



MÁSTER UNIVERSITARIO EN INGENIERÍA INDUSTRIAL

TRABAJO FIN DE MÁSTER

Modelado de la marcha y del consumo de trenes
metropolitanos de conducción automática mediante técnicas
de machine learning

Autor: Sofía Rodríguez Pita

Director: Adrián Fernández Rodríguez

Co-Director: Antonio Fernández Cardador

Madrid

Junio de 2021

Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título
Modelado de la marcha y del consumo de trenes metropolitanos de conducción automática
mediante técnicas de machine learning

en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el
curso académico 2020/21 es de mi autoría, original e inédito y
no ha sido presentado con anterioridad a otros efectos.

El Proyecto no es plagio de otro, ni total ni parcialmente y la información que ha sido
tomada de otros documentos está debidamente referenciada.



Fdo.: Sofía Rodríguez Pita

Fecha: 18/ 06/ 21

Autorizada la entrega del proyecto

EL DIRECTOR DEL PROYECTO



Fdo.: Adrián Fernández Rodríguez

Fecha: 21/ 06/ 21

EL DIRECTOR DEL PROYECTO



Fdo.: Antonio Fernández Cardador

Fecha: 21/ 06/ 21



MÁSTER UNIVERSITARIO EN INGENIERÍA INDUSTRIAL

TRABAJO FIN DE MÁSTER

Modelado de la marcha y del consumo de trenes
metropolitanos de conducción automática mediante técnicas
de machine learning

Autor: Sofía Rodríguez Pita

Director: Adrián Fernández Rodríguez

Co-Director: Antonio Fernández Cardador

Madrid

Junio de 2021

“One of the most telling examples of uses of AI in rail technology is its contribution to the automation of train operation (ATO)” (Nies19)

Agradecimientos

Quiero agradecer en primer lugar a Antonio, Paloma y Adrián su dedicación, su tiempo y su paciencia. Agradecer además a todos los compañeros y profesores del máster industrial y ferroviario el estar ahí día tras día y semana tras semana.

También tengo que agradecer a mi familia su comprensión durante todos estos años. De un modo especial a Fernando, por su ejemplo y su apoyo, aunque sobre todo a Carmen, sin ella no tendríamos en la familia a una persona extraordinaria.

MODELADO DE LA MARCHA Y DEL CONSUMO DE TRENES METROPOLITANOS DE CONDUCCIÓN AUTOMÁTICA MEDIANTE TÉCNICAS DE MACHINE LEARNING

Autor: Rodríguez Pita, Sofía.

Director: Fernández Rodríguez, Adrián.

Director: Fernández Cardador, Antonio.

Entidad Colaboradora: ICAI – Universidad Pontificia Comillas

RESUMEN DEL PROYECTO

En este trabajo, el objetivo es la utilización de machine learning para la predicción de parámetros de marcha de trenes equipados con sistemas de conducción automática ATO. Se busca obtener modelos con errores muy reducidos, que se puedan utilizar como ayuda al diseño eficiente de marchas del ATO, y que mejoren el tiempo de cálculo de los modelos tradicionales basados en simulación. En particular se han desarrollado y validado los siguientes modelos: un modelo de predicción de tiempos de recorrido y consumos, y a continuación, redes neuronales para modelar el comportamiento de un ATO, que posteriormente se integran en un simulador, lo que permite un análisis más detallado de las marchas obtenidas con el modelo anterior.

Palabras clave: ATO, IA, red neuronal

1. Introducción

La necesidad de descarbonizar el transporte hace del ferrocarril un medio muy atractivo. Siendo 2021 el año europeo del ferrocarril se puede esperar un importante avance y desarrollo de este sector. En este contexto se enmarca el presente trabajo, que explora las posibilidades de utilizar la inteligencia artificial para la optimización de la conducción automática ATO en trenes metropolitanos.

Los esfuerzos de la UE para la implantación de sistemas ATO a nivel europeo facilitarán una operación más eficiente desde el punto de vista energético. El ATO automatiza la conducción, haciendo que todos los trenes aceleren y frenen de forma preestablecida. Al tener formas fijas de recorrer la distancia entre dos estaciones, se puede predecir y optimizar su tiempo de recorrido y su consumo.

Incorporar un sistema de conducción automática ATO tiene importantes beneficios. El principal es que permite mejorar la operación de tráfico: puntualidad y regularidad del servicio prestado al viajero. En estaciones en las que hay puertas de andén es imprescindible, ya que necesitan precisiones de 0.2m en la parada (Mont17). Además, otra ventaja importante es el ahorro de energía que consigue, por ejemplo, en Metro de Madrid ahorros del orden de un 18% (Cuca10).

La utilización de la IA para temas relacionados con la conducción se puede ver en (Cast16) y en (Wang20). Otros usos de la misma, ya desarrollados e implantados, incluyen la visión artificial para la detección de obstáculos.

2. Definición del Proyecto

Obtención de modelos de machine learning de predicción de tiempos y consumos de marchas ATO, definidas a partir de sus consignas de conducción. Estos modelos facilitan seleccionar las marchas más eficientes en función del tiempo de recorrido requerido. Sus tiempos de cálculo tan pequeños permiten eliminar de entrada las alternativas menos eficientes, ayudando en el proceso de diseño de marchas y regulación del tráfico. Estos modelos son validados con datos reales de circulación de trenes.

Conjuntamente, modelado de un sistema ATO a través de redes neuronales, para el análisis detallado de las marchas obtenidas utilizando el modelo anterior. Integración de este modelo ATO en un simulador de la dinámica de la marcha de un tren, y cálculo de errores esperables en la conducción por parte del modelo.

Ambos modelos se utilizarán para agilizar las simulaciones realizadas actualmente, y para mejorar el modelado de los algoritmos de conducción ATO.

Finalmente, diseño de interfaces de usuario que posibiliten el uso de dichos modelos de forma independiente.

3. Descripción del modelo/sistema/herramienta

En la Figura R 1 se puede ver un esquema de los puntos desarrollados (en verde). Los modelos se dividen en dos grupos fundamentales: predicción de tiempos y consumos de recorrido, y predicción de salidas del ATO.

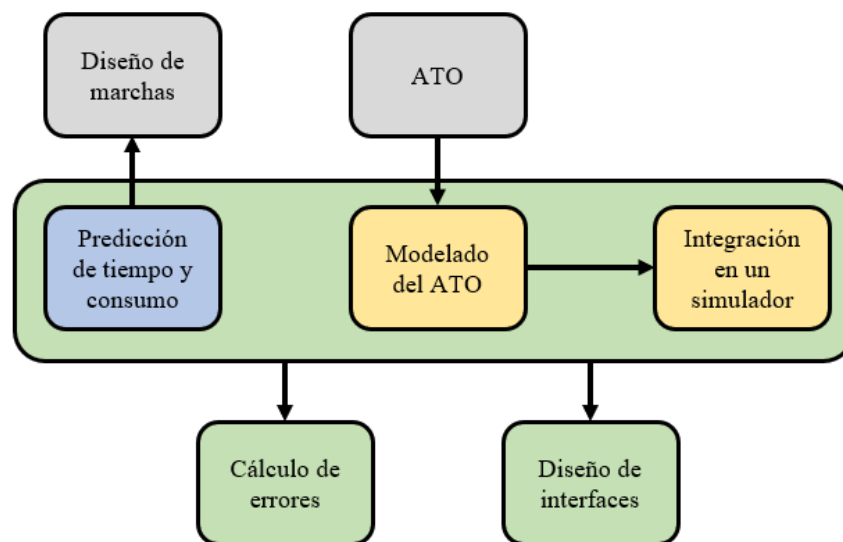


Figura R 1: Esquema de los puntos desarrollados en el trabajo y su relación

El modelo de predicción de tiempo y consumo se utiliza como función de evaluación en el diseño de marchas de conducción. Las marchas son las distintas formas que tiene el equipo de ATO de recorrer la distancia entre estaciones. De su optimización depende el ahorro de energía conseguido.

El modelo de ATO reproduce la funcionalidad de un sistema de conducción automática real con toda su complejidad. Gracias al gran volumen de datos reales de circulación de trenes disponibles, ha sido posible entrenar redes neuronales que replican el comportamiento entrada-salida de este equipo. Los datos de entrenamiento corresponden a los medidos en un tren recorriendo la Línea 1 de Metro de Barcelona.

Además, se han analizado las marchas que se generan cuando conduce un modelo ATO en lugar del sistema de conducción real. Para ello se ha desarrollado un simulador de la marcha del tren. Se han integrado en él los modelos de ATO obtenidos, evaluándolos frente a registros reales de marcha. Para tratar de reducir al máximo los errores cometidos por el modelo se han desarrollado varias soluciones:

- Una única red neuronal
- Dos redes neuronales, una para predecir la parte de tracción y otra la parte de freno
- La incorporación de una red neuronal adicional que prediga la curva de freno cuando el tren está cerca de la estación de destino

De cara a una utilización más cómoda se han diseñado interfaces de usuario para hacer las predicciones y simulaciones. En las interfaces se introducen los datos de entrada para la red neuronal y ésta predice el resultado, que se muestra por pantalla.

Se ha utilizado el lenguaje de programación Python en todo el proyecto. Dentro del machine learning el trabajo se centra especialmente en redes neuronales, aunque en la parte de predicción de tiempos y consumos se han analizado también otros algoritmos. De los estudiados, la red neuronal ha sido la que mejor se ha comportado, con la ventaja adicional de que los tiempos de ejecución son muy pequeños.

4. Resultados

- La predicción de tiempo y consumo entre dos estaciones con una red neuronal se puede ver en la Figura R 2. El error cometido en la predicción del tiempo es de un 0.04%, y en la predicción del consumo de un 0.34%. De entre los cinco tipos de modelos, y de media veinte modelos de cada tipo, esta red neuronal fue la que tuvo un mejor comportamiento. A continuación se detalla el proceso de selección de esta red neuronal.

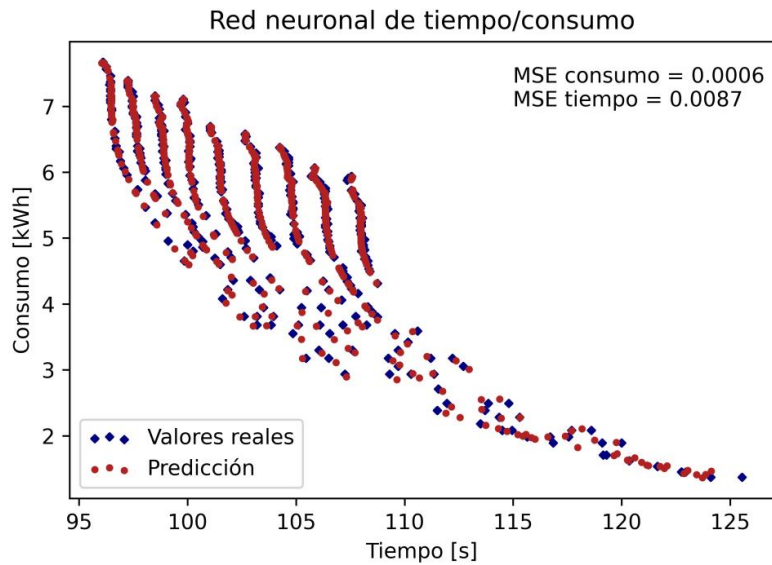


Figura R 2: Predicciones obtenidas de tiempo y consumo

Tras la obtención de errores con todos los algoritmos, se hizo una comparación de la red neuronal y el *RandomForest*. Estas opciones son las que presentaron menos error. La comparación analiza la capacidad de generalización de los modelos. De los nueve valores discretos que puede tomar la entrada ‘Velocidad’ (que controla la velocidad máxima que puede alcanzar el tren), se omitió uno de ellos en el entrenamiento. Los errores cometidos por ambos modelos al predecir el tiempo y el consumo de las marchas que utilizan como consigna de conducción la ‘Velocidad’ omitida se pueden ver en la Tabla R 1.

Tabla R 1: Comparación de la red neuronal y el *RandomForest*

	Tiempo				Consumo			
	MSE	MAE	%	Max	MSE	MAE	%	Max
Red neuronal	0.082	0.2118	0.2060	2.9526	0.0129	0.1031	1.6906	0.2355
RandomForest	1.8266	1.3410	1.3204	1.5500	0.1023	0.3046	4.8645	0.4823

En la Figura R 3 se ve la predicción hecha por ambos algoritmos. Las entradas omitidas en el entrenamiento se muestran en color negro, siendo estos valores los que se pide que predigan los modelos. La red neuronal es capaz de generalizar y predecir resultados cercanos a los reales, mientras que el *RandomForest* se limita a predecir la fila más cercana.

El modelo elegido finalmente para la predicción de tiempos y consumos ha sido la red neuronal, que presenta los errores comentados anteriormente.

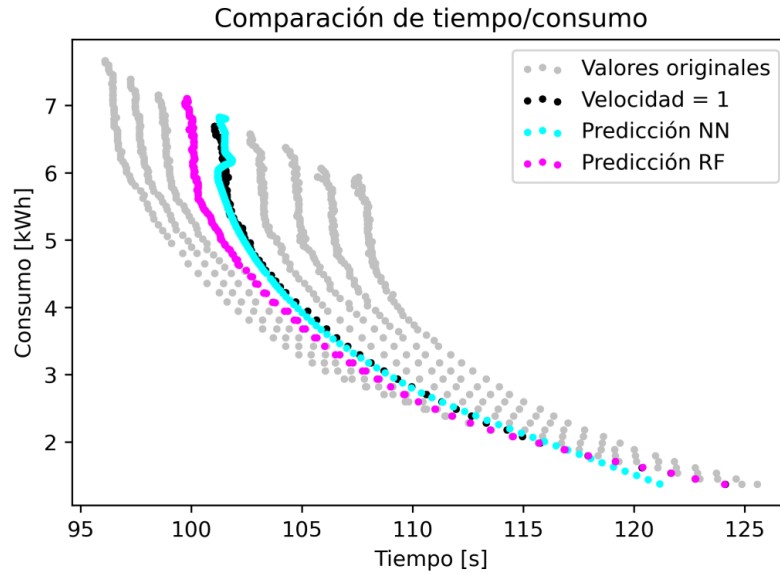


Figura R 3: Comparación de la red neuronal y el RandomForest

- Los modelos de ATO obtenidos se evaluaron mediante la programación de un simulador, que permite analizar las marchas generadas. Se analizaron modelos de 4 tipos de combinaciones estudiadas, escogiendo de entre las 85 redes neuronales las que tenían errores relativamente más bajos. La comparativa, frente a registros reales de velocidad, del recorrido de una interestación se puede ver en la Figura R 4. Se incluyen las salidas de ATO (consigna de aceleración objetivo) asociadas a esta misma interestación en la Figura R 5.

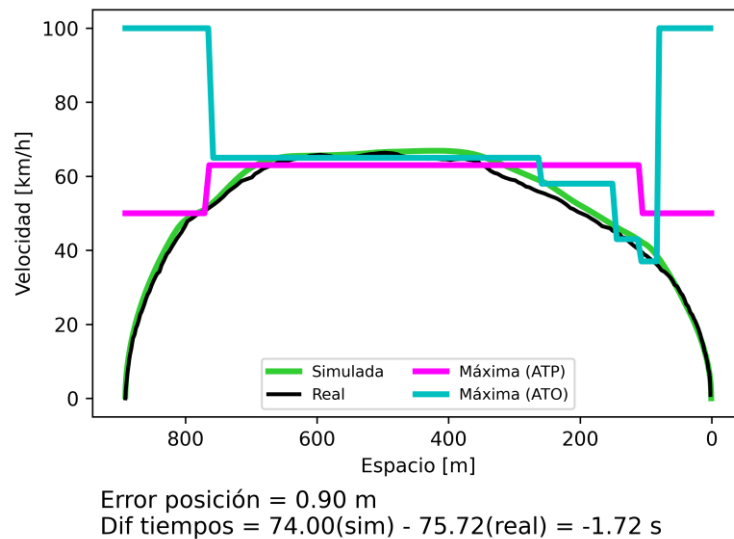


Figura R 4: Comparación de velocidades en una simulación

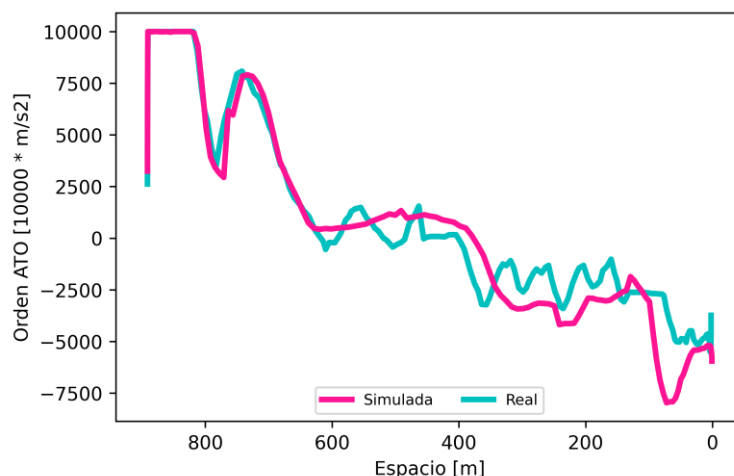


Figura R 5: Salida de ATO simulada comparada con la real en una simulación

En este apartado se estudiaron varias alternativas, desde una única red neuronal para hacer las predicciones, hasta el uso de varias redes neuronales. Una variante utilizaba dos redes neuronales, una que predice las salidas de tracción, y otra que predice la parte de freno. Una evolución posterior, también con dos redes neuronales, utiliza una red principal para predecir durante toda la simulación, y una red auxiliar para predecir el comportamiento del ATO cerca de la estación de destino.

El error medio esperado de posicionamiento en el punto de parada en la estación está entre un 1% y un 6%. El error de tiempo está entre un 0.1% y un 8%. Esta variación tiene en cuenta todos los modelos entrenados.

El modelo que ha obtenido menor error medio es el que combina una red neuronal general (que predice tanto tracción como freno) con una red de curva de freno. Estos errores son de un 0.1% en tiempo y un 1% en distancia al punto de parada. Estos valores de error de tiempo están en el rango de otros simuladores, 1% en (Domí11) y 0.16% en (Jong05).

Este modelo, sin embargo, no es el que tiene menor error máximo individual, siendo de 1.6m respecto al punto de parada. Con otros modelos se han obtenido precisiones de 0.2m, aunque con errores medios más altos.

En cuanto a la alineación con los Objetivos de Desarrollo Sostenible, el trabajo está relacionado con el ODS 11 (Ciudades y comunidades sostenibles) y 12 (Producción y consumo responsables).

El transporte sostenible es clave para cumplir con los objetivos de emisiones de la UE, que está fomentando el uso del ferrocarril sobre otros medios más contaminantes. Para incentivar el uso de este medio de transporte son importantes las mejoras que se pueden conseguir gracias al ATO: mayor capacidad de transporte y regularidad del servicio, y menor consumo de energía.

Dentro del *European Green Deal* se busca que el uso de los recursos sea más responsable. Gracias al ATO se puede predecir con más facilidad el mantenimiento a realizar. Elementos

que duran más y son más fáciles de reparar ayudan a cumplir el *Circular Economy Action Plan*, y acercan a un consumo más responsable.

5. Conclusiones

Los errores obtenidos en la predicción de tiempo/consumo hacen que estos modelos se puedan utilizar de forma fiable en el tramo de vía en el que han sido ajustados. Para otros tramos habría que adaptar el modelo con las soluciones que se proponen en el Capítulo 9. . En un tramo donde hubiera pendientes muy diferentes, por ejemplo, este modelo podría presentar desviaciones más altas con respecto a los valores que se medirían de tiempo y consumo.

De querer entrenar varios modelos para utilizarlos en situaciones diferentes, el *RandomForest* resulta más atractivo. Tiene un tiempo de entrenamiento de 7s frente a los 6.5 min que necesita la red neuronal. Sin embargo, el *RandomForest* se debe usar con cuidado, ya que como se ha comentado tiene problemas para generalizar. La red neuronal es más segura en este aspecto, por lo que debería ser la solución adoptada por defecto.

En el caso del ATO, el objetivo era modelar el equipo ATO embarcado en el tren mediante una red neuronal. Al comparar los resultados de la simulación con las medidas del ATO real se observó un error en el tiempo de recorrido finalmente ejecutado. También que hay algunas interestaciones que se ajustan al punto de parada, consiguiendo las precisiones de entre 0.5m y 0.2m que se le piden a un ATO (Mont17). El error más bajo obtenido en el posicionamiento es de un 1%, y en el tiempo de recorrido de un 0.1%. Los errores conseguidos son razonables para esta aplicación, estando en el rango de otros simuladores existentes basados en simulación detallada.

Los modelos desarrollados, tanto el que se centra en predecir tiempos y consumos como el que predice salidas del ATO, podrán ayudar en el proceso de diseñar las marchas del ATO. Gracias a su rapidez de predicción y precisión, permitirán seleccionar las marchas más eficientes.

6. Referencias

(Cast16) Casteel, Martijn. «Simulation of train driver behaviour using a data-driven agent.» Master's thesis, Utrecht University, Faculty of Science - Department of Information and Computing Sciences, 2016.

(Cuca10). Cucala, Paloma; Fernández, Antonio. Energy efficiency in high-speed train operation. *Global Railway Review*.

(Domí11). Domínguez, M.; Fernández-Cardador, A.; Cucala, A. P.; Lukaszewicz, P.. Optimal Design of Metro Automatic Train Operation Speed Profiles for Reducing Energy Consumption. *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit*, 225, 463-474.

(Jong05). Jong, J. and Chang, S. Algorithms for generating train speed profiles. *Journal of the Eastern Asia Society for Transportation*, 5. doi:10.11175/easts.6.356

(Mont17) Montes, Fernando. *Los Sistemas de Control de Tráfico y Señalización en el Ferrocarril*. Biblioteca Comillas. Ingeniería, 2017.

(Wang20) Wang, Ziran, y otros. «Driver Behavior Modeling Using Game Engine and Real Vehicle: A Learning-Based Approach.» *IEEE Transactions on Intelligent Vehicles* 5 (11 2020): 738-749.

MODELLING THE RUNNING TIME AND ENERGY CONSUMPTION OF AUTOMATIC DRIVING METRO TRAINS THROUGH MACHINE LEARNING TECHNIQUES

Author: Rodríguez Pita, Sofía.

Supervisor: Fernández Rodríguez, Adrián.

Supervisor: Fernández Cardador, Antonio.

Collaborating Entity: ICAI – Universidad Pontificia Comillas

ABSTRACT

In this project, machine learning is used to predict train running parameters of trains equipped with automatic driving system ATO. Models with very low errors are sought, to be used as an aid to efficient design of ATO profiles, and that improve the calculation time of traditional models based in simulation. In particular, the following models have been developed and validated: a first model of running times and consumption prediction; and then neural networks to model the behaviour of an ATO, that will later be integrated in a simulator, which will enable a more detailed analysis of the profiles obtained with the previous model.

Keywords: ATO, AI, neural network

1. Introduction

The need to decarbonize the transport sector makes railways a very attractive means of transportation. With 2021 being the European year of rail, an important development can be expected in this sector. This work is framed within this context, exploring the possibilities of using artificial intelligence to optimize automatic driving ATO in subway trains.

The efforts of the EU to implement an ATO system in a European level will favour a more efficient operation energy-wise. The ATO automates driving, making all trains accelerate and brake in the same way. Because it has predefined ways of covering the distance between two stations, the travel time and energy consumption can be predicted and optimized.

Incorporating an ATO has important benefits. The main one being that it enables an improvement of traffic management: being on time and regularity in the service provided to the commuter. In places where there are platform screen doors it is vital, because a precision of 0.2m is required when stopping before them (Mont17). Additionally, other important benefit is the reduction in energy consumption achieved, for example, in Metro de Madrid savings of around an 18% (Cuca10).

The use of AI for driving related issues can be seen in (Cast16) and in (Wang20). Other uses, already developed and implemented, include artificial vision for obstacle detection.

2. Project definition

Development of machine learning models to predict travel time and consumption of ATO profiles, defined through their driving commands. This model will enable the selection of the most efficient profiles regarding the required travel time. Their small calculation time makes it possible to reject from the start less efficient alternatives, helping in the process of

designing profiles and traffic regulation. These models are validated with real data of trains' driving.

Together with this, modelling of an ATO system through neural networks, to analyze in detail the profiles obtained using the previous model. Integration of this ATO model in a simulation tool of the dynamic behaviour of a train and calculation of the expected driving errors of this model.

Both models will be used to speed up the simulations currently run, and to improve the modelling of the ATO driving algorithms.

Finally, design of user interfaces to enable the use of these models independently.

3. Description of the model

In Figure 1 a diagram of the topics covered in this work has been included, seen in green. The models are divided into two differenced groups: prediction of time and consumption, and prediction of ATO outputs.

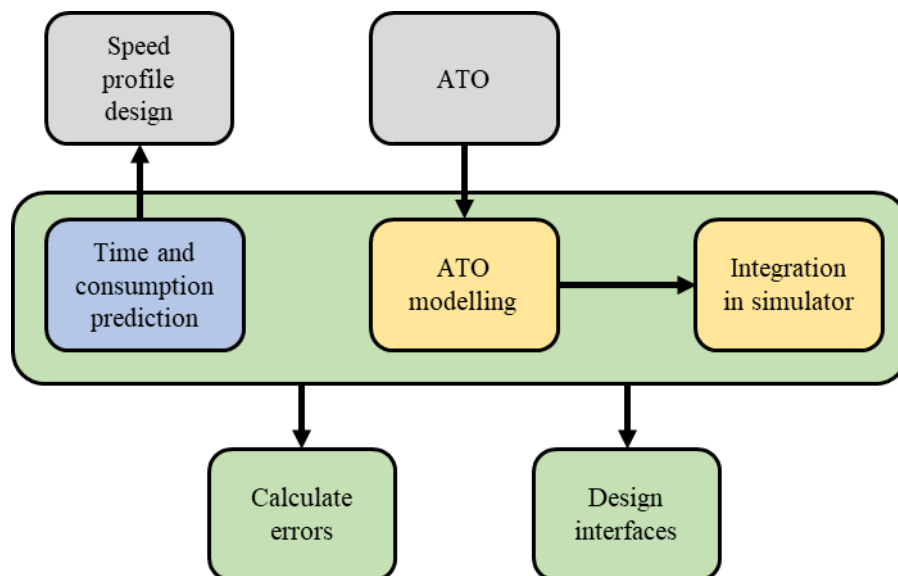


Figure 1: Diagram of topics covered in this project

The time and consumption prediction part is used as an evaluation function in the design of speed profiles. These profiles are the different ways the ATO system has of covering the distance between stations. The energy consumption savings depend on the optimization of this profiles.

The ATO model encompasses the whole ATO inner functioning, and all its complexity. Thanks to the vast amount of data available it has been possible to train neural networks that replicate the input-output behaviour of this system. The data used for training corresponds to the measured data of a train running in L1 of Metro de Barcelona.

When developing these ATO models, it is necessary to evaluate them when driving. To do this, a simulation tool has been developed, to integrate in it the ATO model and evaluate the different models obtained. To try and minimize the errors committed by the models, different solutions have been developed:

- A single neural network
- Two neural networks, one for predicting the traction outputs and another the brake ones.
- Incorporation of an additional neural network to predict the brake curve when the train is closer to the station.

To enable a more comfortable use, user interfaces have been developed to make the predictions and simulations. In the interfaces, the input data for the neural network is introduced, and said network predicts the result, that is shown on the screen.

Programming language Python has been used all throughout the project. Inside machine learning, this project is centred around neural networks, although in the time and consumption prediction part other algorithms have also been analyzed. Of the studied ones, the neural network has been the one with the best behaviour.

4. Results

- The predictions of travel time and consumption with a neural network can be seen in Figure 2. When predicting time, the error incurred in is of 0.04%, and in the consumption prediction a 0.34%. Of the five types of models, and the mean of twenty models per type, this neural network was the one who behaved better. The process of selecting the neural network is shown below.

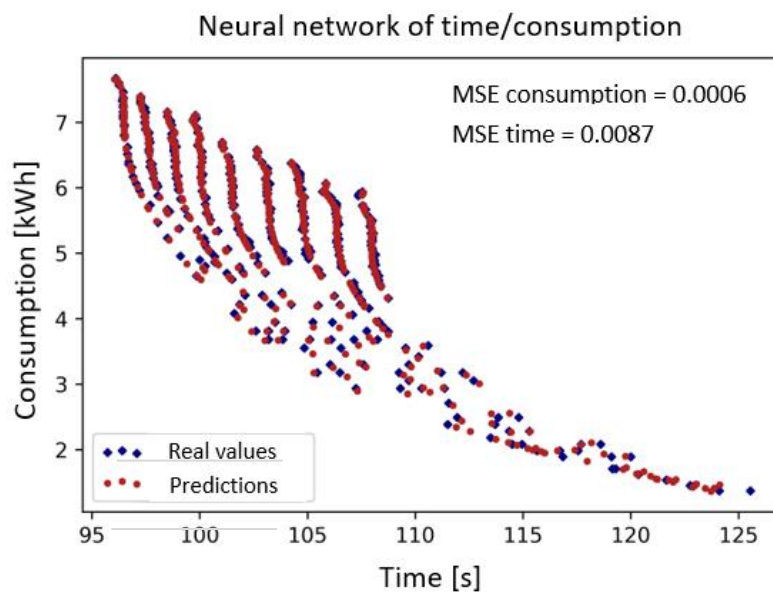


Figure 2: Predictions of time and consumption

After calculating errors of all algorithms tested, a comparison was made between the neural network and the *RandomForest*. These options were the ones with less error. The comparison analyzes the generalization capacity of both models. Of the nine discrete values that the input 'Speed' can take (value that controls the maximum speed the train can achieve), one was omitted from the training data set. The errors incurred by both models when predicting the omitted 'Speed' can be seen in Table 1.

Table 1: Comparison between a neural network and RandomForest

	Time				Consumption			
	MSE	MAE	%	Max	MSE	MAE	%	Max
NeuralNetwork	0.082	0.2118	0.2060	2.9526	0.0129	0.1031	1.6906	0.2355
RandomForest	1.8266	1.3410	1.3204	1.5500	0.1023	0.3046	4.8645	0.4823

In Figure 3 the prediction of both algorithms can be seen. The inputs omitted from the training data set are shown in black, being these values the ones the models are asked to predict. The neural network is able to generalize and predict values close to the real outputs, while the *RandomForest* simply predicts the closest line.

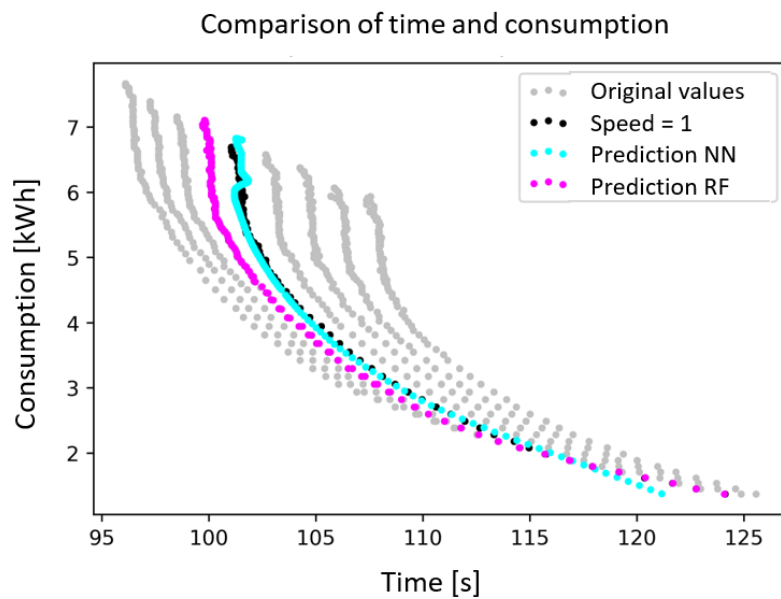
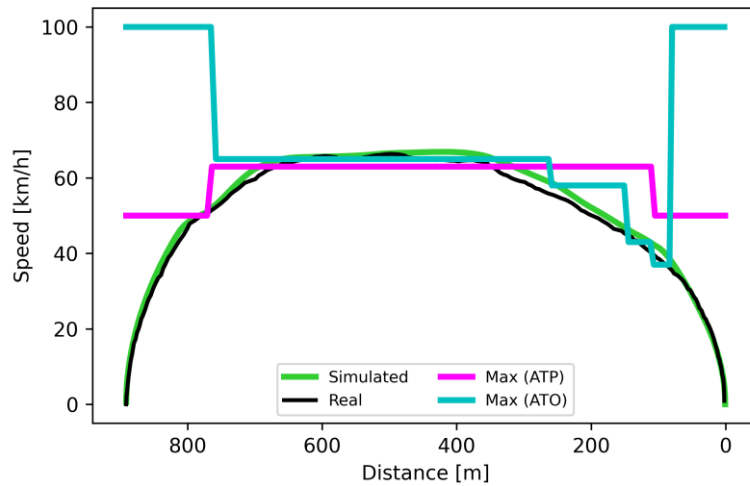


Figure 3: Comparison of the neural network and RandomForest

The model finally selected for the prediction of times and consumptions has been the neural network, characterized by the errors included previously.

- The ATO models were evaluated through the programming of a simulator, which enables an analysis of the generated profiles. Models of the four types of combinations studied were analyzed, choosing from the 85 neural networks the ones with relatively smaller errors.

The comparison when travelling between two stations can be seen in Figure 5. The ATO outputs associated to this same section are included in Figure 4.



Position error = 0.90 m
 Time difference = 74.00(sim) - 75.72(real) = -1.72 s

Figure 5: Comparison of speeds in a simulation

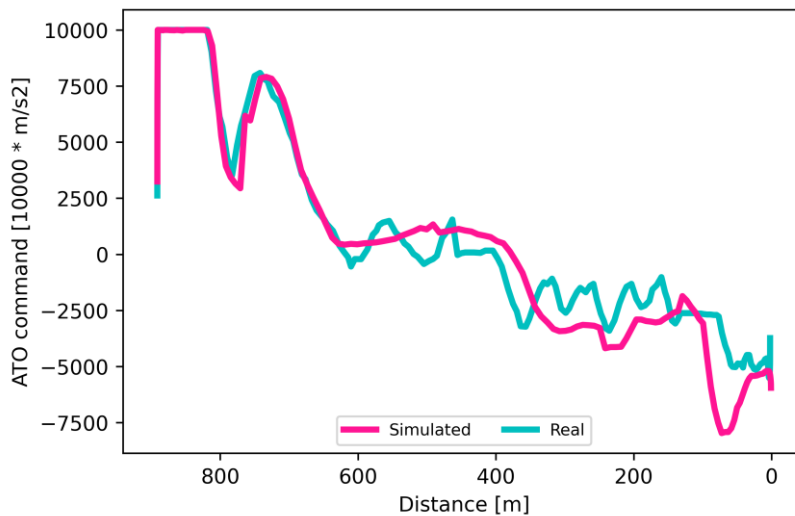


Figure 4: Simulated ATO output compared to the real one

The mean error expected when stopping at the station is between 1% and 6%. This variation is associated to the different types of models trained. The time error is between 0.1% and 8%. These values are similar to other simulators, 1% in (Domí11) and 0.16% in (Jong05).

In this section, various alternatives were studied, from a single neural network to predict, to the use of more than one neural network. An alternative used two neural networks, one to predict traction outputs, and another to predict the braking part. A further development, also with two neural networks, uses a main network to predict throughout the simulation, and an auxiliary network to predict the ATO's behaviour close to the arrival station.

The model that has achieved a smaller mean error is this last one, that combines a general neural network (that predicts both traction and brake) with a brake curve network. Errors were 0.1% in travel time and 1% in distance to the stop point.

This model, however, is not the one with the smallest individual error, being 1.6m to the stop point. With other models precisions of 0.2m have been achieved, although with higher mean errors.

Regarding the alignment with the Sustainable Development Goals, this project is related to SDG 11 (Sustainable cities and communities) and SDG 12 (Responsible consumption and production).

Sustainable transportation is key to meet the emissions targets of the EU, who is promoting the use of the railway over other more polluting means. To encourage the use of this means of transport, the advantages when using an ATO are important: more capacity and regularity and less energy consumption.

A more responsible use of resources is framed within the *European Green Deal*. Thanks to the ATO, it is easier to predict the maintenance to be performed. Elements that last longer and are easier to repair help to achieve the *Circular Economy Action Plan*, and bring us closer to a more responsible consumption.

5. Conclusions

The errors calculated when predicting time and consumption make these models reliable in this track. For other tracks the model would have to be adapted with the solutions included in Capítulo 9. . In a distance where slopes are very different, for example, this model could show a higher deviation with respect to the time and consumption values that would be measured.

With a view to training a set of models to be used in different situations, the *RandomForest* is more attractive. It takes 7s to be trained, opposed to the 6.5 min required by the neural network. However, the *RandomForest* should be used with precaution, it has trouble generalizing as has been mentioned. The neural network is safer in this aspect, so it should be the default solution.

In the case of the ATO, the objective was to model the onboard ATO through a neural network. When comparing the results of the simulations with the measurements taken with a real ATO, an error has been observed in travel time. Also, that there are sections that adjust well to the stop point, achieving the precisions between 0.5m and 0.2m demanded of a real ATO (Mont17). The lowest position error is of 1%, and in travel time a 0.1%. The errors obtained are reasonable for this application, being in the range of other existing simulators for this purpose.

Both models developed, the one that predicts times and consumptions and the one that predicts ATO outputs, will help in designing speed profiles. Thanks to their prediction speed and precision they will enable the selection of the most efficient profiles.

6. References

- (Cast16) Casteel, Martijn. «Simulation of train driver behaviour using a data-driven agent.» Master's thesis, Utrecht University, Faculty of Science - Department of Information and Computing Sciences, 2016.
- (Cuca10). Cucala, Paloma; Fernández, Antonio. Energy efficiency in high-speed train operation. *Global Railway Review*.
- (Domí11). Domínguez, M.; Fernández-Cardador, A.; Cucala, A. P.; Lukaszewicz, P.. Optimal Design of Metro Automatic Train Operation Speed Profiles for Reducing Energy Consumption. *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit*, 225, 463-474.
- (Jong05). Jong, J. and Chang, S. Algorithms for generating train speed profiles. *Journal of the Eastern Asia Society for Transportation*, 5. doi:10.11175/easts.6.356
- (Mont17) Montes, Fernando. *Los Sistemas de Control de Tráfico y Señalización en el Ferrocarril*. Biblioteca Comillas. Ingeniería, 2017.
- (Wang20) Wang, Ziran, y otros. «Driver Behavior Modeling Using Game Engine and Real Vehicle: A Learning-Based Approach.» *IEEE Transactions on Intelligent Vehicles* 5 (11 2020): 738-749.

Índice

Capítulo 1. Introducción.....	1
Capítulo 2. Descripción de las Tecnologías.....	4
2.1 Redes neuronales.....	4
2.2 Otros algoritmos.....	6
2.3 El ATO, centro del proyecto.....	8
Capítulo 3. Estado de la Cuestión.....	11
3.1 Regulación existente.....	12
Capítulo 4. Definición del Trabajo.....	15
4.1 Justificación.....	15
4.2 Objetivos.....	16
4.3 Metodología.....	16
4.4 Planificación y Estimación Económica.....	17
Capítulo 5. Características comunes de los modelos entrenados.....	19
Capítulo 6. Modelo de tiempo-consumo.....	21
6.1 Datos utilizados y estructura.....	21
6.2 Resultados de errores obtenidos.....	23
Capítulo 7. Modelado del ATO y simulador de marcha.....	27
7.1 Modelo de ATO.....	27
7.1.1 Conjunto 1: Modelo único para tracción y freno.....	29
7.1.2 Conjunto 2: Dos modelos.....	32
7.1.3 Conjunto 3: Tres modelos (clasificación + tracción + freno).....	34
7.1.4 Resultados de errores obtenidos.....	35
7.2 Simulador de marcha.....	36
7.2.1 Ecuaciones del movimiento.....	37
7.2.2 Estructura.....	39
7.2.3 Resultados obtenidos.....	41
Capítulo 8. Interfaces de usuario.....	46
8.1 Modelo de tiempo-consumo.....	46
8.2 Modelo de orden ATO.....	47
8.3 Simulador.....	49

Capítulo 9. Análisis de Resultados.....	51
Capítulo 10. Conclusiones y Trabajos Futuros	53
Capítulo 11. Bibliografía	55
ANEXO I: Desarrollo de los modelos	60
Modelos de tiempo/consumo	60
Modelos de ATO	62
ANEXO II: SMOTE (Synthetic Minority Over-sampling Technique)	64
ANEXO III: Alineamiento del proyecto con los ODS	66

Índice de figuras

Figura 1: Visión del tren de RailVision y SBB (Rail19).....	2
Figura 2: Funcionamiento de la red neuronal.....	5
Figura 3: Árbol de decisión para predecir el tiempo	7
Figura 4: Comportamiento en una interestación	9
Figura 5: Grados de automatización (S2R19)	13
Figura 6: Planificación temporal de las actividades	17
Figura 7: Comparación de las búsquedas de los diferentes algoritmos [Google Trends]	22
Figura 8: Modificación de los datos para cálculo de tiempo/consumo	23
Figura 9: Comparación de valores reales y predichos con el modelo de tiempo/consumo.	24
Figura 10: Comparación de generalización entre NN y RF	26
Figura 11: Variación del RandomForest para tratar de corregir el overfitting.....	26
Figura 12: Línea 1 de metro de Barcelona (TMBa20)	27
Figura 13: Recorrido del tren real en una interestación que presenta un tramo anómalo ...	30
Figura 14: Simulación en una interestación con datos reales anómalos.....	31
Figura 15: Tipos de modelo único en función de los datos de entrenamiento	31
Figura 16: Curva de esfuerzo máximo de tracción.....	37
Figura 17: Comparación de velocidades reales y predichas en una interestación.....	42
Figura 18: Comparación de salidas de ATO reales y predichas.....	42
Figura 19: Simulación de un tramo	45
Figura 20: Comparación de fuerzas en la simulación con y sin esfuerzo máximo	43
Figura 21: Interfaz para el cálculo de tiempo-consumo	46
Figura 22: Interfaz para el cálculo de la consigna de ATO.....	47
Figura 23: Gráfico de la pestaña de análisis	48
Figura 24: Pestaña para el análisis de datos dentro de la interfaz de ATO	48
Figura 25: Interfaz para realizar simulaciones	49
Figura 26: Gráfico combinado para el simulador.....	50
Figura 27: Árbol de decisión con una profundidad máxima de 3	61
Figura 28: Funcionamiento del algoritmo SMOTE (Fawc16)	64
Figura 29: Emisiones debidas a los diferentes medios de transporte (EU21)	66

Índice de esquemas

Esquema 1: Relación de conjuntos utilizados	28
Esquema 2: Conjunto 2.1: Dos modelos con tracción + freno	32
Esquema 3: Conjunto 2.2: Dos modelos con modelo de curva de freno.....	33
Esquema 4: Grupo ‘tres modelos’	35
Esquema 5: Funcionamiento del simulador	40

Índice de tablas

Tabla 1: Estimación económica del proyecto.....	18
Tabla 2: Errores cometidos en la predicción de tiempo y consumo.....	23
Tabla 3: Comparativa de errores en predicciones de tiempos y consumos.....	24
Tabla 4: Comparación de generalización entre NN y RF.....	25
Tabla 5: Errores obtenidos al entrenar los modelos de ATO.....	35
Tabla 6: Errores del modelo de clasificación.....	36
Tabla 7: Errores obtenidos en las simulaciones con los diferentes modelos.....	41
Tabla 8: Comparación de la simulación con y sin esfuerzo máximo para todos los modelos.....	43
Tabla 9: Comparación de la simulación con y sin esfuerzo máximo para el mejor conjunto.....	44

Abreviaturas

AdaGrad	Adaptive Gradient
Adam	Adaptive Moment Estimation
ADAS	Advanced Driver-Assistance Systems
ATO	Automatic Train Operation
ATP	Automatic Train Protection
ECMr	Error Cuadrático Medio relativo
EETC	Energy-Efficient Train Control
EETT	Energy-Efficient Train Timetabling
ENIA	Estrategia Nacional de Inteligencia Artificial
ERTMS	European Rail Traffic Management System
ETCS	European Train Control System
GoA	Grade of Automation
GUI	Graphical User Interface
IA	Inteligencia Artificial
kNN	k-Nearest Neighbours
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
MSE	Mean Squared Error
NN	Neural Network
ReLU	Rectified Linear Unit
SBB	Schweizerische Bundesbahnen
SGD	Stochastic Gradient Descent
SNCF	Société nationale des chemins de fer français
SVM	Support Vector Machine
UE	Unión Europea

Capítulo 1. INTRODUCCIÓN

Los trenes metropolitanos tienen un peso considerable en la movilidad española. Dentro del entorno urbano, el transporte de viajeros en metro supone el 40% de los desplazamientos (INE21). Dentro de estas líneas de metro, hay 39 km que están totalmente automatizados (UITP18), y numerosas líneas con un grado intermedio de automatización. Esta automatización se lleva a cabo gracias a un ATO (*Automatic Train Operation*), que lleva a cabo parte, o todas, las funciones del maquinista.

El ferrocarril se hace pieza clave del transporte para reducir las emisiones de gases de efecto invernadero. El transporte por carretera emite 100 gCO₂/pasajero·km, frente a los 30 que emite el ferrocarril (EuEA20). El año europeo del ferrocarril 2021 promueve el desarrollo de nuevas tecnologías como el ATO, que, entre otros beneficios, aumentan la eficiencia energética.

Al ser terrestre y guiado, un problema que tiene este medio de transporte es el coste de las infraestructuras, equipos, etc. Se requieren grandes inversiones que tienen que tener una justificación. Una solución es simular lo que va a ocurrir para detectar problemas, realizar análisis y predecir los resultados de la operación antes de hacer la inversión. También se pueden hacer simulaciones sobre infraestructuras existentes. Esto permite realizar pruebas con un coste mucho más bajo que las pruebas en vía.

Al poner en juego cantidades de dinero tan elevadas, los simuladores tienen que ser muy precisos. Esto los hace complejos y con tiempos de cálculo elevados cuando se quieren hacer optimizaciones, que requieren un gran número de evaluaciones.

El objetivo de este trabajo es la obtención de modelos de machine learning que reproduzcan el comportamiento de trenes conducidos mediante ATO y que permitan obtener resultados similares a los obtenidos en simuladores convencionales. La ventaja de las redes neuronales y otros tipos de modelos utilizados en este trabajo es que tienen tiempos bajos de predicción, entre 10 y 100 veces menores que los tiempos de simulación. Tanto las redes neuronales como la inteligencia artificial en general son ya una realidad en el mundo del ferrocarril. Los administradores de infraestructuras y los operadores están empezando a utilizar IA de forma generalizada.

A través de una red neuronal, por ejemplo, se pueden identificar los elementos que se encuentra el tren en su marcha. La empresa Thales tiene un proyecto en Karlsruhe con trenes que “ven” e identifican obstáculos (Thal19). Rail Vision y SBB tienen un sistema parecido en Gossau (Rail19). Para trenes o metros que se muevan por la superficie es necesario poder detectar obstáculos en la vía.

En la Figura 1 se ha incluido lo que ve el tren de Rail Vision y SBB. Es capaz de detectar obstáculos, así como de anticiparse al movimiento de personas en las inmediaciones de la vía.



Figura 1: Visión del tren de RailVision y SBB (Rail19)

En este contexto la UE está desarrollando nueva regulación. En un entorno en el que prima la seguridad, esta normativa ha de encontrar un equilibrio con el aspecto económico. La regulación debe contemplar que algunas de estas soluciones están implantadas y funcionan, y no cortar la innovación y desarrollo en curso.

El Horizonte Europa, que da continuidad al Horizonte 2020, fue propuesto en 2018 y abarcará de 2021 a 2027. El Pilar II (*Global Challenges and European Industrial Competitiveness*) incluye un apartado de movilidad, reforzado por una de las áreas de desarrollo que es '*Climate-neutral and smart cities*'. Para lograr estos objetivos el ferrocarril será una pieza clave de la movilidad en las ciudades. La comparación de emisiones frente a la carretera se ha comentado anteriormente; y a escala europea, en 2015 el ferrocarril consumió el 2% de la energía final del sector transporte, transportando el 7.6% de los pasajeros, y suponiendo el 3% de las emisiones de CO₂ (UIC17).

La llegada del COVID-19 ha supuesto un cambio en la movilidad de las personas. Se aprecia una tendencia hacia el vehículo privado, principalmente bicicletas y coches en función de la distancia a recorrer. El uso masivo del coche entra en conflicto con el objetivo de la UE de ser *carbon neutral* en 2030. Los datos disponibles de contagio en el transporte público son alentadores. En Japón, Francia o Colorado la probabilidad de infectarse en un metro es del 0%. En Reino Unido es de un 0.005% (UITP20). El ferrocarril es seguro, pero será necesario un impulso en el sector. Gracias al año europeo del ferrocarril se pueden esperar medidas que promuevan el uso de este medio de transporte.

Aunque la población siga viajando en metro, se observa un cambio en la forma de hacerlo. Para evitar aglomeraciones habría que aumentar las frecuencias donde sea posible, pero para eso se necesitan más trenes y maquinistas, además de en ciertos casos aumentar la capacidad

de transporte máxima de las líneas ferroviarias. Tener un ATO GoA4 (*unattended*) facilitaría en gran medida una adaptación flexible a la demanda, ya que se podría variar el número de trenes sin restricciones externas, y se incrementaría la capacidad y la regularidad, mejorando la calidad del servicio.

Tener un tren autónomo supone que ha de estar más sensorizado. La IA aporta una gran capacidad de análisis de los datos que se generan, facilitando tanto la operación eficiente como el mantenimiento.

En este trabajo se obtendrán modelos de machine learning que permitan seleccionar la conducción óptima de los trenes en función de las condiciones del tráfico. También se desarrollan modelos que repliquen el comportamiento de un ATO, y se integran en un simulador para evaluar en detalle las conducciones seleccionadas con el modelo anterior. Adicionalmente se han diseñado interfaces para interactuar con cada una de estas soluciones.

Capítulo 2. DESCRIPCIÓN DE LAS TECNOLOGÍAS

El machine learning es una rama de las ciencias de la computación que se engloba en la inteligencia artificial. Busca obtener algoritmos que aprendan a través de un entrenamiento. Los modelos generados mediante estos algoritmos se pueden utilizar para hacer predicciones o clasificaciones. En casos en los que el código para resolver un problema es complejo, el machine learning puede aplicarse para obtener soluciones, con la ventaja de que puede ser reentrenado fácilmente para tener en cuenta nuevos casos.

En todo el trabajo se ha utilizado aprendizaje supervisado. Los algoritmos se entrenan viendo las entradas y las salidas correctas que les corresponden. Lo contrario sería aprendizaje no supervisado, donde al algoritmo se le dan entradas que no están clasificadas, para que busque patrones entre ellas.

Dentro del aprendizaje supervisado los modelos se pueden dividir en clasificaciones o regresiones. En este caso se entrenará a los algoritmos para hacer una regresión. Se trata de aprendizaje supervisado, por lo que al algoritmo se le dice cuál es la salida que corresponde a cada entrada, tiene que saber cuál es la relación entre ellas.

2.1 REDES NEURONALES

Dentro del machine learning una técnica muy habitual son las redes neuronales. Una red neuronal está formada por capas, y dentro de estas neuronas (o nodos). La salida se obtiene a base de ir avanzando por las neuronas. Cada neurona coge los inputs que le llegan y los combina para obtener una salida.

En la Figura 2 se ha incluido un diagrama del funcionamiento de la red neuronal, donde e son los valores que toman las entradas a la red neuronal, y w los pesos de cada una de las conexiones entre neuronas. En todos los pasos entre capas el funcionamiento es el mismo, la entrada a una neurona son las salidas de las anteriores, multiplicadas por los pesos y sumadas. Internamente la neurona obtiene la salida usando una función de activación, que puede ser diferente para cada capa. En la Figura 2 esta función de activación se representa con una a .

Los pesos de las conexiones marcan cómo de influyente es esa rama. Un peso más grande que otro quiere decir que esa entrada es más importante o influyente en esa neurona. Un peso pequeño o cero significa que esa entrada no afecta a la salida.

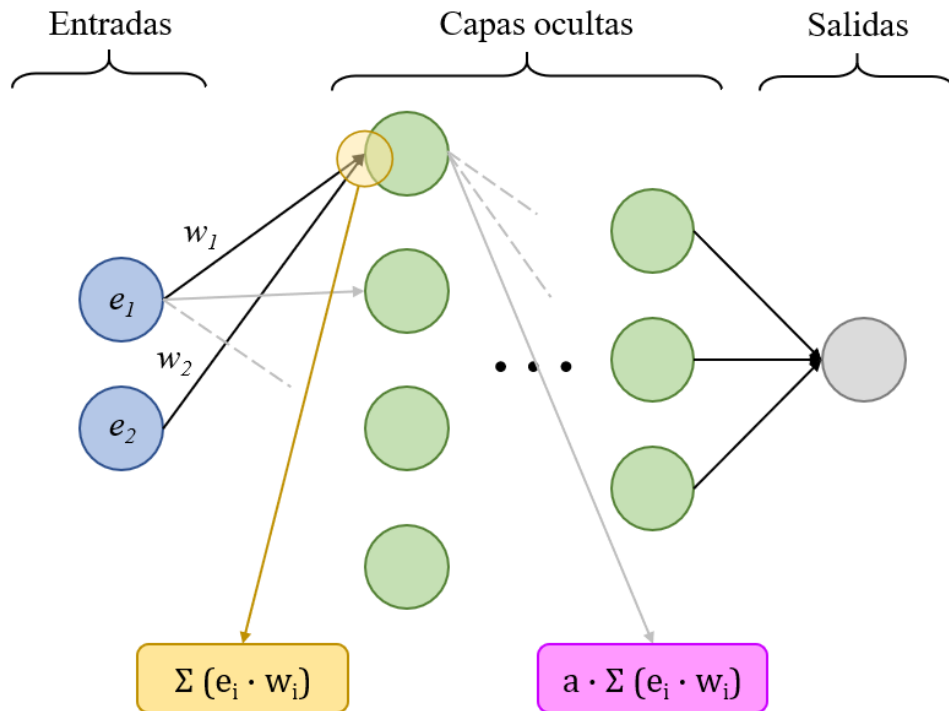


Figura 2: Funcionamiento de la red neuronal

En este trabajo las funciones de activación más utilizadas han sido *linear* y *ReLU*. Otras activaciones no resultaron en menores errores.

- *Linear activation*
La salida es igual a la entrada, puede ser positiva o negativa.
- *ReLU*
En este caso, sólo se puede usar en las capas internas de la red neuronal. Devuelve el máximo entre las entradas y 0. Se quieren predecir valores positivos y negativos, por lo que de usarla en la última capa se perderían los negativos.

La red aprende evaluando el error cometido y reasignando los pesos hasta que obtiene el error más bajo. El error cometido se calcula a través de una *loss*, que mide lo que se desvía la predicción del valor real, pudiendo penalizar más o menos las desviaciones más altas. En este caso se ha usado el *mean squared error (mse)*. El valor de la función *loss*, utilizando el *mse*, es

$$(\text{valor real} - \text{predicción})^2$$

En algunos modelos se utilizó el *mean absolute error (mae)*, con valores de errores parecidos. En el simulador los mejores modelos fueron los entrenados con *mse*. Las ecuaciones de ambos se pueden ver en el Capítulo 5. , estos errores también se utilizaron para evaluar de forma global los modelos entrenados.

El parámetro que une el error cometido con la actualización de los parámetros es el optimizador. Aunque se han utilizado varios, sobre todo *Stochastic Gradient Descent (SGD)* al principio, los que finalmente dieron menores errores fueron *Adam (King14)* y *AdaGrad (Duch10)*. Los optimizadores tratan de minimizar la función *loss* a través de su gradiente, dirigiéndose hacia los gradientes decrecientes.

Mientras que *SGD* utiliza una *learning rate* constante, en *Adam* y *AdaGrad* es variable y se adapta para encontrar el óptimo antes. La *learning rate* marca el ritmo al que los parámetros se actualizan. Si es alta se da un paso más grande hacia el gradiente decreciente, y puede que se pase del mínimo. Si es pequeña va más despacio, y puede que se quede en un mínimo local.

Un peligro de todos los modelos y algoritmos es el *overfitting*. Cuando esto ocurre el modelo no es capaz de generalizar. En el entrenamiento ha aprendido los datos “de memoria”, es decir, ha aprendido los datos de entrenamiento incluyendo el ruido que puedan tener incorporado. Para prevenir el *overfitting* se evalúan los modelos con datos que no se han utilizado para el entrenamiento (Capítulo 5.). También se realiza un análisis de la capacidad de generalizar de los modelos que predicen tiempos y consumos (Capítulo 6.).

Se han utilizado *feedforward neural networks*, las neuronas están conectadas a la capa anterior y posterior, y el flujo es unidireccional. Lo opuesto a una *feedforward neural network* sería una *recurrent neural network (RNN)*, en la que hay conexiones entre las neuronas de una misma capa y salidas que pueden ir a neuronas de capas previas. Las RNN se utilizan típicamente para el reconocimiento de voz y de caracteres. Son más complejas y por lo tanto suelen tardar más en entrenarse.

Esta herramienta se ha utilizado para modelar un ATO, aprovechando su ventaja de rapidez de predicción. Adicionalmente se han utilizado para predecir tiempos y consumos. Para la predicción de tiempos y consumos se han utilizado también los algoritmos que se incluyen a continuación.

2.2 OTROS ALGORITMOS

- DecisionTree

El primer algoritmo para hacer una regresión con un árbol de decisión apareció en 1963, bajo el nombre de AID (*Automatic Interaction Detection*) (Morg63). El nombre árbol de decisión se empezó a utilizar en 1984 (Brei84). Es un algoritmo que se utiliza tanto para regresiones como clasificaciones. En ambos casos, el árbol de decisión predice a base de hacer preguntas. La respuesta a la pregunta hace que el árbol se divida en ramas. Cuando se llega al final de la rama está la hoja, que es la predicción, y el árbol deja de dividirse.

En la Figura 3 se incluye un árbol de decisión simplificado para la predicción del tiempo de recorrido. Para que se aprecien los pasos en el gráfico se ha reducido la profundidad máxima

del árbol a 2. En el ‘ANEXO I: Desarrollo de los modelos’ se puede ver el gráfico de un árbol más profundo.

Si comparación es verdadera (velocidad ≤ 0.063), evoluciona hacia la izquierda. Y si es falsa hacia la derecha. En los nodos finales, u hojas, se encuentra la predicción se sacará el modelo. Si la (velocidad ≤ 0.063) y la (deriva es ≤ -0.475) el árbol predecirá que la salida para estos valores de entradas es un tiempo de 0.365.

Los árboles de decisión tienen la ventaja de que se puede ver fácilmente cómo llegan a la predicción. Se puede seguir el proceso para obtener cada una de las salidas y eso los hace más transparentes. Sin embargo, tienen problemas importantes como su tendencia al *overfitting*.

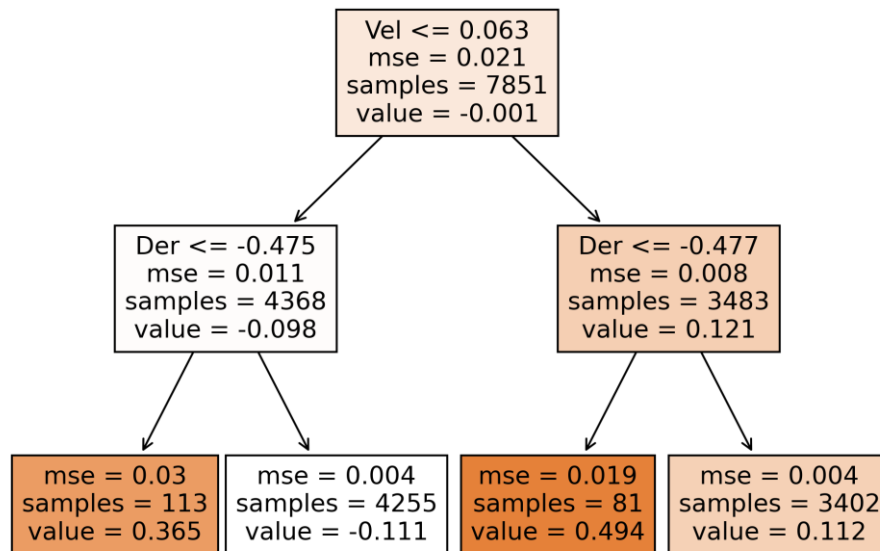


Figura 3: Árbol de decisión para predecir el tiempo

○ *RandomForest*TM (Brei01)

Es una variante del árbol de decisión, que intenta resolver el problema del *overfitting*. Entrena varios árboles y cada uno hace una predicción, siendo la salida la media de lo que han dicho los árboles. La idea es que, por la ley de los grandes números, si se hace la media de lo que dicen un gran número de árboles, la predicción va a estar más cerca del valor real. El *RandomForest* también se puede representar gráficamente. En este caso la figura que se obtiene es igual que la Figura 3. De querer representar todos los árboles juntos se obtendría una figura en la que aparecerían uno al lado del otro.

○ kNN

Es un algoritmo que apareció en 1951 para clasificación (Fix51), aunque el nombre de kNN no apareció hasta (Cove67), donde se usan los *nearest neighbours* también para una clasificación. Aunque el uso más frecuente es la clasificación, también se puede usar para regresión, e incluso para aprendizaje no supervisado.

El algoritmo tiene un mapa de todos los puntos de entrenamiento. Cuando le llegan las coordenadas del punto que tiene que predecir, lo ubica y obtiene sus k vecinos más próximos. El resultado que predice es el combinado de los vecinos. Se puede especificar que los que están más cerca contribuyan más a la solución que los que están más lejos.

Este algoritmo funciona bien en el rango de los datos de entrada. No es capaz de predecir fuera de esta zona.

2.3 EL ATO, CENTRO DEL PROYECTO

El ATO (*Automatic Train Operation*) es un sistema electrónico opcional del tren que permite automatizar todo o parte de la conducción del mismo. Las ventajas de usar un ATO se ven sobre todo en ferrocarriles suburbanos (Cercanías) y metropolitanos (metros). Entre ellas se encuentran la mejora del confort del viajero, una mejora de la puntualidad y de la capacidad de transporte, y la mejora en la eficiencia energética. Un ATO es especialmente fácil de implantar en redes cerradas tipo metro, en las que circula un único tipo de material rodante, y en las que no hay elementos como pasos a nivel, que necesitarían un mayor grado de inteligencia.

La implantación de un ATO necesita un trabajo de configuración de la conducción a realizar. En general tiene preconfiguradas cuatro marchas, o perfiles de velocidad, que son las posibles formas de conducir el tren en un tramo entre dos estaciones, ejecutándose la que le indican desde el centro de control. Estas marchas van de la más rápida a otras más lentas, en las que se ahorra energía. El diseño óptimo de estas marchas es fundamental. Metro de Madrid en hora punta mueve 325 trenes (MdeM20). Cualquier mejora en el consumo, por pequeña que sea, tiene un gran impacto, Metro de Madrid ha conseguido ahorros a través de la optimización de la conducción del orden de un 18% (Cuca10).

A lo largo del recorrido, la velocidad puede estar limitada por el sistema ATP (*Automatic Train Protection*) o el sistema ATO. La velocidad máxima del ATP viene condicionada por la seguridad, de forma que si el tren excede su velocidad máxima este sistema ordena freno de emergencia. La del ATO está relacionada con el confort de los pasajeros. La velocidad máxima del ATO no suele afectar a la marcha, siendo superior a la del ATP. Hay zonas, sin embargo, en las que para tener mayor confort de los pasajeros se reduce la velocidad del ATO. Son tramos en los que se podría ir más rápido de forma segura, pero los pasajeros perderían confort.

Para poder anticiparse a las reducciones de velocidad, el ATO conoce la velocidad y distancia objetivo. Representan el próximo cambio de velocidad que verá el tren, ya sea un aumento o reducción. La velocidad objetivo es la próxima que se encontrará el tren, y la distancia objetivo indica la distancia en metros al próximo cambio de velocidad que se encontrará el tren. Hay una para ATP y otra para ATO, aunque al llegar al final de la línea toman valores por defecto ya que no hay una velocidad siguiente.

El ATO conoce como frena el tren, y calcula una curva de freno espacio-velocidad que debe seguir el tren para cumplir con el objetivo de velocidad en cada momento.

En la Figura 4 se puede ver como el ATO acelera cuando aumenta la velocidad máxima permitida y como hace una curva de freno para cumplir con la reducción. Al llegar a la estación también hace una curva de freno, esta vez independiente de la velocidad máxima.

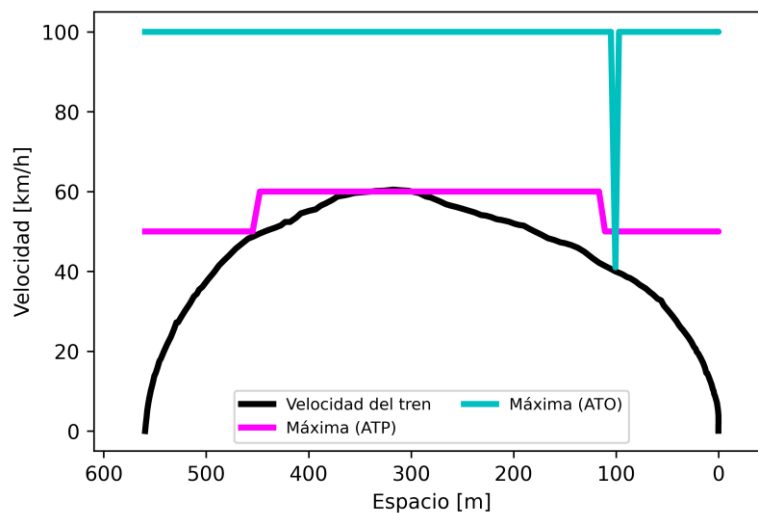


Figura 4: Comportamiento en una interestación

El ATO hace que todos los trenes frenen y aceleren igual. Es más homogéneo y se pueden tener aceleraciones de frenado más altas sin tener que sacrificar el confort de los pasajeros. Indirectamente también mejora el mantenimiento, ya que al frenar de forma más homogénea es más fácil controlar el desgaste.

Como conoce la posición de los trenes y cuánta distancia necesita para frenar, puede hacer que los trenes vayan más cerca unos de otros, aumentando la capacidad (trenes/h). También permite reducir el consumo de energía, ya que puede acelerar sólo cuando es necesario.

Un buen ejemplo es la línea C5 de Cercanías de Madrid, que gracias a un ATO consigue tiempos de recorrido muy estables. Aunque la inversión en equipos de ATO embarcados y

de vía es importante, los incrementos de capacidad y los ahorros de energía hacen que un año o año y medio se recupere la inversión realizada para instalarlo.

En este trabajo se predecirán valores de tiempo de recorrido y consumo esperados de conducciones ATO en función de los parámetros de conducción del ATO a seleccionar para el diseño de perfiles de velocidad. También se modelarán los algoritmos de conducción de un ATO real para ser integrados en un simulador de la dinámica del tren. Todo ello utilizando las técnicas de inteligencia artificial que se han comentado.

Capítulo 3. ESTADO DE LA CUESTIÓN

Existen diversas aplicaciones de la IA en el entorno ferroviario. Por ejemplo, (Wang20) utiliza redes neuronales y el algoritmo kNN para predecir los errores que tendrá un maquinista. El modelo de kNN clasifica a los maquinistas según su estilo de conducción para saber qué red neuronal aplicarles. Se centra en ADAS (*Advanced Driver-Assistance Systems*), que no llega a ser un ATO¹. Un ADAS es un sistema que ayuda a los maquinistas, mostrándoles un perfil de velocidad y cuándo tienen que acelerar. Como las personas no se ajustan perfectamente a lo que recomienda ADAS, se van acumulando errores. La red neuronal se utiliza, en este caso, para predecir las desviaciones que tendrá el maquinista y actualizar el perfil de velocidad que se le muestra.

Combinando redes neuronales con cálculos de tiempos y consumos, en (Mart19) se utiliza una red neuronal para calcular el consumo. En él se trata el ATO como un elemento a parte, que forma parte de las entradas a un simulador. Del simulador se obtiene el tiempo de recorrido y las entradas a la red neuronal, que se utiliza para calcular el consumo del tren a partir de su perfil de velocidad. Con esta red neuronal se obtienen errores medios de un 1,5% en el consumo.

(Cast16) también modeliza el comportamiento de un maquinista para utilizarlo en un simulador, aunque no hace referencia al tipo de modelo utilizado (desde el punto de vista de si es una red neuronal, etc.).

A parte de para modelar la conducción, las redes neuronales en el ámbito ferroviario se están utilizando para predecir retrasos, hacer mantenimiento predictivo, analizar video de cámaras de vigilancia o predecir el asentamiento de terraplenes.

Hay un gran interés por el *eco-driving*: conducir consumiendo la menor energía posible pero manteniendo la calidad del servicio (Fern18). El caso más básico es la conducción en llano, para la que se calculan la aceleración y freno máximo, mantenimiento de la velocidad y deriva óptimos. Sobre este caso base se pueden añadir más características, como considerar que la vía no es llana, o que hay velocidades máximas. Añadir estas restricciones hace que la solución se vaya complicando. En (Sche17) se hace una recopilación de los estudios realizados sobre *EETC* y *EETT* (*energy-efficient train control* y *timetabling*).

Se han realizado estudios de optimización de *eco-driving*, basados en simulación detallada de los perfiles de velocidad (Domí14) (Domí11), característica fundamental para diseñar con precisión las marchas de ATO. Este simulador tiene unos errores medios de tiempo de un 1% y de consumo de un 4.2%. El estudio de (Domí11) consiguió un ahorro de energía del 13% en la red de Metro, con la consiguiente modificación de las marchas para implantarlo.

¹ El cruise control de los coches es un ADAS

En cuanto a la parte de simulación, existen simuladores como FRISO, desarrollado por INCONTROL para ProRail, ferrocarriles holandeses (Inco20). Permite simular todos los aspectos de una línea, tanto los trenes como el comportamiento de la infraestructura, cambio de agujas, etc. En este hay un módulo que simula un ATO, y permite estudiar cómo afectaría hacer cambios (como introducir un ATO) en la operación.

OpenTrack también es un simulador de marcha de los trenes, en él se simula cómo se comporta el sistema cuando se le da un horario y se introducen retrasos. Es el simulador que utilizan operadores como SNCF y SBB (OTra20). Estos simuladores no incorporan IA, y al actualizar su estado por tiempo discreto (del orden de 0.25 s), son relativamente lentos en su ejecución para aplicaciones de optimización, que requieren una gran cantidad de evaluaciones.

3.1 REGULACIÓN EXISTENTE

Con el auge de la inteligencia artificial, la UE está llevando a cabo una importante labor regulatoria. En 2018 se publicó '*Artificial intelligence for Europe*' (CE18(1)) junto con un plan de acción (CE18(2)). En ellos se trata la inteligencia artificial (IA) de forma general, aunque se hace mención a un transporte autónomo y más seguro, en el que la IA tendrá un papel relevante.

En 2019 se publicó el briefing (Nies19), en el que se hace un breve análisis del impacto de la IA en el sector del transporte por ferrocarril. Se centra más en usos como el mantenimiento, y hace hincapié en el *European Rail Traffic Management System* (ERTMS) como requisito para el futuro de la IA ferroviaria. La idea del ERTMS está presente desde 1990. Es un sistema que intenta que el tráfico ferroviario en Europa sea interoperable. En este trabajo no se menciona el modelado de un ATO, pero sí puntos muy importantes como la necesidad de poder geolocalizar al tren con precisión, o la ciberseguridad.

Se está trabajando en un ATO para incorporarlo al ámbito del ERTMS en proyectos europeos como el X2Rail. Tal y como corresponde a una iniciativa de este tipo, ha de ser totalmente interoperable. Este ATO se podrá utilizar para todo tipo de tráfico, y sólo se le pedirá la automatización de la conducción. La seguridad del tren sigue estando a cargo del ATP (*Automatic Train Protection*) que incorpora el Sistema ERTMS.

Los grados de automatización se denominan GoA (*Grade of Automation*) y se pueden ver en la Figura 5. En un GoA1, el maquinista se encarga de toda la conducción. En un GoA2 pasa a ordenar la apertura y cierre de puertas y la salida de la estación. Una preocupación de los operadores es la reducción en la carga de trabajo de los maquinistas (ERAA18). Tener un ATO hace que pasen a un rol de espera. Esto hace que en una emergencia tenga que cambiar instantáneamente a una actitud mucho más activa, atendiendo a la señalización. Para implantar un GoA2 hay que tener en cuenta el tiempo de adaptación de los maquinistas y reducirlo al mínimo.

Uno de los objetivos de Shift2Rail, proyecto de la UE enmarcado en el Horizonte 2020, es conseguir un ATO apoyado en ETCS (*European Train Control System*) que llegue hasta GoA3 o 4. Siendo el GoA4 el grado más alto de automatización (Figura 5).

El proyecto X2RAIL-4, dentro de Shift2Rail, tiene como objetivo obtener un ATO GoA4. Se espera que finalice para 2023 (S2R21). A más corto plazo, se espera tener las especificaciones del ATO GoA2 listas para finales de 2021 (EUAR20), para poder añadirlas a las Especificaciones Técnicas de Interoperabilidad en 2022.

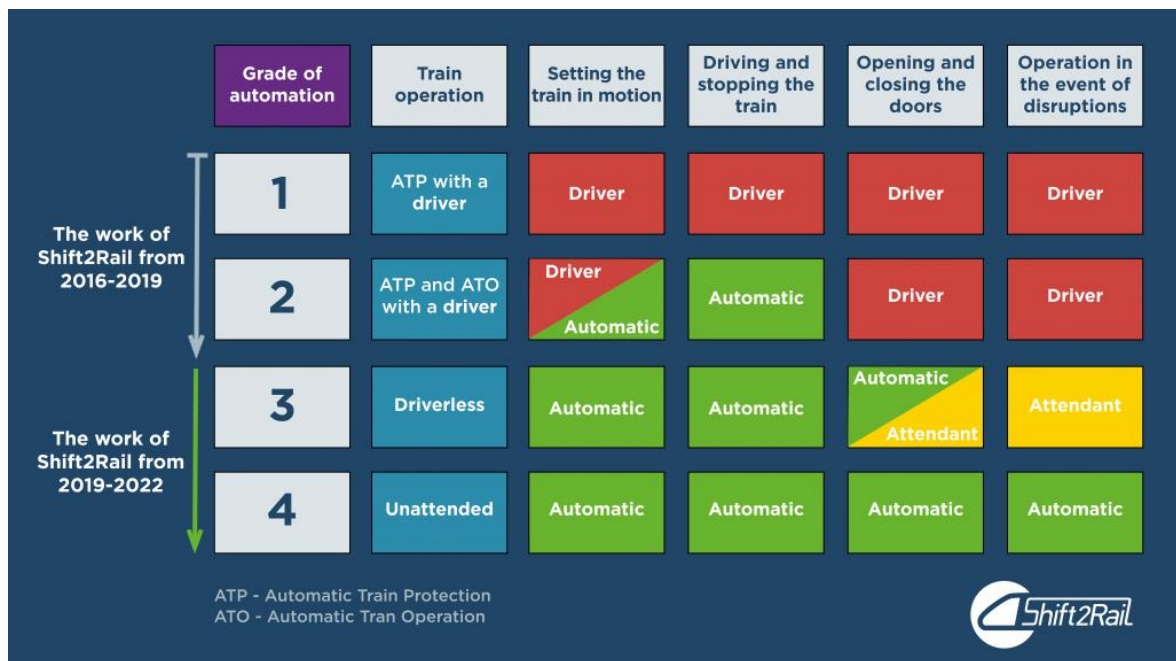


Figura 5: Grados de automatización (S2R19)

La responsabilidad que a veces se confiere a sistemas de inteligencia artificial hace que tengan que ser muy seguros. Aunque pueden ayudar al desarrollo de nuevas soluciones, hay que tener en cuenta la opinión de los operadores de estos sistemas.

No hay ventajas en un centro de control si el operador no confía en el sistema, ya que acabará por hacer el trabajo manualmente. Esta operación manual por parte del operador aumenta su carga de trabajo. Esto añadido a que es el responsable de la operación conduce inevitablemente a altos niveles de estrés.

En *'Building Trust in Human Centric Artificial Intelligence'* (CE19) se habla de la necesidad de hacer inteligencia artificial fiable. Los sistemas no sólo tienen que ser seguros, las personas deben entender cómo funcionan y ser capaces de seguir el proceso de decisión. En línea con esto, se indica que los sistemas considerados de alto riesgo deben ser revisados cada seis meses (Rodr20). La delimitación de lo que se considera un sistema de alto riesgo (*high-risk*) no está cerrada. La Comisión, sin embargo, considera por ahora que el sector en el que se va a aplicar y los usos que se le van a dar tengan un cierto riesgo (EUPa20). El

modelo de ATO desarrollado se puede calificar de alto riesgo. De querer comercializar soluciones de este tipo en el mercado europeo, sería necesario comprobar que cumplen los requerimientos de la UE.

En el ámbito español también está habiendo un impulso en materia de IA. La Agenda España Digital 2025, a través de la ENIA (Estrategia Nacional de Inteligencia Artificial), busca que para 2025 haya por lo menos un 25% de las empresas que utilicen IA y Big Data (GobE20(1)). Hoy en día sólo el 19% de las empresas españolas de transporte utilizan IA (GobE20(2)). La Secretaría de Estado de Digitalización e Inteligencia Artificial trata de fomentar una utilización más extensa de estas tecnologías en la industria española.

Capítulo 4. DEFINICIÓN DEL TRABAJO

4.1 JUSTIFICACIÓN

Se ha realizado un modelo de predicción de tiempos y consumos de marchas ATO. Esta predicción se hace en función de las consignas de conducción entre dos estaciones, es decir, se ha desarrollado el módulo de evaluación de marchas que podría ser usado por algoritmos de optimización. Cuando se tiene un ATO es fundamental ser capaz de predecir el tiempo y el consumo con suficiente precisión, ya que el diseño de las marchas está íntimamente ligado con ellos. En este trabajo se trata de obtener un modelo que prediga rápido el tiempo y consumo de unas consignas de conducción en un tramo determinado. No se optimizará el comportamiento del sistema, sólo se pedirá que con unas entradas determinadas obtenga los valores de las salidas.

A la hora de optimizar marchas, este modelo permitirá descartar de entrada las peores. Las que pasen este cribado inicial merecerán ser analizadas en más detalle. Debido a esto se necesita un modelo más específico, y utilizar un simulador de marcha cuya pieza fundamental es el modelo de ATO desarrollado en este trabajo.

Los estudios y trabajos realizados hasta el momento optimizan el entorno del ATO, o modelan su funcionamiento, pero con simulaciones muy costosas en tiempo de cálculo. Otro problema de los modelos de simulación de ATO clásicos es que requieren un conocimiento completo del funcionamiento del ATO de cada fabricante. En este caso se tratará de obtener una red neuronal que se ajusta automáticamente a partir de registros reales de operación. El objetivo será englobar todo lo que es el ATO en un modelo, que a partir de unos datos de entradas prediga la salida de ATO, es decir, la orden de esfuerzo de tracción/freno que deben cumplir los motores del tren en cada momento.

Este modelo se podrá utilizar como alternativa rápida en un simulador de la marcha del tren. Con él se podrán calcular los resultados preliminares antes de utilizar un módulo de ATO más complejo. Una diferencia con otros simuladores existentes es que el modelo que reproduce el ATO está embebido en el propio simulador.

A diferencia del estudio realizado por (Mart19), en este caso se utilizará Python. Es un lenguaje muy ligado al machine learning y tiene disponibles librerías y recursos que con Matlab resultan complicados de implementar. Además, se utilizarán datos reales obtenidos de un Sistema de ATO operando en una línea de metro en España.

Como se vio en el Capítulo 3. las técnicas de machine learning empiezan a generalizarse en el ámbito ferroviario. El gran volumen de datos disponible en este caso hace que se considere plantear un modelo de este tipo.

4.2 OBJETIVOS

- Obtener un modelo de machine learning de predicción de tiempo-consumo esperado en el recorrido del tren entre dos estaciones. Con este modelo se obtendrán el consumo (en kWh) y el tiempo de recorrido (en segundos) a partir de las consignas de conducción del ATO: la distancia de deriva y reducción de velocidad.
- Diseño de una interfaz de usuario que permita obtener resultados con el modelo anterior.
- Obtener un modelo de machine learning que prediga las órdenes de ATO (consigna de tracción/freno para los motores) en función de la posición y estado actual del tren, calculando la orden de tracción o freno que tiene que aplicar el ATO a lo largo del recorrido hasta llegar a la estación.
- Diseño de una interfaz de usuario mediante la cual se puedan calcular las salidas de ATO.
- Programación de un simulador de la marcha del tren que integre el modelo de machine learning del ATO. Evaluación para calcular errores de tiempo y distancia al usar los modelos frente a los datos reales.
- Diseño de una interfaz para el simulador.

4.3 METODOLOGÍA

La metodología seguida empieza con la definición de lo que se quiere predecir con cada modelo. Se tienen dos modelos. Uno predice tiempos de recorrido y consumos de la conducción de un tren en un tramo, y el otro predice consignas del ATO, que deben cumplir los motores del tren en cada momento.

Para entrenar los modelos se tienen datos reales de circulación de trenes. El siguiente paso es separar los datos de entrada y las salidas de otros datos registrados, que no se utilizarán. Las entradas empleadas para cada modelo se pueden ver en el apartado 6.1 y 7.1, dentro del subapartado de cada uno.

A continuación se desarrollan y entrenan los modelos de machine learning, utilizando y comparando las alternativas que mejor se adapten al tipo de problema que se quiere resolver. En el modelo de tiempo-consumo se han utilizado varios algoritmos para comparar los errores obtenidos con cada uno. Para predecir las consignas de ATO desde el principio se ha establecido el uso de una red neuronal.

Una vez obtenidos los modelos es necesario evaluar la bondad de sus resultados. Se calculan cinco tipos de errores teóricos cometidos: *mean squared error*, error cuadrático medio relativo, *mean absolute error*, el error máximo y el error porcentual (MAPE). Adicionalmente, para el modelo de ATO se han de calcular los errores al conducir el tren.

El establecimiento de un simulador permite calcular errores adicionales de los modelos de ATO. Los errores se evalúan en primer lugar con los datos de test y, posteriormente, se evalúan los modelos una vez incluidos en el simulador de marcha, ya que el objetivo es

obtener un simulador lo más preciso posible. Este paso es necesario para seleccionar el mejor modelo. Los modelos con errores teóricos más pequeños no son necesariamente los que tienen menores errores de conducción una vez que se integra el modelo del ATO en el simulador. Al simular, la salida anterior condiciona los valores de la entrada a la red neuronal, por lo que el modelo tiene que tratar de ser lo más general posible, evitando el *overfitting*.

Finalmente, para poder trabajar con los modelos se han diseñado interfaces de usuario. Mediante éstas se pueden hacer cálculos y obtener resultados de forma sencilla.

Para todo el trabajo se ha utilizado el lenguaje de programación Python y el entorno Spyder. Para diseñar la parte gráfica de las interfaces se ha utilizado Qt y QtDesigner.

4.4 PLANIFICACIÓN Y ESTIMACIÓN ECONÓMICA

En la Figura 6 se muestran las actividades realizadas a lo largo del trabajo. La planificación inicial incluida en el Anexo B sufrió ciertas modificaciones. La programación del simulador y cálculo de errores se adelantaron. Surgió la necesidad de evaluar los modelos a través de los errores cometidos en la simulación además de los errores teóricos.

La interfaz para el ATO incluye la del simulador, alargándose su duración frente a la de tiempos/consumos.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Entrenar modelos de tiempo-consumo y calcular errores														
Diseño de una interfaz para el modelo de tiempo-consumo														
Entrenar modelos de ATO y calcular errores														
Diseño de una interfaz para el modelo de ATO														
Programación del simulador de marcha con el modelo de ATO														
Cálculo de errores con el simulador de marcha														
Redacción del Anexo B														
Redacción de la memoria														

	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Entrenar modelos de tiempo-consumo y calcular errores																
Diseño de una interfaz para el modelo de tiempo-consumo																
Entrenar modelos de ATO y calcular errores																
Diseño de una interfaz para el modelo de ATO																
Programación del simulador de marcha con el modelo de ATO																
Cálculo de errores con el simulador de marcha																
Redacción del Anexo B																
Redacción de la memoria																

Figura 6: Planificación temporal de las actividades

La estimación económica del proyecto se encuentra en la Tabla 1. Se ha estimado una dedicación de los directores de un 10% de la duración total del trabajo. El consumo de electricidad del ordenador se ha incluido en el presupuesto. Se trata de un proyecto realizado

íntegramente con herramientas de software, en el que los tiempos para entrenar las redes neuronales y hacer las simulaciones son significativos.

El coste de la electricidad para el ordenador se ha estimado utilizando *OuterVision® Power Supply Calculator* (Extr20), tomando un precio de la electricidad de 0.16 €/kWh, tarifa de Iberdrola (Iber21).

Los costes indirectos se han tomado como el 20% de los costes de personal (CE13).

Tabla 1: Estimación económica del proyecto

	Coste unitario	Cantidad	Total [€]
Alumno	15€/h	350 h	5250
Directores	65€/h	35 h	2275
Ordenador	0,16€/kWh		7.5
Costes indirectos			1505
Total			9037.5
IVA (21%)			1897.86
Total con IVA			10935.38

Capítulo 5. CARACTERÍSTICAS COMUNES DE LOS MODELOS ENTRENADOS

En este capítulo se presentan los dos tipos de modelos entrenados. Para las redes neuronales se ha utilizado el módulo *Keras* de *TensorFlow* (Chol15), y para el resto de modelos *Scikit-learn* (Pedr11).

Ambos modelos tienen características en común. Los datos de entrada han sido normalizados. Estos modelos aprenden mejor cuando los datos se suministran en un rango entre 0 - 1 o -1 - 1. En este caso se han utilizado los datos entre -1 y 1. Se entrenaron varios modelos con los dos tipos de normalizaciones y se observaron errores más bajos en los entrenados con datos entre -1 y 1.

Esta normalización (*mean normalization*) se lleva a cabo utilizando la siguiente fórmula

$$\frac{x - \bar{x}}{x_{max} - x_{min}}$$

Donde \bar{x} es la media de la categoría.

Para todos los modelos, se separaron los datos de entrada en dos conjuntos (Géro17):

- Uno de entrenamiento (80%) que se utiliza para que el modelo aprenda a predecir los resultados
- Uno de validación (20%) que se utiliza para calcular los errores cometidos por el modelo

Estos datos de validación se cogen de forma aleatoria dentro del conjunto de datos de entrada, poniendo una semilla de forma que siempre coja los mismos. La primera vez sí se cogen al azar, pero todas las siguientes se seleccionan los mismos puntos. Haciendo esto se evita que a la larga los modelos acaben por ver todos los datos.

Los errores calculados son los siguientes, donde p son las predicciones y r los valores reales de las salidas:

- Mean squared error

$$mse = \frac{\sum (r - p)^2}{n^{\circ} \text{ datos}}$$

- Error Cuadrático Medio relativo

$$ECMr = \frac{mse}{Var}$$

El cálculo del ECMr se ha hecho de acuerdo a (Mart19), considerando un valor menor o igual a 0.25 como aceptable. El ECMr es otra medida de la desviación de las predicciones.

- Mean absolute error

$$mae = \frac{\sum |r - p|}{n^{\circ} \text{ datos}}$$

Se ha calculado el *mae* como alternativa al *mse*. El *mae* tiene la ventaja de que las desviaciones afectan de forma proporcional al error. En el caso del *mse* las desviaciones más grandes influyen más, penalizando los valores atípicos (*outliers*).

- Error máximo

$$e_{Max} = \max (|r - p|)$$

- Error porcentual

$$e_{\%} = \frac{\sum \left| \frac{r - p}{r} \right|}{n^{\circ} \text{ datos}} \cdot 100$$

El error porcentual calculado es el *mean absolute percentage error (MAPE)*. Como no se puede calcular cuando el valor real es 0, en estos casos se omitirá el punto al calcular el error. Las salidas con valor 0 sólo se encuentran en el modelo de ATO y son minoritarias (un 2.5% del total de datos).

Se toma el valor absoluto porque al tratarse de puntos aislados importa más la desviación producida que el sentido en el que se produzca. Si las desviaciones negativas se oponen a las positivas el error calculado no representa lo bien que el modelo se ajusta a las salidas reales. Un modelo con 0% de error puede ser perfecto, o simplemente tener la misma desviación positiva que negativa.

Para la realización de todos los gráficos que se incluyen en los siguientes dos capítulos se ha utilizado la librería *matplotlib* (Hunt07).

Capítulo 6. MODELO DE TIEMPO-CONSUMO

Se describe en primer lugar el modelo que predice los tiempos de recorrido y consumos. Este modelo se ha entrenado para ayudar en el proceso de diseño de marchas, siendo la entrada al modelo las consignas de conducción del ATO, la forma de recorrer un tramo concreto. En este caso se probaron diferentes alternativas antes de seleccionar de forma definitiva una red neuronal.

6.1 DATOS UTILIZADOS Y ESTRUCTURA

Los datos de entrada utilizados son las consignas de conducción del ATO: velocidad y distancia de deriva (Apartado 4.2). Los valores que toma la entrada ‘Velocidad’ son discretos, ya que representa cuánto se separa de la velocidad máxima. Varía entre -3 (3 km/h por encima de la velocidad máxima) y 5.

La distancia de deriva marca el punto en el que el tren deja de traccionar. Los valores varían entre 15m y 1105m, en incrementos de un metro. Siendo 1105m la longitud del tramo, por lo que esta distancia equivale a no hacer deriva. El no hacer deriva supone que se llega antes, pero se consume más. Las salidas del modelo son el tiempo de recorrido en segundos y el consumo en kWh.

Las combinaciones de entradas que daban lugar a marchas no confortables no se han incluido en el entrenamiento. Los datos que se han utilizado corresponden a un recorrido por la L1 de Metro de Barcelona, entre las estaciones de Fabra i Puig y Sant Andreu.

Para realizar este modelo se estudiaron diferentes alternativas. Dentro de las posibilidades que ofrece *scikit learn* se escogieron las que mejor se adaptaban al problema a resolver, teniendo en cuenta también el interés que han tenido en el último año. Una dificultad presente en este tipo de soluciones es la obsolescencia. Hay determinadas funcionalidades que dejan de tener soporte en versiones nuevas. En la Figura 7 se muestra la comparación de las búsquedas de los algoritmos utilizados, añadiendo el que más interés generó de los restantes de *scikit learn* (*Naive Bayes*).

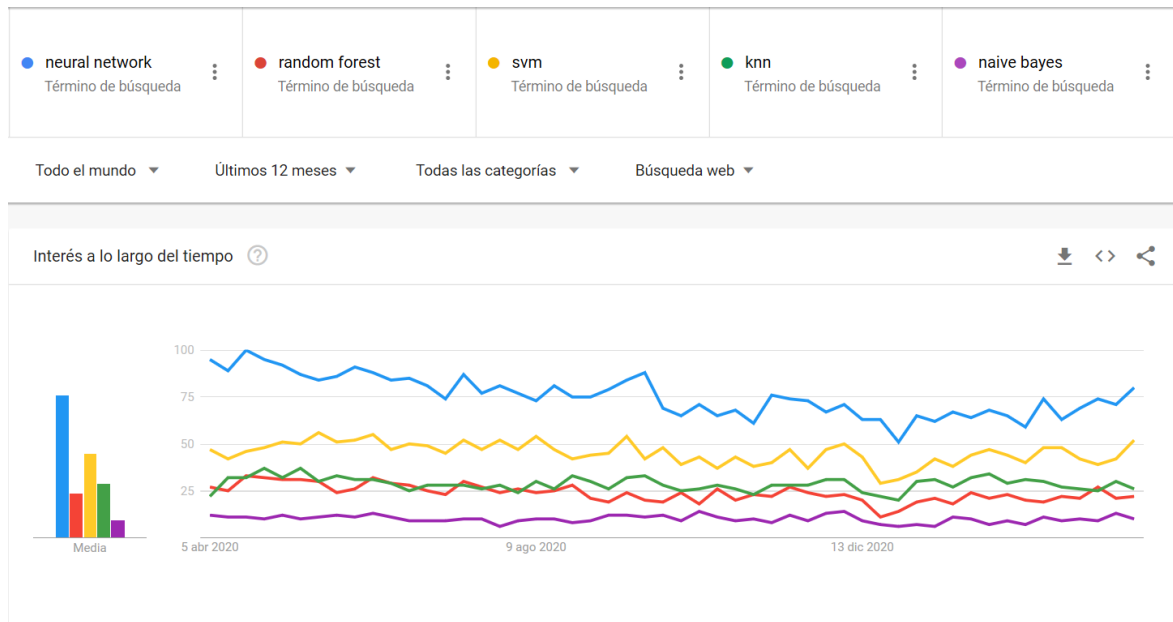


Figura 7: Comparación de las búsquedas de los diferentes algoritmos [Google Trends]

Adicionalmente se accedió a las fichas de cada algoritmo en *scikit learn* para comprobar que los excluidos no presentaban ventajas importantes a la hora de resolver un problema como este. Los algoritmos finalmente seleccionados para ser entrenados son los siguientes:

- Red neuronal
- Árbol de decisión
- RandomForest
- kNN
- Support Vector Regression

Se han calculado errores con todos para ver cuál era la solución más apropiada. El código de los modelos analizados se puede ver en el ANEXO I.

Una vez calculados estos errores se ve la necesidad de seleccionar una de las alternativas. Aparte de seleccionar las alternativas que presentan un error más pequeño, se ha comparado la capacidad de generalización de los modelos. Para realizar esta comparación se han entrenado los mejores con un *set* de datos en el que se ha eliminado un valor de consigna de reducción de velocidad.

La estructura es la misma y la única diferencia es que a los datos de entrenamiento les falta un valor de reducción de velocidad máxima. Se han eliminado las entradas en las que el código de velocidad era 1. Este corte se ha realizado seleccionando un código de velocidad al azar.

En la Figura 8 se muestran las salidas correspondientes a este *set*. El eliminar una velocidad hace que desaparezca una línea entera de datos. De esta forma se busca que los modelos sean

capaces de imaginarse cómo tiene que ser la salida de la ‘Velocidad’ = 1 a partir de las velocidades 0 y 2, que son las contiguas. Para analizar los modelos se entrenan con estos datos y luego se validan haciéndoles predecir la fila que falta.

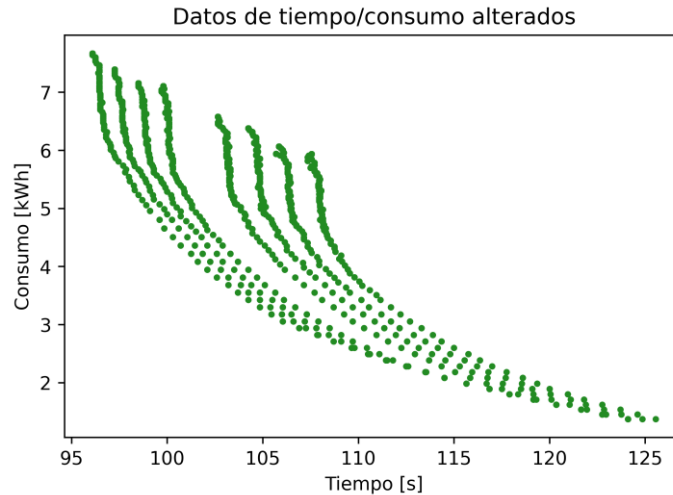


Figura 8: Modificación de los datos para cálculo de tiempo/consumo

6.2 RESULTADOS DE ERRORES OBTENIDOS

Se muestra en primer lugar la primera parte del proceso: el entrenamiento y cálculo de errores con todos los modelos. En la Tabla 2 se recoge una comparativa de los errores obtenidos con los diferentes tipos de modelos entrenados. En esta tabla se reflejan los resultados de los modelos entrenados con todos los códigos de velocidad. En la Figura 9 se incluye una gráfica de comparación entre los valores reales de tiempo/consumo y los predichos por la red neuronal.

Tabla 2: Errores cometidos en la predicción de tiempo y consumo

	Tiempo					Consumo				
	MSE	ECMr	MAE	%	Max	MSE	ECMr	MAE	%	Max
Red neuronal	0.0087	0.0005	0.0416	0.0394	1.4232	0.0006	0.0006	0.0181	0.3383	0.1200
Decision tree	0.0162	0.0009	0.0196	0.0181	3.0000	0.0002	0.0002	0.0029	0.0528	0.2780
RandomForest	0.0051	0.0003	0.0148	0.0136	0.9683	0.0001	0.0001	0.0026	0.0480	0.1224
SVM (Reg)	1.9720	0.1047	0.4033	0.3711	16.0847	0.0628	0.0642	0.0873	2.2198	6.0799
kNN	0.5370	0.0285	0.0566	0.0526	23.3051	0.0368	0.0377	0.0134	0.3052	5.9634

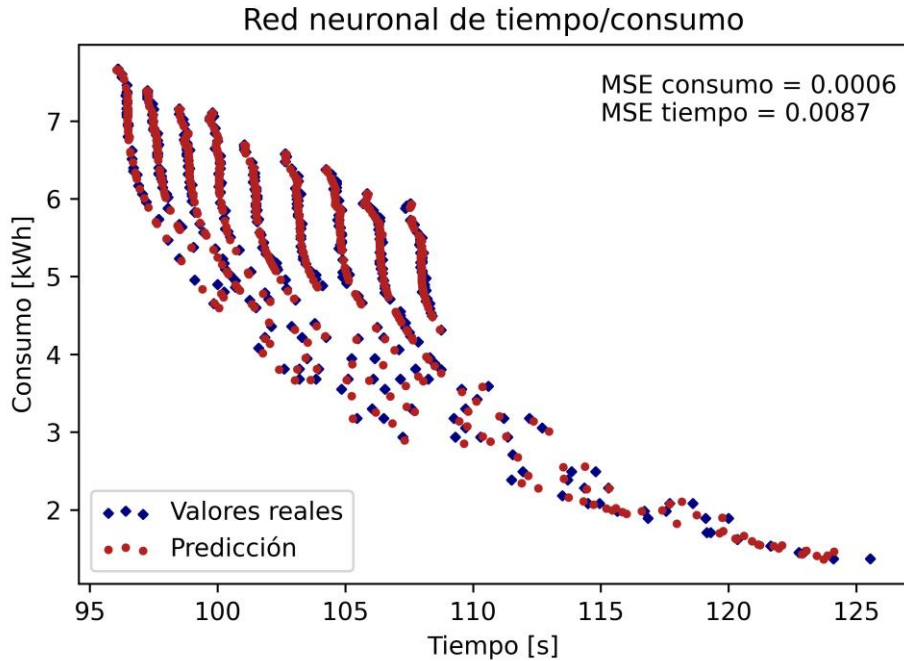


Figura 9: Comparación de valores reales y predichos con el modelo de tiempo/consumo

Los valores de ECMr que se observan en la Tabla 2 están muy por debajo de los 0.25 que considera aceptables (Mart19), de hecho en (Mart19) se obtienen ECMr entre 0.14 y 0.2. En este caso el *mse* es muy bajo por lo que el ECMr también. En la Tabla 3 se muestra una comparativa de los errores de diferentes soluciones. Los errores obtenidos en este trabajo no son realmente comparables con los de otros trabajos presentados en el Capítulo 3. El modelo desarrollado es diferente a otras alternativas existentes.

Tabla 3: Comparativa de errores en predicciones de tiempos y consumos

	% Tiempo	% Consumo
(Domí11)	1	4.2
(Jong05)	0.16	-
(Mart19)	-	1.49
Red Neuronal	0.04	0.34

Como segunda parte del proceso, se hace un análisis de la capacidad de generalización. Los algoritmos con peor comportamiento se descartaron. kNN sólo puede predecir en el rango de entradas, por lo que generaliza mal, y no se incluyó. Viendo los valores de los errores, la *SupportVectorRegression* fue rechazada desde un principio.

Entre un árbol de decisión y un *RandomForest* se seleccionó el *RandomForest*, por ser una evolución del árbol de decisión y tener mejores resultados. A pesar de que es el que tiene errores más bajos, hay que tener en cuenta que el *RandomForest* es un algoritmo con alta tendencia al *overfitting*, igual que los árboles de decisión. Gracias al análisis de generalización se puede ver si está aprendiendo los valores de memoria.

Los resultados obtenidos de este análisis de capacidad de generalización se pueden ver en la Tabla 4 y en la Figura 10. La Figura 10 muestra la comparación entre las predicciones obtenidas por una red neuronal y un *RandomForest*. Como se ha mencionado en el apartado 6.1, los datos con código de velocidad 1 se dejaron fuera del entrenamiento, y se utilizaron para hacer las predicciones.

Como se puede ver, la red neuronal (NN) es capaz de generalizar y predecir razonablemente bien las salidas correctas. El *RandomForest* (RF), sin embargo, predice las salidas más cercanas, que ha aprendido de memoria. Los errores correspondientes a las predicciones de este caso se pueden ver en la Tabla 4.

Tabla 4: Comparación de generalización entre NN y RF

	Tiempo				Consumo			
	MSE	MAE	%	Max	MSE	MAE	%	Max
NN	0.082	0.2118	0.2060	2.9526	0.0129	0.1031	1.6906	0.2355
RandomForest	1.8266	1.3410	1.3204	1.5500	0.1023	0.3046	4.8645	0.4823

Es por esto que entre los diversos algoritmos y una red neuronal, se ha preferido ésta última, ya que es más adaptable a situaciones diferentes. Aunque se ha seleccionado la red neuronal, hubo varios intentos por mejorar la capacidad de predicción del *RandomForest*. Desde el punto de vista de rapidez de entrenamiento habría sido interesante su uso. La red neuronal tarda unos 6.5 min en ser entrenada, frente a los 7 s que tarda el *RandomForest*.

Para resolver el *overfitting* observado en el *RandomForest* se utilizaron los métodos habituales para reducir este problema. La *max_depth* se redujo y *min_samples_leaf* se aumentó.

Con *max_depth* se controla la profundidad del árbol, que afecta a las divisiones que tiene y a su capacidad para aprender los datos. Si este valor es alto se aprende mejor los datos, y deja de generalizar. Con *min_samples_leaf* se controla con cuántos valores deja de dividir las ramas. Si una división va a dar por resultado que en una hoja haya menos puntos que los que marca *min_samples_leaf*, la división no se hace. En la Figura 11 se muestra el resultado de aplicar estas medidas, los cambios en los parámetros llegaron siempre a formas parecidas. El árbol de decisión y el *RandomForest* no son adecuados para generalizar en este problema.

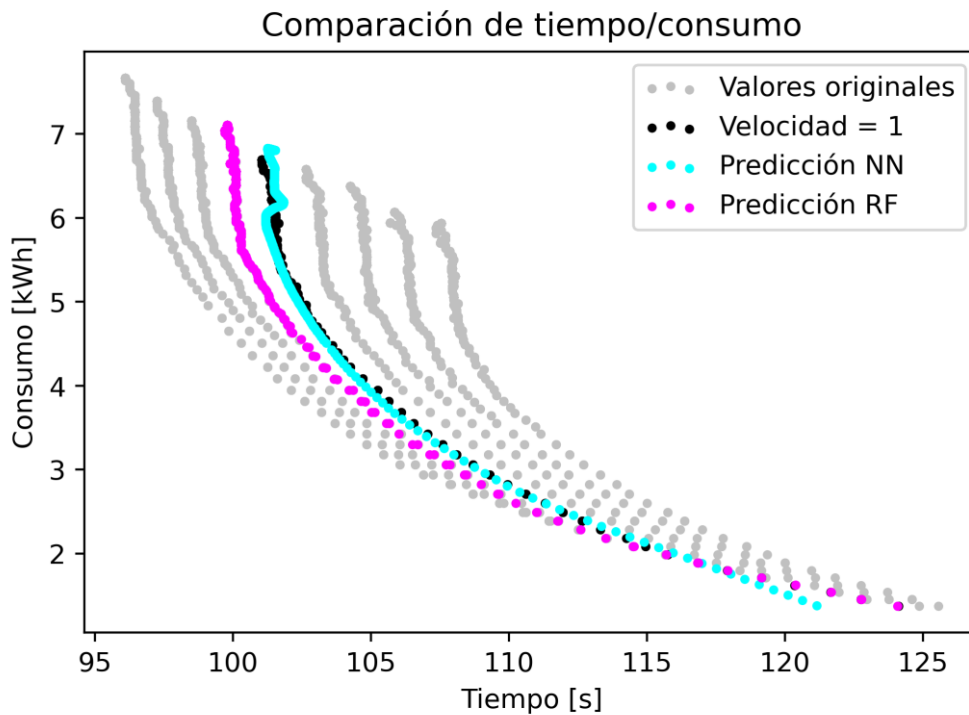


Figura 10: Comparación de generalización entre NN y RF

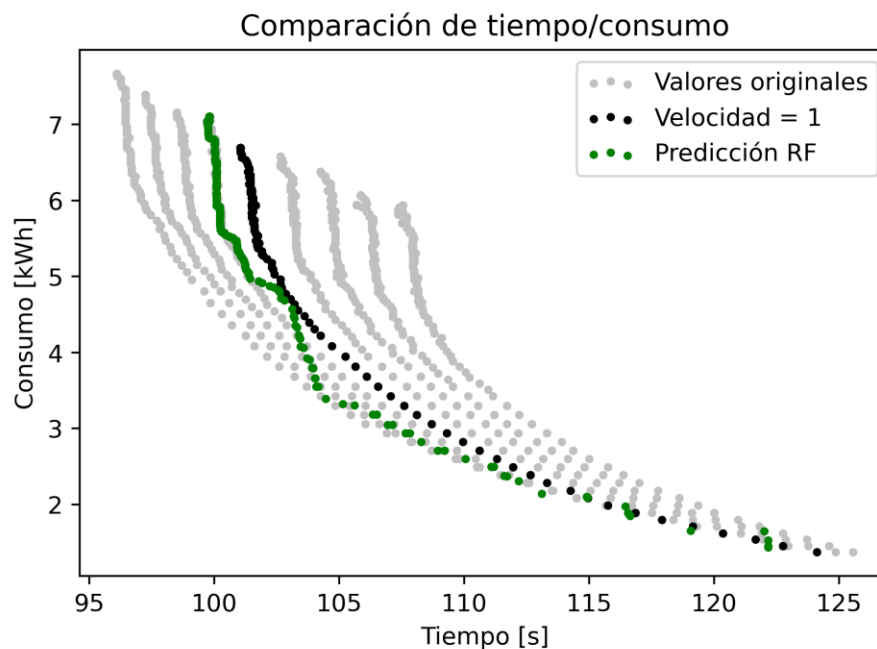


Figura 11: Variación del RandomForest para tratar de corregir el overfitting

Capítulo 7. MODELADO DEL ATO Y SIMULADOR DE MARCHA

Tras la descripción del primer tipo de modelo, que predice tiempos y consumos de las marchas, a continuación se presenta el segundo tipo. En este caso se trata también de una regresión, aunque con un único *output*, que predice las salidas del ATO (*Automatic Train Operation*). Lo puesto en el Capítulo 5. sigue siendo válido en este apartado.

7.1 MODELO DE ATO

Este modelo es el que luego se integrará en el simulador de marcha, por lo que la salida que proporciona es una orden de ATO, es decir, la demanda de esfuerzo de tracción/freno que deben cumplir los motores del tren en cada momento. El código de los modelos se puede ver en el ANEXO I. Se trata en todos los casos de redes neuronales, salvo un modelo auxiliar que hace una clasificación y es un *RandomForest*. Las entradas a todos estos modelos son las mismas, la velocidad y aceleración del tren, las velocidades máximas del ATP (*Automatic Train Protection*) y ATO, las velocidades y distancias objetivo del ATP y ATO, la distancia a la estación y el gradiente de la vía.

Para entrenar la red neuronal se han utilizado datos de una línea recorrida en sentido ascendente y descendente. Los datos corresponden a la línea 1 de Metro de Barcelona (Hospital de Bellvitge-Fondo), que tiene una longitud de unos 20 km, y en la que hay 30 estaciones (Figura 12). Se dispone de los datos de la línea, en los que se encuentra cada estación asociada al punto kilométrico en el que está, así como las distintas declividades en los tramos de la línea.



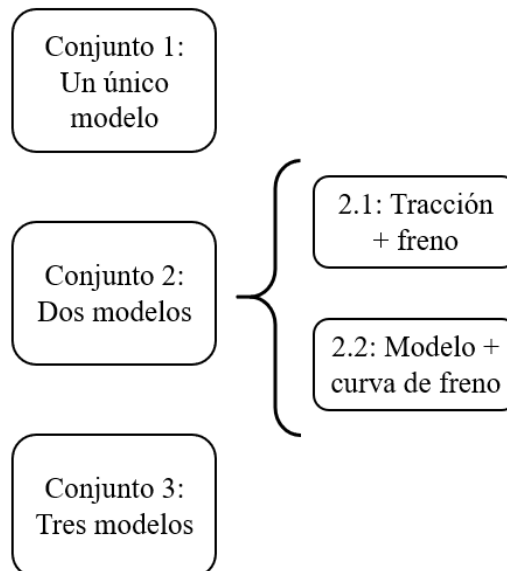
Figura 12: Línea 1 de metro de Barcelona (TMBa20)

Para cada una de las interestaciones se tiene un registro de velocidades, aceleraciones y distancias según va avanzando el tren en su marcha. Se dispone también del valor de gradiente que considera el ATO en sus cálculos. Este dato aparece en el mapa de vía del ATO y puede ser ligeramente diferente del dato de pendiente real.

La obtención de estos modelos fue de la mano del desarrollo del simulador. Desde el principio se vio la necesidad de evaluar los modelos viendo lo bien que conducían además de sus errores.

Para calcular los errores teóricos de los modelos se utilizan datos separados al azar del conjunto. Esto hace que no sean representativos de cómo se va a comportar luego en el simulador. Los errores se han utilizado de forma orientativa para descartar los peores modelos. Se ha comprobado que los modelos con errores más altos son los que se comportan peor en el simulador.

A continuación, se describen los diferentes modelos empleados. En algunos casos se utilizan varios modelos para hacer una única predicción. En el Esquema 1 se muestra un resumen de los conjuntos utilizados. El primero utiliza una única red neuronal, el segundo y el tercero, dos. Dentro del segundo se divide a su vez en función del tramo en el que se utiliza cada modelo. En el 2.1 ambos modelos predicen todos los puntos posibles. En el 2.2 el modelo llamado ‘*de curva de freno*’ sólo predice puntos en los que la distancia a la estación es de menos de 150m. El tercer conjunto utiliza dos redes neuronales como las del 2.1, y un modelo adicional de clasificación.



Esquema 1: Relación de conjuntos utilizados

Como entrada al modelo se utilizan los parámetros de:

- Velocidad actual del tren
- Aceleración actual del tren
- Velocidad máxima del ATP
- Velocidad y distancia objetivo del ATP
- Velocidad máxima del ATO
- Velocidad y distancia objetivo del ATO
- Gradiente considerado por el ATO

La salida que devuelve el modelo es la orden de ATO, en forma de una aceleración objetivo que debe seguir el tren. Esta aceleración se tiene en los datos de entrada y varía entre 10000 y -10000 [$10000 \cdot \text{m/s}^2$], aunque como se ha comentado en el Capítulo 5. las entradas al modelo están normalizadas. El rango en el que se mueven las salidas reales de ATO que le entran a la red neuronal es entre -0.6 y 0.5 (después de la normalización). En este caso sólo se ha considerado el uso de redes neuronales.

Tras entrenar modelos con distintos optimizadores como *Adam*, *SGD* o *Adadelta*, se comprobó que *AdaGrad* era el que presentaba menores errores en el simulador.

7.1.1 Conjunto 1: Modelo único para tracción y freno

En estos modelos, una única red neuronal predice una única salida, que puede ser una orden de tracción o freno.

Los datos utilizados se han dividido siguiendo las modificaciones en el *set* con el que se entrenaban las redes neuronales. Aunque en un principio se utilizaron todos los datos disponibles, con el tiempo se vio la necesidad de seccionar este *set* de datos. En la Figura 15 se recoge un resumen de los tipos de modelo único que se describen a continuación.

- La primera familia de modelos es la entrenada con todos los datos disponibles. (Modelo 1)
- Una segunda familia son los entrenados con datos ampliados. Al ver los errores obtenidos en la frenada, se probó a utilizar el algoritmo SMOTE² para introducir más datos en la zona de frenado. Estos modelos tenían errores significativamente menores que los entrenados anteriormente. Sin embargo, esta opción se abandonó tras algunos modelos. Desde el punto de vista de esta aplicación, la solución de introducir nuevos valores en los datos de entrenamiento se consideró poco ortodoxa. Los nuevos valores no son datos reales medidos, son puntos intermedios que pueden no tener sentido. (Modelo 2)

² Una explicación del funcionamiento del algoritmo SMOTE se puede ver en el ANEXO II.

- Tras la experiencia de SMOTE se decidió utilizar solamente una parte de las interestaciones a la hora de entrenar. Esto permite comparar cómo se comportan los modelos al correr en estaciones totalmente nuevas. Es también una forma de medir que en el entrenamiento no está habiendo *overfitting*. La zona que se retiró de los datos de entrenamiento fueron las estaciones entre Bellvitge y Santa Eulalia (6 interestaciones). Estos datos de entrenamiento son una sección del *set* completo de datos. (Modelo 3)
- A continuación, se suprimieron datos de entrenamiento que presentaban comportamientos anómalos, tales como frenadas inesperadas o velocidades anormalmente bajas. Estos datos se eliminaron para que el modelo no aprendiera de ellos. (Modelo 4)

En la Figura 13 se incluye un ejemplo de uno de estos casos. Según los registros de la marcha del tren, se comporta como si hubiera una limitación de velocidad a 40 km/h.

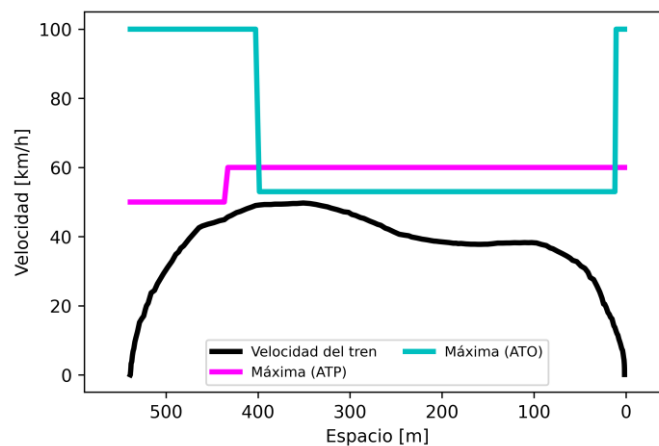


Figura 13: Recorrido del tren real en una interestación que presenta un tramo anómalo

Estos datos del *set*, junto con otros en los que se superaba la velocidad máxima, fueron eliminados. En la Figura 14 se puede ver la simulación de otro ejemplo de una de estas interestaciones. La simulada acelera durante más tiempo al principio, cayendo después de forma uniforme. Aunque la parte de simulación pertenece al 7.2, se incluye en este apartado una figura para ejemplificar el problema que tenían los datos eliminados.

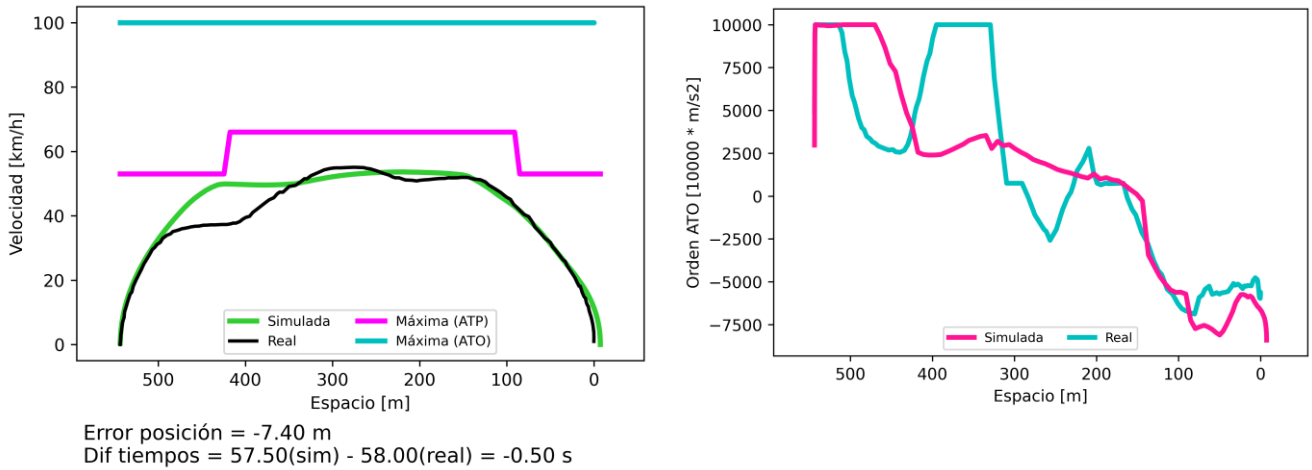


Figura 14: Simulación en una interestación con datos reales anómalos

→ Viendo los problemas de los modelos anteriores para respetar las velocidades máximas en el simulador, se entrenó otra variante con una única velocidad máxima. (Modelo 5)

Esta velocidad máxima se cogió como la mínima entre la del ATP y el ATO.






Tipos de modelo único en función de los datos de entrenamiento	Modelos 1.x Todos los datos	Modelos 2.x Todos los datos + SMOTE
		
Modelos 3.x Omitir interestaciones enteras	Modelos 4.x Suprimir datos anómalos	Modelos 5.x Una velocidad máxima
		

Figura 15: Tipos de modelo único en función de los datos de entrenamiento

7.1.2 Conjunto 2: Dos modelos

Para tratar de reducir los errores, durante el frenado fundamentalmente, se entrenaron modelos diferentes para la tracción y el freno. La salida global es la más baja de las dos, como se verá a continuación. Se han desarrollado dos variantes:

- Tracción + freno
- Modelo + curva de freno

En el Esquema 2 y Esquema 3 se ha incluido cómo se hacen las predicciones con ambas variantes de este conjunto.

7.1.2.1 Datos utilizados para el entrenamiento

Se crearon tres *datasets* a partir de los datos de entrada. Cada uno se utilizó para entrenar uno de los modelos.

→ Modelo de tracción (Trac)

Modifica la columna de orden de ATO que hay en los datos de entrenamiento, de forma que sólo se queda con las salidas positivas del ATO

Si la salida de ATO en los datos de entrenamiento es positiva: no se modifica

Si la salida de ATO es negativa: se sustituye por un 0

→ Modelo de freno (Fren)

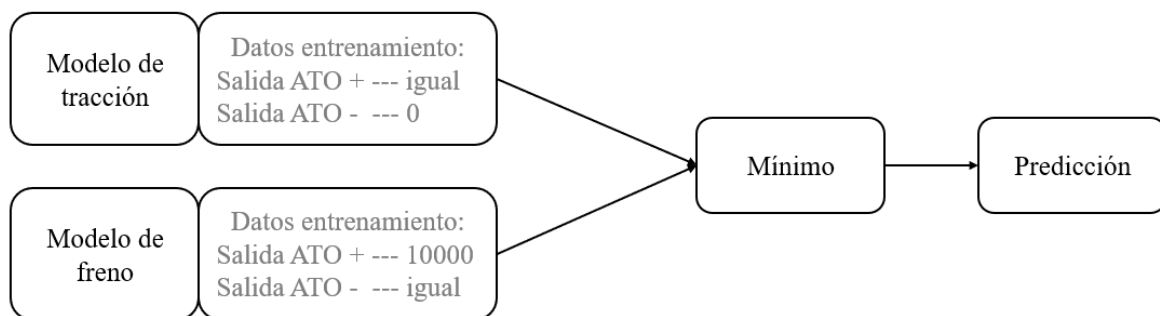
Se queda con las ordenes de ATO negativas

Si la salida de ATO es positiva: se sustituye por 10000

Si la salida de ATO es negativa: no se modifica

❖ Modelo de curva de freno (BC)

Entrenado con los datos que están a menos de 150m de la estación, sin hacer distinciones cuando tracciona o frena.



Esquema 2: Conjunto 2.1: Dos modelos con tracción + freno

7.1.2.2 Estructura

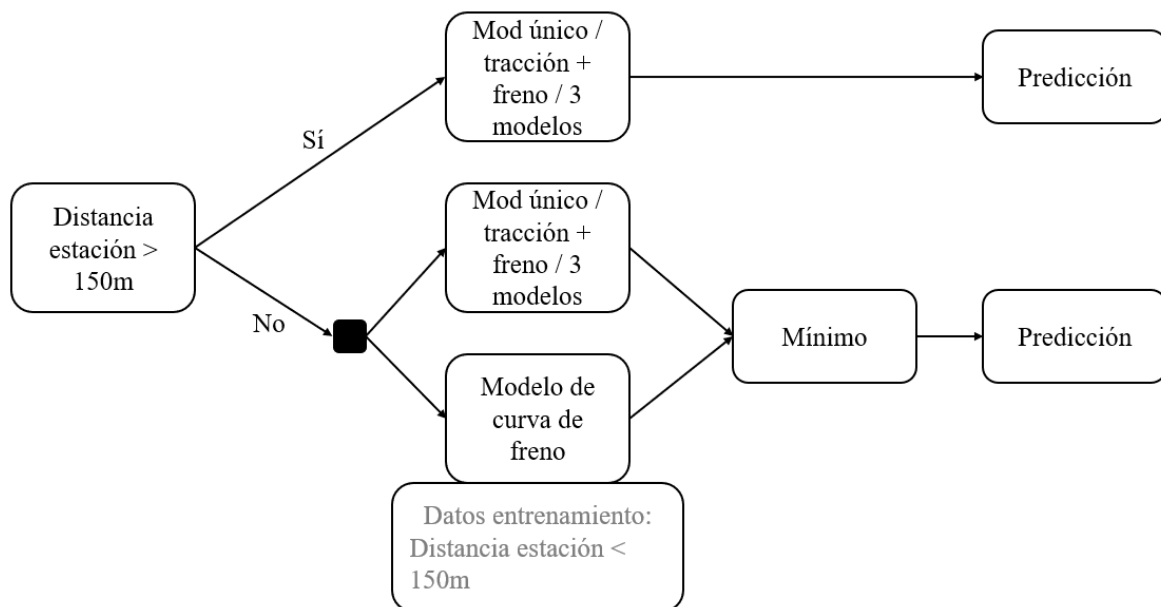
La estructura de la red neuronal es como la del modelo único. La diferencia se encuentra en la salida del modelo.

En el caso de tracción + freno, los datos de entrada mencionados en el apartado 7.1.1 se pasan a cada uno de los modelos, que predicen de forma independiente (Esquema 2). La salida global es la más baja de las predicciones, los frenados siempre tienen preferencia.

Si la salida es de tracción, el modelo de tracción predice la orden de ATO que hay que aplicar, y el de freno un número muy elevado para que no limite la marcha. La más baja de ambas es la salida de tracción correcta. En el caso del freno, el modelo de tracción predice 0, y el de freno una orden negativa, siendo la más pequeña la negativa.

El método para predecir con el modelo de curva de freno se muestra en el Esquema 3. Con esta opción se puede utilizar cualquier modelo o combinación de modelos. Este modelo principal es el que se utiliza para predecir las salidas de ATO de todos los puntos del recorrido, siendo de hecho un modelo cualquiera de los expuestos anteriormente. La diferencia con las otras formas de predicción ocurre cuando se llega a menos de 150m de la estación de destino. De este punto en adelante, hace una predicción simultánea con el modelo principal y el de curva de freno, quedándose con la salida más baja como ocurre con el conjunto 2.1.

De esta forma, a menos de 150m de la estación de destino prevalece el modelo de curva de freno, salvo que el modelo principal frene por sí mismo.



Esquema 3: Conjunto 2.2: Dos modelos con modelo de curva de freno

7.1.3 Conjunto 3: Tres modelos (clasificación + tracción + freno)

Como variante a la serie de modelos presentada en el apartado 7.1.2, se desarrolló una versión en la que un modelo clasifica los datos. Esta clasificación se realiza en función de si la salida va a ser de tracción o freno. En el Esquema 4 se muestra el proceso de predicción de este conjunto.

Se estudió esta alternativa para sortear el error debido al modelo de freno. Para salidas que deberían ser de tracción, si el modelo de freno predice una salida negativa se coge esa como salida global.

7.1.3.1 Datos utilizados para el entrenamiento

→ Modelo de clasificación (Clas)

Modifica la columna de orden de ATO de la siguiente forma:

Si la salida de ATO es positiva: se sustituye por 1

Si la salida de ATO es negativa: se sustituye por 0

→ Modelo de tracción

Igual que el presentado en 7.1.2.1

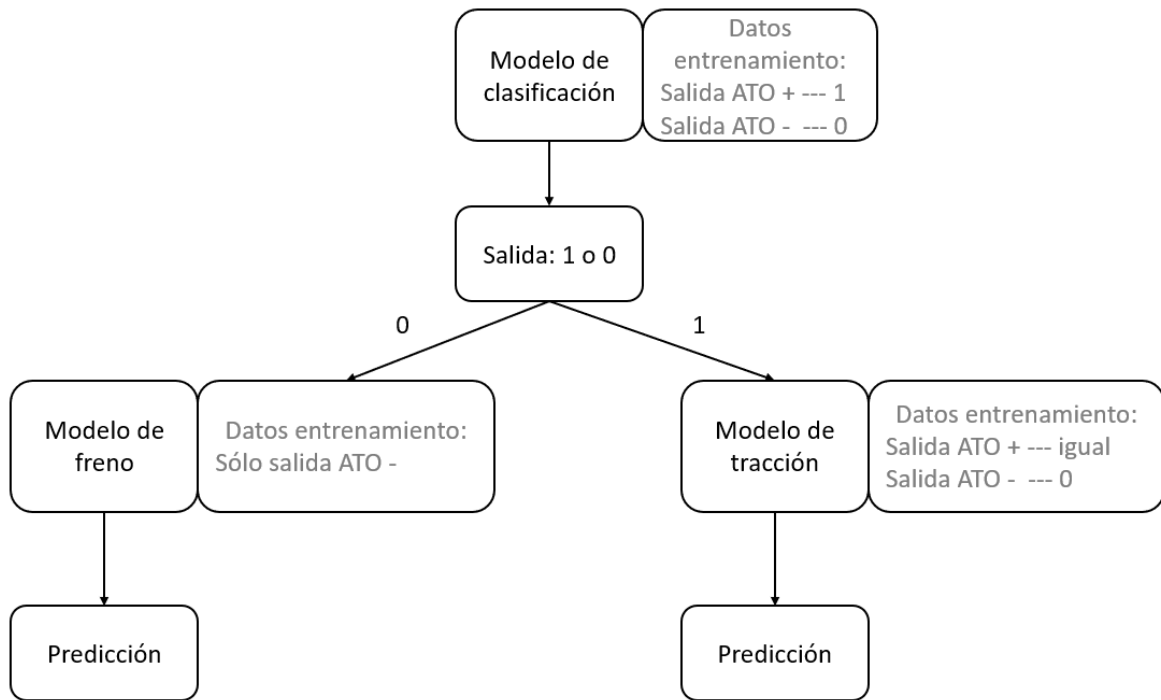
→ Modelo de freno (Reg fren)

Entrenado únicamente con los datos con salidas negativas, sin modificarlos

7.1.3.2 Estructura

Los modelos de tracción y freno son redes neuronales como las de los apartados anteriores. La única diferencia son los datos utilizados para el entrenamiento de la red de freno. Para hacer la clasificación se utilizaron diversos algoritmos además de la red neuronal, obteniéndose los menores errores con un *RandomForest*.

En este conjunto, los datos de entrada se pasan por el modelo de clasificación. Este modelo ha sido entrenado para, en función de las entradas, predecir si la salida tiene que ser de tracción o freno. En función de lo que prediga este primer modelo se transfieren los datos a la red neuronal de tracción o a la de freno (Esquema 4). Siendo la salida del conjunto la que prediga la red neuronal correspondiente.



Esquema 4: Grupo ‘tres modelos’

7.1.4 Resultados de errores obtenidos

En la Tabla 5 se pueden ver los errores cometidos por los diferentes tipos de modelos entrenados. Se han incluido sólo los modelos que dieron menos error. Al hacer las simulaciones se observó que los modelos con errores más bajos no son necesariamente los que se comportan mejor en el simulador.

Tabla 5: Errores obtenidos al entrenar los modelos de ATO

	Max Error	% Error	MAE	MSE	ECMr
Modelo único de tracción y freno					
Modelo 1	4115.77	36.05	331.11	235022.40	0.0078
Modelo 2	3712.21	23.37	275.54	201261.19	0.0054
Modelo 3	3678.87	38.28	334.55	255344.61	0.0082
Modelo 4	3317.09	23.04	259	153252.27	0.0049
Modelo 5	1911.37	25.2	256.8	145591.77	0.0046
Dos modelos					
Trac	3496.25	17.03	137.11	117331.55	0.0076
Fren	10900.63	269.52	773.64	2921406.07	0.0615

BC	2372.61	8.58	247.23	133059.37	0.0472
Tres modelos					
Clas	1.00	1.42	0.02	0.02	0.0809
Reg fren	1872.24	33.38	215.58	104648.10	0.0237

Los modelos de freno presentan errores más altos. La diferencia entre los valores de freno y el 100000 establecido para los de tracción hace que cuando se equivoca el error que ve sea mucho mayor (caso de *Fren*).

El valor tan bajo de ECMr se debe a la varianza tan grande que tienen los datos, entre 10000 y -10000.

Para los modelos del conjunto de dos o tres no hay comparación en los valores de los errores. Estos modelos están entrenados con un *set* de datos diferente. Los modelos del conjunto de modelo único tienen los mismos datos de entrenamiento, aunque en algunos casos se hayan suprimido o añadido puntos. Para los otros modelos se altera el valor de la columna de órdenes de ATO en los datos de entrenamiento, dejando de ser comparables.

En la Tabla 6 se han incluido los errores de los modelos entrenados para hacer la clasificación (apartado 7.1.3). El elegido finalmente fue el *RandomForest*. Las diferencias con la red neuronal son más acusadas que en el modelo de tiempo y consumo. Esto sumado a que el uso que se le iba a dar era experimental hizo que se utilizara el *RandomForest*.

Si este conjunto hubiera sido el mejor en el simulador se habría analizado el posible *overfitting*. Sin embargo, como se verá en el apartado 7.2.3 este no es el caso. No se utilizará este modelo para hacer predicciones, por lo que no es necesario comprobar si generaliza bien o no.

Tabla 6: Errores del modelo de clasificación

	Max Error	% Error	MAE	MSE
Red Neuronal	1.43	43.47	1.02	1.29
Decision Tree	1.00	3.16	0.04	0.04
RandomForest	1.00	1.42	0.02	0.02

7.2 SIMULADOR DE MARCHA

El simulador de marcha del tren es el último paso en la evaluación de los modelos de ATO. Se han utilizado los modelos con errores más bajos para construir el simulador. Las nuevas entradas del modelo ATO se van calculando a partir de las salidas que proporciona el simulador. Hay una relación entre las predicciones y las entradas que no se evalúa a través

de los errores calculados anteriormente. Esto hace que los errores más bajos en el simulador de marcha no correspondan necesariamente a los modelos ATO con errores teóricos más bajos.

El simulador funciona resolviendo en cada paso de simulación la dinámica del tren a partir de la orden de aceleración objetivo que proporciona el ATO. En un primer momento se obtienen los datos de la línea, pasando a continuación a simular cada una de las interestaciones,

Se ha hecho en todo momento la simplificación de considerar el tren una masa puntual. No se tienen en cuenta esfuerzos debidos a las curvas ni a las transiciones entre pendientes. Las pendientes cambian de manera discreta. Tampoco se ha evaluado el modelo desde el punto de vista del confort de los pasajeros.

7.2.1 Ecuaciones del movimiento

En este apartado se incluyen las ecuaciones utilizadas para resolver la dinámica del tren. Al esfuerzo de tracción se oponen una serie de fuerzas, que determinan la aceleración del tren.

- Esfuerzo de tracción

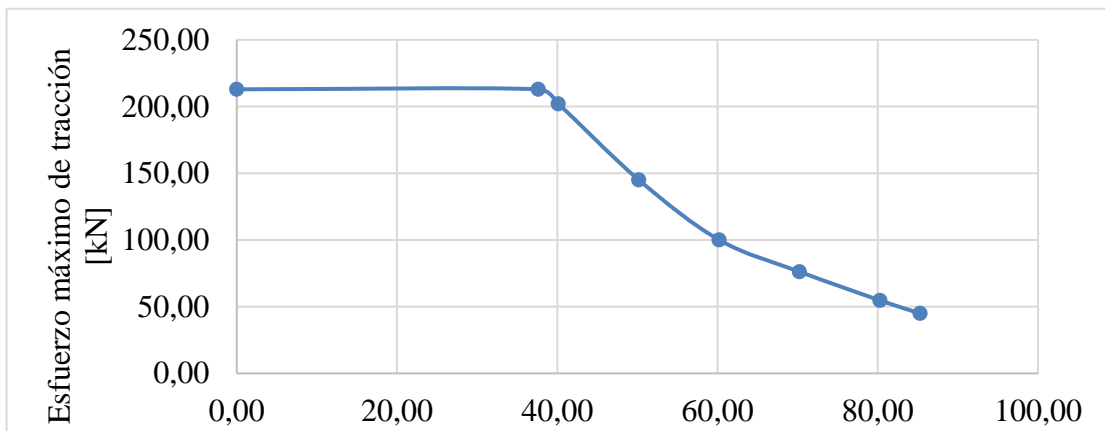


Figura 16: Curva de esfuerzo máximo de tracción

El máximo esfuerzo de tracción que es capaz de dar el tren depende de la velocidad.

$$ET_{MAX} = \begin{cases} 212,91 [kN] & \text{si } v \leq 37,63 \frac{km}{h} \\ 734,92 \cdot e^{-0,033v} [kN \text{ con } v \text{ en } km/h] & \text{si } v > 37,63 \frac{km}{h} \end{cases}$$

En la Figura 16 se puede ver la curva del esfuerzo máximo de tracción que es capaz de dar el tren, característica del tren con el que se obtuvieron los datos de los registros. El esfuerzo de tracción resultante de la salida del ATO se compara con el máximo para obtener el aplicable. El tren no puede dar más esfuerzo de tracción que el máximo, las ruedas patinarían

o los motores se encuentran a potencia máxima. Debajo de la curva de tracción se puede estar sin restricciones

El esfuerzo de tracción que ordena el ATO se obtiene al multiplicar la masa total por la salida del ATO. La salida del ATO es una aceleración. Los datos tanto de la masa en vacío como la inercial, asociada a las masas rotatorias del tren, son conocidos.

$$ET_{ATO} = (M_{vacio} + M_{inercial}) \cdot Orden\ ATO$$

Cuando el valor del esfuerzo de tracción del ATO (ET_{ATO}) es negativo, el resultado se ha multiplicado por 0.9, ya que el esfuerzo máximo de freno considerado es el 90%.

El esfuerzo de tracción que finalmente se utiliza es el que solicita el ATO, limitado por el esfuerzo máximo que pueden proporcionar los motores

$$ET = \min \{ET_{ATO}, ET_{MAX}\}$$

- Fuerzas que se oponen al movimiento
 - Resistencia al avance, que sigue la ecuación de Davis (Davi26)

$$F_{RR} = A + B \cdot v + C \cdot v^2$$

Donde A, B y C dependen del tren y son conocidos al igual que el esfuerzo máximo de tracción.

- Gradiente de la vía

$$F_{slope} = M_{vacio} \cdot g \cdot gradiente$$

Siendo g la aceleración de la gravedad y el gradiente el correspondiente al tramo en el que se encuentra el tren. No se ha tenido en cuenta la longitud del tren, por lo que éste ve un único gradiente en cada momento, positivo en rampa y negativo en pendiente. En función de si el tren se encuentra en una rampa o una pendiente, esta fuerza puede oponerse o no al movimiento, teniendo en cuenta este factor a través del signo.

- Aceleración

Resulta al resolver la segunda Ley de Newton considerando el esfuerzo de tracción y las fuerzas que se oponen al movimiento en cada paso de simulación

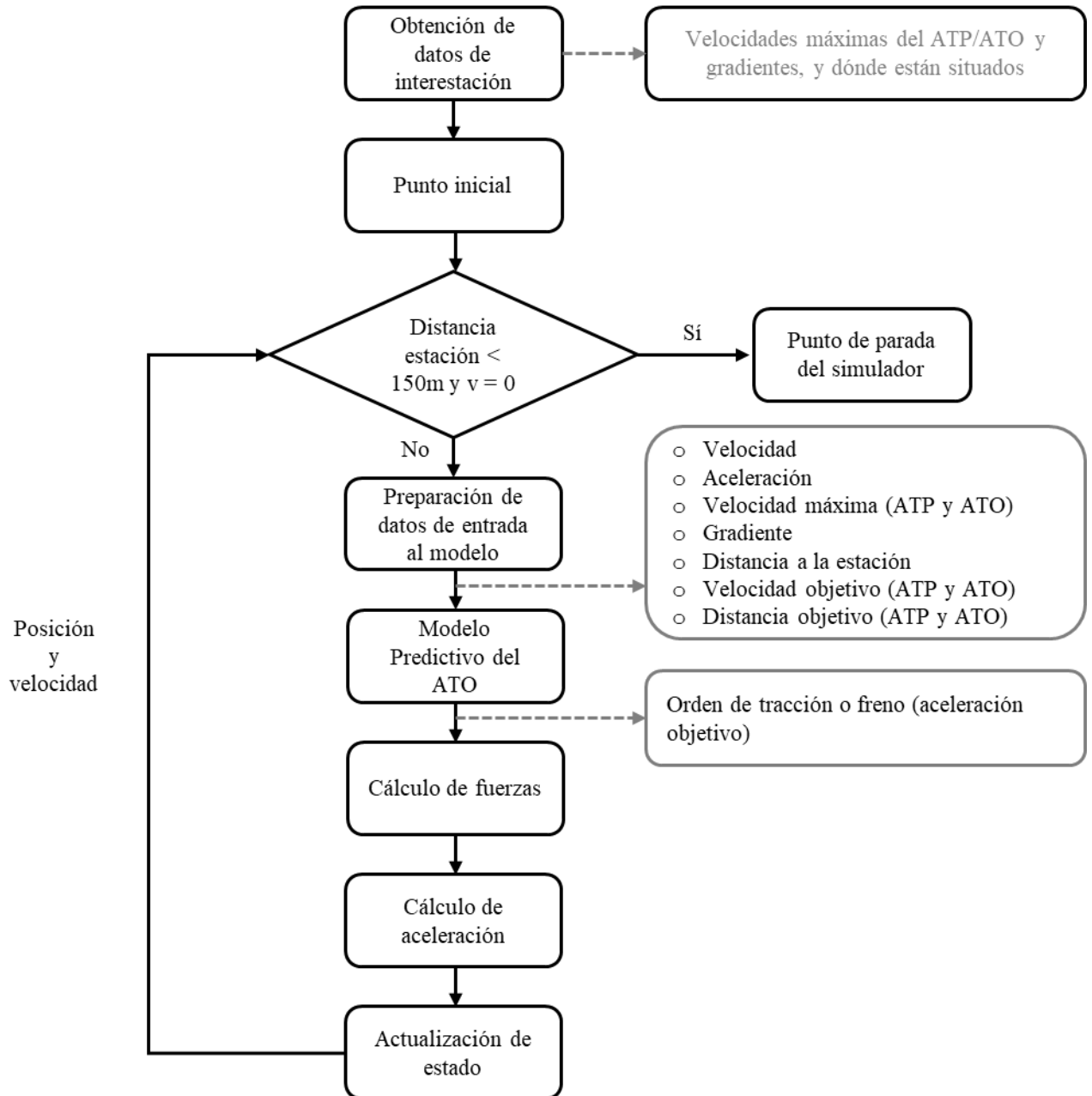
$$a = \frac{ET - F_{RR} - F_{slope}}{M_{vacio} + M_{inercial}}$$

Los valores de la velocidad y distancia recorrida en el siguiente paso de simulación se han obtenido de las ecuaciones de movimiento rectilíneo uniforme y uniformemente acelerado.

El paso de tiempo considerado ha sido 0.5s, igual a la diferencia de tiempos que tienen los registros originales.

7.2.2 Estructura

Una vez se ha calculado la nueva velocidad y posición del tren, se pueden utilizar los valores correspondientes a ese punto como entrada para la red neuronal. Con la salida de la red se calcula el nuevo esfuerzo de tracción, y se repite el proceso hasta que llega a la estación. En el Esquema 5 se puede ver el funcionamiento del simulador.



Esquema 5: Funcionamiento del simulador

7.2.3 Resultados obtenidos

Los errores medios en distancia y tiempo de los mejores modelos entrenados se presentan en la Tabla 7. En los errores porcentuales se trata del *mean absolute percentage error* (MAPE) presentado en el Capítulo 5. El error de tiempo se ha calculado con respecto a los valores que hay registrados del tren real recorriendo la L1. El error de distancia es con respecto al punto de parada en la estación. Se ha preferido utilizar un error porcentual frente a un error medio, para independizar el resultado de la longitud de las interestaciones.

Para complementar los errores porcentuales, se incluyen también el máximo y mínimo en la parada en la estación. Sin estos valores, el conjunto de tres modelos sería el segundo mejor en distancias. Al tener el máximo error cometido se puede ver que es uno de los que más se desvía en algunas interestaciones. Estos errores corresponden a las simulaciones de las interestaciones de forma individual.

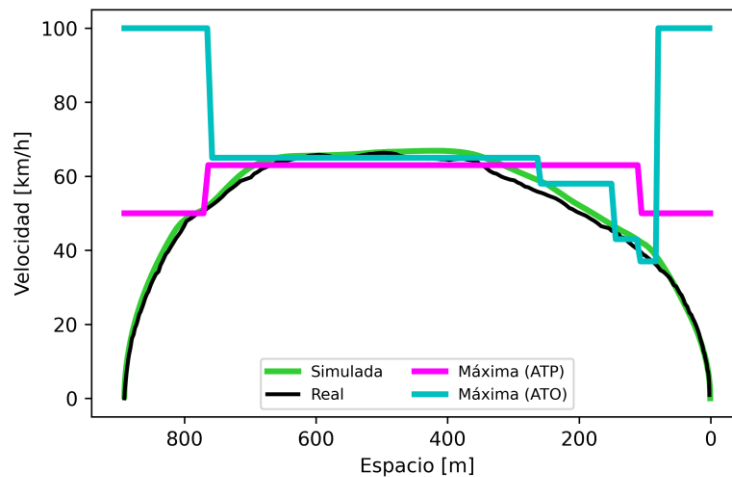
Tabla 7: Errores obtenidos en las simulaciones con los diferentes modelos

	Error tiempo [%]	Error distancia [%]	Mínimo error [m]	Máximo error [m]
Modelo único de tracción y freno				
Modelo 2	6.9%	7.6%	1.40	161.20
Modelo 3a	3.3%	4.9%	2.30	154.90
Modelo 3b	7.0%	5.2%	3.00	112.90
Modelo 4	4.7%	8.2%	4.40	116.87
Modelo 5	4.7%	4.0%	0.97	89.14
Dos modelos				
Trac + Fren	0.3%	5.8%	1.2	179.4
Modelo 4 + BC	0.1%	1.0%	1.60	39.19
Tres modelos				
Clas +Trac + Reg Fren	4.8%	2.1%	0.23	175.2

El error porcentual en este caso se ha calculado teniendo en cuenta todas las simulaciones de las interestaciones. En la Figura 17 se muestra el comportamiento del simulador en una interestación en la que se ve condicionado tanto por la velocidad máxima del ATP como la del ATO. Aunque no es el mejor caso de funcionamiento, se incluye esta interestación por presentar la particularidad de que la velocidad máxima está condicionada por el ATO.

Complementariamente, en la Figura 18 se puede ver la comparación de salidas del ATO para el caso de la interestación de la Figura 17. Por lo general la aceleración del principio es igual

en la simulación y en los datos reales. Es una zona fácil de predecir porque al salir de la estación acelera todo lo que puede para coger velocidad.



Error posición = 0.90 m
Dif tiempos = 74.00(sim) - 75.72(real) = -1.72 s

Figura 17: Comparación de velocidades reales y predichas en una interestación

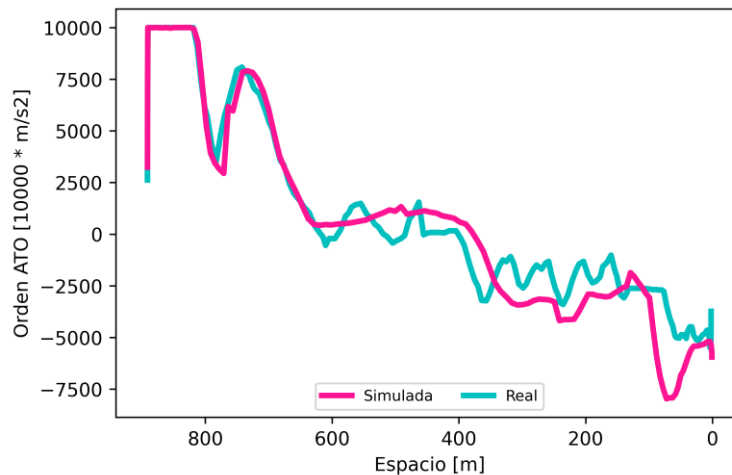


Figura 18: Comparación de salidas de ATO reales y predichas

El esfuerzo de tracción se ve limitado por el máximo que es capaz de dar el tren. En la Figura 19 se incluyen los esfuerzos de tracción aplicados. Estos son los que entran en el balance después de comparar el que quiere aplicar el ATO con el máximo. Como se puede ver, el esfuerzo de tracción máximo condiciona la marcha en algunos puntos. Aunque el ajuste a la curva no es perfecto, de no limitar el esfuerzo de tracción se obtendrían resultados como el que se muestra con los cuadrados rojos.

La diferencia entre uno y otro caso no es muy acusada. Se han realizado simulaciones en los dos supuestos:

- El esfuerzo máximo de tracción limita el aplicable
- No hay esfuerzo máximo de tracción

Se han comparado los resultados obtenidos para medir la influencia de la curva de esfuerzo máximo en la marcha del tren. La diferencia observada se puede ver en la Tabla 8 y en la Tabla 9.

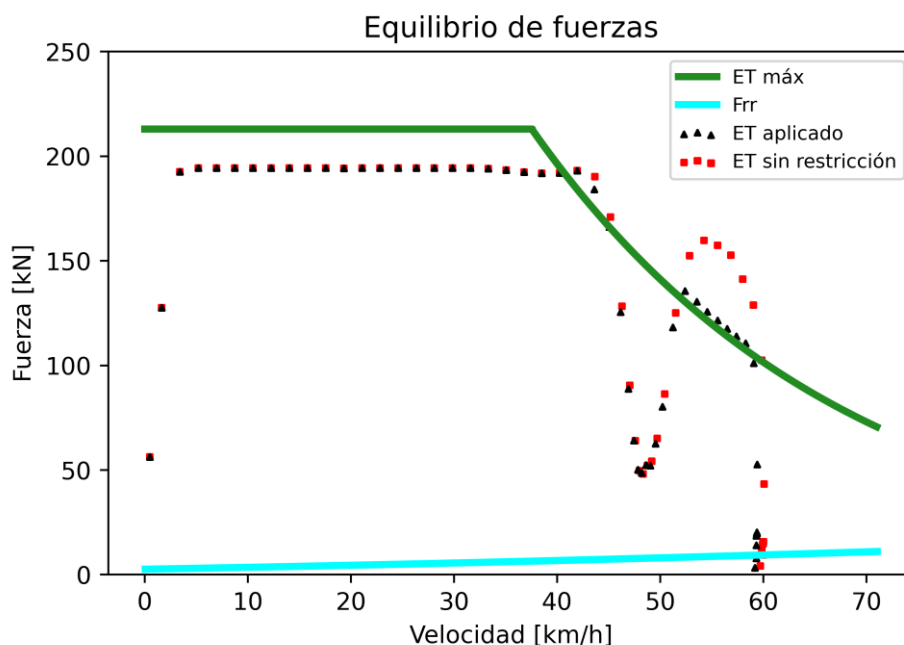


Figura 19: Comparación de fuerzas en una simulación con y sin esfuerzo máximo

En la Tabla 8 se muestra la comparación de errores si se utiliza el esfuerzo máximo de tracción o si se ignora. Para obtenerlo se han tenido en cuenta las simulaciones de todos los modelos presentados. Como se puede ver, si no hubiera esfuerzo máximo de tracción los errores serían más bajos.

Tabla 8: Comparación de la simulación con y sin esfuerzo máximo para todos los modelos

	Error medio [s]	Error medio [m]	Error % tiempo	Error % distancia
Con esfuerzo máximo	3.34	36.7	4.0%	4.8%
Sin esfuerzo máximo	3.28	30.1	3.9%	3.8%

Por contra, en la Tabla 9 se incluyen estos mismos errores pero sólo para el mejor conjunto (el de curva de freno). En este caso se aprecian menos errores al utilizar un esfuerzo máximo de tracción. Estos resultados no son inesperados, ya que los modelos se entrenan con datos reales en los que sí que hay un esfuerzo máximo. Los resultados presentados en la Tabla 8 contienen la agrupación de todos los modelos, entre los que hay algunos muy imprecisos como el 1.

Tabla 9: Comparación de la simulación con y sin esfuerzo máximo para el mejor conjunto

	Error medio [s]	Error medio [m]	Error % tiempo	Error % distancia
Con esfuerzo máximo	-0.05	-5.99	0.1%	0.9%
Sin esfuerzo máximo	-0.43	-7.76	0.6%	1.2%

La deceleración media obtenida a la hora de parar en la estación es de 0.6 m/s^2 , condicionada por las pendientes. Como se busca que el ATO mantenga un esfuerzo de frenado independiente de la pendiente, en tramos en los que es pronunciada el tren frena con una deceleración menor, para no superar el esfuerzo de freno máximo. En los datos utilizados para el entrenamiento se observan deceleraciones más bajas en zonas con pendientes acusadas.

En metros, con distancias cortas, se supone que los pasajeros toleran aceleraciones y deceleraciones más altas. La deceleración máxima con el modelo de ATO es de unos 0.9 m/s^2 , por lo que no habría problemas de confort de los viajeros.

Aunque en este trabajo se han simulado únicamente las interestaciones por separado, existe la posibilidad de haber simulaciones de tramos completos. En la Figura 20 se incluye la gráfica de la simulación de un tramo de más de una interestación.

Las predicciones individuales son casi instantáneas (0.016s), pero el simulador tarda una media de 15s por interestación. El proceso de desnormalización de los resultados supone 0.05s. Por esta razón, para realizar el diseño de marchas, donde se necesitan muchas simulaciones, la mejor opción es utilizar el modelo de tiempos y consumos, y utilizar este modelo para el análisis en más detalle de marchas en concreto.

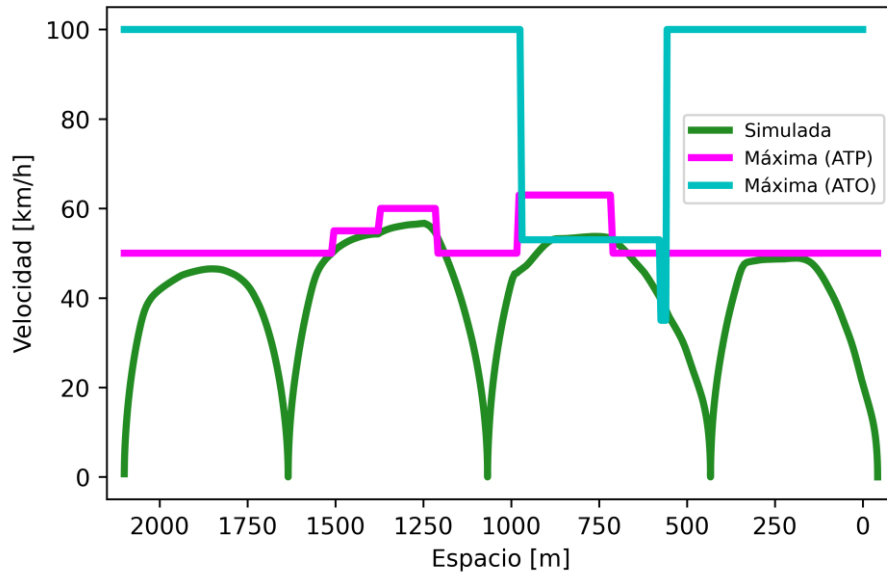


Figura 20: Simulación de un tramo

Capítulo 8. INTERFACES DE USUARIO

Los tres tipos de interfaces han sido diseñados utilizando *PyQt5* (Qt), pensada para diseñar GUIs (Graphical User Interface). La herramienta QtDesigner que incorpora fue muy útil en el diseño de la parte gráfica. Las interfaces empezaron con un diseño muy sencillo. El tamaño y tipo de letra por defecto y los colores grises que ofrece la herramienta también por defecto. Posteriormente se añadió el logo de Comillas de fondo y más colores. Para los colores se utilizaron sobre todo el amarillo y el blanco por ser los colores corporativos de la universidad.

El objetivo de estas interfaces es poder hacer predicciones con los modelos sin necesidad de abrir el código y cambiar valores. Al ser de uso puramente para pruebas son muy sencillas.

Las interfaces, al igual que el resto del trabajo, han sido programadas utilizando Python. El resultado de la programación es un archivo ejecutable *.exe* que se puede utilizar en cualquier ordenador, no requiriendo tener instalado Python.

8.1 MODELO DE TIEMPO-CONSUMO

La interfaz para el cálculo del tiempo y el consumo se puede ver en la Figura 21. Dispone de dos casillas para introducir los valores de código de velocidad y distancia de deriva. Para las entradas, al ser sólo dos valores, se han utilizado *Spin Box*. El cuadro de salida no permite escribir sobre él, y sobrescribe los valores con cada nueva predicción.

Se incluye un ejemplo de cálculo con la aplicación. En este caso el modelo cargado es el de kNN, aunque se puede utilizar cualquiera de ellos. Los valores reales de tiempo y consumo que corresponden a estas entradas son 104.8 s y 5.34 kWh.

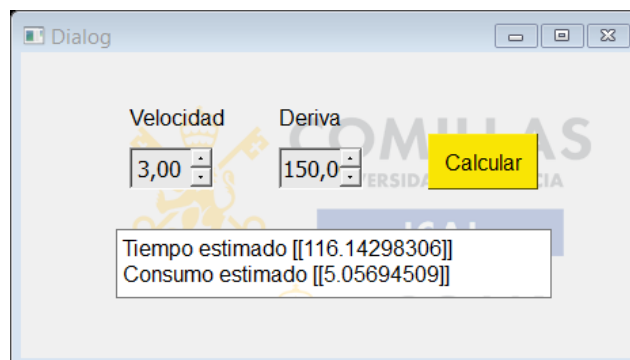


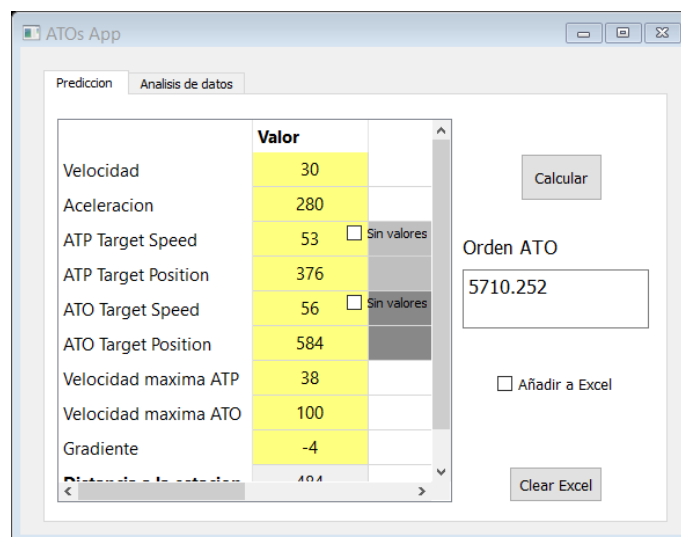
Figura 21: Interfaz para el cálculo de tiempo-consumo

8.2 MODELO DE ORDEN ATO

La interfaz que permite el cálculo de las salidas de ATO se puede ver en la Figura 22. Los valores de las entradas se introducen a través de la tabla. Al haber gran cantidad de entradas es más cómoda, ya que permite desplazarse con las flechas del teclado. La opción de sin valores cubre los casos en los que el punto está cerca del final de la línea. En éstos las velocidades y distancias objetivo toman valores por defecto.

Dentro de la interfaz existe la posibilidad de añadir los cálculos a una hoja Excel. En ella se escriben los valores de las entradas y las salidas que ha predicho el modelo. Hay asimismo la opción de borrar el Excel de resultados.

Una mejora posterior fue la posibilidad de tomar las entradas desde una hoja Excel en vez de utilizar la tabla. En el Excel se ponen los valores de las entradas en filas consecutivas y la aplicación predice las salidas y las escribe al lado.



	Valor	
Velocidad	30	
Aceleracion	280	
ATP Target Speed	53	<input type="checkbox"/> Sin valores
ATP Target Position	376	
ATO Target Speed	56	<input type="checkbox"/> Sin valores
ATO Target Position	584	
Velocidad maxima ATP	38	
Velocidad maxima ATO	100	
Gradiente	-4	
Distancia de frenado	404	

Calcular

Orden ATO: 5710.252

Añadir a Excel

Clear Excel

Figura 22: Interfaz para el cálculo de la consigna de ATO

Dentro de la misma aplicación se ha incluido una pestaña que permita hacer un análisis de los datos (Figura 24). En ella se pueden representar gráficos 2D o 3D de cualquier combinación de variables. Estas variables son las correspondientes a los datos originales que se tienen de línea 1 de Metro de Barcelona, en formato de hoja Excel. Se podrían añadir datos a este Excel o sustituirlo por otro de otra línea.

Para la realización de estos gráficos se ha utilizado *plotly*, una librería *open source* (Plot15). En la Figura 23 se incluye un ejemplo de un gráfico 3D obtenido con *plotly*, dentro de la pestaña de análisis. El gráfico se puede rotar libremente.

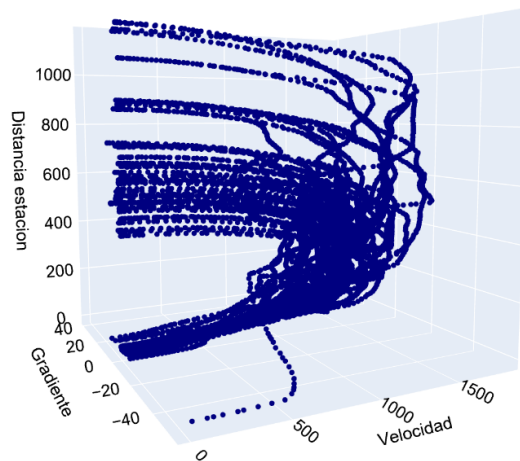


Figura 23: Gráfico de la pestaña de análisis

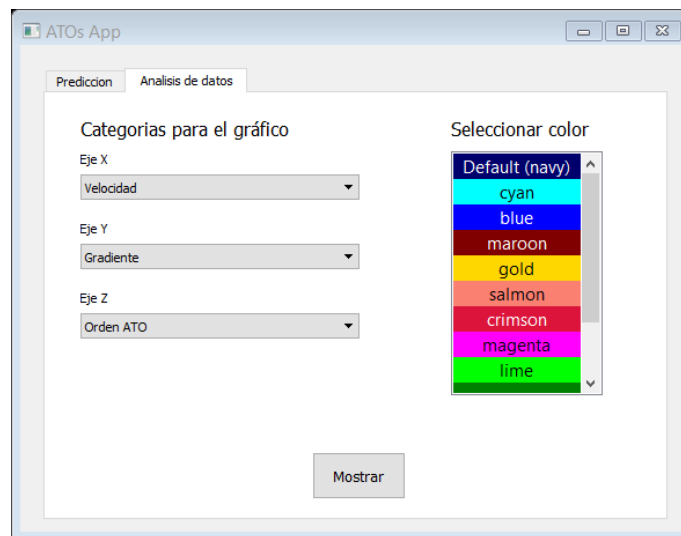


Figura 24: Pestaña para el análisis de datos dentro de la interfaz de ATO

8.3 SIMULADOR

En la Figura 25 se puede ver la interfaz para utilizar el simulador desarrollado. En las casillas para las estaciones se deben introducir los códigos de las mismas. En este caso, al ser de nuevo sólo dos valores, se han utilizado *Spin Box*. Es posible hacer simulaciones de más de una interestación. Basta con poner códigos de estaciones que no estén consecutivas.

Existe la posibilidad de realizar cuatro tipos de gráficos diferentes, con la opción de hacer una única figura con los cuatro. En la Figura 26 se ha incluido uno de estos gráficos integrados. El de tiempo-espacio en estos casos no es relevante. Al tratarse de una única interestación no aporta información. Este gráfico está pensado para hacer simulaciones de varias interestaciones seguidas. En un tramo se ve la diferencia en los tiempos de parada y cómo se separan la simulada y la real.

Aunque tiene cargados los datos de la línea 1 de Metro de Barcelona, se podría introducir otro Excel con otra línea. El único requisito es que los códigos de las estaciones sean diferentes entre sí.

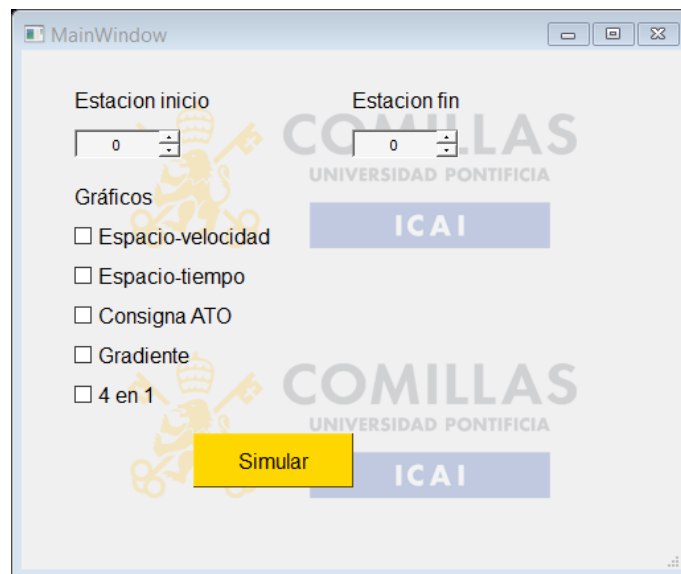


Figura 25: Interfaz para realizar simulaciones

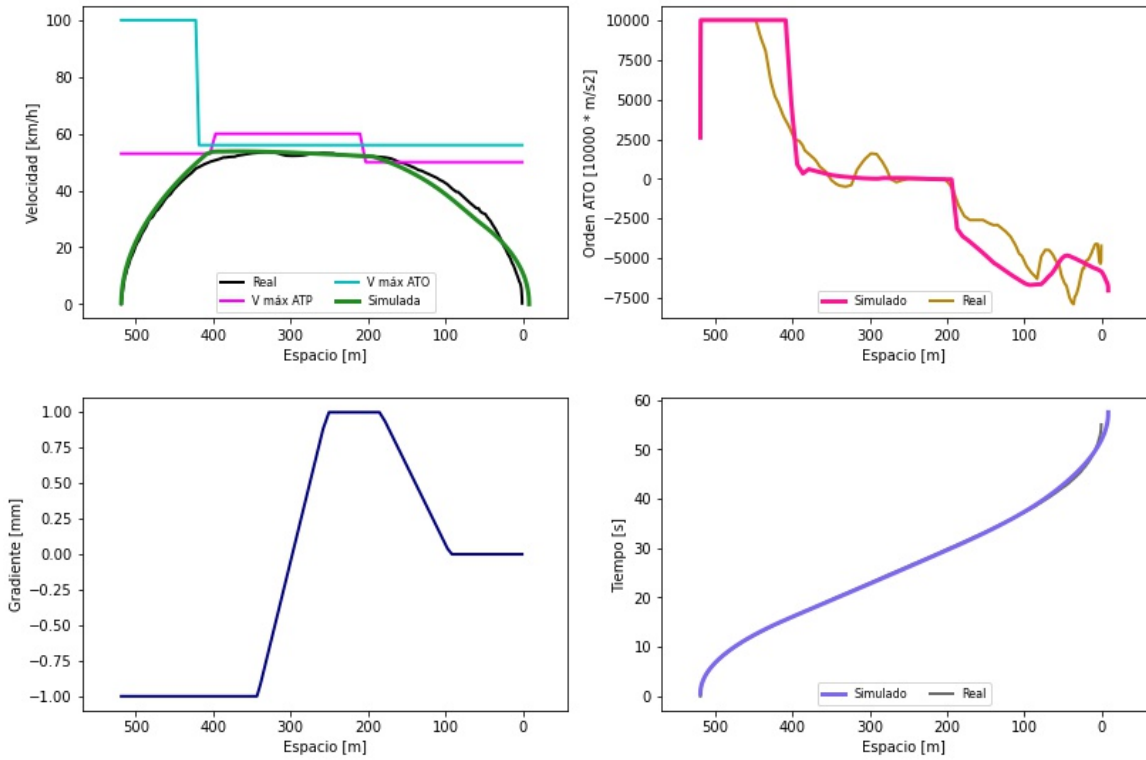


Figura 26: Gráfico combinado para el simulador

Capítulo 9. ANÁLISIS DE RESULTADOS

En el modelo de predicción de tiempos y consumos, los errores tan bajos obtenidos hacen que se pueda utilizar de forma fiable en este tramo. Considerando los mejores algoritmos entrenados, el error en la predicción del tiempo está entre un 0.01% y un 0.05%. El error de consumo entre un 0.05% y un 0.3%.

Estos modelos podrían presentar errores más altos en recorridos diferentes. Las salidas dependen de otros parámetros propios de cada interestación, principalmente las pendientes de la vía, que no se han tenido en cuenta. Esta limitación es fácilmente solventable. Se podrían añadir más datos al conjunto de entrenamiento, ampliando el número de entradas al modelo.

Otra solución sería entrenar modelos para cada interestación. El tiempo necesario para entrenar el *RandomForest*, por ejemplo, fue de 7s. El de la red neuronal fue de 6.5 min, su tiempo de entrenamiento es mayor que el del resto de algoritmos. De querer entrenar muchos modelos diferentes sería más económico en cuanto a tiempos utilizar un *RandomForest*. Si no hay gran necesidad de generalización podría ser una opción. Para obtener modelos más robustos habría que atajar el problema del *overfitting*. Como se comentó en el apartado 6.2 este proceso no es sencillo y sería necesario un avance importante.

Se ha podido comprobar que varios algoritmos diferentes proporcionan errores similares. Aunque se ha elegido la red neuronal como solución ganadora, en un futuro se podrían utilizar las otras opciones.

En la parte de modelado de un sistema de conducción automática ATO, realizado sólo a partir de registros entrada-salida, se han obtenido modelos con un error medio en el posicionamiento en la estación de entre un 1% y un 8%. Los errores de tiempo obtenidos van de un 0.1% a un 8%.

En otros simuladores se han obtenido errores de tiempo de un 1% (Domí11) y de un 0.16% en (Jong05). En este rango sólo se encuentran los modelos del denominado conjunto 2. El 2.1, que combina dos redes neuronales, una para tracción y otra para freno, tiene un error de tiempo del 0.3%. Y el 2.2, que es el de un modelo principal más un modelo sólo para la curva de freno, un error del 0.1%.

A su vez, dentro de estos dos subconjuntos destaca el de curva de freno (2.2), denominado 4 + BC. Tiene un error en la distancia al punto de parada de un 1%, y el error máximo más bajo de todos (40m).

Las otras combinaciones consideradas no lograron tener errores sensiblemente menores que un único modelo. El aumento de complejidad en el entrenamiento y simulador no se ve compensado por una mejora del error. Estas combinaciones, sin embargo, fueron las que

desembocaron en el desarrollo del conjunto con curva de freno. Su desarrollo fue útil como paso intermedio.

Como se ha podido comprobar, la existencia de un esfuerzo máximo de tracción es beneficiosa. En el conjunto $4 + BC$ se obtienen errores más bajos si se utiliza. Hay que tener en cuenta que el entrenamiento se realiza con datos en los que hay un esfuerzo máximo detrás.

Por último, las interfaces obtenidas son multiuso. Cambiando los ficheros Excel se pueden utilizar para otros casos. La única condición es que el nuevo archivo tenga el mismo nombre y estructura que el anterior. También se podría cambiar el modelo utilizado, y cargar uno más refinado y con menos error.

Capítulo 10. CONCLUSIONES Y TRABAJOS FUTUROS

Los modelos obtenidos presentan una importante ventaja frente a otros modelos o simuladores existentes: su rapidez. El modelo que predice tiempos y consumos tiene una precisión adecuada, y es útil para el diseño de marchas. Se pueden hacer optimizaciones más rápidas, incluso sería posible realizarlas en aplicaciones de tiempo real.

En cuanto al modelo del sistema de conducción, al estar basado en registros disponibles se trata de un modelo general. Los simuladores convencionales son más lentos que las soluciones de machine learning, con el inconveniente adicional de que el módulo de ATO es propio de cada fabricante. Las predicciones con el modelo de ATO desarrollado son más rápidas que las basadas en simulación detallada, y no es necesario conocer el funcionamiento interno del ATO para hacer simulaciones.

Se han cubierto todos los objetivos del proyecto. En el caso del modelo de predicción de tiempos y consumos en una intersección, se han obtenido varios tipos de modelos con algoritmos diferentes para poder comparar alternativas. La red neuronal ha sido elegida sobre otros algoritmos como *RandomForest*.

La red neuronal presenta mayor capacidad de generalizar, y resulta más conveniente para predecir en un rango amplio de entradas. De querer entrenar gran cantidad de modelos en poco tiempo, sería necesario utilizar un *RandomForest*. El entrenamiento de la red neuronal lleva 56 veces más tiempo.

En cuanto a la parte de modelado del sistema de conducción automática ATO, se han entrenado una gran variedad de modelos. Además de los simples cambios en la estructura de la red, se han barajado otras opciones integrando varios modelos en un conjunto para tratar de reducir los errores. Como se comentó en el Capítulo 9, los errores más bajos corresponden a las combinaciones de dos modelos.

Estos errores están en el rango de otros simuladores existentes. Los modelos y simulador desarrollados se pueden utilizar para ayudar en el diseño de marchas y en la regulación del tráfico. Como se ha comentado, la gran ventaja que proporcionan los modelos de machine learning es su rapidez. Ambos modelos, el de predicción de tiempos y consumos, y el de modelado del ATO, se pueden usar como filtro. Permiten rechazar rápido de entrada las alternativas que presenten peores resultados.

El simulador en sí y las posibilidades que ofrece podría tener refinamientos sucesivos, para hacerlo cada vez más realista. Un trabajo futuro podría ser la ampliación del simulador. Sería necesaria la incorporación de la resistencia en curvas y la adherencia. Un paso más sería considerar el tren con su longitud, en vez de una masa puntual, e introducir el uso del freno regenerativo. Otro trabajo futuro podría ser incluir un modelo de machine learning adicional

para modelar la cadena de tracción del tren, de manera que se modele el consumo y los retardos en la aplicación de las consignas de tracción/freno.

Ahora mismo es posible hacer la simulación de tramos completos de varias interestaciones. De querer explotar esta característica en un futuro sería necesario un mayor diseño de la parte gráfica. Los gráficos obtenidos de tramos son los mismos que los de interestaciones. Esto hace que al simular más de cuatro o cinco interestaciones las líneas se junten y se confundan. Para utilizarlo con tramos largos habría que incorporar un *slider widget* (disponible entre los *widgets* de *matplotlib*), de forma que se pueda desplazar el gráfico y ver las interestaciones en un tamaño razonable.

Los dos tipos de modelos y el simulador han sido integrados en interfaces que permitan obtener resultados sin tener que abrir el código. Un paso más sería la integración de todos estos modelos y fragmentos de interfaces en una única aplicación. Esta aplicación tendría más funcionalidades y posibilidades que las interfaces sueltas. Además de facilitar su uso englobando todos los modelos.

Capítulo 11. BIBLIOGRAFÍA

- Brei01. (10 de 2001). Breiman, Leo. Random Forests. *Machine Learning*, 45, 5-32. Obtenido de <https://doi.org/10.1023/A:1010933404324>
- Brei84. (1984). Breiman, Leo and Friedman, Jerome and Stone, Charles J and Olshen, Richard A. *Classification and regression trees*. CRC press.
- Cast16. (10 de 2016). Casteel, Martijn. Simulation of train driver behaviour using a data-driven agent. Master's thesis, Utrecht University, Faculty of Science - Department of Information and Computing Sciences.
- CE13. (2007-2013). Comisión Europea. How to avoid common errors identified in cost claims. Note for beneficiaries in the 7th Framework Programme for Research and Innovation.
- CE18(1). (2018). Comisión Europea. Artificial Intelligence for Europe. Communication from the Commission, Bruselas.
- CE18(2). (2018). Comisión Europea. Coordinated Plan on Artificial Intelligence. Communication from the Commission, Bruselas.
- CE19. (2019). Comisión Europea. Building Trust in Human-Centric Artificial Intelligence. Communication from the Commission , Bruselas.
- Chol15. (2015). Chollet, François et al.. Keras.
- Cove67. (1967). Cover, T. and Hart, P. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1), 21-27. doi:10.1109/TIT.1967.1053964
- Cuca10. (4 de Agosto de 2010). Cucala, Paloma; Fernández, Antonio. Energy efficiency in high-speed train operation. *Global Railway Review*.
- Davi26. (1926). Davis, William J.. The tractive resistance of electric locomotives and cars. General Electric.
- Domí1. (9 de 2011). Domínguez, M.; Fernández-Cardador, A.; Cucala, A. P.; Lukaszewicz, P.. Optimal Design of Metro Automatic Train Operation Speed Profiles for Reducing Energy Consumption. *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit*, 225, 463-474.
- Domí14. (2014). Domínguez, María; Fernández-Cardador, Antonio; Cucala, Asunción P.; Gonsalves, Tad; Fernández, Adrián. Multi objective particle swarm optimization

- algorithm for the design of efficient ATO speed profiles in metro lines. *Engineering Applications of Artificial Intelligence*, 29, 43-53. doi:<https://doi.org/10.1016/j.engappai.2013.12.015>
- Duch10. (3 de 2010). Duchi, John; Hazan, Elad; Singer, Yoram. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. Tech. rep., EECS Department, University of California, Berkeley. Obtenido de <http://www2.eecs.berkeley.edu/Pubs/TechRpts/2010/EECS-2010-24.html>
- Dumo09. (1 de 2009). Dumont, Marie; Marée, Raphaël; Wehenkel, Louis; Geurts, Pierre. Fast Multi-class Image Annotation with Random Subwindows and Multiple Output Randomized Trees., 2, págs. 196-203.
- ERA18. (2018). European Railway Agency - Arcadis. Feasibility study reference system ERTMS - Digitalisation of CCS and migration to ERTMS. Final Report.
- EU21. (2021). Unión Europea. European year of rail - A little bit of history! Recuperado el Febrero de 2021, de https://europa.eu/year-of-rail/why-rail_en
- EUAR20. (25-11-2020). European Union Agency for Railways. Single Programming Document 2021-2023. MB Decision n° 243 – Annex. 54th MB meeting .
- EuEA20. (s.f.). European Environment Agency. Specific CO2 emissions per passenger-km and per mode of transport in Europe. Obtenido de https://www.eea.europa.eu/data-and-maps/daviz/specific-co2-emissions-per-passenger-3#tab-chart_2
- EUPa20. (Plenary – October II 2020). European Parliament. An EU framework for artificial intelligence .
- Extr20. (s.f.). Extreme Outer Vision. OuterVision® Power Supply Calculator. Recuperado el Diciembre de 2020, de <https://outervision.com/power-supply-calculator>
- Fawc16. (8 de 2016). Fawcett, Tom. Learning from Imbalanced Classes. Obtenido de <https://www.svds.com/learning-imbalanced-classes/>
- Fern18. (2018). Fernández Rodríguez, Adrián. Train eco-driving optimisation based on simulation models. Master's thesis, Universidad Pontificia Comillas, Escuela Técnica Superior de Ingeniería (ICAI).
- Fix51. (1951). Fix, E. and Hodges, J. L. Discriminatory Analysis - Nonparametric discrimination: consistency properties. University of California, Berkeley, USAF School of Aviation Medicine. Randolph Field, Texas.
- Géro17. (2017). Géron, Aurélien. Hands-On Machine Learning with Scikit-Learn and TensorFlow.

- GobE20(1). (2020). Gobierno de España. MINECO. Informe España Digital 2025.
- GobE20(2). (2020). Gobierno de España. MINECO. ENIA - Estrategia Nacional de Inteligencia Artificial.
- Hunt07. (2007). Hunter, J. D.. Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*, 9, 90-95. doi:10.1109/MCSE.2007.55
- Iber21. (2021). Iberdrola. Hogares - Luz - Plan Estable. Obtenido de <https://www.iberdrola.es/luz/plan-estable>
- Inco20. (2020). Incontrol Simulation Software. Recuperado el Noviembre de 2020, de <https://www.incontrolsim.com/application-areas/railway-simulation/>
- INE21. (2005-2020). INE. Estadísticas de transporte de viajeros.
- Jong05. (2005). Jong, J. and Chang, S. Algorithms for generating train speed profiles. *Journal of the Eastern Asia Society for Transportation*, 5. doi:10.11175/easts.6.356
- King14. (12 de 2014). Kingma, Diederik; Ba, Jimmy. Adam: A Method for Stochastic Optimization. *International Conference on Learning Representations*.
- Mart19. (2019). Martínez, Pablo. Estudio y optimización del consumo energético del ferrocarril mediante redes neuronales y algoritmos heurísticos. Master's thesis, Universitat Politècnica de València. doi:<https://doi.org/10.4995/Thesis/10251/134018>
- MdeM20. (2020). Metro de Madrid. Quiénes somos. Recuperado el Enero de 2021, de <https://www.metromadrid.es/es/quienes-somos/somos-centenarios>
- Mont17. (2017). Montes, Fernando. Los Sistemas de Control de Tráfico y Señalización en el Ferrocarril. Biblioteca Comillas. Ingeniería.
- Morg63. (1963). Morgan, James N and Sonquist, John A. Problems in the analysis of survey data, and a proposal. *Journal of the American statistical association*, 58(302), 415-434.
- Nies19. (3 de 2019). Niestadt, Maria; Debyser, Ariane; Scordamaglia, Damiano; Pape, Marketa. Artificial intelligence in transport - Current and future developments, opportunities and challenges. EPRS - European Parliamentary Research Service.
- OTLE21. (2021). OTLE - Jornada Anual (8º edición) - Informe Anual 2020.
- OTra20. (2020). OpenTrack Railway Technology. Recuperado el Noviembre de 2020, de http://www.opentrack.ch/opentrack/opentrack_e/opentrack_e.html

- Pedr11. (2011). 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, Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12, 2825-2830.
- Plot15. (2015). Plotly Technologies Inc. Collaborative data science. Obtenido de <https://plot.ly>
- Powe15. (6 de 2015). Powell, J. P.; Palacín, R.. Passenger Stability Within Moving Railway Vehicles: Limits on Maximum Longitudinal Acceleration. *Urban Rail Transit*, 1, 95-103. doi:10.1007/s40864-015-0012-y
- Qt. (s.f.). Qt Group. <https://www.qt.io/>.
- Rail19. (2019). Rail Vision, SBB. Rail Vision Shunting Yard Solution. <https://railvision.io/>: <https://vimeo.com/294567425>.
- Rodr20. (2020). Pilar Rodríguez. Cátedra Telefónica Economía Digital (ETSIT UPM). Actuaciones del Parlamento Europeo en materia de Inteligencia Artificial. Obtenido de <http://economydigital.etsit.upm.es/actuaciones-del-parlamento-europeo-en-materia-de-inteligencia-artificial/>
- S2R19. (30 de Septiembre de 2019). Shift2Rail. Innovation in the Spotlight: Towards unattended mainline train operations (ATO GoA 4). Obtenido de <https://shift2rail.org/highlight/innovation-in-the-spotlight-towards-unattended-mainline-train-operations-ato-go4/>
- S2R19. (30 de Sept de 2019). Shift2Rail. Innovation in the Spotlight: Towards unattended mainline train operations (ATO GoA 4). Obtenido de <https://shift2rail.org/highlight/innovation-in-the-spotlight-towards-unattended-mainline-train-operations-ato-go4/>
- S2R21. (2021). Shift2Rail. X2RAIL-4. Recuperado el Enero de 2021, de https://projects.shift2rail.org/s2r_ip2_n.aspx?p=X2RAIL-4
- Sche17. (2017). Scheepmaker, Gerben M.; Goverde, Rob M. P.; Kroon, Leo G.. Review of energy-efficient train control and timetabling. *European Journal of Operational Research*, 257, 355-376. doi:<https://doi.org/10.1016/j.ejor.2016.09.044>
- Thal19. (2019). Thales. Artificial intelligence for the autonomous trams. Obtenido de <https://www.thalesgroup.com/en/germany/news/artificial-intelligence-autonomous-trams>
- TMBa20. (2020). Transports Metropolitans de Barcelona. Mapa metro Barcelona. Obtenido de <https://www.tmb.cat/es/transporte-barcelona/mapa/metro>

- UIC17. (2017). UIC; IEA. Railway Handbook - Energy consumption and CO2 emissions (Focus on passenger rail services).
- UITP18. (2018). UITP. World report on metro automation.
- UITP20. (2020). UITP. Public Transport is COVID-safe.
- Wang20. (11 de 2020). Wang, Ziran; Liao, Xishun; Wang, Chao; Oswald, David; Wu, Guoyuan; Boriboonsomsin, Kanok; Barth, Matthew; Han, Kyungtae; Kim, BaekGyu; Tiwari, Prashant. Driver Behavior Modeling Using Game Engine and Real Vehicle: A Learning-Based Approach. IEEE Transactions on Intelligent Vehicles, 5, 738-749. doi:10.1109/TIV.2020.2991948

ANEXO I: DESARROLLO DE LOS MODELOS

MODELOS DE TIEMPO/CONSUMO

Se incluye el código de los modelos utilizados para predecir tiempos y consumos. En aquellos algoritmos pensados para hacer una regresión con una única salida se ha utilizado *MultiOutputRegressor*. Esta clase permite obtener más de una salida de algoritmos que por naturaleza sólo predicen una salida. Funciona ajustando un modelo por salida. Ambos modelos comparten el mismo código, pero son entrenados de forma individual. El *MultiOutputRegressor* hace de envoltorio, de forma que externamente se tiene un modelo que devuelve varias salidas.

- NeuralNetwork

Se ha utilizado el código de la siguiente red neuronal. Se ha entrenado con los datos normalizados usando la *mean normalization*. Es la red que ha tenido menores errores.

```
model = Sequential()  
model.add(Dense(512, input_dim=2, activation='relu'))  
model.add(Dense(512, activation='relu'))  
model.add(Dense(2, activation='linear'))  
model.compile(loss='mean_squared_error', optimizer='adam')
```

- DecisionTree

Este algoritmo apareció en (Dumo09). En la Figura 27 se puede ver el esquema de la predicción de un árbol con una profundidad máxima de 3. Con profundidades mayores el texto es difícil de leer en un documento.

```
dt_reg = DecisionTreeRegressor(max_depth=17)
```

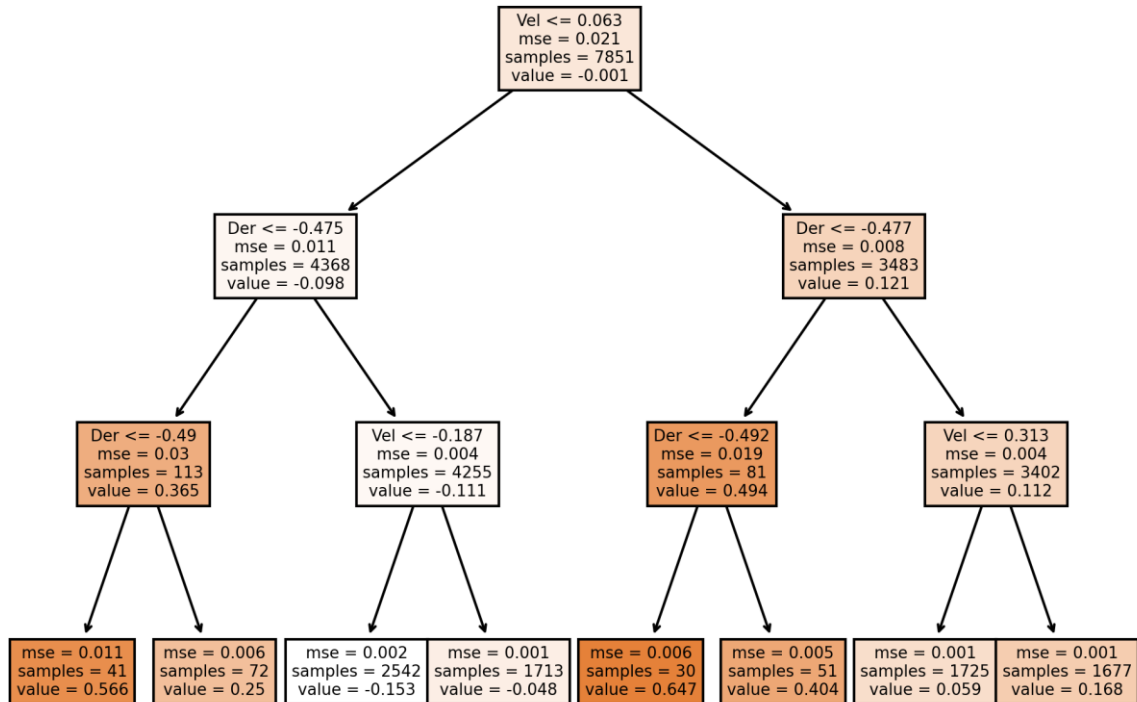



Figura 27: Árbol de decisión con una profundidad máxima de 3

- Support Vector Regression

```
svr = svm.SVR(kernel='rbf', gamma = 'scale', tol=0.00001, epsilon = 0.01)
```

- RandomForest

Este algoritmo apareció en (Brei01), y es propiedad de Leo Breiman y Adele Cutler

```
rf_reg = RandomForestRegressor(n_estimators=150, max_depth=20)
```

- kNN

```
neigh = KNeighborsRegressor(n_neighbors=10, weights = 'distance', algorithm='auto',  
metric='minkowski', p=1)
```

MODELOS DE ATO

En este caso todos los modelos son redes neuronales. Para optimizar el tiempo dedicado al entrenamiento se ha utilizado el siguiente *callback*

```
callback = EarlyStopping(monitor='loss',  
    patience=10,  
    restore_best_weights=True)
```

La función del *EarlyStopping* es detener el entrenamiento cuando la *loss* no disminuye, en este caso tras 10 iteraciones. El parámetro adicional de *restore_best_weights* recupera los pesos de la iteración con menos *loss*. Gracias a esta función es posible entrenar modelos más rápido, asegurando que los pesos finalmente utilizados son los de la mejor iteración.

- Modelo 1

```
model = Sequential()  
model.add(Dense(500, input_dim=10, activation='relu'))  
model.add(Dense(350, activation='relu'))  
model.add(Dense(150, activation='relu'))  
model.add(Dense(25, activation='relu'))  
model.add(Dense(1, activation='linear'))  
model.compile(loss='mean_squared_error', optimizer='adam')
```

- Modelo 2 y Fren

```
model = Sequential()  
model.add(Dense(200, input_dim=10, activation='relu'))  
model.add(Dense(100, activation='relu'))  
model.add(Dense(50, activation='relu'))  
model.add(Dense(1, activation='linear'))  
model.compile(loss='mean_squared_error', optimizer='adam')
```

- Modelo 3 y Trac

```
tf.keras.optimizers.Adagrad(  
    learning_rate=0.003,  
    initial_accumulator_value=0.001,  
    epsilon=1e-07,  
    name="adagrad"  
)  
  
model = Sequential()  
model.add(Dense(200, input_dim=10, activation='relu'))  
model.add(Dense(150, activation='relu'))  
model.add(Dense(100, activation='relu'))  
model.add(Dense(50, activation='relu'))  
model.add(Dense(25, activation='relu'))  
model.add(Dense(1, activation='linear'))  
model.compile(loss='mean_squared_error', optimizer='adagrad')
```

○ Modelo 4 y Reg fren

Este modelo utiliza los valores de *AdaGrad* que vienen por defecto.

```
tf.keras.optimizers.Adagrad(  
    learning_rate=0.001,  
    initial_accumulator_value=0.1,  
    epsilon=1e-07,  
    name="adagrad"  
)  
  
model = Sequential()  
model.add(Dense(200, input_dim=10, activation='relu'))  
model.add(Dense(150, activation='relu'))  
model.add(Dense(100, activation='relu'))  
model.add(Dense(50, activation='relu'))  
model.add(Dense(25, activation='relu'))  
model.add(Dense(1, activation='linear'))  
model.compile(loss='mean_squared_error', optimizer='adagrad')
```

○ Modelo 5

```
model = Sequential()  
model.add(Dense(300, input_dim=9, activation='relu'))  
model.add(Dense(200, activation='relu'))  
model.add(Dense(150, activation='relu'))  
model.add(Dense(75, activation='relu'))  
model.add(Dense(25, activation='relu'))  
model.add(Dense(1, activation='linear'))  
model.compile(loss='mean_squared_error', optimizer='adagrad')
```

○ BC

```
model = Sequential()  
model.add(Dense(200, input_dim=9, activation='relu'))  
model.add(Dense(100, activation='relu'))  
model.add(Dense(50, activation='relu'))  
model.add(Dense(1, activation='linear'))  
model.compile(loss='mean_squared_error', optimizer='adagrad')
```

○ Clas fren

```
rfclf = RandomForestClassifier(max_depth=100, random_state=0)
```

ANEXO II: SMOTE (SYNTHETIC MINORITY OVER-SAMPLING TECHNIQUE)

Es un algoritmo que se utiliza para añadir datos a un *dataset*. Usando SMOTE se puede resolver el problema del *unbalanced learning*. Si hay muchas muestras de un tipo (por ejemplo A) y muy pocas de otro (B), la red aprende a predecir el mayoritario. Se equivoca muy pocas veces si dice siempre que la solución es A. Este problema es típico en el diagnóstico de enfermedades como el cáncer.

En este caso, hay más muestras en las que el tren tracciona que en las que frena. Para intentar que aprendiera a predecir mejor la frenada, se entrenaron algunos modelos con datos rellenos.

En vez de replicar datos en las zonas despobladas o quitarlos en las masificadas, SMOTE crea nuevos datos a partir de los que ya hay. La opción de replicar datos hace que cambie la varianza, afectando a las predicciones del modelo. La opción de eliminar datos reduce el tamaño del espacio de entradas, con el riesgo de eliminar valores importantes que se quiere enseñar al modelo.

En SMOTE, en vez de duplicar los datos minoritarios, se generan nuevos datos a partir de los que ya hay. Se coge cada punto y se sacan sus k-NearestNeighbours. Con esos vecinos se crean nuevos datos interpolando, de forma que los nuevos datos quedan en algún punto de la “línea” que une el punto original con cada vecino (Figura 28).

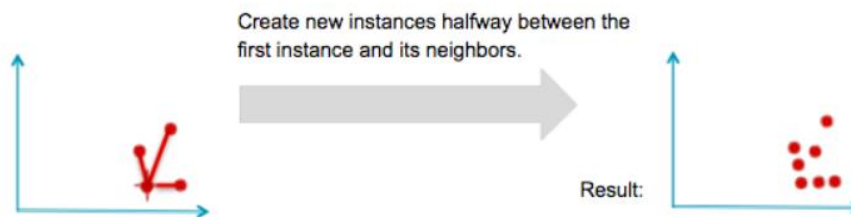


Figura 28: Funcionamiento del algoritmo SMOTE (Fawc16)

Una limitación de SMOTE es que como interpola entre los datos, todos los nuevos que saca están dentro de los bordes del conjunto de datos minoritarios, no crea datos fuera del espacio de los que ya hay.

Al ser SMOTE un algoritmo pensado para clasificación, hubo que hacer una adaptación para usarlo en una regresión. La solución utilizada fue añadir al *dataset* una columna extra en la que se clasificaban los datos en función de si eran mayoritarios o no. SMOTE se aplicaba usando esta columna extra.

No se ha llegado a utilizar ADASYN, una variante de SMOTE mejorada. En vez de crear los nuevos datos en la línea que une los Neighbours, añade un poco de desviación/ruido para que sea más realista. El uso de SMOTE se abandonó antes de probar esta alternativa.

Tanto SMOTE como su variante ADASYN pertenecen al paquete de *imbalanced learning* de Python.

ANEXO III: ALINEAMIENTO DEL PROYECTO CON LOS ODS

El trabajo está relacionado con los ODS 11 y 12.



Con el 11 porque este objetivo busca un transporte más sostenible. El ferrocarril es un medio de transporte muy limpio. Todas las mejoras en el servicio (como usar un ATO) ayudan a que el tren sea más atractivo que otros medios de transporte más contaminantes.

En la Unión Europea, la fracción de transporte de viajeros que se hace por carretera es un 82.2%, mientras que por ferrocarril es un 7.6% (UIC17). En España la situación es similar, el 6.4% de los viajeros utilizan el ferrocarril, frente al 85.5% que utilizan la carretera. Incluso dentro de los modos de transporte colectivos el tren es de los menos utilizados, superando únicamente al transporte marítimo (OTLE21). Hay margen para reconducir parte de ese tráfico hacia el tren.

El transporte por carretera emite 100 gCO₂/pasajero·km, frente a los 30 que emite el ferrocarril (EuEA20). Todo lo que incentive el uso del ferrocarril sobre la carretera/transporte aéreo (todavía más contaminante, 250 gCO₂/pasajero·km) favorece la consecución del ODS 11. En la Figura 29 se puede ver la comparación de las emisiones de los medios de transporte en el UE.

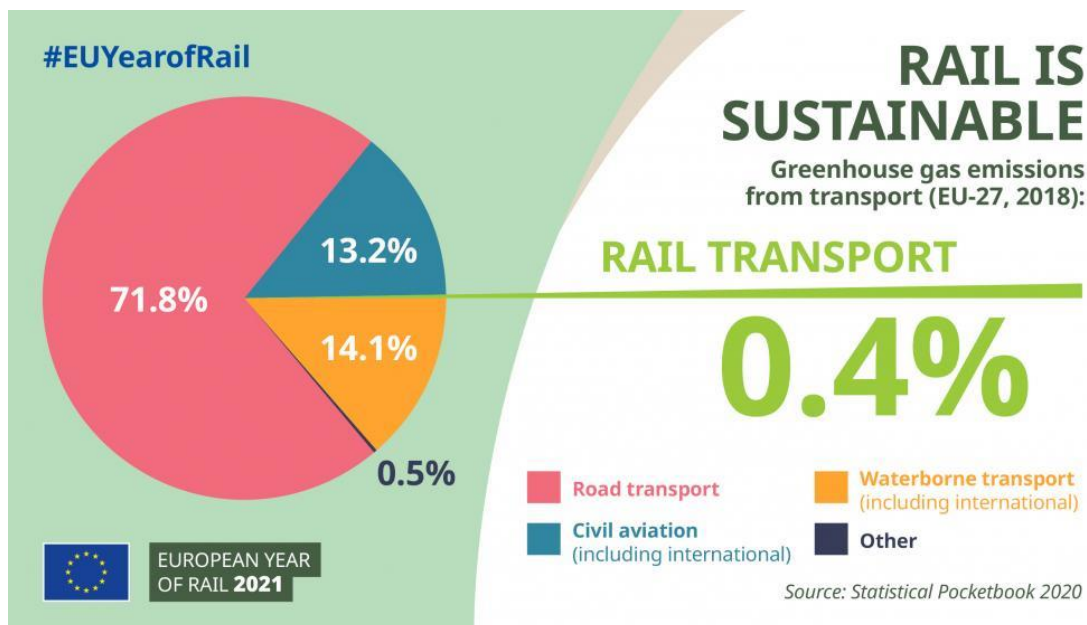


Figura 29: Emisiones debidas a los diferentes medios de transporte (EU21)

El objetivo de la UE de ser *carbon neutral* en 2050 hará que propuestas como la que se desarrolla en este trabajo se generalicen. El *European Green Deal* contempla el desarrollo de un transporte público más limpio. La declaración de 2021 como Año Europeo del Ferrocarril también será un impulso al ATO, dentro del marco de interoperabilidad europea.



Con el objetivo 12 está relacionado porque este objetivo se preocupa por el consumo de energía. Un tren necesita unas cantidades de energía del orden de 350 kJ/pasajero·km³ en Europa (UIC17), usando un ATO se puede reducir este consumo. En Shift2Rail planean tener para 2022 un ATO que ahorrará hasta un 45% de energía (S2R19).

También dentro del *European Green Deal* se busca que el uso de los recursos sea más responsable. Gracias al ATO se puede predecir con más facilidad el mantenimiento a realizar. Elementos que duran más y son más fáciles de reparar ayudan a cumplir el *Circular Economy Action Plan*, y acercan a un consumo más responsable

³ A modo de comparación, al ir andando se consumen unos 200 kJ/km

