permite emplear redes convolucionales (por sus siglas en inglés, CNN) para realizar predicciones. En los resultados de ambos experimentos, tal como se puede consultar en las Tablas 2 y 3, el resultado del experimento es que las CNN son capaces de obtener menores errores cuadráticos medios, incluso en predicciones a largo plazo. Este resultado es debido a que estas redes son las únicas del estudio que tienen en cuenta la correlación espacial. Sin embargo, la precisión de los modelos basados en RNR son mayores en comparación a predicciones a corto plazo. El error cuadrático medio de las RNN se va incrementando a medida que las predicciones se hacen a tiempos más largos.

Tabla 2: Comparativa de resultados del error cuadrático medio para predicción con redes convolucionales realizada en el estudio de Ma et al. (2017). Las columnas representan el número de pasos temporales a los que se proyecta a futuro la predicción. En negrita está marcada la técnica con los mejores resultados. Datos extraídos de la misma fuente.

Modelo	Tarea 1	Tarea 2	Tarea 3	Tarea 4
<b>CNN</b>	<b>22.825</b>	<b>24.345</b>	<b>30.593</b>	<b>31.424</b>
OLS	27.047	31.273	41.334	48.107
KNN	51.700	55.708	60.256	64.132
RF	35.092	35.431	40.476	40.638
ANN	67.764	52.339	58.797	57.225
SAE	60.751	69.082	65.292	68.326
RNN	33.408	36.833	40.551	39.038
LSTM NN	37.759	33.218	42.909	42.865

Otro de los trabajos que exploran la correlación entre las dinámicas espacial y temporal es el de Yao et al. (2018), en el que aplican aprendizaje profundo a la predicción de la demanda de taxis, un problema similar a la predicción de la densidad de tráfico. Se desarrolla un modelo híbrido, con la parte espacial modelada por redes convolucionales y la parte temporal con LSTM, tal como se puede apreciar en la Figura 16. Además, la inter-dependencia entre estas



Tabla 3: Comparativa de resultados del error cuadrático medio para predicción con modelos de aprendizaje profundo, llevado a cabo en el estudio de S. Sun et al. (2019). El número entre paréntesis presente en las columnas representa el número de pasos temporales a los que se proyecta la predicción. En negrita está marcada la técnica con los mejores resultados. Datos extraídos de la misma fuente.

Modelo	Tarea 1 (6)	Tarea 2 (8)	Tarea 3 (12)	Tarea 4 (18)
<b>CNN</b>	<b>38.50</b>	<b>36.97</b>	<b>27.11</b>	<b>15.69</b>
RNN	44.58	43.73	26.75	27.00
LSTM	47.88	47.95	35.42	23.81
GRU	48.39	44.77	37.21	30.49

dos partes es incluida a través de una arquitectura llamada Semantic Network Embedding, que se encarga de correlacionar áreas con patrones temporales similares.

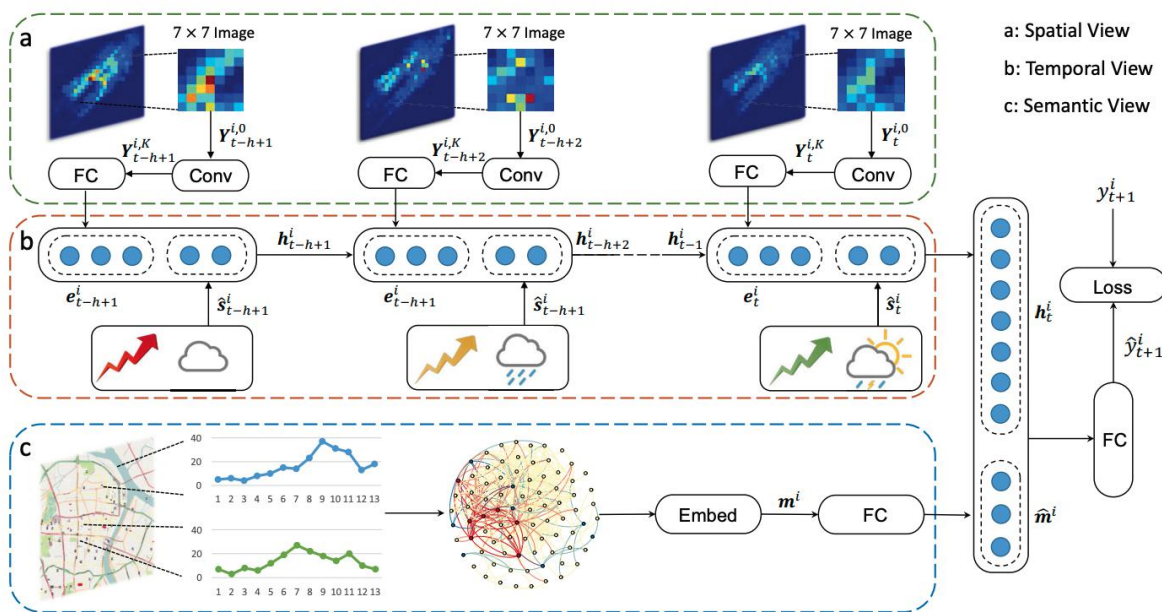


Figura 16: Diagrama de la arquitectura del modelo llamado DVMST, realizado en Yao et al. (2018). Figura reproducida de la misma fuente.

Una de las aproximaciones que asumía este último trabajo es la naturaleza estática de la parte espacial. Por ello, en un estudio posterior (Yao et al. 2019), los autores desarrollaron un modelo llamado Spatial-Temporal Dynamic Network (STDN), en la que, como se puede observar en la Figura 17, las contribuciones espacial y temporal son modeladas por CNN y RNR, respectivamente. Además, un mecanismo de flujo modulado (flow-gated mechanism) busca similitudes en parte espacial para describir su dinámica, mientras que un mecanismo de atención mejora las predicciones de la parte temporal.

No obstante, se ha de señalar que la principal limitación de estos modelos para capturar la dinámica espacio-temporal es que las redes convolucionales implícitamente asumen una geometría euclídea y ortogonal. Es decir, de la misma manera las CNN procesan imágenes como



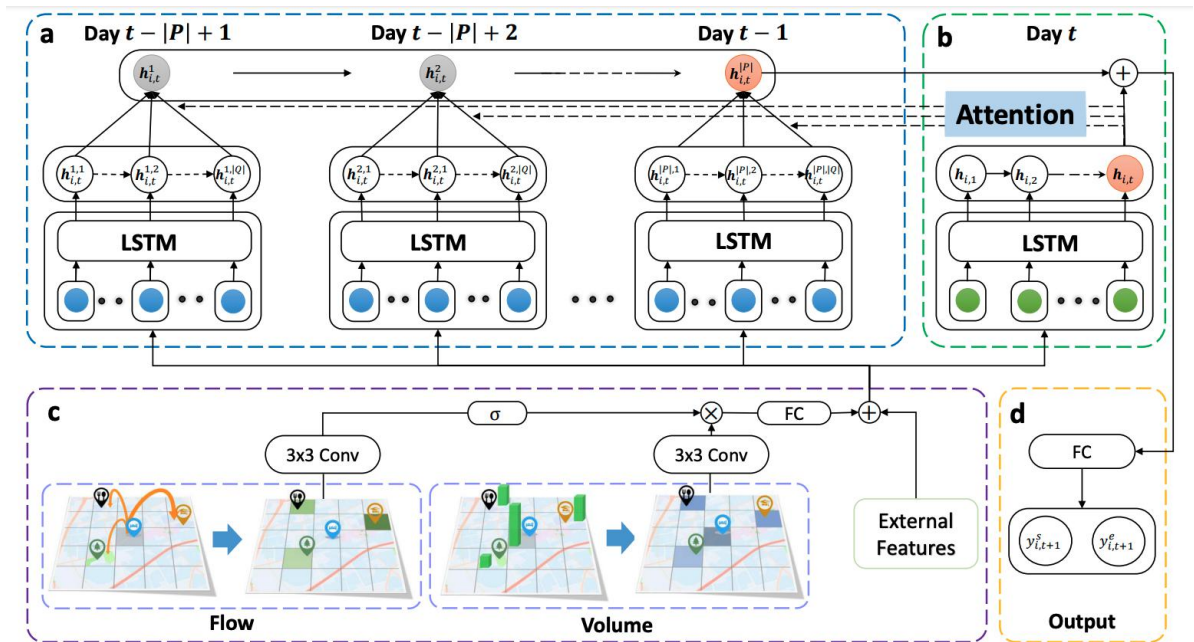


Figura 17: Diagrama de la arquitectura del modelo STDN, desarrollado en Yao et al. (2019). Figura reproducida de la misma fuente.

cuadrículas de píxeles ortogonales y contiguos, estas redes también asumen el mismo tipo de topología entre diferentes puntos de medida del tráfico. Para poder describir de forma más precisa la complejidad de la dinámica espacial recientemente se han empezado a utilizar redes neuronales basadas en grafos.

### 2.2.3. Modelos basados en redes neuronales de grafos

Durante los últimos años se ha popularizado el uso de redes neuronales basadas en grafos. Estas redes explotan el concepto matemático de grafo, un tipo de objeto matemático que describe una serie de objetos, llamados nodos, a través de sus relaciones, llamadas vértices. El concepto de grafo es particularmente poderoso aplicado a grandes conjuntos de datos que se interrelacionan de manera compleja, como es el caso de individuos dentro de redes sociales, átomos dentro de moléculas (Carleo et al. 2017; Pfau et al. 2019; Stokes et al. 2020), o incluso partículas dentro de sistemas físicos (Pfaff et al. 2021; Sanchez-Gonzalez et al. 2020). Entonces, las redes neuronales basadas en grafos (RNG) no son más que redes neuronales aplicadas al dominio de los grafos (Zhou et al. 2021). El desarrollo reciente de las RNG ha sido motivado por la expansión de las CNN, surgiendo como una generalización del concepto de redes convolucionales. Como se ha mencionado en la Subsección anterior, las CNN solo operan con datos estructurados de forma regular, como se muestra a la izquierda de la Figura 18. En cambio, las RNG amplían el dominio de aplicación de topologías Euclídeas a No-Euclídeas. Es por esto que las RNG se muestran especialmente útiles para describir relaciones espaciales

entre nodos como los de una red de carreteras, en las que los distintos nodos del tráfico no se disponen de manera adyacente y regular.

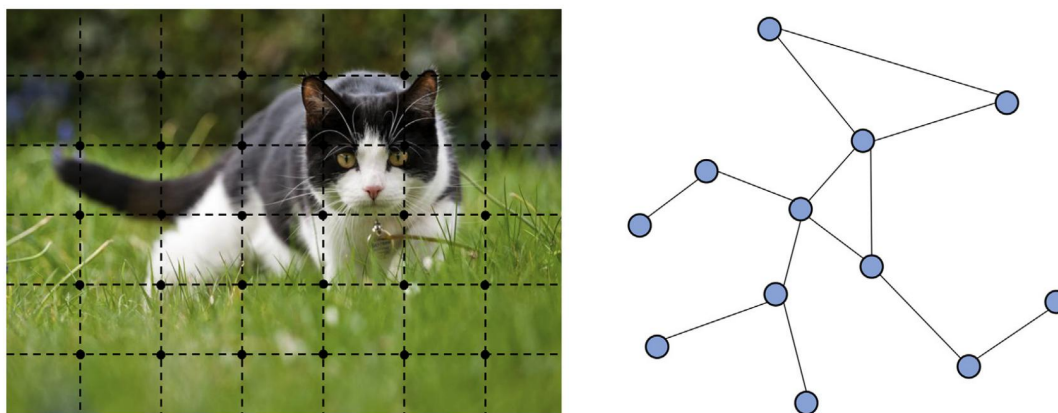


Figura 18: Comparación de una estructura regular de datos y un grafo. En el lado izquierdo se muestra una foto, ejemplo de estructura de datos (píxeles) regular y bi-dimensional. A la izquierda se muestra un ejemplo de grafo, cuyos nodos no siguen un ordenamiento regular. Figura reproducida de Zhou et al. (2020).

Una problemática común que abordan las redes basadas en grafos en la de la correlación espacio-temporal de los distintos nodos de una red, como ocurre en la predicción del tráfico. Tal como se puede observar en la Figura 19, el histórico del tráfico se puede entender como una secuencia de grafos ordenados en el eje temporal.

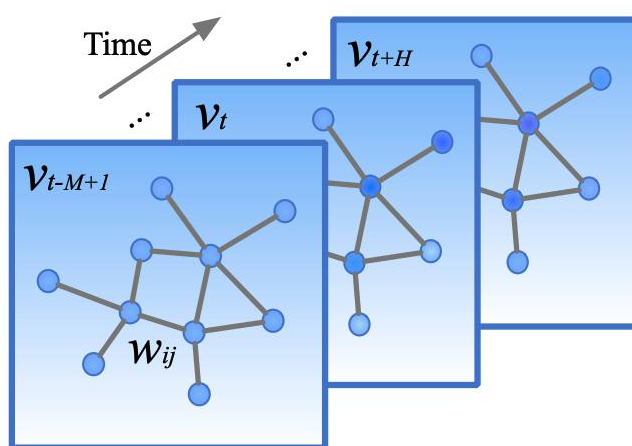


Figura 19: Representación de una secuencia temporal de grafos, en la que a cada instante de tiempo  $v_t$  se le asocia una matriz de adyacencia  $W$ . Figura reproducida de Yu et al. (2018).

La convolución es una operación fácilmente generalizable desde las habituales redes convolucionales hasta las redes basadas en grafos. Para esta operación existen dos aproximaciones: la espacial y la espectral. Esta última aproximación es la que usan la mayoría de trabajos sobre predicción del tráfico, y se caracteriza por definir la convolución en el dominio espectral. Dos de

los estudios más importantes en esta área son las ChebNets (Defferrard et al. 2016), que usan una expansión truncada de polinomios de Chebyshev para escribir los filtros de la convolución, y Kipf et al. (2017), que usan aproximaciones de primer orden para los filtros. Es este último estudio el que da nombre a las *Graph Convolutional Networks* o GCN de forma abreviada.

Precisamente, son estos dos últimos trabajos en los que se basa la implementación de Yu et al. (2018), en la que se emplean ChebNets y GCN como extractores de características de la parte espacial. A diferencia de otros trabajos, la parte temporal es procesada con CNN, para después ser fusionada con la parte espacial en bloques residuales, en un conjunto llamado *Spatio-Temporal Graph Convolutional Networks* (STGCN).

Uno de las recientes investigaciones más influyentes en predicción del tráfico es el de Li et al. (2018), ya que ha servido de punto de referencia para siguientes estudios. Los autores desarrollan un tipo de convolución sobre grafos llamada convolución por difusión. Esta técnica, que guarda analogías con las ChebNets (Defferrard et al. 2016), es la base para describir la parte espacial del modelo, mientras que la parte temporal es tratada con una RNR basadas en GRU. El modelo, llamado *Diffusion Convolutional Recurrent Neural Network* (DCRNN) se puede ver en la Figura 20.

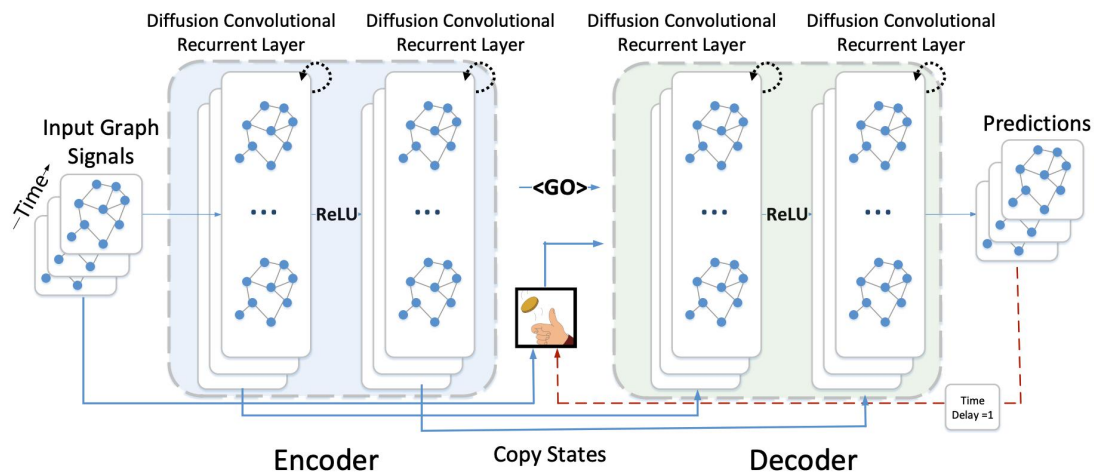


Figura 20: Diagrama de la arquitectura del modelo DCRNN, implementado del trabajo Li et al. (2018). Figura reproducida de la misma fuente.

Otro aspecto a remarcar en este trabajo es la creación de dos de los datasets públicos más utilizados en pronóstico del tráfico. El primer dataset es METR-LA, compuesto de medidas recogidas a partir de un conjunto de 207 sensores colocados en autopistas del condado de Los Ángeles a lo largo de cuatro meses, desde marzo a junio de 2012. El otro dataset se llama

Tabla 4: Comparativa del error cuadrático medio donde se comparan los modelos DCRNN y STGCN con otros modelos basados en aprendizaje profundo. Las columnas representan el tiempo al que se proyecta la predicción, 15, 30 y 45 minutos. En negrita está señalado el método con mejores resultados. Reproducido de Yu et al. (2018).

Model	15	30	45
FNN	2.74	4.02	5.04
LSTM	3.57	3.94	4.16
DCRNN	2.37	3.31	4.01
<b>STGCN (ChebNet)</b>	<b>2.25</b>	<b>3.03</b>	<b>3.57</b>
STGCN (1 <sup>st</sup> )	2.26	3.09	3.79

PEMS-BAY, con medidas de 325 sensores localizados en el área de la bahía de San Francisco durante 6 meses, de enero a mayo de 2017.

En la Tabla 4 se muestra una comparativa de resultados para las dos variantes de STGCN, DCRNN, LSTM y redes neuronales convencionales, FNN (cuyas siglas corresponden a *Feed-forward Neural Networks*). Se puede observar cómo la variante basada en ChebNets de STGCN obtiene un error cuadrático medio ligeramente inferior a la variante con aproximación de primer orden, y con muy poca diferencia con el modelo DCRNN. Para las predicciones a 15 minutos, las FNN tienen un rendimiento comparable a las DCRNN e incluso mayor que las LSTM. Sin embargo, a medida que el tiempo de predicción va aumentando, el error crece más que en el caso de la LSTM. En este sentido, las redes basadas en grafos son las que menores errores presentan en predicciones a largo plazo, siendo la reducción del error en predicciones a 45 minutos entre un 14 % para las LSTM y hasta un 29 % para las FNN.

Más recientemente, algunos trabajos han introducido mecanismos de atención en redes basadas en grafos (Veličković et al. 2018). Tanto en Do et al. (2019) como en S. Guo et al. (2019), la parte espacial de modelo cuenta con un mecanismo de atención, una matriz cuyos elementos son coeficientes que representan la correlación entre los distintos nodos, siendo de mayor magnitud cuanto más relación guarden dos nodos entre sí. Esta matriz de atención opera junto con la matriz de adyacencias del grafo. Para la parte temporal, el modelo también cuenta con un mecanismo de atención, para posteriormente fusionar las dos partes mediante convoluciones. Como conclusiones de estos estudios, además de disminuir el error de las predicciones a largo plazo, también ayuda a mejorar la interpretabilidad de los resultados del modelo.

Sin embargo, estas técnicas sufren de una limitación computacional en el grado de vecindad de los nodos involucrados en la convolución. Además, y de forma más importante, el grafo sobre el que se opera es estático durante todo el cálculo, lo que restringe el descubrimiento de

nuevas topologías sobre los nodos existentes. Formalmente podemos decir que estos modelos necesitan previamente del conocimiento de la matriz de adyacencias asociada al grafo, la cual, al mantenerse fija durante todo el proceso, es incapaz de reflejar la dinámica espacial entre los nodos. Entre los trabajos que intentan superar esta limitación está el de Wu et al. (2019), cuyo modelo Graph WaveNet (basado en el modelo WaveNet para la generación de audio (van den Oord et al. 2016)) implementa un grafo auto-adaptativo para la parte espacial que es capaz de aprender la matriz de adyacencias durante el entrenamiento, otorgando la capacidad de capturar correlaciones subyacente entre los nodos. Posteriormente, otros investigadores han trabajado desde este modelo para introducir modificaciones que mejoran su rendimiento (Shleifer et al. 2019), y también muy recientemente en Tian et al. (2021) para introducir módulos de auto-atención en su parte espacial, como se puede observar en el diagrama de la Figura 21.

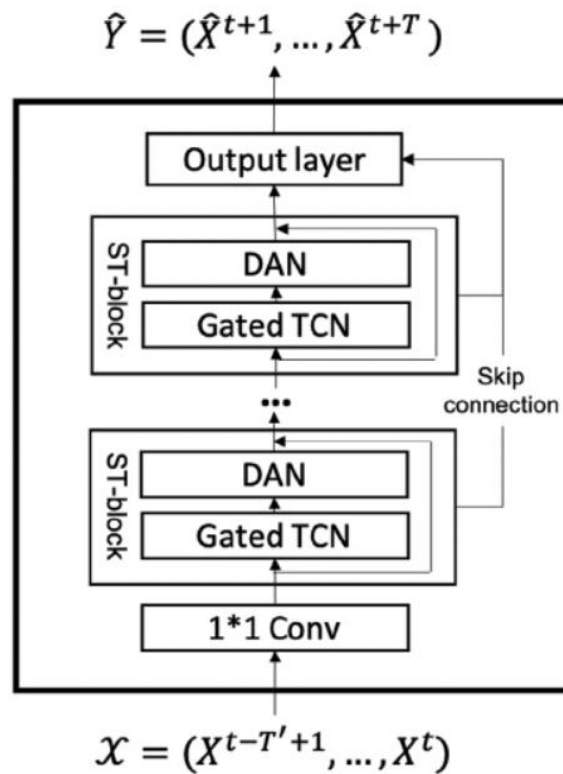


Figura 21: Diagrama de la arquitectura del modelo STAWNet, implementado del trabajo Tian et al. (2021). La arquitectura consta de una serie de módulos, llamados bloques espacio-temporales (ST), que constan de una convolución temporal seguida de un módulo de atención. Figura reproducida de la misma fuente.

### 2.3. Conclusiones

A lo largo de este capítulo se ha hecho un breve repaso por el reciente estado del arte de la predicción del estado del tráfico por medio de técnicas de aprendizaje automático. En la Ta-

Tabla 5: Resumen de los modelos repasados en el estado del arte. En las diferentes columnas figura el algoritmo utilizado, el nombre del modelo si lo tuviera, el año de la publicación de la implementación y la referencia al artículo.

Algoritmo	Modelo	Año	Referencia
KNN	-	2016	P. Cai et al. (2016)
SVM	-	2019	Tang et al. (2019)
KRR	-	2019	S. Guo et al. (2019)
LSTM	-	2017	Zhao et al. (2017)
GRU	-	2017	Zhang et al. (2017)
CNN	-	2017	Ma et al. (2017)
CNN	-	2019	S. Sun et al. (2019)
CNN + LSTM + SNE	DVMST	2018	Yao et al. (2018)
Gated CNN + LSTM	STDN	2019	Yao et al. (2019)
GCN	DCRNN	2017	Li et al. (2018)
GCN	STGCN	2017	Yu et al. (2018)
CGN + FNN	ASTGCN	2019	S. Guo et al. (2019)
CGN + GRU	STANN	2019	Do et al. (2019)
GCN + attention	STAWnet	2021	Tian et al. (2021)

En la [Tabla 5](#) se puede consultar el listado de los distintos modelos que han sido revisados. En primer lugar, en la Subsección [2.2.1](#) se han repasado tres trabajos que empleaban técnicas de aprendizaje automático clásico, siendo respectivamente k-NN, SVM y Regresión de Kernel de Rigde (KRR). La principal limitación de estos modelos es que requieren de una elección de características adecuada para su correcto rendimiento. Esto supone una gran dificultad en su aplicación a cantidades masivas de datos, como las provenientes de fuentes más modernas de colecta de datos. En segundo lugar, se han discutido trabajos en los que se empleaban técnicas de aprendizaje profundo en la Subsección [2.2.2](#). Entre las técnicas vistas en esta parte están las puramente basadas en redes recurrentes, como LSTM y GRU, las que únicamente están implementadas usando CNN, además una serie de modelos híbridos formados de CNN y RNN. La desventaja más importante que encuentran los algoritmos descritos en esta Subsección es la limitación que sufren al captar la correlación espacio-temporal de los diferentes puntos del tráfico, propios de un sistema de carreteras realista. Por último, en la Subsección [2.2.3](#) se han repasado los algoritmos de redes neuronales basadas en grafos. Esta familia de algoritmos se pueden entender como una generalización natural del concepto de redes convolucionales, ya que permiten trabajar con estructuras de datos con topologías no-Euclidianas. Son precisamente los grafos lo que mejor describe una red de carreteras, permitiendo captar su naturaleza altamente no-lineal.

Sin embargo, todos los modelos aquí expuestos se basan en aprendizaje a través de datos, por lo que los datasets empleados son de vital importancia. Es necesario para la generalización

de los modelos que estos aprendan a partir de conjuntos de datos que contengan un rango diverso de ejemplos, reproduciendo la naturaleza compleja del tráfico. Por ello, es necesario evaluar los modelos en datasets que reflejen distintos escenarios para determinar su extensión y aplicabilidad en multitud de circunstancias.





### **3. Objetivos y metodología**

En este Capítulo se enunciarán los objetivos del presente trabajo, tanto el objetivo principal que se persigue como los objetivos específicos que se derivan del principal. Seguidamente se expondrá la metodología seguida, en la que se discutirán los requisitos previos a la ejecución de la investigación, las herramientas elegidas y una descripción de las etapas en las que se ha desglosado el estudio. En la Sección 3.1 se expondrá el objetivo general, mientras que en la Sección 3.2 se presentarán los objetivos específicos. Por último, en la Sección 3.3 se describirá brevemente la metodología seguida.

#### **3.1. Objetivo general**

El presente trabajo tiene como objetivo general evaluar la precisión y eficiencia de una pareja de modelos de predicción del tráfico basados en redes neuronales de grafos para el caso de la ciudad de Madrid, mediante la creación de un dataset específico obtenido a partir de datos abiertos publicados en fuentes institucionales, utilizando herramientas de código abierto como el lenguaje de programación Python y distintas librerías implementadas en ese lenguaje para el procesamiento y análisis de los datos, así como también del entrenamiento y evaluación de los modelos resultantes.

#### **3.2. Objetivos específicos**

Con el fin de alcanzar el objetivo general previamente expuesto en la Sección 3.1, a continuación se presentarán los objetivos específicos a realizar:

- Analizar la problemática del tráfico y sus consecuencias negativas como contaminación, accidentes y congestión circulatoria.
- Estudiar el estado del arte en predicción del tráfico y, más específicamente, investigar sobre estudios recientes de modelos predictivos basados en aprendizaje automático.

- Diseñar una comparativa de modelos basados en redes neuronales de grafos para un estudio de aplicación al caso de la ciudad de Madrid.
- Crear un dataset con datos relativos a la ciudad de Madrid para el entrenamiento de modelos basados en aprendizaje automático. Este objetivo conlleva buscar fuentes de datos válidas de naturaleza abierta, capturar los conjuntos de datos de interés, seleccionar un subconjunto de los datos y, finalmente, limpiar y transformar los datos seleccionados a formatos válidos.
- Entrenar los modelos con el dataset creado. Estudiar la influencia de los hiperparámetros en el entrenamiento. Obtener métricas para la evaluación de los modelos.
- Realizar la comparativa. Validar las características de cada uno de los modelos y comparar las métricas obtenidas de cada uno de los modelos entre sí.
- Extraer conclusiones. Resumir los principales resultados. Resaltar y discutir los hechos más importantes encontrados en la comparativa.

### 3.3. Metodología

A continuación se hará un breve repaso de las etapas seguidas durante el desarrollo de este trabajo.

#### 3.3.1. Identificación de requisitos

En primer lugar es preciso identificar los requisitos necesarios para la realización de este trabajo. Estos requisitos se distinguen en dos categorías: la selección de los modelos a evaluar y los conjuntos de datos con los que se nutrirán los modelos.

##### Modelos

Para este estudio se han seleccionado tres modelos de aprendizaje profundo basados en grafos. Estos modelos son DCRNN (Li et al. [2018](#)), Graph-WaveNet (Shleifer et al. [2019](#); Wu et al. [2019](#)) y STAWnet (Tian et al. [2021](#)). Se ha juzgado interesante comparar estos tres modelos al representar tres aproximaciones que, una tras otra, incorporan mejoras cualitativas respecto al anterior. El primer modelo, DCRNN, se ha escogido por ser uno de los trabajos basados en grafos que sirven de referencia a multitud de estudios posteriores, y en el que presentan dos de los datasets públicos más utilizados para evaluar modelos predictivos del tráfico, METR-LA y PEMS-BAY. Seguidamente, Graph-WaveNet ha sido escogido por ser el modelo que presentar

la ventaja de prescindir de la necesidad del conocimiento previo de la matriz de adyacencia del grafo. Esto, como ya se mencionó en la Sección 2.2, aporta una mayor flexibilidad al calcular de forma adaptativa dicha matriz. Por último se incluye el modelo STAWnet, puesto que además de compartir la naturaleza adaptativa de Graph-WaveNet, el modelo que incorpora módulos de atención. Todas las arquitecturas anteriores cuentan con código abierto disponible en el repositorio GitHub (Github Inc., 2021) para la implementación de sus respectivos modelos.

### Fuentes de datos

En cuanto a los datos necesarios para implementar los modelos, se han empleado dos conjuntos de datos provenientes de una fuente institucional, el portal de datos abiertos del Ayuntamiento de Madrid (Ayuntamiento de Madrid, 2021). Estos conjuntos de datos son el histórico del tráfico en la ciudad de Madrid desde 2013 y la ubicación de los puntos de medida del tráfico.

### Tecnología

Las tecnologías empleadas en la realización de este trabajo se basan en el lenguaje de programación Python. Este es un lenguaje de programación creado por Guido Van der Rossum en 1991. Es un lenguaje interpretado de alto nivel que soporta diferentes paradigmas de programación, como son la programación orientada a objetos y la programación funcional. Uno de los aspectos principales de Python es su simplicidad y legibilidad. Este espíritu está resumido en el manifiesto *The Zen of Python* (Peters, 2004), un documento formado de 19 aforismos sobre la escritura y el estilo de la escritura de código.

Las cualidades antes mencionadas han hecho de Python uno de los lenguajes más populares del momento, y también unos de los de mayor crecimiento y expansión de los últimos años (ver Figura 22). Sin duda, sus módulos dedicados al cálculo científico a la ciencia de datos han contribuido notablemente a esa expansión: entre los más destacados se cuentan NumPy (Oliphant, 2006), SciPy (McKinney, 2010), pandas (McKinney, 2010, 2011), matplotlib (Hunter, 2007) o scikit-learn (Pedregosa et al. 2011).

A pesar del hecho de que los lenguajes interpretados son a priori más lentos en la ejecución de su código en comparación con los lenguajes de bajo nivel (como C), muchos de estos módulos actúan como interfaces a una gran cantidad de funciones matemáticas escritas en C y FORTRAN. Este es particularmente el caso de Numpy (Oliphant, 2006), que contiene cientos de rutinas de cálculo científico de alto rendimiento, cubriendo áreas desde el álgebra lineal hasta

el análisis numérico.

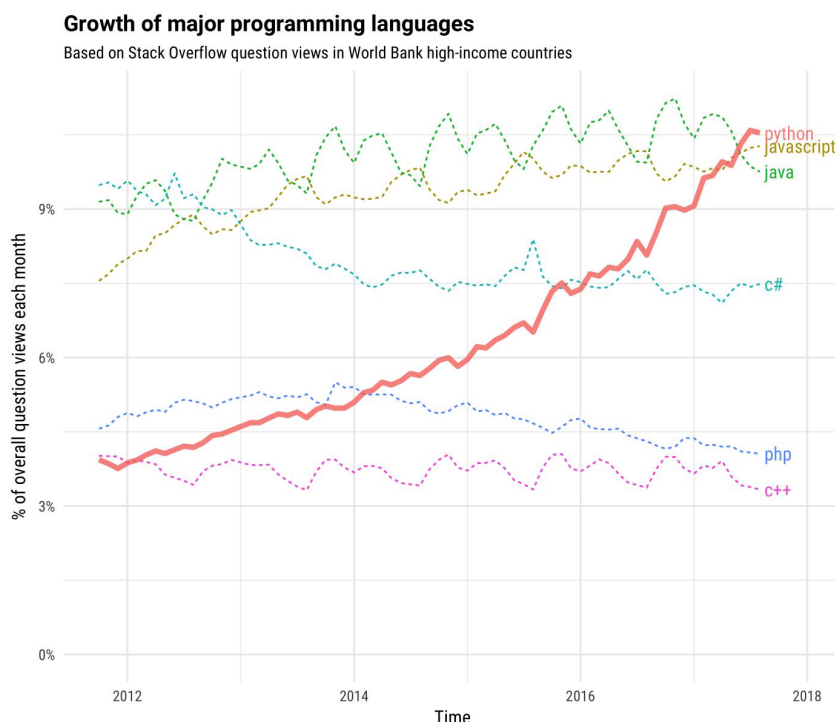


Figura 22: Crecimiento de los lenguajes de programación más usados en los países ricos en los últimos años. La línea gruesa continua representa Python. Como se aprecia en el gráfico, es el lenguaje que más rápido crecimiento ha experimentado en los años que abarca la figura, desde 2012 hasta 2018. Reproducido de StackOverflow (Robinson, [2017](#)).

Para la manipulación de los conjuntos de datos y la compilación del dataset usado en los modelos se utilizarán las herramientas NumPy y Pandas. Como se ha adelantado antes, NumPy (Harris et al. [2020](#)) es una librería de Python de código abierto enfocada al cálculo numérico escrita en Python, C y C++, creada por Travis Oliphant en 2005 (Oliphant, [2006](#)). Su principal característica es su implementación centrada en arrays multidimensionales, los cuales se almacenan de forma contigua en memoria para ser manipulados de una forma mucho más veloz y eficiente que las simples listas de Python. Su rapidez, facilidad de uso, versatilidad, así como su vasto catálogo de funciones y rutinas matemáticas hacen de NumPy una de las librerías más populares en el mundo del cálculo científico, sobre todo el ámbito de la ciencia de datos, tal como se puede apreciar en la Figura [23](#).

Pandas es una librería de Python pensada para el análisis de datos creada por Wes McKinney en 2008 mientras trabajaba en AQR Capital Management (Pandas development team, [2020](#)). Ofrece una serie de herramientas y utilidades para la manipulación de tablas de datos y series temporales. La librería se centra en la creación de objetos llamados `DataFrame`, que sirven como contenedores de tablas de datos y series temporales. También se proporcionan

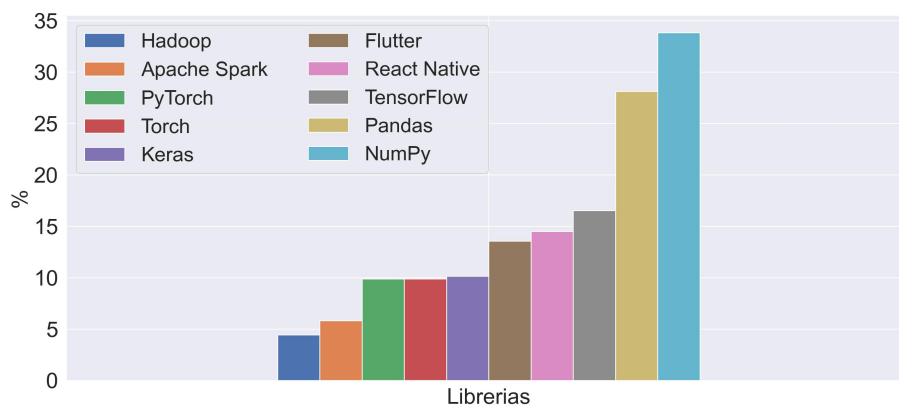


Figura 23: Librerías y frameworks más populares, según la encuesta anual realizada en mayo de 2021 por el portal StackOverflow a cerca de 80,000 desarrolladores de todo el mundo (stackoverflow, 2021).

herramientas y utilidades para su manipulación, propias del análisis y consulta de bases de datos (McKinney, 2010). Por tanto, tanto NumPy como Pandas resultan por tanto una elección adecuada para el uso en la manipulación de los conjuntos de datos con los que se tratará en este trabajo.

Como se ha hecho mención antes, Python es un lenguaje especialmente popular en el área de la ciencia de datos. De hecho, se ha convertido en el lenguaje de preferencia para la implementación de modelos de aprendizaje profundo, ya que cuenta con las librerías más ampliamente utilizadas, tanto en la academia como en la industria, como son TensorFlow y PyTorch (stackoverflow, 2021). TensorFlow (Google Inc., 2019; Martín Abadi et al. 2015) es un *framework* de aprendizaje profundo de código abierto desarrollado y mantenido por Google. Está principalmente escrito en C, con su API principal escrita en Python, junto con otras en C, Javascript, Golang y Swift. La principal característica de TensorFlow es la implementación de redes neuronales a través de grafos computacionales. El mismo nombre de TensorFlow hace referencia a dos elementos clave: los tensores, los cuales son arrays multidimensionales que describen los nodos del grafo y flujo (*flow* en inglés), que se refiere a la ejecución sub-secuencial de las operaciones que se suceden a lo largo del grafo. Estos grafos pueden ser manipulados de forma transparente al usuario en múltiples arquitecturas, desde procesadores (CPU), tarjetas gráficas (GPU) (usando la librería de programación en GPU NVIDIA CUDA) hasta las recientes Unidades de Procesado Tensorial (Tensor-Processor Units en inglés, TPU).

A su vez, PyTorch es una librería de código abierto para aprendizaje automático basada en Torch, una librería escrita en Lua y C dedicada al cálculo tensorial. PyTorch (Facebook Inc.,

2019) fue originalmente desarrollada por los laboratorios de investigación en Inteligencia Artificial de Facebook (FAIR). Aunque esta librería ha sido menos popular que la ya mencionada TensorFlow, en los últimos años está ganando cada vez más aceptación en el mundo del desarrollo y de la Inteligencia Artificial, como se puede observar en la Figura 24. De hecho, PyTorch es la librería escogida en importantes compañías como Facebook OpenAI (OpenAI, 2020), Tesla (Karpathy, 2019) o Uber (Goodman, 2017). De la misma manera que TensorFlow, PyTorch se basa en el cálculo con tensores. Sin embargo, su filosofía de uso no está tan centrada en la ejecución de grafos, como TensorFlow, y sí más cercana a una librería de cálculo numérico como NumPy, con una fuerte dedicación a explotar la computación en tarjetas gráficas (GPU). Una de las características de PyTorch es su mayor enfoque en la programación orientada a objetos a través de la creación clases heredadas de la clase `nn.Module`. Estos módulos, que pueden representar conjuntos de capas o redes neuronales enteras, cuentan con la capacidad del cálculo implícito de cada uno de los gradientes asociados a los tensores definidos. Esta capacidad es debida a que PyTorch contiene un módulo de diferenciación automática (Paszke et al. 2017), la cual permite de una forma eficiente el aprendizaje por diversos métodos basados en el descenso del gradiente de una forma transparente para el usuario. La elección de tanto PyTorch como TensorFlow nos permite hacer uso de las implementaciones originales de los modelos estudiados en este trabajo, ya que cuentan con código abierto publicado por sus autores con al menos una de las dos librerías.

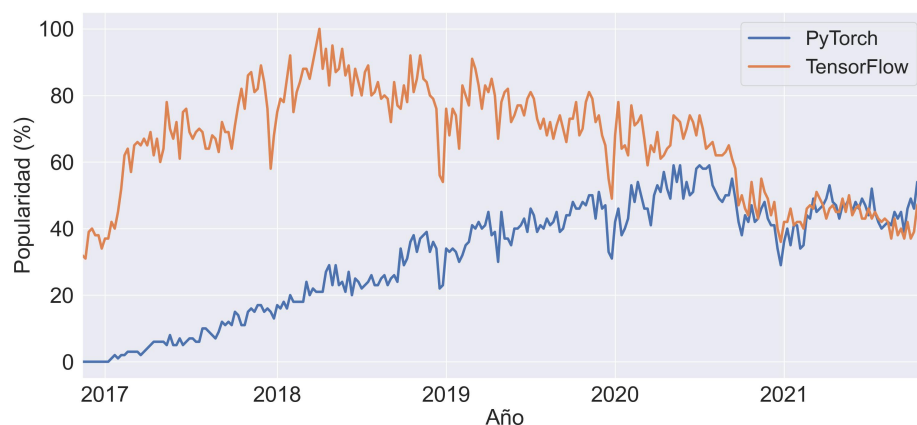


Figura 24: Porcentaje de popularidad de TensorFlow y PyTorch en los últimos cinco años. Datos extraídos de Google Inc. (2021).

En resumen, el lenguaje Python y las librerías mencionadas se ajustan adecuadamente al caso de uso que representa este estudio. Para ilustrar la elección entre otra de las opciones mayoritarias en ciencia y análisis de datos, en la Tabla 6 se muestra una comparativa entre los

Tabla 6: Comparativa entre los lenguajes de programación Python y R. (IBM, 2021)

	Python	R
Velocidad de ejecución	✓	✗
Focalizado en manipulación de datos	✓	✗
Focalizado en análisis estadístico	✗	✓
Disponibilidad de código	✓	✗
Flexibilidad de uso	✓	✗
Uso extendido en AI	✓	✗
Uso en producción	✓	✗

lenguajes Python y R.

### 3.3.2. Descripción del desarrollo

El desarrollo de este trabajo constará de las siguientes etapas:

1. Captura de datos y creación del dataset.
2. Entrenamiento y evaluación de los modelos. Estudio de hiperparámetros y obtención de métricas.
3. Evaluación de los modelos. Comparativa de modelos entre sí. Discusión de resultados.
4. Extracción de conclusiones.

En la Figura 25 se muestra un diagrama de flujo de la realización de las diferentes etapas. En la primera etapa, captura de datos y creación del dataset, se procederá a la adquisición de los datos necesarios mediante descarga. Para la creación del dataset, se acotará un subconjunto del total de los puntos de medida disponibles, además de determinar temporalmente el periodo que comprenderá el dataset. Posteriormente, los datos deberán ser procesados. Del conjunto relativo a los puntos de medida del tráfico se extraerán aquellos que pertenezcan al subconjunto acordado durante el periodo que comprende el dataset. Seguidamente, se filtrará el histórico para obtener solo aquellas medidas de parámetros correspondientes a los puntos de medida seleccionados anteriormente. A continuación, se creará la tabla de velocidades, cuyos ejes serán el tiempo y los puntos de medida, y se guardará el resultado. Finalmente, se realizará un análisis exploratorio del dataset, para conocer mejor la naturaleza y distribución de los datos de trabajo, así entender los posibles casos de éxito o fallo de los modelos.

En la segunda etapa se pasará a entrenar los modelos de aprendizaje basados en grafos. Cada uno de los modelos será entrenado por separado, realizando un pequeño estudio de la

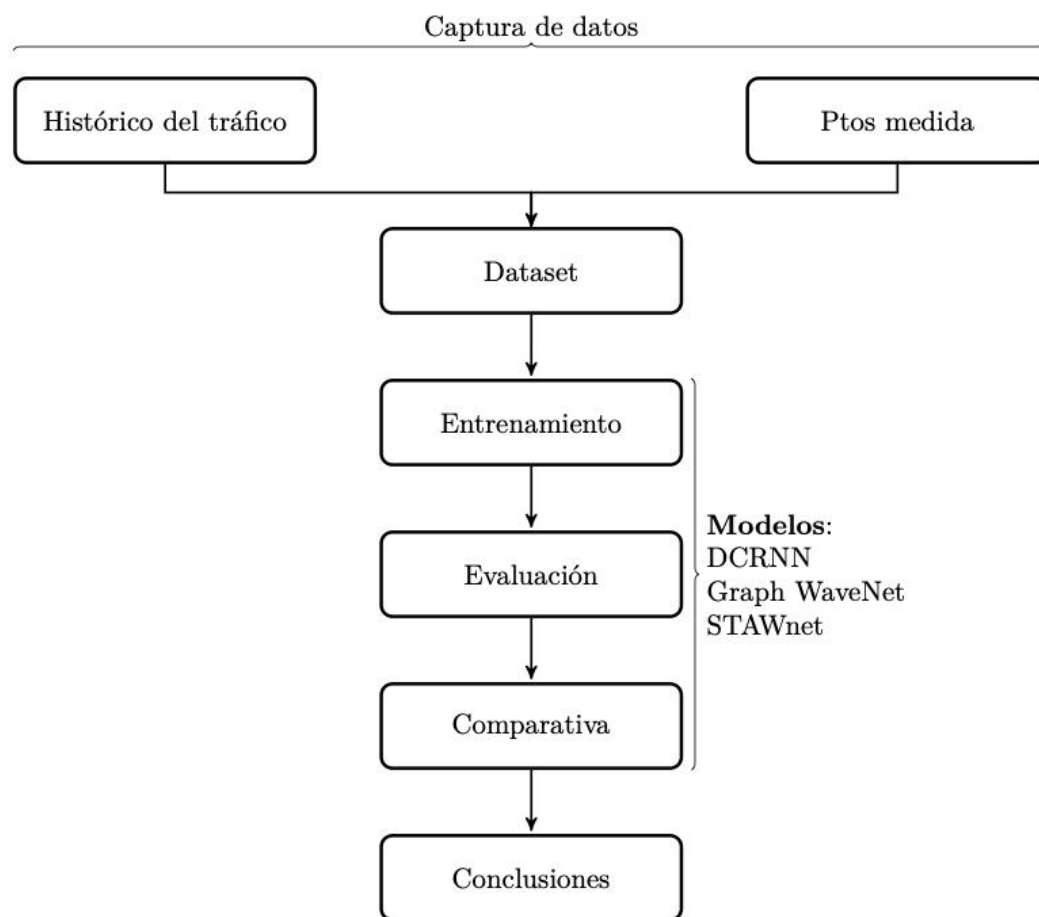


Figura 25: Diagrama de flujo de las tareas del trabajo.

influencia de algunos hiperparámetros en el aprendizaje de cada modelo. Se partirá de una estrategia de entrenamiento análoga al de los artículos originales de los modelos, para explorar alguna dependencia que se presente de forma particular para el dataset creado en este trabajo.

En la tercera etapa, en primer lugar se realizará una evaluación de cada uno de los modelos por separado. Basadas en las métricas de los trabajos originales estudiaremos los resultados de las evaluaciones, analizando las principales diferencias encontradas con respecto a la literatura existente. Posteriormente se hará una comparativa de los tres modelos, resaltando aquel con el que se hayan obtenido los mejores resultados.

Por último, en la cuarta etapa se extraerán las principales conclusiones del trabajo. Se dará una valoración de los resultados obtenidos en comparación a los resultados publicados en la literatura correspondiente a los tres modelos. Se señalarán las principales diferencias obtenidas, intentando relacionar estas con el caso particular de este dataset. También se revisará uno por uno el cumplimiento de los objetivos marcados al comienzo de este Capítulo, en la Sección 3.1



y Sección 3.2. Finalmente se apuntarán las claves para futuro trabajo que continúe y complete el presente estudio.

El código implementado y utilizado en este trabajo puede encontrarse en el siguiente repositorio: <https://github.com/itsAlexNoir/bovra>



## **4. Desarrollo específico de la contribución**

En el presente Capítulo se van a presentar los principales descubrimientos y resultados de este trabajo. En la Sección 4.1 se presentará el conjunto de datos, señalando sus fuentes en la Subsección 4.1.1, seguido en la Subsección 4.1.2 de un pequeño análisis exploratorio previo al pre-procesado y compilación de los datos en la Subsección 4.1.3. Más adelante en la Sección 4.2 se hará una descripción detallada de los modelos empleados. Se discutirá una breve definición del problema de la predicción del tráfico en la Subsección 4.2.1, para después detallar los modelos DCRNN y GWN en las Subsecciones 4.2.2 y 4.2.3 respectivamente. A continuación en la Sección 4.3 se presentará una evaluación de cada uno de los modelos, en las Subsecciones 4.3.1 y 4.3.2. Finalmente se cerrará el Capítulo con una comparativa de los modelos en la Subsección 4.3.3.

### **4.1. Implementación del dataset**

En esta Sección se van a exponer los principales requisitos para la realización de este trabajo. En primer lugar, en la Subsección 4.1.1 se mostrará la principal fuente de datos desde donde se ha extraído el histórico del tráfico que sirve de base de este trabajo. En la Subsección 4.1.2 se presentará un pequeño estudio estadístico del conjunto de datos relativo a la M30. Finalmente en la Subsección 4.1.3 se describirá el preprocesamiento realizado para la obtención del dataset de entrenamiento y validación.

#### **4.1.1. Fuente de datos**

Para el aprendizaje de los modelos utilizados en este trabajo se han utilizado dos conjuntos de datos presentes en el Portal de datos abiertos del Ayuntamiento de Madrid (Ayuntamiento de Madrid, 2021). Esta página web está dedicada a facilitar y promover hacia la ciudadanía el uso de datos provenientes del gobierno municipal del Ayuntamiento de Madrid.

En concreto, los conjuntos de datos utilizados son:

Tabla 7: Esquema del conjunto de datos del histórico del tráfico. En la primera columna se puede ver el nombre del atributo, en la segunda columna su tipo, y en la tercera columna su descripción. En todos los atributos un valor negativo significa ausencia de datos. La *vmed* solo está disponible para pmed tipo M30. Todos los valores están integrados para un periodo de 15 minutos. Extraído de Ayuntamiento de Madrid (2021).

Nombre	Tipo	Descripción
idelem	Entero	Identificación única del pmed
fecha	Fecha	Fecha y hora local en Madrid
identif	Texto	Identificador del pmed (Sist. Tráfico)
tipo_elem	Texto	Tipo de pmed: URB o M30
Intensidad	Entero	Intensidad del pmed (vehículos/hora).
Ocupación	Entero	Tiempo de ocupación del pmed ( %)
Carga	Entero	Carga de vehículos de la vía [0,100].
vmed	Entero	Velocidad media (km/h).
error	Texto	Calidad de la medición.
periodo_integración	Entero	Total muestras para el periodo de integración.

- Histórico de datos del tráfico desde 2013.
- Ubicación de los puntos de medida del tráfico.

En las Subsecciones siguientes se presentarán los conjuntos de datos, junto con su esquema:

### Histórico de datos del tráfico desde 2013

Serie histórica de las medidas recogidas de los puntos de medida (*pmed*) del tráfico desde julio de 2013 a la actualidad. La información está ordenada por mes y año, siendo posible su descarga en formato csv dentro de un archivo comprimido con formato *zip*. En la Tabla 7 se reproduce un esquema del conjunto de datos.

Según se especifica en la documentación asociada a este conjunto de datos, la información recogida en los puntos de medida se vuelca e integra en la base de datos central SICTRAM en periodos de 15 minutos. En los casos en los que el sensor no registre información, ésta no se integrará en la base de datos. Sin embargo, si el sensor proporciona información que no cumple los niveles de calidad marcados, ésta se recogerá, aunque marcada como posiblemente errónea. Los posibles valores de este parámetro son:

- *N*: Sin errores
- *E*: alguna de las muestras que conforman la medida no cumplen los parámetros de calidad.
- *S*: alguna de las muestras era inequívocamente errónea y no se ha integrado.

Tabla 8: Esquema del conjunto de datos sobre los puntos de medida. En las columnas están especificadas correspondientemente el nombre del atributo, el tipo de dato y una breve descripción. Para el campo nombre, las vías urbanas se nombran con la calle y sentido de la orientación. Para las vías rápidas se nombran con el punto kilométrico, calzada y si es o bien vía central, de servicio o un enlace. Extraído de Ayuntamiento de Madrid (2021).

Nombre	Tipo	Descripción
<i>cod_cent</i>	Texto	Código de centralización
<i>id</i>	Entero	Identificación única del pmed
<i>Nombre</i>	Texto	Denominación del pmed
<i>tipo_elem</i>	Texto	Tipo de pmed: URB o M30
<i>x</i>	Real	Coordenada UTM del eje x
<i>y</i>	Real	coordenada UTM del eje y

Las posibles fuentes de error en este dataset son la recogida de datos fuera de los rangos establecidos, o la falta de información de uno de los sensores que componen el punto de medida (ej. uno de los sensores de un punto de medida en una vía de cuatro carriles falla).

### Ubicación de los puntos de medida del tráfico

Información actualizada mensualmente sobre los puntos de medida desde dónde se recogen los datos del tráfico. La información puede descargarse por meses en formato *xlsx* y *csv*. Además en formato comprimido se suministran varios archivos para la visualización de los puntos de medida por medio de una herramienta de Sistemas de Información Geográfica (SIG).

Estos puntos de medida son unas espiras enterradas bajo el pavimento que funcionan por inducción electromagnética, de forma que cuando un cuerpo metálico cruza el área en el que está depositada la espira este cuerpo se detecta con precisión.

En la Tabla 8 se ofrece un resumen de los atributos del conjunto de datos.

#### 4.1.2. Análisis exploratorio

En este apartado se realizará un pequeño análisis exploratorio del conjunto de datos seleccionado para el dataset. Como se ha discutido previamente en la Subsección 4.1.3, el histórico de datos comprende un periodo de seis meses, de enero a junio de 2019. En este periodo el histórico acumula un total de 17,373 entradas para todos los puntos de medida seleccionados. En este trabajo solo se han seleccionado puntos de medida situados en las vías de la M30, los cuales pueden verse en la Figura 26. La razón de esta limitación en el número de puntos, y por consiguiente en el número de nodos que forman el grafo, es reducir el tamaño del dataset y así el coste computacional de entrenar un modelo.

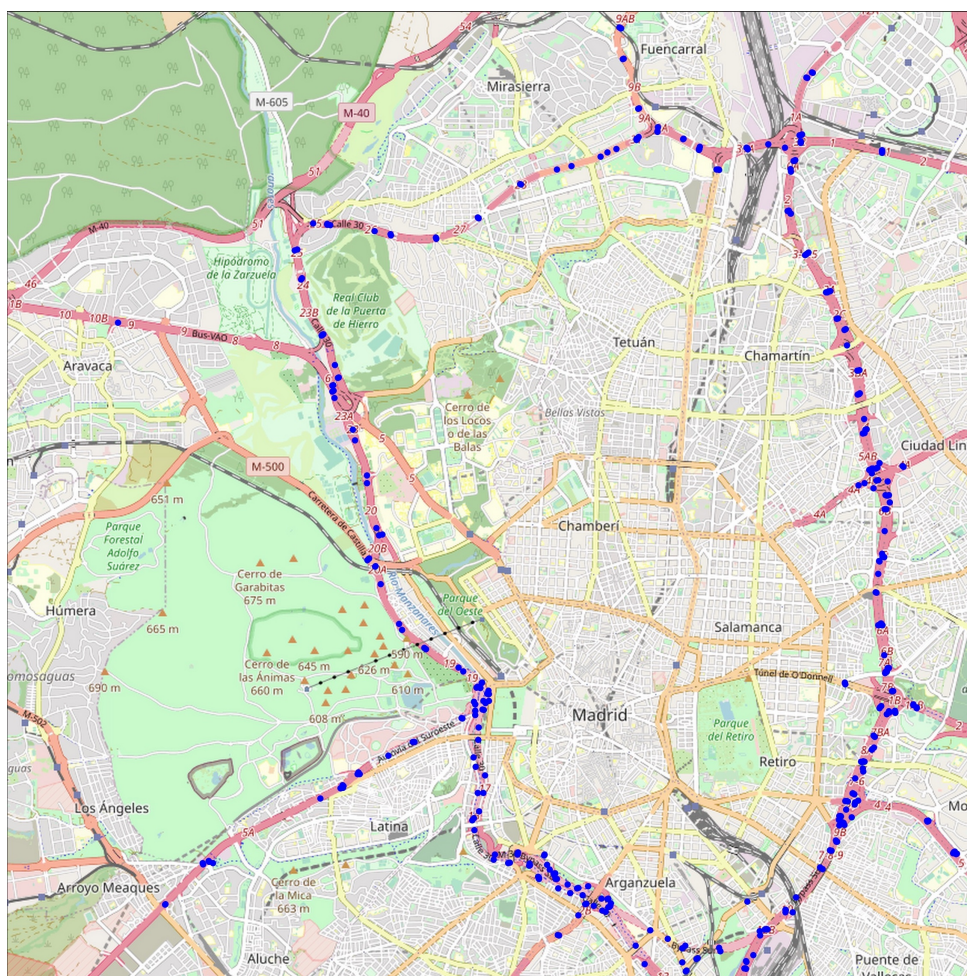


Figura 26: Mapa de la localización de los puntos de medida del dataset. En la Figura se puede apreciar que todos están localizados en vías de la M30.

El total de puntos de medida en el histórico original asciende a un total de 6,072, de los cuales solo 421 corresponden a puntos situados en la M30. Esto equivale aproximadamente a un 7 % del total de puntos de medida (Tabla 9). Este número de nodos, 421, es comparable al número de nodos en los datasets de referencia de estado del arte, donde el conjunto METR-LA cuenta con 207 nodos, mientras que el conjunto PEMS-BAY cuenta con 325 nodos. Un pequeño resumen de las características de estos datasets se puede consultar en la Tabla 10.

Tabla 9: Estadística del número de puntos de medida del dataset.

Tipo	Número	Porcentaje
URB	5649	93 %
M30	423	7 %
Total	6072	100 %

En la Tabla 11 se muestran los valores de los principales estimadores para los parámetros del dataset. Es interesante señalar que en el caso de la *intensidad* los valores de la media



Tabla 10: Principales estadísticas del dataset *M30* (nuestro), y de los datasets METR-LA y PEMS-BAY. En la tabla se puede ver el número de nodos y aristas que conforman los distintos grafos, y también el número de pasos temporales que contienen.

Dataset	Nodos	Aristas	Pasos temporales
METR-LA	207	1515	34272
PEMS-BAY	325	2369	52116
<i>M30</i>	417	67983	17373

Tabla 11: Estadística de los principales estimadores para el conjunto de datos seleccionado que forma el dataset utilizado en los experimentos. Cada columna contienen los resultados para cada una de las variables del conjunto de datos original.

	<i>vmed</i>	<i>intensidad</i>	<i>ocupación (%)</i>	<i>carga (%)</i>
Media	64.4	1309.6	5.9	27.1
Mediana	67.0	817.0	4.0	15.0
Desviación	20.8	1344.5	7.3	29.7
Min.	0.0	0.0	0.0	0.0
Max.	200.0	13428	100.0	100.0

y la mediana difieren un 37 %, lo que indica una clara presencia de valores extremos en el conjunto de datos que modifican al alza el valor de la media. También indicar que el valor máximo registrado para la velocidad media, *vmed*, es de 200 km/h, un valor muy por encima de la velocidad máxima permitida de 120 km/h en este tipo de vías.

En la Figura 27 podemos observar los histogramas correspondientes a las variables del dataset. En la Figura 27a se aprecia que el máximo de cuentas para la velocidad media está centrado en torno a los 65 km/h, valor que coincide con la velocidad media indicada en la Tabla 11. Para el resto de histogramas, en las Figuras 27b, 27c y 27d, el máximo se localiza en torno al valor cero, decayendo el número de cuentas para valores mayores. El hecho de que el valor más registrado para la *carga*, *ocupación* e *intensidad* sea cero apunta a que los sensores no registran flujo de tráfico la mayor parte del tiempo.

En las Figuras 28 y 29 se pueden observar las señales recogidas durante la semana del 14 al 21 de enero de 2019 en 6 puntos de medida aleatorios para las variables de *vmed* en la Figura 28a, *intensidad* en la Figura 28b, *ocupación* en la Figura 29a y *carga* en la Figura 29b. En todas las figuras podemos apreciar el carácter periódico de las señales, con una frecuencia aproximada de un día. Aunque existen variaciones entre las señales de los distintos sensores, en casi todos ellos se pueden apreciar dos máximos diarios que se corresponden con las horas puntas de entrada y salida del trabajo, una por la mañana y otra por la tarde. También está muy marcado el mínimo de la señal, localizado en torno a la medianoche. Igualmente es remarcable en la Figura 28a, la actividad correspondiente al fin de semana, como una desaparición de los dos picos matutino y vespertino. Estos picos en el fin semana sí son apreciables en *intensidad* en la Figura 28b, aunque son de menor magnitud comparados con los días laborables.

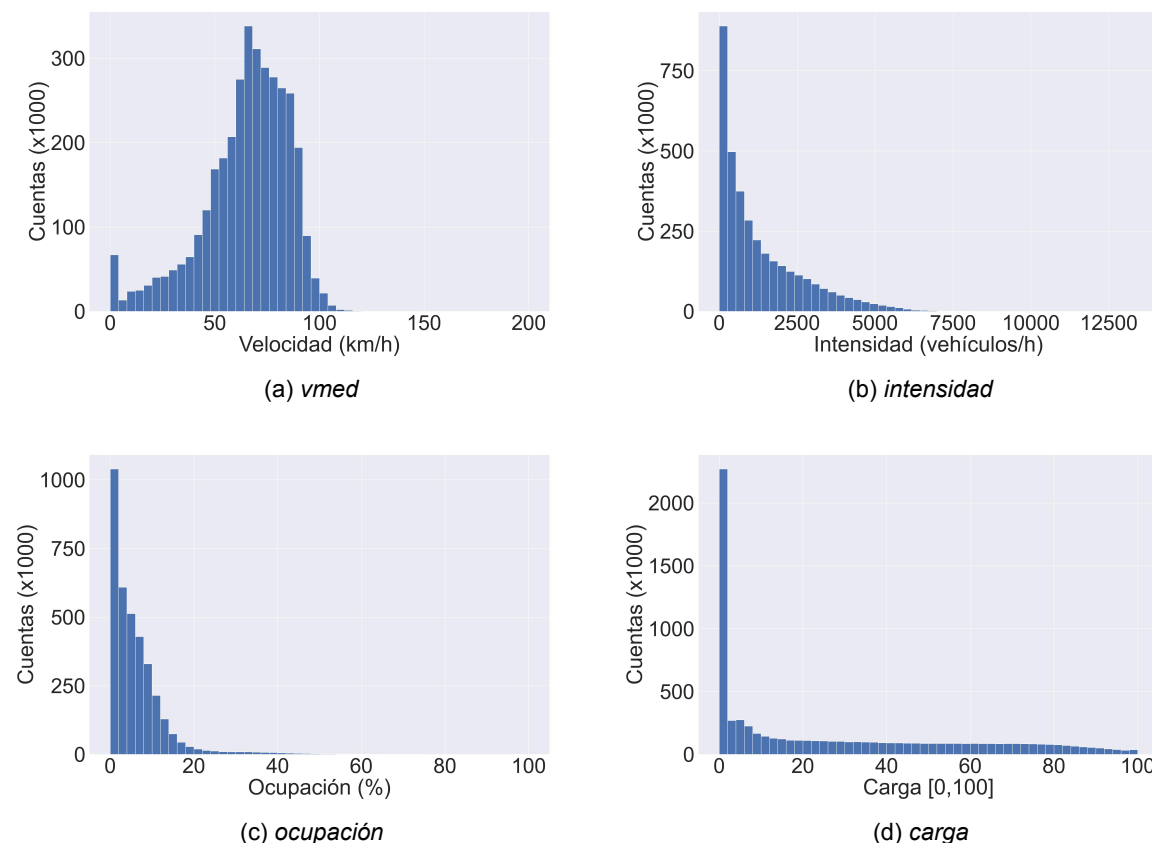


Figura 27: Histogramas de las variables del conjunto de datos *M30*, correspondientes a (a) *vmed*, (b) *intensidad*, (c) *ocupación* y (d) *carga*.

En la Figura 30 es posible apreciar los valores medios diarios del conjunto de datos *M30*, correspondiendo a los valores de la *vmed*, *intensidad*, *ocupación* y *carga* a las Figuras 30a, 30b, 30c y 30d respectivamente. En ellas se puede apreciar más claramente apuntado en la Figuras 28 y 29: una caída nocturna de los valores entre la medianoche y las 6 de la mañana, seguido de un máximo en torno a las 8 de la mañana. Posteriormente los valores caen ligeramente. En el caso de la Figura 30a, la *vmed* se mantiene estable durante la tarde para volver a caer en torno a las 20 horas, y subir en las dos últimas horas del día. La *ocupación* cuenta con una oscilación más marcada, con picos a las 8 de la mañana, 2 y 8 de la tarde. La *intensidad* y *carga* presentan una curva idéntica, debido a la definición análoga de las dos variables. Los mínimos de estas dos variables son complementarios a los máximos de la *ocupación*.

A continuación, en la Figura 31 se pueden ver los valores medios semanales del dataset. En ellos se puede apreciar una fuerte distinción entre los valores correspondientes a los días laborales y los correspondientes al fin de semana. Por ejemplo, los valores de la *vmed* oscilan entre una velocidad media de 65 km/h, estando entre los 62 y 64 km/h de lunes a viernes y



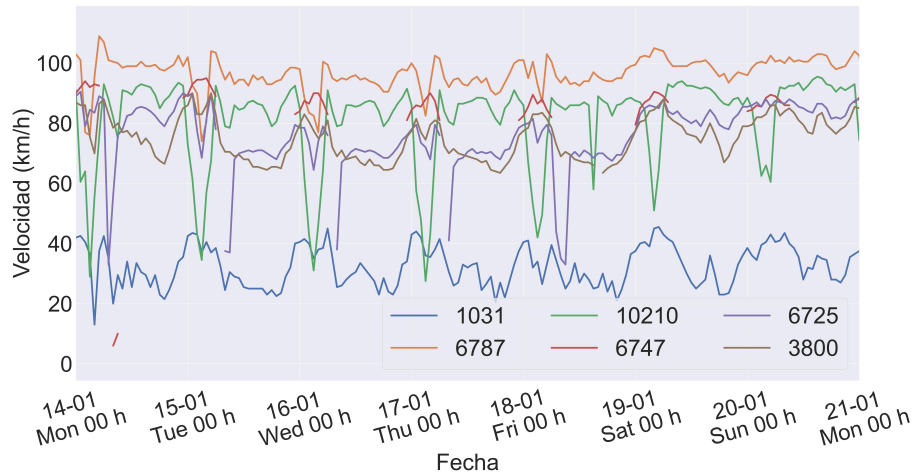
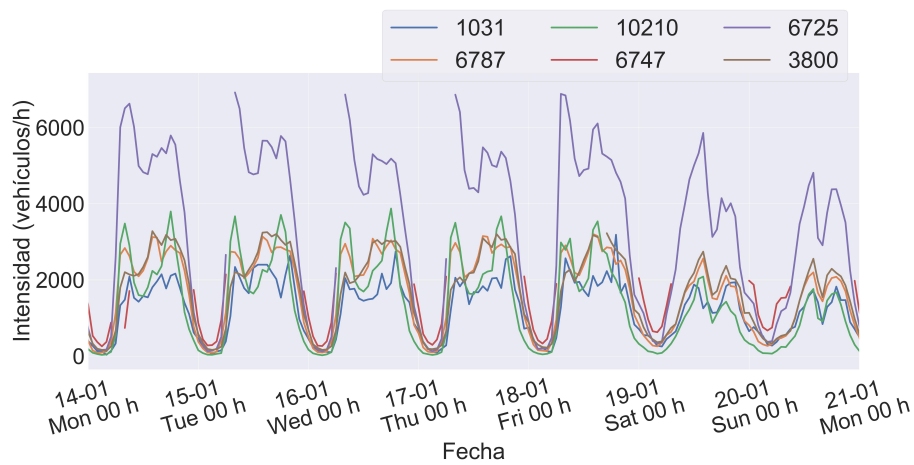
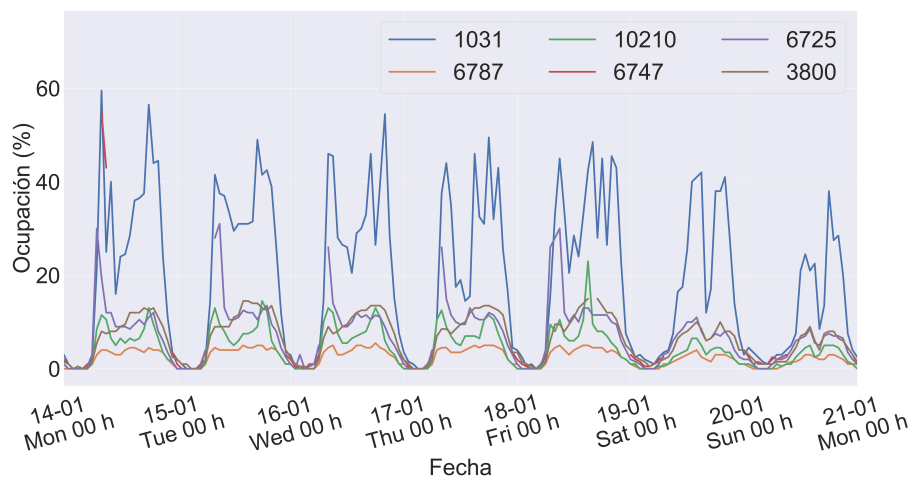
(a) *vmed*(b) *intensidad*

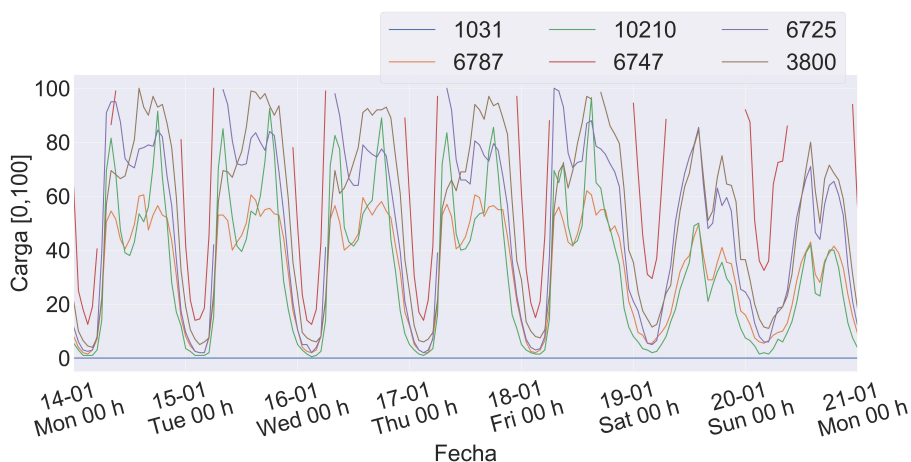
Figura 28: Señales recogidas durante la semana del 14 al 21 de enero de 2019 en seis puntos de medida aleatorios de la M30. Las variables corresponden a (a) *vmed*, (b) *intensidad*.

ascendiendo a los 68 km/h en sábado y domingo. La tendencia inversa puede verse en el resto de variables: existe una mayor congestión en las vías durante los días laborables, donde la *intensidad* ronda los 1,400 vehículos a la hora, y una ocupación en torno al 6.5 %. Esta congestión baja en el fin de semana, lo que facilita una circulación a mayor velocidad (valores de la *vmed* más altos). Una vez más es posible apreciar la correlación entre la velocidad media de circulación y la ocupación de las vías.

En cuanto a los valores medios mensuales, mostrados en la Figura 32 no se aprecian tendencias tan claras como las vistas para las medias diarias y semanales. Los valores parecen mostrar un comportamiento periódico, con una periodicidad aproximada cada cuatro días. Es interesante notar mínimo en torno al día 17, que coincide con el comienzo de la segunda quincena de cada mes. Estas bajadas son más apreciables en la *intensidad* (Figura 32b) y la *carga* (Figura 32d).



(a) ocupación



(b) carga

Figura 29: Señales recogidas durante la semana del 14 al 21 de enero de 2019 en seis puntos de medida aleatorios de la M30. Las variables corresponden a (a) *ocupación* y (b) *carga*.

Para concluir, en la Figura 33a se muestran los valores medios anuales del dataset, en las que a simple vista no se parecía ninguna estacionalidad en las tendencias de los datos. La velocidad media, mostrada en la Figura 33a, posee una naturaleza monotónica ascendente constante, es decir, la velocidad media de los vehículos aumenta ligeramente de principios a mediados de año. La *intensidad*, *ocupación* y *carga* (Figuras 33b, 33c y 33d) por el contrario experimenta una subida en enero, seguida de una meseta en los meses de febrero y marzo y una caída en abril. Quizá esta caída es debida a las vacaciones de Semana Santa, ocurridas entre el 14 y el 21 de abril de 2019. Es estas fechas cuando una porción significativa de la población de Madrid sale de la ciudad por vacaciones, desocupando las vías de circulación interurbana.

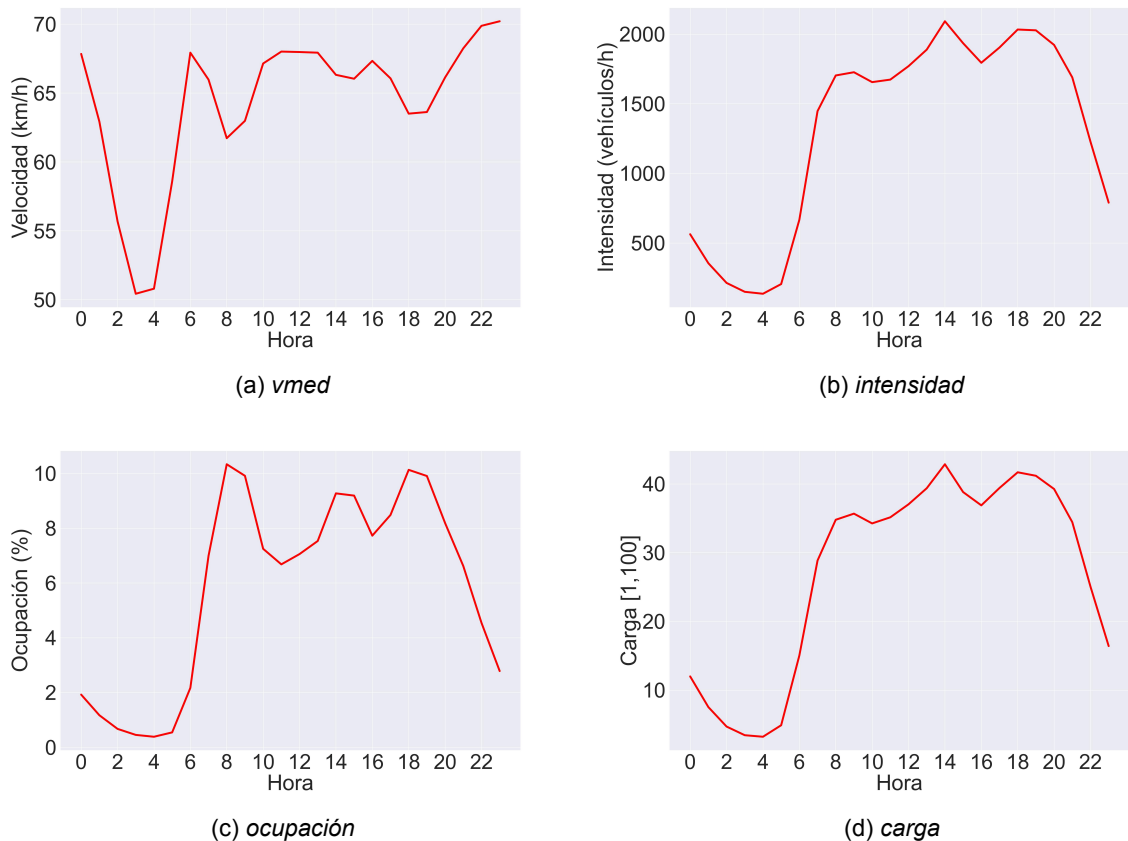


Figura 30: Valores medios diarios del conjunto de datos de la M30. Las figuras corresponden a (a) *vmed*, (b) *intensidad*, (c) *ocupación* y (d) *carga*. En todas las gráficas se puede apreciar una caída de la actividad del tráfico durante la madrugada, entre la medianoche y las cuatro de la mañana, y su repunte entre las seis y las ocho. Existe también una oscilación diurna, cuyos máximos coinciden con las horas de entrada y salida de la jornada laboral.

#### 4.1.3. Preprocesamiento y creación del dataset

En esta Subsección se detallará el proceso de preprocesamiento y compilación del dataset que se empleará en este trabajo. En adelante, el dataset obtenido recibirá el nombre de dataset *M30*.

En primer lugar, pasamos a descargar los conjuntos de datos de sus correspondientes páginas, cuyas URL son:

- Ubicación de los puntos de medida: <https://bit.ly/33qEtul>
- Histórico del tráfico: <https://bit.ly/3oZ1v4m>

En ambos casos, la sección de descargas de ambas páginas contienen los ficheros de datos ordenados por meses y años. Pinchamos en aquellos enlaces correspondientes a los archivos que comprenden el periodo entre enero y junio de 2019. En el caso del histórico del tráfico, el fichero es un archivo comprimido con formato *zip* que contienen los datos en formato *csv*.

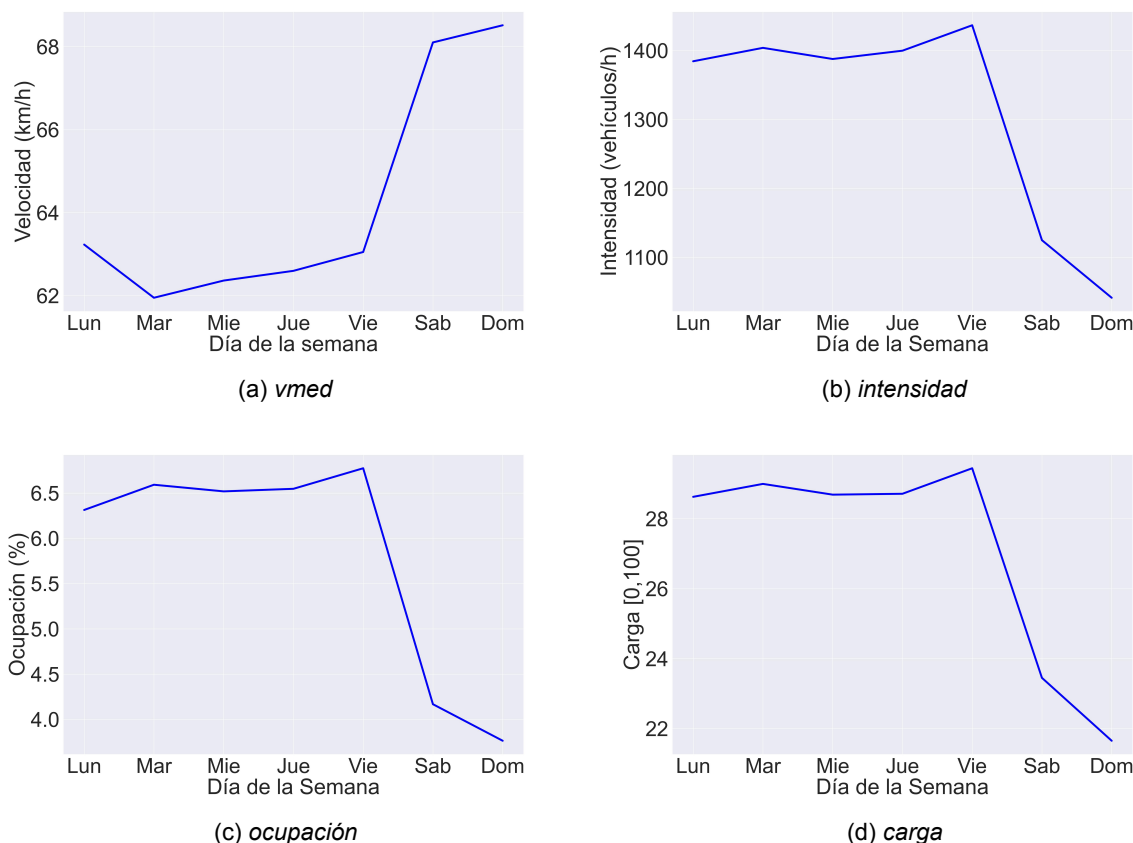


Figura 31: Valores medios semanales del conjunto de datos de la *M30*. Las figuras corresponden a (a) *vmed*, (b) *intensidad*, (c) *ocupación* y (d) *carga*. En todas las figuras se aprecia una clara diferencia entre los periodos de los días laborables y el fin de semana, donde en (a) la velocidad media sube y en (b), (c) y (d) la *intensidad*, *ocupación* y *carga* descienden.

En el caso de la ubicación de los puntos de medida, cada mes ofrece los datos en tres formatos diferentes, *csv*, *xlsx* y *zip*. Tal como se ha mencionado en la Subsección 4.1.1 el archivo comprimido contiene los archivos necesarios para una visualización de los datos a través de programas SIG. Sin embargo, en el presente trabajo solo se harán uso de los archivos con formato *csv*.

Una vez descargados los conjuntos de datos, se pasará a extraer el subconjunto de puntos de interés. Como se ha mencionado en la Subsección 4.1.2, este subconjunto está formado solamente por puntos pertenecientes a la vía *M30*. Para obtener esa lista, se utilizará el script de *python* `1_getting_M30_pmed.py`, ubicado en la carpeta `datasets` del código del proyecto. Este script implementa los siguientes pasos:

1. Lectura de todos los *csv* correspondientes a los meses de enero hasta junio de 2019. La lectura se hace a través de la librería *Pandas*, la cual carga estos ficheros como un objeto *DataFrame*. Este objeto es equivalente a una tabla relacional de datos.

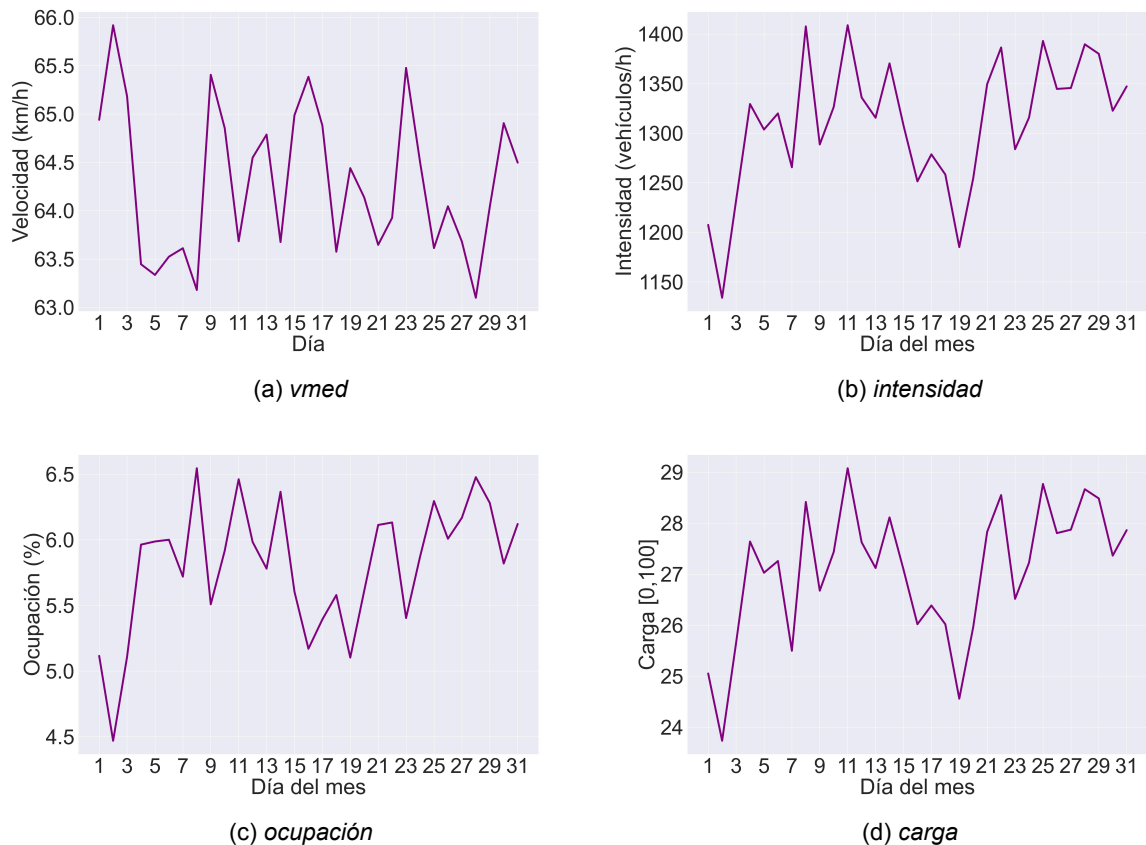


Figura 32: Valores medios mensuales del conjunto de datos de la M30. Las figuras corresponden a (a) *vmed*, (b) *intensidad*, (c) *ocupación* y (d) *carga*.

2. En el mismo proceso de lectura se concatenan todos los Dataframes de los distintos meses en una sola tabla.
3. Se filtran todas las filas cuyo valor del campo *tipo\_elem* es igual a *M30*.
4. Con la tabla resultante, se buscan entradas duplicadas, correspondientes al mismo punto de medida registrado en distintos meses. Estas entradas duplicadas se eliminan.
5. También se buscan entradas con campos que contengan nulos, para ser eliminadas. No se encuentran entradas de este tipo.
6. Se guarda en disco un archivo en formato csv con las entradas obtenidas. Los campos que se registran en este archivo son: *id*, *utm\_x*, *utm\_y*, *longitud* y *latitud*.

A continuación se pasa a extraer el histórico de datos relativo al subconjunto de puntos *M30*. Análogamente, se emplea el script `2_extract_pmed_from_historic.py` para ello. Los pasos seguidos son:

1. En primer lugar, cargamos el fichero generado previamente que contiene el listado de puntos de medida de la *M30*.

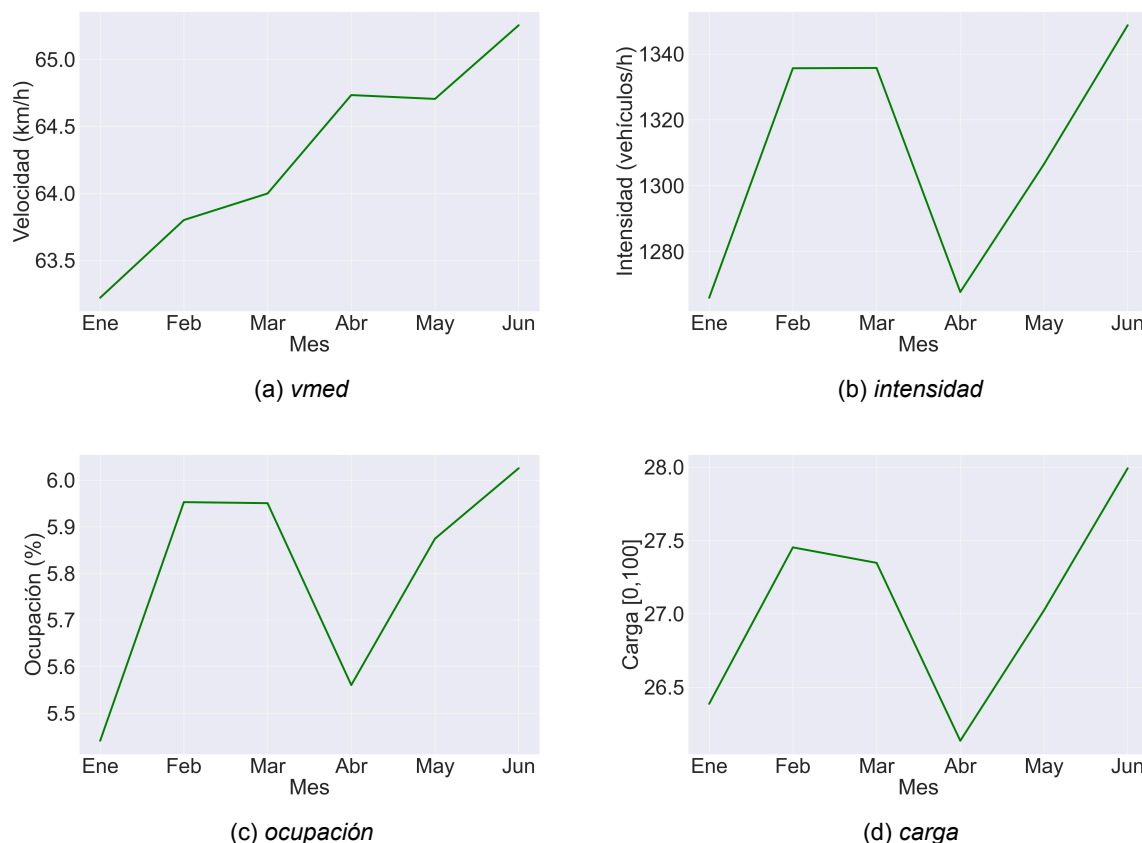


Figura 33: Valores medios anuales del conjunto de datos de la M30. Las figuras corresponden a (a) *vmed*, (b) *intensidad*, (c) *ocupación* y (d) *carga*. En (b), (c) y (d) se puede apreciar una bajada del flujo del tráfico en abril, mes que coincide con las vacaciones de Semana Santa.

- Seguidamente, se leen los ficheros en formato csv del histórico sirviéndose de la librería Pandas. En este caso cada uno de los ficheros mensuales tiene un tamaño en disco en torno a los 650 MB, por lo que concatenar todos los datos en un único DataFrame generaría un objeto muy grande, en torno a los 3.5 GB de datos, que sería muy pesado de almacenar y procesar. Por ello, al leer cada archivo csv, se filtra por punto de medida para retener únicamente aquellas entradas con un punto de medida en la M30.
- Los DataFrames filtrados, obtenidos para cada mes son concatenados en un único DataFrame. Este resultado se guarda en disco en formato csv.

Una vez se ha obtenido el conjunto de entradas del histórico que corresponden a los puntos de medida de la M30, se pasa a crear el fichero que formará el dataset. El campo seleccionado para la predicción es *vmed*, que corresponde con la velocidad media registrada en cada punto para cada instante temporal. Este proceso se realiza a través del script llamado `3_create_dataset.py`. El resultado es una tabla, en la que las medidas de los distintos nodos se disponen a lo largo de las columnas, mientras que cada fila representa un instante temporal. En la Tabla 12 se pueden observar las cinco primeras filas de la tabla. Esta matriz se guarda

en formato *HDF5*. La elección de este formato para expresar el dataset corresponde al mismo utilizado en los datasets de METR-LA y PEMS-BAY, de forma que el archivo creado sea compatible con los modelos originales.

Tabla 12: Muestra de las cinco primeras filas y seis primeras columnas del dataset *M30*. Las columnas corresponden a distintos puntos de medida, mientras que las filas son los distintos instantes temporales, separados en un intervalo de 15 minutos.

fecha id	1001	1002	1003	1006	1009	1010
2019-01-01 00:00:00	63.0	76.0	90.0	70.0	77.0	65.0
2019-01-01 00:15:00	63.0	78.0	93.0	70.0	74.0	72.0
2019-01-01 00:30:00	63.0	77.0	87.0	66.0	29.0	71.0
2019-01-01 00:45:00	63.0	74.0	79.0	72.0	62.0	57.0
2019-01-01 01:00:00	63.0	75.0	80.0	70.0	75.0	55.0

Seguidamente, se ejecuta el script `4_getting_sensor_graph_info.py`, el cual a partir del archivo del histórico filtrado para la *M30*, genera dos archivos necesarios para el funcionamiento de los modelos:

1. Por un lado, se genera un archivo llamado `graph_sensor_ids.txt` en formato de texto plano `txt`, cuyo contenido es cada uno de los valores del campo `id` (que identifica de manera única cada uno de los puntos de medida) separados por comas en una única fila.
2. Otro lado, se calculan las distancias (en metros) de todos los puntos con el resto. El cálculo de esta matriz es necesario para a su vez el cálculo de la matriz de adyacencias del grafo. Estas distancias se calculan a partir de las columnas `utm_x` y `utm_y`, que guardan las coordenadas del punto de medida en el sistema UTM. Se han elegido los parámetros de UTM, ya que este sistema de coordenadas es ortogonal, por lo que no tenemos que dar cuenta de la curvatura la Tierra en el cálculo de la distancia. El resultado se guarda en un archivo `csv` que contiene tres columnas: `to`, `from` y `cost`, donde la columna `cost` denota la distancia entre los puntos indicados en las columnas `to` y `from` de la misma fila.

La matriz de adyacencias del grafo se calcula usando la técnica de kernel gaussiano truncado (Thresholded Gaussian kernel) (Shuman et al. 2013), donde el peso de cada arista del grafo  $W_{ij}$  se calcula siguiendo la función gaussiana:

$$W_{ij} = \begin{cases} \exp -\frac{\nu_{ij}^2}{\sigma^2} & \text{si } \nu_{ij} \leq k \\ 0 & \text{si } \nu_{ij} > k \end{cases} \quad (4.1)$$

donde  $\nu_{ij}$  es la distancia entre el sensor  $i$  y el sensor  $j$ ,  $\sigma$  es la desviación estándar de las distancias y  $k$  es el umbral escogido. Este cálculo es realizado por el script `gen_adj_mx.py`,

incluido en el código de los modelos.

Por último, se realiza una partición del dataset en tres conjuntos disjuntos, que serán utilizados en las distintas etapas de entrenamiento, validación y evaluación o testeo de un modelo de aprendizaje automático. Se nombran a estos conjuntos por su nombre habitual de *train*, *val* y *test*. La partición se realiza siguiendo el orden cronológico de los datos, a los que también se les aplica una normalización *Z-score*. Los porcentajes de la partición se pueden consultar en la Tabla 13. Este proceso se realiza ejecutando el script `generate_training_data.py`, que comparten en común los modelos empleados.

Tabla 13: Partición de los diferentes subconjuntos *train* / *test* / *val* del dataset *M30*.

Partición	<i>train</i>	<i>test</i>	<i>val</i>
Porcentaje	70	20	10

## 4.2. Modelos predictivos

En la presente Sección se presentarán brevemente los modelos de predicción que se utilizarán en el estudio. Aunque en la Sección 2.2, en la discusión del estado del arte, ya se introdujeron estos modelos, a continuación vamos a describir formalmente dichos métodos sin entrar en detalles técnicos. En la Subsección 4.2.1 se introducirá una definición formal al problema de la predicción del tráfico, mientras que en las Subsecciones 4.2.2 y 4.2.3 se describirán los algoritmos correspondientes a los modelos DCRNN, y Graph WaveNet respectivamente.

### 4.2.1. Definición del problema

En primer lugar, es necesario definir la tarea a resolver en este trabajo, el llamado problema del pronóstico del tráfico. El principal objetivo es predecir los valores futuros de una de las variables asociadas al tráfico, como la velocidad, ocupación o densidad, a partir observaciones pasadas de estos valores. En el presente caso se pretende predecir la velocidad media de los vehículos que cruzan los sensores de tráfico.

Se puede definir la red de carreteras como un grafo dirigido y ponderado  $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{W})$ , donde el conjunto de sensores del tráfico son los nodos  $\mathcal{V}$ , con un total de  $N$  nodos,  $\mathcal{E}$  son los vértices del grafo que conectan los distintos sensores, y  $\mathbf{W} \in \mathbb{R}^{N \times N}$  es la matriz de adyacencia ponderada, que representa la correlación de los sensores entre sí.



Es posible expresar un observable asociado a los nodos como  $\mathbf{X} \in \mathbb{R}^{N \times P}$ , donde  $P$  es el número de características del observable. Si dicho observable es asociado al tráfico en un instante  $t$  como  $\mathbf{X}^{(t)}$ , el problema de la predicción se reduce a aprender una función  $h$  que a partir de un conjunto histórico  $T'$  de señales es posible obtener otro conjunto de señales en  $T$  instantes futuros:

$$[\mathbf{X}^{(t-T'-1)}, \dots, \mathbf{X}^{(t)}] \xrightarrow{h} [X^{(t+1)}, \dots, X^{(t+T)}] \quad (4.2)$$

#### 4.2.2. Diffusion Convolutional Recurrent Neural Network (DCRNN)

El modelo DCRNN fue el primero en emplear convoluciones en grafos para tratar el problema del pronóstico del tráfico. En concreto, el flujo del tráfico es descrito como un proceso de difusión, con transiciones aleatorias representadas por la matriz de transición de estados  $D_0^{-1}\mathbf{W}$ , siendo  $D_0^{-1} = \text{diag}(\mathbf{W}\mathbf{1})$  y  $\mathbf{1}$  un vector formado por unos. La operación de difusión por convolución está formada principalmente por dos términos, el término ya visto de  $D_0^{-1}\mathbf{W}$ , y el término de flujo inverso  $D_I^{-1}\mathbf{W}^T$ , que tiene en cuenta la difusión del flujo en sentido inverso. Entonces, la convolución sobre un observable  $\mathbf{X}$  con un filtro  $f_\theta$  se puede escribir como

$$\mathbf{X}_{:,p} * f_\theta = \sum_{k=0}^{K-1} \left( \theta_{k,1} (D_0^{-1}\mathbf{W})^k + \theta_{k,2} (D_I^{-1}\mathbf{W}^T)^k \right) \mathbf{X}_{:,p} \quad p \in \{1, \dots, P\} \quad (4.3)$$

donde  $K$  es el orden de la aproximación para la difusión,  $\theta_{1,k}$  y  $\theta_{2,k}$  son los parámetros de los filtros de la convolución y  $\mathbf{X}$  es la cantidad observada.

En la parte temporal se emplea una red recurrente tipo GRU, donde las multiplicaciones de cada bloque temporal son sustituidas por la convolución de la Eq. (4.3). La arquitectura final sigue el esquema de codificador-descodificador descrito en Sutskever et al. (2014), que se puede consultar en la Figura 34.

#### 4.2.3. Graph Wavenet (GWN)

En comparación con el trabajo de Li et al. (2018), Graph Wavenet (en adelante GWN) (Wu et al. 2019) introduce dos importantes cambios:

- Para construir la matriz de adyacencias del grafo, en DCRNN se asume que dos nodos cercanos están siempre fuertemente correlacionados, incluso si entre ellos apenas se registra paso de tráfico entre uno y otro. En cambio, GWN evita asumir a priori una correlación espacial. Asignado un vector de características a cada nodo, construye la matriz de adyacencias del grafo por aprendizaje. Esta matriz de adyacencias aprendida

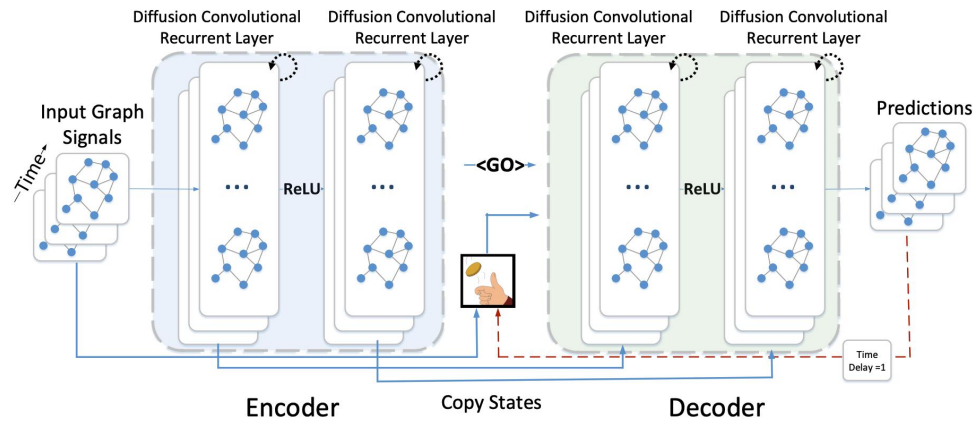


Figura 34: Arquitectura del modelo *Diffusion Convolutional Recurrent Neural Network*, compuesto de dos partes, el codificador (encoder) a la izquierda y el decodificador (decoder) a la derecha. Una serie histórica sirve de entrada para el codificador, el cual produce unos estados intermedios que sirven de entrada al decodificador. La arquitectura está basada en el modelo *Sequence to Sequence* (Sutskever et al. 2014). Figura tomada de Li et al. (2018).

puede también ser usada en combinación con una matriz construida como en DCRNN.

- GWN utiliza convoluciones de 1D y 2D para la parte temporal, con bloques al estilo de van den Oord et al. (2016) (ver Figura 35). Esto reduce considerablemente el coste computacional de DCRNN. Además de esto, GWN implementa una predicción múltiple de 12 pasos temporales a la vez, lo que reduce aún más el tiempo de ejecución comparado con DCRNN, el cual predecía cada paso temporal de forma recursiva.

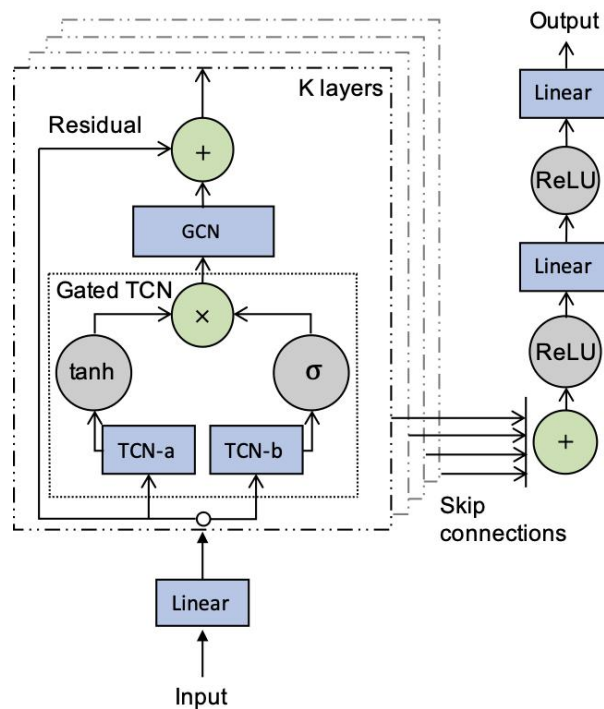


Figura 35: Bloque perteneciente a la arquitectura del modelo Graph Wavenet, basado en el trabajo de van den Oord et al. (2016). Figura tomada de Wu et al. (2019).

### 4.3. Evaluación de resultados

A continuación se expondrán los resultados obtenidos del entrenamiento de los modelos con el conjunto de datos obtenido en la Subsección 4.1. En la Subsección 4.3.1 se mostrarán los resultados obtenidos con el modelo DCRNN, descrito en la Subsección 4.2.2, y en la Subsección 4.3.2 se hará lo propio con los resultados del modelo GWN, detallado en la Subsección 4.3.2. En último lugar se realizará una comparativa del conjunto de los modelos en la Subsección 4.3.3.

#### 4.3.1. DCRNN

En la Figura 36 se pueden consultar la evolución de las pérdidas obtenidas para el conjunto de entrenamiento y validación durante el entrenamiento del modelo DCRNN. Las diferentes líneas corresponden a distintos cálculos a los que se les ha incluido los términos directo e inverso en la convolución de la Eq. (4.3), denotados como  $D_0^{-1}W$  y  $D_I^{-1}W^T$ , llamado *Bi* en el gráfico, incluyendo solo el término directo, denotado como *Uni*, y despreciando ambos términos, lo que equivale a no realizar una convolución espacial. Como se puede apreciar, no se encuentran diferencias entre los tres entrenamientos, lo que significa que la convolución espacial no tiene ningún efecto sobre el aprendizaje del modelo. Se ha de mencionar que, como se puede observar la figura, el entrenamiento para el caso bi-direccional solo duró unas 12,500 iteraciones, lo que equivale a unas 64 épocas. Esto es debido a que, en todos los casos en los que se ejecutó este cálculo, el entrenamiento recibía una orden de abortar la ejecución (*SIGKILL*) en torno a este número de épocas. Se desconoce con seguridad el origen de esta señal de terminar el proceso, ya que no se ha tenido tiempo de explorar el código en profundidad, pero una pequeña monitorización de los entrenamientos mostró una demanda en memoria RAM alta y creciente en el tiempo, posible motivo por el que el sistema operativo puede haber abortado el entrenamiento mucho después de haberse iniciado y alocado en memoria.

Otro hecho a destacar es que el entrenamiento de estos modelos ha sido con una notable diferencia el más largo de toda la evaluación, ya que en el caso del cálculo bi-direccional el entrenamiento tuvo una duración de 5 días, 2 horas, 38 minutos y 37 segundos, quedando aun así incompleto como se ha explicado antes. Esto nos hace pensar que el código original, además de los inconvenientes algorítmicos mencionados en la Subsección 4.2.3, la implementación es inestable en su gestión de la memoria.

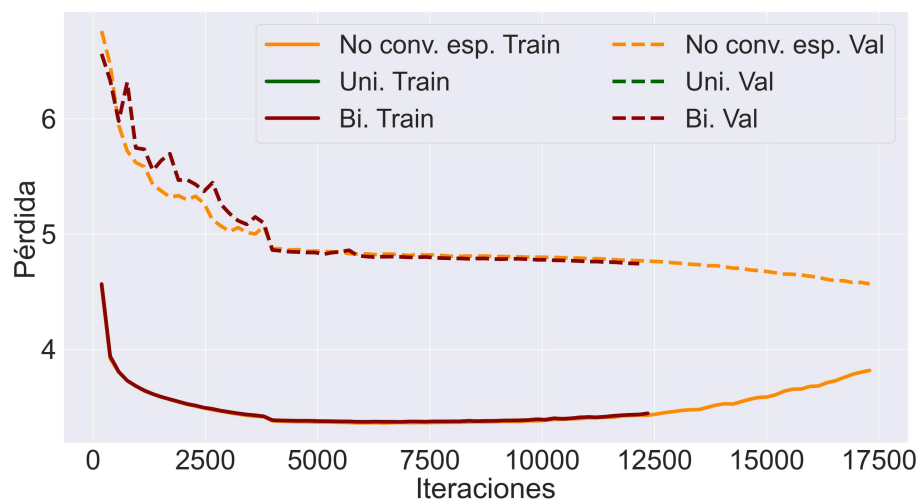


Figura 36: Valores de la pérdida para los conjuntos de entrenamiento y validación durante el entrenamiento del modelo DCRNN. Las líneas corresponden a *Bi*: Convolución bi-direccional, *Uni*: Convolución unidireccional y *NoConv*: Sin convolución espacial. Las líneas sólidas representan las pérdidas del conjunto de entrenamiento, mientras que las líneas discontinuas representan las pérdidas del conjunto de validación. Se puede apreciar que apenas hay diferencia entre los tres entrenamientos.

En la Tabla 14 se pueden consultar las métricas obtenidas para los entrenamientos mostrados en la Figura 36, donde se muestran el error absoluto medio (MAE) y el error cuadrático medio (RMSE) en cada uno de los cálculos para el conjunto del horizonte de predicciones. Este horizonte tiene un intervalo temporal de 15 minutos, abarcando desde una predicción a 15 minutos hasta una a 180 minutos (3 horas). Es una sorpresa comprobar como la media tanto el MAE y como del RMSE es más baja en el caso en el que no hay convolución espacial, seguida por el caso en el que solo se incluye la convolución directa y, por último, la convolución bi-direccional. No solo las convoluciones en una o ambas direcciones no tienen ninguna influencia, como apuntaba la Figura 36, sino que parece que cada uno de los términos de la Eq. (4.3) impacta negativamente en el resultado final.

Para comparar cómo evoluciona el error de la predicción para un horizonte cada vez mayor, hemos representado el error cuadrático medio en un diagrama de barras para los tres cálculos en la Figura 37. De este gráfico podemos ver que para horizontes a tiempos cortos en modelo bi-direccional tiene errores más pequeños, pero éstos crecen a tiempos más largos. Es el caso contrario del modelo uni-direccional, que es mayor a tiempos cortos y disminuye para tiempos largos. En el caso sin convolución espacial el error se reduce ligeramente para tiempos largos. Sin embargo, ninguno de los tres modelos muestran una tendencia clara con respecto al horizonte de predicciones.

Tabla 14: Error absoluto medio (MAE) y el error cuadrático medio para los entrenamientos del modelo DCRNN. Las columnas en cada una de estas métricas corresponden a *Bi*: Convolución bi-direccional, *Uni*: Convolución unidireccional y *NoConv*: Sin convolución espacial. El horizonte de predicciones de valores futuros tiene un intervalo temporal de 15 minutos. En negrita está marcada la aproximación con la mejor métrica.

Horizonte (min.)	MAE			RMSE		
	Bi	Uni	NoConv	Bi	Uni	NoConv
15	6.59	6.42	6.29	10.75	14.23	13.30
30	6.79	6.26	6.13	11.58	14.01	13.58
45	7.13	5.73	5.64	12.02	14.43	14.08
60	6.97	5.52	5.45	13.79	13.44	10.59
75	7.28	5.93	5.83	15.23	13.73	13.85
90	4.61	6.11	5.99	13.39	13.20	14.28
105	5.08	5.33	5.26	12.43	12.90	11.41
120	5.34	6.73	5.03	15.01	12.18	11.80
135	6.37	6.59	4.58	14.76	12.55	13.06
150	6.13	5.07	6.58	12.92	10.64	12.78
165	5.59	6.86	6.44	14.14	11.44	12.10
180	5.86	4.60	6.71	14.47	11.88	12.45
Media	6.15	5.93	<b>5.83</b>	13.38	12.89	<b>12.77</b>

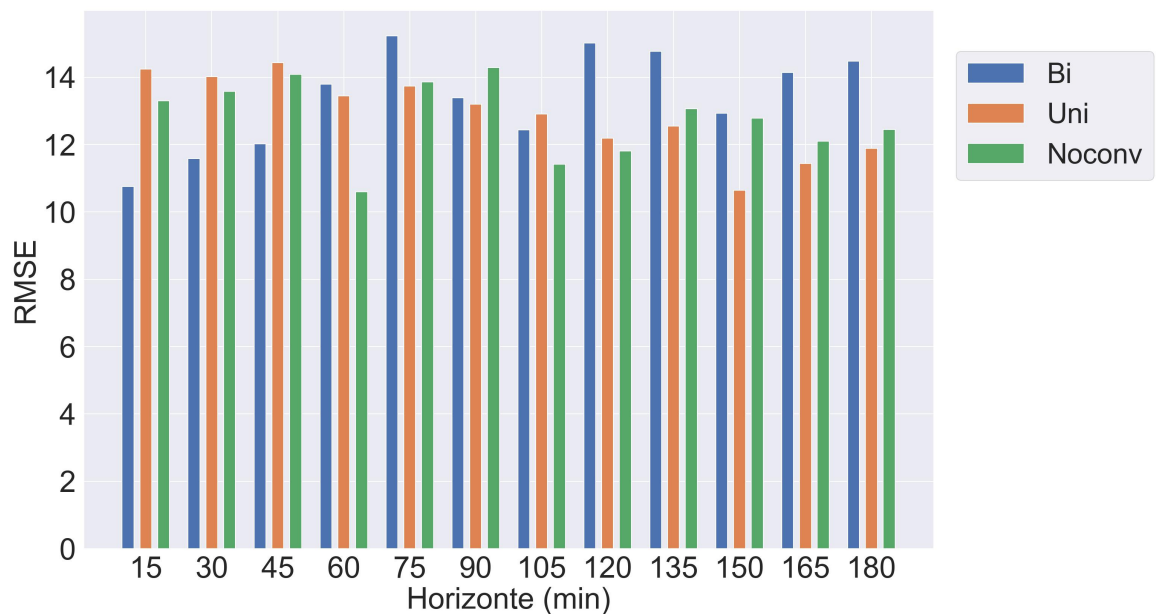


Figura 37: Error cuadrático medio sobre el conjunto de test en función del horizonte de predicciones para el modelo DCRNN. El horizonte de predicciones posee un intervalo temporal de 15 minutos. En ninguno de los casos el modelo muestra una tendencia clara de dependencia con respecto al horizonte de predicciones.

#### 4.3.2. Graph Wavenet

En la evaluación para el modelo Graph Wavenet (GWN) se ha incluido el entrenamiento del modelo con una serie de aproximaciones: el modelo puede contar con los términos de la convolución directa e inversa y sin convolución espacial, análogamente al caso del modelo DCRNN en la Subsección 4.3.1. Además de estas aproximaciones, el modelo puede contar con un tér-

mino adaptativo, ya mencionado en la descripción del modelo en la Subsección 4.2.3. En total, contamos con seis aproximaciones: Bi-direccional además de adaptativo (*Bi-A*), Uni-direccional además de adaptativo (*Uni-A*), solo bi-direccional (*Bi*), solo unidireccional (*Uni*), solo adaptativo (*A*) y sin convolución espacial (*UniConv*).

Las pérdidas de los conjuntos de entrenamiento y validación para el entrenamiento del modelo Graph Wavenet (GWN) se muestran en la Figura 38. En el gráfico se puede apreciar que las menores pérdidas son alcanzadas en el caso en que las aproximaciones bidireccional y unidireccional contienen un término adaptativo, y curiosamente también cuando el cálculo no incluye una convolución espacial. La línea de las pérdidas prácticamente se solapa en estos tres casos. Con una pérdida mayor se encuentran las aproximaciones no adaptativas, las cuales también se solapan. Con una pérdida claramente mayor al resto se encuentra el caso que solo contiene términos adaptativos. Esto indica que aunque un conocimiento previo de la matriz de adyacencias es crucial para obtener unos mejores resultados, no obstante es posible obtener un modelo capaz de aprender la tarea de predicción en este dataset. Todos los entrenamientos muestran la misma convergencia, con un primer descenso muy rápida hasta la décima época, seguida de una convergencia más lenta entre las épocas 10 y 40 épocas.

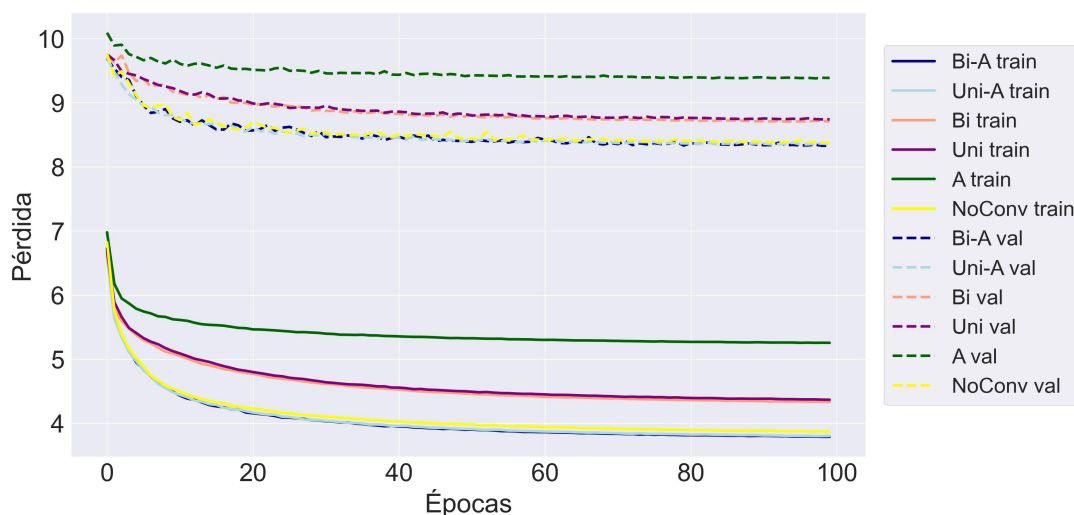


Figura 38: Pérdidas de los conjuntos de entrenamiento y validación para el entrenamiento del modelo Graph Wavenet (GWN). Las líneas corresponden a los siguientes entrenamientos: bidireccional adaptativo (*Bi-A*), unidireccional adaptativo (*Uni-A*), solo bidireccional (*Bi*), solo unidireccional (*Uni*), solo adaptativo (*A*) y sin convolución espacial (*NoConv*). Las líneas sólidas representan las pérdidas sobre el conjunto de entrenamiento, mientras que las líneas discontinuas las representan el conjunto de validación.

En las Tablas 15, 16 y 17 se muestran resultados para el RMSE, MAE y MAPE del entrenamiento de los modelos de GWN. En todos ellos el modelo que obtiene es el caso bidireccional adaptativo, seguido del unidireccional adaptativo y el que carece de convolución espacial. Es-

tos resultados están en consonancia con lo visto las pérdidas de entrenamiento mostradas en la Figura 38. Como era de esperar, el mejor resultado se obtiene con el modelo que no asume aproximaciones, el caso bidireccional adaptativo, seguido del modelo unidireccional adaptativo. Es curioso notar que el otro modelo con resultado muy similar en cifras es aquel que no incluye convolución espacial. Esto puede deberse a que en las vías de la M30 no carriles solo tienen un sentido de la marcha, siendo necesarios coger una salida e incorporarse de nuevo para cambiar de sentido. Esto hace que a través de los carriles solo fluya tráfico en una dirección, haciendo despreciable el término de flujo inverso de la convolución. También es interesante notar de nuevo, que aunque las métricas relativas al caso solo adaptativo están por apreciablemente por encima del resto de aproximaciones, es de nuevo posible obtener un modelo predictivo sin un conocimiento a priori del grafo, lo cual en situaciones en los que se desconoce la estructura del grafo puede ser de enorme utilidad.

Tabla 15: Error cuadrático medio del modelo GWN sobre el conjunto de test para diferentes tipos de convolución. En orden, las abreviaturas corresponden a: bidireccional adaptativo (*Bi-A*), unidireccional adaptativo (*Uni-A*), solo bidireccional (*Bi*), solo unidireccional (*Uni*), solo adaptativo (*A*) y sin convolución espacial (*NoConv*). El horizonte de predicciones de valores futuros tiene un intervalo temporal de 15 minutos. En negrita está marcada la aproximación con la mejor métrica.

Horizonte (min.)	<i>Bi-A</i>	<i>Uni-A</i>	<i>Bi</i>	<i>Uni</i>	<i>A</i>	<i>NoConv</i>
15	11.29	11.30	11.37	11.38	11.52	11.31
30	11.59	11.61	11.74	11.76	12.06	11.64
45	11.79	11.84	12.09	12.12	12.67	11.86
60	11.72	11.76	11.95	11.97	12.40	11.78
75	11.86	11.92	12.22	12.27	12.93	11.93
90	11.92	11.99	12.34	12.40	13.17	11.99
105	11.96	12.04	12.45	12.51	13.37	12.04
120	12.00	12.08	12.54	12.62	13.55	12.06
135	12.04	12.12	12.63	12.71	13.70	12.09
150	12.08	12.16	12.72	12.81	13.85	12.12
165	12.12	12.21	12.82	12.90	13.97	12.16
180	12.19	12.27	12.93	13.01	14.09	12.21
Media	<b>11.88</b>	11.94	12.32	12.37	13.11	11.93

Por otro lado, es posible visualizar la variación del error en una de las métricas para predicciones a tiempos más largos. Es el caso de la Figura 39, donde se muestra la variación del error cuadrático medio para distintos horizontes. En primer lugar en cada uno de los horizontes se confirma el resultado que se pudo ver tanto en la Figura 38 como en las Tablas 15, 16 y 17: la superioridad de los modelos que combinan convolución espacial directa e inversa con un tratamiento adaptativo junto con el modelo que carece de convolución espacial. De hecho, en la Figura se revela que para estos modelos la variación del error para horizontes cada vez mayores es menor que en el resto de casos. Incluso, el caso con peores resultados, el modelo solo adaptativo muestra peores resultados para todos los horizontes, mostrando un incremento



Tabla 16: Error absoluto medio (MAE) del modelo GWN sobre el conjunto de test para diferentes tipos de convolución. En orden, las abreviaturas corresponden a: bidireccional adaptativo (*Bi-A*), unidireccional adaptativo (*Uni-A*), solo bidireccional (*Bi*), solo unidireccional (*Uni*), solo adaptativo (*A*) y sin convolución espacial (*NoConv*). El horizonte de predicciones de valores futuros tiene un intervalo temporal de 15 minutos. En negrita está marcada la aproximación con la mejor métrica.

Horizonte (min.)	<i>Bi-A</i>	<i>Uni-A</i>	<i>Bi</i>	<i>Uni</i>	<i>A</i>	<i>NoConv</i>
15	7.89	7.90	7.97	7.98	8.10	7.91
30	8.11	8.12	8.26	8.28	8.51	8.15
45	8.19	8.22	8.40	8.43	8.77	8.24
60	8.24	8.27	8.51	8.53	8.97	8.30
75	8.28	8.32	8.60	8.64	9.17	8.35
90	8.32	8.37	8.69	8.73	9.34	8.39
105	8.35	8.40	8.77	8.81	9.49	8.42
120	8.38	8.43	8.83	8.89	9.62	8.44
135	8.40	8.46	8.90	8.95	9.74	8.46
150	8.44	8.49	8.97	9.02	9.84	8.49
165	8.47	8.53	9.04	9.09	9.94	8.52
180	8.53	8.58	9.12	9.16	10.01	8.56
Media	<b>8.30</b>	8.34	8.67	8.71	9.29	8.35

Tabla 17: Error porcentual absoluto medio (MAPE) del modelo GWN sobre el conjunto de test para diferentes tipos de convolución. En orden, las abreviaturas corresponden a: bidireccional adaptativo (*Bi-A*), unidireccional adaptativo (*Uni-A*), solo bidireccional (*Bi*), solo unidireccional (*Uni*), solo adaptativo (*A*) y sin convolución espacial (*NoConv*). El horizonte de predicciones de valores futuros tiene un intervalo temporal de 15 minutos. En negrita está marcada la aproximación con la mejor métrica.

Horizonte (min.)	<i>Bi-A</i>	<i>Uni-A</i>	<i>Bi</i>	<i>Uni</i>	<i>A</i>	<i>NoConv</i>
15	0.14	0.14	0.14	0.14	0.14	0.14
30	0.15	0.15	0.15	0.15	0.16	0.15
45	0.15	0.15	0.16	0.16	0.17	0.15
60	0.15	0.15	0.16	0.16	0.17	0.15
75	0.15	0.16	0.16	0.16	0.18	0.15
90	0.15	0.16	0.17	0.17	0.19	0.16
105	0.16	0.16	0.17	0.17	0.20	0.16
120	0.16	0.16	0.17	0.18	0.20	0.16
135	0.16	0.16	0.18	0.18	0.20	0.16
150	0.16	0.16	0.18	0.18	0.21	0.16
165	0.16	0.17	0.18	0.18	0.21	0.16
180	0.16	0.17	0.18	0.19	0.22	0.16
Media	<b>0.16</b>	0.16	0.17	0.17	0.19	0.16

del error mucho mayor que el resto.

A continuación se presentan resultados de la influencia del modelo completo de GWN (bidireccional adaptativo) con la longitud de la secuencia de entrada, donde también la secuencia pronosticada varía en longitud a la vez que la secuencia de entrada. Las longitudes de entrada y salida son de 4, 6, 8, 10 y 12 elementos lo que equivale a secuencias de duraciones de 60, 90, 120, 150 y 180 minutos respectivamente, siendo la de 180 minutos la longitud original de la muestra en los experimentos anteriores. Las pérdidas de los conjuntos de entrenamiento y validación pueden observarse en la Figura 40. Se puede apreciar cómo, exceptuando la longitud



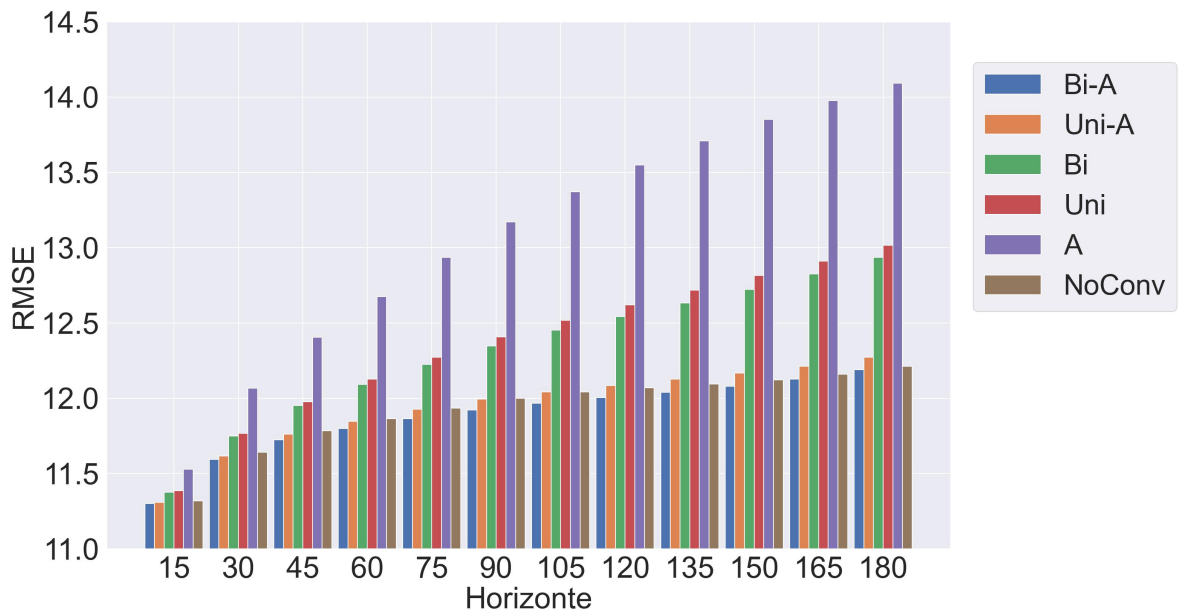


Figura 39: Error cuadrático medio sobre el conjunto de test en función del horizonte de predicciones para el modelo GWN. El horizonte de predicciones posee un intervalo temporal de 15 minutos.

de 12, la pérdida es menor cuanto menor es la longitud de la secuencia. La razón por la que la menor pérdida se obtenga a partir de la más larga secuencia es quizá producida por el hecho de que los hiperparámetros seleccionados son los mismos que se encontraron al minimizar la pérdida de un modelo con una secuencia de 12 pasos temporales.

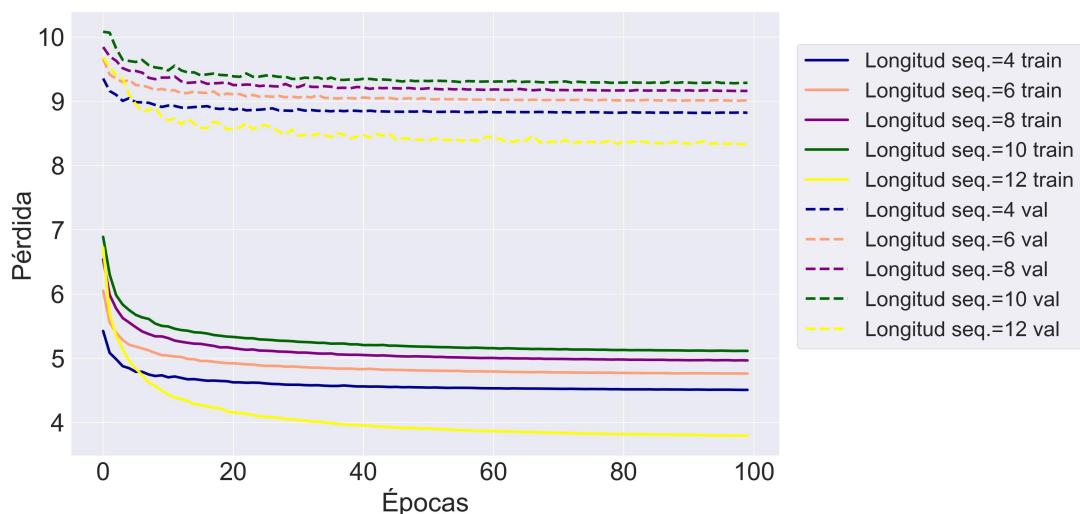


Figura 40: Pérdidas de validación para el entrenamiento del modelo Graph Wavenet (GWN) para distintas longitudes de la secuencia de entrada. Las longitudes de las secuencias son de 4, 6, 8, 10 y 12, lo que corresponde también a duraciones de 60, 90, 120, 150 y 180 minutos.

En la Figura 18 se muestran resultados de las métricas para el experimento con diferentes longitudes de secuencia de entrada y salida. En concordancia con el resultado mostrado en la Figura 40, las mejores métricas se obtienen para el caso de una secuencia de 180 minutos de duración, unas 3 horas. Uno de los posibles motivos de este comportamiento es el apuntado

anteriormente, ya que los hiperparámetros utilizados son los que minimizan la pérdida para el caso de 180 minutos. Además, comprobamos que para secuencias cortas la calidad de la predicción es mayor empeorando a para tiempos más largos.

Tabla 18: Error cuadrático medio del modelo GWN para diferentes longitudes de secuencia. Las distintas columnas muestran resultados para secuencias de 60, 90, 120, 150 y 180 minutos de duración. El horizonte de predicciones de valores futuros tiene un intervalo temporal de 15 minutos. En negrita está marcada la aproximación con la mejor métrica.

Horizonte (min.)	60	90	120	150	180
15	11.58	11.54	11.51	11.50	11.29
30	12.21	12.14	12.09	12.06	11.59
45	12.63	12.53	12.46	12.41	11.72
60	12.98	12.85	12.77	12.70	11.79
75	-	13.16	13.05	12.98	11.86
90	-	13.42	13.31	13.22	11.92
105	-	-	13.53	13.42	11.96
120	-	-	13.72	13.61	12.00
135	-	-	-	13.77	12.04
150	-	-	-	13.92	12.08
165	-	-	-	-	12.12
180	-	-	-	-	12.19
Media	12.36	12.61	12.81	12.96	<b>11.88</b>

En la Figura 41 se puede ver cambio de la métrica RMSE para predicciones a cada vez tiempos más largos. Es posible apreciar cómo las predicciones con secuencias de entrada más largas consiguen un error menos, siendo la predicción con secuencias de 180 minutos notablemente mejor. Es interesante ver cómo, en cada modelo, el error de la predicción se va haciendo mayor, teniendo una pequeña variación en el error en el modelo de 180 minutos.

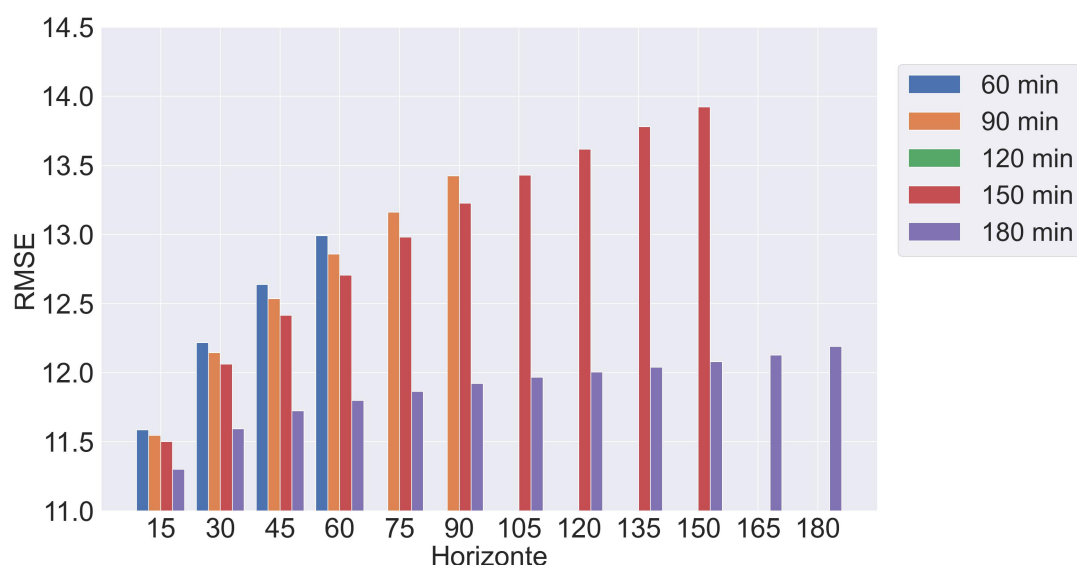


Figura 41: Error cuadrático medio sobre el conjunto de test para las predicciones del modelo GWN, con diferentes longitudes de secuencia de entrada y salida.

### 4.3.3. Comparativa

Finalmente en esta Subsección se presentará una comparativa de los resultados obtenidos con los modelos DCRNN y GWN. Los experimentos correspondientes a estos modelos son aquellos que incluyen todos los términos de la convolución, con flujo directo e inverso para el modelo DCRNN, y además el término adaptativo para el modelo GWN. Para mayor completitud, se incluyen otros métodos clásicos de predicción de series temporales que servirán de base para la comparativa. Los métodos son:

- Estático: Consiste solamente en asumir los mismos valores del histórico con una ventana del tiempo marcada por el horizonte de predicción. Es decir, si tenemos un valor determinado en el primer lugar de la secuencia, y la predicción se hace para tres pasos temporales, entonces se asume que el cuarto lugar de la secuencia tendrá el mismo valor que el primero. Es decir, que se considera que la serie permanece estática para una determinada predicción a futuro.
- Promedio histórico (*historical average*, HA): Este método calcula una media ponderada de los valores de un histórico de datos.
- VAR (*Vector auto-regression*): Es un modelo auto-regresivo, que además de incluir el histórico de valores de esa variable como una combinación lineal, también incluye una combinación lineal del histórico de otras variables. El retraso (o lag en inglés) es el orden, o el número de entradas previas del histórico que se incluyen en el cálculo. En este caso está fijado a 3 (lag=3).

En la Tabla 19 se muestran los resultados de la comparativa. En los métodos de base solo se han incluido predicciones a 1, 3, 6, y 12 pasos temporales, lo que equivale a 15, 45, 90, 180 minutos. Los resultados muestran los métodos de base son capaces de realizar mejores predicciones a tiempos cortos, ya que de ellos se obtiene métricas más bajas. Sin embargo, los modelos basados en grafos muestra un menor error para predicciones largas, y un menor crecimiento del error a medida que aumentamos el horizonte.

La estabilidad de estos modelos a tiempos largos puede estar causada principalmente por dos factores: el primero es la importancia de una correcta descripción espacial de la correlación entre los nodos; y el segundo es un tratamiento temporal que capta la naturaleza no-lineal a medida que los pronósticos son cada vez más lejanos en el tiempo. No obstante, la similitud de los resultados vistos en las Subsecciones 4.3.1 y 4.3.2 revela una prácticamente nula influencia del flujo en dirección inversa en la correlación espacial, por lo que podemos concluir

que la estabilidad de los modelos es debida en mayor parte a la descripción no-lineal de la dependencia temporal.

Tabla 19: Resultados de la comparativa de modelos. En la columna de la izquierda se representa el horizonte de la predicción en minutos, que abarca desde 15 hasta 180 minutos. Para cada fila, la cifra marcada en negrita es la mejor métrica.

Hor. (min)	MAE					RMSE				
	Est	HA	VAR	DCRNN	GWN	Est	HA	VAR	DCRNN	GWN
15	<b>4.42</b>	4.97	4.58	6.59	7.89	9.04	8.69	<b>7.65</b>	10.75	11.29
30	-	-	-	<b>6.79</b>	8.11	-	-	-	<b>11.58</b>	11.59
45	6.05	<b>4.97</b>	5.11	7.13	8.19	11.98	8.69	<b>8.41</b>	12.02	11.79
60	-	-	-	<b>6.97</b>	8.24	-	-	-	13.79	<b>11.72</b>
75	-	-	-	<b>7.28</b>	8.28	-	-	-	15.23	<b>11.86</b>
90	8.06	4.97	5.64	<b>4.61</b>	8.32	15.27	<b>8.69</b>	9.02	13.39	11.92
105	-	-	-	<b>5.08</b>	8.35	-	-	-	12.43	<b>11.96</b>
120	-	-	-	<b>5.34</b>	8.38	-	-	-	15.01	<b>12.00</b>
135	-	-	-	<b>6.37</b>	8.40	-	-	-	14.76	<b>12.04</b>
150	-	-	-	<b>6.13</b>	8.44	-	-	-	12.92	<b>12.08</b>
165	-	-	-	<b>5.59</b>	8.47	-	-	-	14.14	<b>12.12</b>
180	10.65	4.97	6.53	<b>5.86</b>	8.53	18.95	8.69	<b>10.11</b>	14.47	12.19

La Tabla 20 muestra los tiempos de ejecución de los experimentos mostrados en la Tabla 19 correspondientes a los modelos DCRNN y GWN. Sorprende la gran diferencia de tiempos que muestra el entrenamiento de modelo GWN con respecto al DCRNN, que ya se mencionó en la Subsección 4.3.1. Esta diferencia, de días frente a horas, hace mucho más útil el modelo GWN para una aplicación práctica.

Por último, en las Tablas 21 y 21 se muestran comparados los resultados reportados de estos modelos para los datasets públicos METR-LA y PEMS-BAY, ya mencionados en la Sección 4.1. Tanto el MAE como el RMSE son mucho más bajos para los otros datasets. Sin embargo, hay un par de aspectos que diferencian estos conjuntos públicos con el conjunto de datos de este trabajo.

- En el caso de METR-LA y PEMS-BAY existe una mayor influencia en la correlación espacial entre nodos, por lo que existen márgenes de mejor en estos datasets debido a los términos de flujo inverso de la convolución. Como hemos comprobado en la Subsección 4.3.1 para el modelo DCRNN y en la Subsección 4.3.2 para el GWN, la influencia espacial es despreciable en los resultados obtenidos.

Tabla 20: Tiempos ejecución de los experimentos mostrados en la Tabla 19 para los modelos DCRNN y GWN.

Duración	DCRNN	GWN
	5 días 2 h 38 min. 38 s	1 h 58 min. 57 s

- En segundo lugar estos dataset representan grafos más pequeños como se puede comparar en la Tabla 10, 207 y 325 nodos METR-LA y PEMS-BAY respectivamente frente a los 417 nodos de M30. Es más, el número de aristas en METR-LA y PEMS-BAY es solo el 2.2 % y 3.4 % respectivamente.

Tabla 21: Error absoluto medio (MAE) para los modelos DCRNN y GWN sobre tres conjuntos de datos diferentes: METR-LA, PEMS-BAY y M30.

Hor. (min)	DCRNN			GWN		
	METR-LA	PEMS-BAY	M30	METR-LA	PEMS-BAY	M30
15	2.77	1.38	6.59	2.69	1.30	7.89
30	3.15	1.74	6.79	3.07	1.63	8.11
60	3.60	2.07	6.97	3.53	1.95	8.24

Tabla 22: Error cuadrático medio (RMSE) para los modelos DCRNN y GWN sobre tres conjuntos de datos diferentes: METR-LA, PEMS-BAY y M30.

Hor. (min)	DCRNN			GWN		
	METR-LA	PEMS-BAY	M30	METR-LA	PEMS-BAY	M30
15	5.38	2.95	10.75	5.15	2.74	11.29
30	6.45	3.97	11.58	6.22	3.70	11.59
60	7.60	4.74	13.79	7.37	4.52	11.72



## 5. Conclusiones y trabajo futuro

Finalmente, en este Capítulo se presentarán las conclusiones del presente trabajo, en el que se repasarán los principales hallazgos, tanto del análisis del dataset compilado a partir de datos publicados para la ciudad de Madrid, como para las evaluaciones de los modelos predictivos con dicho conjunto de datos. Por último, se concluirá este trabajo apuntando a posibles líneas futuras para la continuación de este estudio, ya sea extendiendo la investigación en modelos predictivos basados en aprendizaje profundo, o ampliando a otros conjuntos de datos de naturaleza distinta. En la Sección 5.1 se expondrán las conclusiones, mientras que en la Sección 5.2 se realizará un comentario de las posibles líneas futuras de trabajo.

### 5.1. Conclusiones

El presente trabajo ha tenido como objeto el evaluar la precisión y eficiencia de una pareja de modelos de predicción de aprendizaje automático basados en grafos sobre un conjunto de datos recogidos de sensores situados en la M30 de la ciudad de Madrid. A continuación se repasarán las conclusiones alineadas con los principales objetivos propuestos en este trabajo:

- En la Sección 2.1 se ha analizado la problemática del tráfico, describiendo las principales consecuencias negativas que conllevan para las sociedades modernas: la congestión, los accidentes y la contaminación. Los estudios relativos a la congestión mencionados en la Subsección 2.1.1 señalan las cada vez mayores pérdidas asociadas en los atascos, no solo de tiempo por parte de los usuarios de medios de transporte rodado, sino también su coste económico, un 1 % de PIB europeo según cálculos de la Comisión Europea. Se estima que en la ciudad de Madrid el tiempo perdido en un viaje de 30 minutos es de 17 minutos por las mañanas y 14 por las tardes, lo que equivale a unos cuatro días y 21 horas al año.

Los accidentes de tráfico son otros de los problemas derivados del tráfico, que en España ascendieron a 104,080 en 2019 según datos oficiales. De estos, casi el doble ocurrió

en vías urbanas (66,741) que en carreteras convencionales (37,339). Esta diferencia es incluso mayor en el caso de Madrid, donde la proporción es de 4 accidentes en vías urbanas por cada uno en inter-urbanas. La dependencia de diversos factores del tráfico con los accidentes ha sido ampliamente estudiada en el pasado, estableciendo una relación directa de la siniestralidad con la congestión del tráfico.

Por último, pero no menos importante, otra de las consecuencias adversas del tráfico es la contaminación del aire, la cual la Organización Mundial de la Salud estima que causa unas cuatro millones de muertes prematuras al año, de las cuales medio millón se produjeron en Europa en 2015. Entre los principales factores que afectan a la calidad del aire de las ciudades está el tráfico rodado. Es por ello que gobiernos locales y estatales han trazados planes de regulación del tráfico en núcleos urbanos, como los sucesivos protocolos implantados en la ciudad de Madrid, que han conseguido disminuir los niveles de  $\text{NO}_2$  un 20 % entre los años 2010 y 2019. Por todo lo anterior, investigaciones y desarrollo de nuevos modelos matemáticos cada vez más precisos y robustos son de vital importancia para que instituciones y organismos públicos puedan proponer regulaciones y soluciones adecuadas y eficaces a los problemas causados por el tráfico rodado.

- Se ha repasado el estado del arte en la Sección 2.2, la cual se ha centrado en modelos de aprendizaje automático clásico y profundo, en las Subsecciones 2.2.1, 2.2.2 y 2.2.3 respectivamente. En cuanto a los primeros, la principal limitación de los modelos clásicos es que necesitan de una ingeniería manual sobre las características de entrada que hace que no sea una técnica factible para un volumen de datos elevado. Para los modelos de aprendizaje profundo, los modelos de redes recurrentes han supuesto un avance notable en el desarrollo de modelos del tráfico, ya que al procesar una serie de datos de forma secuencial, permiten captar las dependencias temporales de las distintas variables del tráfico. Sin embargo, estos modelos fallan a la hora de captar las complejas inter-dependencias espaciales entre los distintos puntos de medida del tráfico, siendo insuficientes a la hora de adaptarse al cambiante flujo del tráfico. Es aquí donde los modelos de aprendizaje basados en grafos aportan su ventaja, al incorporar una correlación entre los nodos de una red de carreteras por medio de un grafo. El conocimiento de este grafo se puede tener de manera previa, como en el caso de modelos como DCRNN, o puede ser desconocido a priori, como el caso de modelos como GWN y sus sucesores, que incorporan el grafo a través de una matriz de adyacencias aprendida durante el entrenamiento del modelo.



- En la Sección 4.1 se ha descrito la creación de un conjunto de datos nuevo relativo a la ciudad de Madrid, a partir de datos publicados en fuentes institucionales. Estos datos provienen de mediciones de sensores situados en las carreteras de la M30, una vía de circulación rápida alrededor del distrito centro de la ciudad de Madrid. El grafo que representa la red de carreteras descrita está compuesta por un total de 417 puntos de medida, y que comprenden un periodo de seis meses, de enero a junio de 2019.

En la Subsección 4.1.1 se han descrito los dos conjuntos de datos a partir de los cuales se ha construido el dataset: el conjunto relativo a los puntos de medida y el histórico del tráfico desde 2013.

A continuación, en la Subsección 4.1.2 se ha realizado un análisis exploratorio de los datos. En este pequeño estudio se ha mostrado el carácter periódico de los datos, con dos claras frecuencias: una diaria y otra semanal. En la diaria existen claramente máximos de actividad alrededor de las horas punta de entrada y salida del trabajo, y un mínimo de actividad de madrugada, entre la medianoche y las cuatro de la mañana. En la frecuencia semanal existen claras diferencias entre los valores de los días laborables y los relativos al fin de semana, con una fuerte bajada de la actividad durante los sábados y domingos. En cuanto a la frecuencia anual, al restringirse este trabajo a solo los seis primeros meses del año no ha sido posible observar una estacionalidad, aunque sí ha sido interesante notar un descenso de actividad en abril, atribuible con las vacaciones de Semana Santa.

La Subsección 4.1.3 describe el pre-procesado y limpieza de los datos con los scripts de *python* creados en este trabajo para generar el conjunto de datos que sirva a los modelos predictivos.

- Se han descrito los modelos elegidos para este trabajo en la Sección 4.2. Estos modelos han sido seleccionados por mostrar una evolución en el estado del arte respecto a la predicción basada en grafos. En la Subsección 4.3.1 y 4.3.2 se han mostrado los resultados obtenidos con los modelos DCRNN y GWN, del cual de este primero se ha de destacar el ingente tiempo de ejecución requerido para su entrenamiento, lo cual dificulta en gran medida su experimentación con él. Los resultados muestran una influencia nula de la dependencia espacial en la predicción, ya que es inapreciable la diferencia entre las métricas en el caso que incluye la convolución espacial y la que no lo incluye. Este es debido probablemente a la naturaleza del grafo que representa el conjunto de datos, ya que la M30 es un conjunto de vías de único sentido de la circulación, y para la que un cambio de sentido implica un cambio de carretera, no de carril, por lo que no existe

tráfico cruzado en ambos sentidos dentro de una misma vía.

Además, para el modelo GWN la inclusión del término adaptativo mejora sensiblemente la preedición, pero por sí solo no es capaz de igualar la precisión en el caso en que sí contamos con la matriz de adyacencias. Esto apunta no solo a que el conocimiento previo del grafo es crucial para un pronóstico más preciso, sino también que es posible entrenar un modelo predictivo que tenga en cuenta la correlación espacial sin tener a priori la matriz de adyacencias.

- Por último, en la comparativa final de la Subsección 4.3.3 ha sido posible apreciar que, aunque los modelos de base, mucho más sencillos, son capaces de realizar mejores predicciones a tiempos cortos, los modelos basados en grafos son capaces de realizar pronósticos a tiempos mucho más largos sin incrementar su error. La estabilidad en la predicción (el hecho de que cometan un error constante para horizontes cada vez mayores) puede ser debido a su tratamiento de la correlación espacial y temporal. Sin embargo, como se ha expuesto antes, la correlación espacial tiene una dependencia despreciable en los resultados de este trabajo, por lo que concluimos que la descripción no-lineal llevada a cabo por estos modelos juega un papel esencial para realizar predicciones a tiempos cada vez más futuros.

## 5.2. Trabajo futuro

Una de las líneas más inmediatas de trabajo es la ampliación de la evaluación a un mayor número de modelos predictivos, tanto de modelos basados en grafos, como de otros basados en otras arquitecturas. Entre los modelos que merecen una exploración con el conjunto de datos de la M30 están los que construyen una correlación espacio-temporal explícita, como el de Yu et al. (2019) o el más reciente de Yang et al. (2021); o también que emplean mecanismos de atención como S. Guo et al. (2019) y Do et al. (2019).

Sin embargo, en los últimos años las redes llamadas *transformers* (Vaswani et al. 2017), basadas en mecanismos de atención, están revolucionando el panorama del aprendizaje profundo, tanto en lenguaje (Brown et al. 2020; Devlin et al. 2019), como en visión (Chen et al. 2020; Dosovitskiy et al. 2020; Ramesh et al. 2021). Aunque *STAWnet* (Tian et al. 2021) empleaba mecanismos de atención, existen trabajos recientes en los que emplean arquitecturas basadas en *transformers*, como el de L. Cai et al. (2020). En futuras evaluaciones se podría plantear

una comparativa de modelos basados *transformers* contra aquellos basados en grafos.

Por otro lado, este trabajo se ha restringido a las vías de la M30, un conjunto de carreteras de circulación rápida alrededor del distrito centro de la ciudad de Madrid. Como se discutió en la Subsección 4.1.2, los datos relativos a estas vías se caracterizan por reflejar fuertemente su naturaleza. En futuros estudios sería interesante ampliar el estudio al resto de datos de tráfico interurbano de Madrid. En los datos provenientes de tráfico interurbano convencional esperamos ver una correlación entre más puntos de media, en la que el flujo del tráfico en ambos sentidos juegue en mayor papel que el visto en este estudio.



## Bibliografía

- Amidi, A., & Amidi, S. (2018). VIP Cheatsheet: Recurrent Neural Networks.
- Ayuntamiento de Madrid. (2019). *Calidad del Aire 2019* (inf. téc. N.º 1). Dirección General de Sostenibilidad y Control Ambiental. Ayuntamiento de Madrid.
- Ayuntamiento de Madrid. (2021). *Portal de datos abiertos de Madrid*. Consultado el 4 de noviembre de 2021, desde [datos.madrid.es](https://datos.madrid.es)
- Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., ... Amodei, D. (2020). Language Models are Few-Shot Learners [eprint: 2005.14165]. <http://arxiv.org/abs/2005.14165>
- Cadar, R., Boitor, M. R., & Dumitrescu, M. (2017). Effects of traffic volumes on accidents: the case of romania's national roads.
- Cai, L., Janowicz, K., Mai, G., Yan, B., & Zhu, R. (2020). Traffic transformer: Capturing the continuity and periodicity of time series for traffic forecasting. *Transactions in GIS*, 24(3), 736-755. <https://doi.org/https://doi.org/10.1111/tgis.12644>
- Cai, P., Wang, Y., Lu, G., Chen, P., Ding, C., & Sun, J. (2016). A spatiotemporal correlative k-nearest neighbor model for short-term traffic multistep forecasting. *Transportation Research Part C Emerging Technologies*, 62, 21-34. <https://doi.org/10.1016/j.trc.2015.11.002>
- Carleo, G., & Troyer, M. (2017). Solving the quantum many-body problem with artificial neural networks. *Science*, 355(6325), 602-606. <https://doi.org/10.1126/science.aag2302>
- Chen, M., Radford, A., Child, R., Wu, J., Jun, H., Dhariwal, P., & Luan, D. (2020). Image GPT. *lcm1*.
- Comisión Europea. (2017). *European Urban Mobility. Policy Context* (inf. téc. N.º 1). European Commission.
- Defferrard, M., Bresson, X., & Vandergheynst, P. (2016). Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering. *Proceedings of the 30th International Conference on Neural Information Processing Systems*, 3844-3852.

- Devlin, J., Chang, M., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. En J. Burstein, C. Doran & T. Solorio (Eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)* (pp. 4171-4186). Association for Computational Linguistics. <https://doi.org/10.18653/v1/n19-1423>
- DGT. (2019). *Anuario Estadístico de accidentes 2019* (inf. téc. N.º 1). Dirección General de Tráfico.
- DGT. (2020). *Las principales cifras de la Siniestralidad Vial. España 2019* (inf. téc. N.º 1). Dirección General de Tráfico.
- DGT. (2021a). *El Gobierno impulsa un nuevo modelo de seguridad vial para reducir los accidentes de tráfico*. <https://www.lamoncloa.gob.es/consejodeministros/resumenes/Paginas/2020/101120-cministros.aspx>
- DGT. (2021b). *Tablas estadísticas 2019*. <https://www.dgt.es/es/seguridad-vial/estadisticas-e-indicadores/accidentes-30dias/tablas-estadisticas/2019/>
- Do, L., Vu, H., Vo, B., Liu, Z., & Phung, D. (2019). An effective spatial-temporal attention based neural network for traffic flow prediction. *Transportation Research Part C: Emerging Technologies*, 108, 12-28. <https://doi.org/10.1016/j.trc.2019.09.008>
- Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., & Houlsby, N. (2020). An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale [eprint: 2010.11929]. <http://arxiv.org/abs/2010.11929>
- EEA. (2020). *Air Quality in Europe. 2020 Report* (inf. téc. N.º 12). European Environment Agency.
- Facebook Inc. (2019). *Pytorch*. Consultado el 23 de noviembre de 2019, desde <https://pytorch.org>
- Github Inc. (2021). *GitHub*. <https://github.com/>
- Goodman, N. (2017). *Uber ai labs open sources pyro, a deep probabilistic programming language*. <https://eng.uber.com/pyro/>
- Google Inc. (2019). *TensorFlow*. Consultado el 23 de noviembre de 2019, desde <https://tensorflow.org>
- Google Inc. (2021). *Google Trends*. Consultado el 12 de noviembre de 2021, desde <https://trends.google.es/>
- Guo, S., Lin, Y., Feng, N., Song, C., & Wan, H. (2019). Attention Based Spatial-Temporal Graph Convolutional Networks for Traffic Flow Forecasting. *Proceedings of the AAAI Conference*

- rence on Artificial Intelligence, 33(01), 922-929. <https://doi.org/10.1609/aaai.v33i01.3301922>
- Guo, Z., Zhao, X., Chen, Y., Wu, W., & Yang, J. (2019). Short-term passenger flow forecast of urban rail transit based on GPR and KRR. *IET Intelligent Transport Systems*, 13(9), 1374-1382. <https://doi.org/https://doi.org/10.1049/iet-its.2018.5530>
- Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., del Río, J. F., Wiebe, M., Peterson, P., ... Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, 585(7825), 357-362. <https://doi.org/10.1038/s41586-020-2649-2>
- Head A, J. (1959). Predicting traffic accidents from roadway elements on urban extensions of state highways. *Highway Research Board Bulletin*, 208(208), 45-63. <https://trid.trb.org/view/111231>
- Hunter, J. D. (2007). Matplotlib: A 2D Graphics Environment. *Computing in Science & Engineering*, 9(3), 90-95. <https://doi.org/10.1109/MCSE.2007.55>
- IARC. (2016). *Outdoor Air Pollution* (Vol. 109). International Agency for Research on Cancer.
- IBM. (2021). *Python vs. R: What's the Difference?* Consultado el 23 de marzo de 2021, desde <https://trends.google.es/>
- Karpathy, A. (2019). *Pytorch at Tesla*. Consultado el 6 de noviembre de 2019, desde <https://www.youtube.com/watch?v=oBklItKXtDE>
- Kipf, T. N., & Welling, M. (2017). Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.
- Li, Y., Yu, R., Shahabi, C., & Liu, Y. (2018). Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting. *International Conference on Learning Representations*. <https://openreview.net/forum?id=SJiHXGWAZ>
- Ma, X., Dai, Z., He, Z., Ma, J., Wang, Y., & Wang, Y. (2017). Learning Traffic as Images: A Deep Convolutional Neural Network for Large-Scale Transportation Network Speed Prediction. *Sensors*, 17(4). <https://doi.org/10.3390/s17040818>
- Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Jia, Y., Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, ... Xiaoqiang Zheng. (2015). TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems.

- McKinney, W. (2010). Data Structures for Statistical Computing in Python. En S. van der Walt & J. Millman (Eds.), *Proceedings of the 9th Python in Science Conference* (pp. 51-56).
- McKinney, W. (2011). pandas: a Foundational Python Library for Data Analysis and Statistics.
- Oliphant, T. E. (2006). *A guide to NumPy* (Vol. 1). Trelgol Publishing USA.
- OpenAI. (2020). *OpenAI Standardizes on PyTorch*. <https://openai.com/blog/openai-standardizes-on-pytorch/>
- Pandas development team. (2020). *pandas-dev/pandas: Pandas* (Ver. latest). Zenodo. <https://doi.org/10.5281/zenodo.3509134>
- Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., & Lerer, A. (2017). Automatic differentiation in PyTorch. *NIPS-W*.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12, 2825-2830.
- Peters, T. (2004). *The Zen of Python*. Consultado el 19 de noviembre de 2019, desde <https://www.python.org/dev/peps/pep-0020/>
- Pfaff, T., Fortunato, M., Sanchez-Gonzalez, A., & Battaglia, P. W. (2021). Learning Mesh-Based Simulation with Graph Networks.
- Pfau, D., Spencer, J. S., Alexander, A. G., & Foulkes, W. M. (2019). Ab-initio solution of the many-electron schrödinger equation with deep neural networks [Publisher: American Physical Society \_eprint: 1909.02487]. *arXiv*, 033429. <https://doi.org/10.1103/physrevresearch.2.033429>
- Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., & Sutskever, I. (2021). Zero-Shot Text-to-Image Generation. En M. Meila & T. Zhang (Eds.), *Proceedings of the 38th International Conference on Machine Learning* (pp. 8821-8831). PMLR. <https://proceedings.mlr.press/v139/ramesh21a.html>
- Retallack, A. E., & Ostendorf, B. (2019). Current Understanding of the Effects of Congestion on Traffic Accidents. *International journal of environmental research and public health*, 16(18), 3400.
- Robinson, D. (2017). *The Incredible Growth of Python*. Consultado el 19 de noviembre de 2019, desde <https://stackoverflow.blog/2017/09/06/incredible-growth-python/>
- Sanchez-Gonzalez, A., Godwin, J., Pfaff, T., Ying, R., Leskovec, J., & Battaglia, P. (2020). Learning to Simulate Complex Physics with Graph Networks. En H. D. III & A. Singh (Eds.),



- Proceedings of the 37th International Conference on Machine Learning* (pp. 8459-8468). PMLR. <https://proceedings.mlr.press/v119/sanchez-gonzalez20a.html>
- Shleifer, S., McCreery, C., & Chitters, V. (2019). Incrementally Improving Graph WaveNet Performance on Traffic Prediction.
- Shuman, D. I., Narang, S. K., Frossard, P., Ortega, A., & Vandergheynst, P. (2013). The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE Signal Processing Magazine*, 30(3), 83-98. <https://doi.org/10.1109/MSP.2012.2235192>
- stackoverflow. (2021). *2021 Developer Survey*. Consultado el 1 de enero de 2021, desde <https://insights.stackoverflow.com/survey/2021#most-popular-technologies-misc-tech>
- Stokes, J. M., Yang, K., Swanson, K., Jin, W., Cubillos-Ruiz, A., Donghia, N. M., MacNair, C. R., French, S., Carfrae, L. A., Bloom-Ackermann, Z., Tran, V. M., Chiappino-Pepe, A., Badran, A. H., Andrews, I. W., Chory, E. J., Church, G. M., Brown, E. D., Jaakkola, T. S., Barzilay, R., & Collins, J. J. (2020). A Deep Learning Approach to Antibiotic Discovery. *Cell*, 180(4), 688-702.e13. <https://doi.org/10.1016/j.cell.2020.01.021>  
doi: 10.1016/j.cell.2020.01.021
- Sun, J., Li, T., Li, F., & Chen, F. (2016). Analysis of safety factors for urban expressways considering the effect of congestion in Shanghai, China. *Accident Analysis & Prevention*, 95, 503-511.
- Sun, S., Chen, J., & Sun, J. (2019). Traffic congestion prediction based on GPS trajectory data. *International Journal of Distributed Sensor Networks*, 15(5), 1550147719847440. <https://doi.org/10.1177/1550147719847440>
- Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to Sequence Learning with Neural Networks. En Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence & K. Q. Weinberger (Eds.), *Advances in Neural Information Processing Systems*. Curran Associates, Inc. <https://proceedings.neurips.cc/paper/2014/file/a14ac55a4f27472c5d894ec1c3c743d2-Paper.pdf>
- Tang, J., Chen, X., Hu, Z., Zong, F., Han, C., & Li, L. (2019). Traffic flow prediction based on combination of support vector machine and data denoising schemes. *Physica A: Statistical Mechanics and its Applications*, 534, 120642. <https://doi.org/10.1016/j.physa.2019.03.007>
- TfL. (2021). *Traffic Modelling Guidelines* (inf. téc. N.º 4). Transport for London.
- The Economist. (2021). *The hidden cost of congestion*. <https://www.economist.com/graphic-detail/2018/02/28/the-hidden-cost-of-congestion>

- Tian, C., & Chan, W. K. (2021). Spatial-temporal attention wavenet: A deep learning framework for traffic prediction considering spatial-temporal dependencies. *IET Intelligent Transport Systems*, 15(4), 549-561. <https://doi.org/10.1049/itr2.12044>
- TomTom. (2021a). *TomTom Madrid Traffic Index*. [https://www.tomtom.com/en\\_gb/traffic-index/madrid-traffic](https://www.tomtom.com/en_gb/traffic-index/madrid-traffic)
- TomTom. (2021b). *TomTom Traffic Index Ranking*. [https://www.tomtom.com/en\\_gb/traffic-index/ranking/](https://www.tomtom.com/en_gb/traffic-index/ranking/)
- van den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., & Kavukcuoglu, K. (2016). WaveNet: A Generative Model for Raw Audio. *Arxiv*. <https://arxiv.org/abs/1609.03499>
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is All you Need. En I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan & R. Garnett (Eds.), *Advances in Neural Information Processing Systems*. Curran Associates, Inc. <https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf>
- Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., & Bengio, Y. (2018). Graph Attention Networks.
- Vitaliano, D. F., & Held, J. (1991). Road accident external effects: an empirical assessment. *Applied Economics*, 23(2), 373-378. <https://doi.org/10.1080/00036849100000146>
- Wang, C., Quddus, M., & Ison, S. (2013). A spatio-temporal analysis of the impact of congestion on traffic safety on major roads in the UK. *Transportmetrica A: Transport Science*, 9(2), 124-148.
- WHO. (2018a). Air pollution. *World Health Organization Media Centre*. Consultado el 19 de septiembre de 2021, desde <https://www.who.int/health-topics/air-pollution>
- WHO. (2018b). Ambient (outdoor) air pollution. *World Health Organization Media Centre*. Consultado el 19 de septiembre de 2021, desde [https://www.who.int/news-room/fact-sheets/detail/ambient-\(outdoor\)-air-quality-and-health](https://www.who.int/news-room/fact-sheets/detail/ambient-(outdoor)-air-quality-and-health)
- WHO. (2021a). *Air pollution*. Consultado el 19 de septiembre de 2021, desde [https://www.who.int/data/gho/data/indicators/indicator-details/GHO/ambient-air-pollution-attributable-death-rate-\(per-100-000-population\)?bookmarkId=cf8ed905-7131-44b7-9586-0653529af631](https://www.who.int/data/gho/data/indicators/indicator-details/GHO/ambient-air-pollution-attributable-death-rate-(per-100-000-population)?bookmarkId=cf8ed905-7131-44b7-9586-0653529af631)
- WHO. (2021b). *WHO global air quality guidelines* (inf. téc. N.º 1). World Health Organization.
- Woo, J. C. H. (1957). Correlation of Accident Rates and Roadway Factors.

- Wu, Z., Pan, S., Long, G., Jiang, J., & Zhang, C. (2019). Graph WaveNet for Deep Spatial-Temporal Graph Modeling. *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, 1907-1913. <https://doi.org/10.24963/ijcai.2019/264>
- Yang, S., Liu, J., & Zhao, K. (2021). Space Meets Time: Local Spacetime Neural Network For Traffic Flow Forecasting.
- Yao, H., Tang, X., Wei, H., Zheng, G., & Li, Z. (2019). Revisiting Spatial-Temporal Similarity: A Deep Learning Framework for Traffic Prediction. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01), 5668-5675. <https://doi.org/10.1609/aaai.v33i01.33015668>
- Yao, H., Wu, F., Ke, J., Tang, X., Jia, Y., Lu, S., Gong, P., Ye, J., & Li, Z. (2018). Deep Multi-View Spatial-Temporal Network for Taxi Demand Prediction. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1). <https://ojs.aaai.org/index.php/AAAI/article/view/11836>
- Yu, B., Yin, H., & Zhu, Z. (2018). Spatio-Temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting. *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, 3634-3640. <https://doi.org/10.24963/ijcai.2018/505>
- Yu, B., Yin, H., & Zhu, Z. (2019). ST-UNet: A Spatio-Temporal U-Network for Graph-structured Time Series Modeling.
- Zhang, D., & Kabuka, M. R. (2017). Combining Weather Condition Data to Predict Traffic Flow: A GRU Based Deep Learning Approach. *2017 IEEE 15th Intl Conf on Dependable, Automatic and Secure Computing, 15th Intl Conf on Pervasive Intelligence and Computing, 3rd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress(DASC/PiCom/DataCom/CyberSciTech)*, 1216-1219. <https://doi.org/10.1109/DASC-PiCom-DataCom-CyberSciTec.2017.194>
- Zhang, Z., Li, M., Lin, X., Wang, Y., & He, F. (2019). Multistep speed prediction on traffic networks: A deep learning approach considering spatio-temporal dependencies. *Transportation Research Part C: Emerging Technologies*, 105, 297-322. <https://doi.org/https://doi.org/10.1016/j.trc.2019.05.039>
- Zhao, Z., Chen, W., Wu, X., Chen, P. C. Y., & Liu, J. (2017). LSTM network: a deep learning approach for short-term traffic forecast. *IET Intelligent Transport Systems*, 11(2), 68-75. <https://doi.org/https://doi.org/10.1049/iet-its.2016.0208>

- Zhou, J., Cui, G., Hu, S., Zhang, Z., Yang, C., Liu, Z., Wang, L., Li, C., & Sun, M. (2020). Graph neural networks: A review of methods and applications. *AI Open*, 1, 57-81. <https://doi.org/https://doi.org/10.1016/j.aiopen.2021.01.001>
- Zhou, J., Cui, G., Hu, S., Zhang, Z., Yang, C., Liu, Z., Wang, L., Li, C., & Sun, M. (2021). Graph Neural Networks: A Review of Methods and Applications.