Escuela Técnica Superior de
Ingeniería de Caminos, Canales y Puertos
UNIVERSIDAD DE CANTABRIA

COMPARATIVE ANALYSIS OF TIME SERIES MODELS FOR SHORT-TERM URBAN TRAFFIC FORECASTING

Trabajo realizado por:
Carmen Ortega Salan

Dirigido:
Borja Alonso Oreña
Andrés Rodríguez Gutiérrez

Titulación:
Máster Universitario en Ingeniería de Caminos, Canales y Puertos

Santander, julio de 2025

Abstract

With the increased growth of urban areas, the management of traffic flow in an efficient manner is becoming more and more complex. Traffic forecasting tools have proven to be quite useful in the decision-making process. In this paper we will be studying short-term traffic in one of the main arteries of the city of Santander, Spain. We will be developing and comparing the results from three different prediction models, ARIMA, Holt-Winters and Prophet, which have shown to be reliable in the past. We will then choose the most reliable one and try to develop it further, applying different techniques and studying which ones work well for our type of data and area.

Our results show us that Prophet, even though visually correct and user-friendly, doesn't give us accurate results for short-term high-frequency data. ARIMA is able to predict the general trends, but on the other hand it requires much more work from the user with fine tuning and it can struggle with seasonality.

Holt Winters was chosen as the most appropriate model, it showed accuracy in its results, interpretability and higher reliability overall. It was able to adapt to the weekly seasonality and the sudden traffic fluctuations. We concluded then the study by taking the Holt-Winters model further, with more refined parameters and trying to predict for longer periods of time.

It proved to be a reliable tool for this kind of urban traffic prediction and a good base for more complex case studies.

Resumen

Con el aumento del crecimiento de las zonas urbanas, la gestión del flujo de tráfico de manera eficiente se presenta cada vez más compleja. Las herramientas de predicción de tráfico han demostrado ser bastante útiles en el proceso de toma de decisiones, ya sea para la regulación del tráfico, para la instauración de nuevas medidas, o para el desarrollo de sistemas de transporte más ecológicos. En este trabajo estudiaremos el tráfico a corto plazo en una de las principales arterias de la ciudad de Santander, España. Desarrollaremos y compararemos los resultados de tres modelos de predicción diferentes, ARIMA, Holt-Winters y Prophet, los cuales ya han demostrado ser fiables en el pasado. Elegiremos el más adaptado a nuestro caso e intentaremos desarrollarlo más, aplicando diferentes técnicas y estudiando cuáles funcionan bien para nuestro tipo de datos y zona.

Nuestros resultados nos muestran que Prophet, a pesar de ser visualmente correcto y fácil de usar, no nos da resultados precisos para los datos de alta frecuencia a corto plazo. ARIMA es capaz de predecir las tendencias generales, pero por otro lado requiere mucho más trabajo por parte del usuario con el calibrado de los diferentes parámetros y puede tener problemas con la estacionalidad.

Holt Winters ha sido identificado como el modelo más adecuado, ya que mostró precisión en sus resultados, interpretabilidad y fiabilidad en general. Fue capaz de adaptarse a la estacionalidad semanal y a las fluctuaciones repentinas del tráfico. Concluimos entonces el estudio desarrollando el modelo Holt-Winters más en profundidad, con parámetros más refinados y tratando de predecir para periodos de tiempo más largos.

Ha demostrado ser una herramienta fiable para este tipo de predicción del tráfico urbano y una buena base para estudios de casos más complejos.

Index

Figure index

Table index

Introduction and objective

One of the main issues that is faced by modern cities today is the need to improve mobility and reduce traffic congestion. With the exponential growth of urban areas, the traffic demands are increasing accordingly, and it becomes more and more necessary to ensure that the traffic is managed properly and efficiently for many reasons: to reduce delays, to reduce fuel consumption which allows for a less polluted city, but to also improve safety, improving the lives of the citizens. Being able to anticipate traffic patterns, even if it is just a few hours in advance allows cities to respond in advance to potential bottlenecks, rather than reacting to these problems once they have already appeared.

Short-term traffic forecasting has many practical applications. It can be used by real-time traffic control systems, for tasks such as the dynamic adjustment of traffic lights or to feed algorithms for navigation apps. It can also inform decisions on public transport scheduling and maintenance planning. At a broader level, traffic forecasting contributes to smart city projects, where solutions based on real data are applied to create more adaptive and resilient urban environments.

In this study, we will be working with data from loop detectors installed in a certain area of the city of Santander, Spain. These sensors count the number of vehicles that pass through and give us an realistic and detailed picture of how traffic behaves over time. Traffic in the city of Santander follows generally a regular urban area weekly pattern, but there are also small variations from day to day.

The main goal of our project is to compare three known time series models, ARIMA, Holt-Winters, and Prophet, in terms of their ability to predict short-term traffic intensity. We will go further than just trying to find the one that gives us the lowest error, we want to understand the behavior of each model, how user-friendly it is, and how well it handles some of the most common issues in real-world data, like missing values or unexpected fluctuations.

In order to guide our analysis, we will be trying to answer the following questions:

- Which of the three models can give us the most accuracy in terms of short-term traffic forecasting?
- How are these models different in their complexity, and difficulty of use?

- How robust is each model when the data contains noise, is incomplete, or the traffic data is irregular?

In the end, our aim is to find a model that is able to accomplish a good balance between performance and practicality, something that could realistically be used by a city like Santander to support traffic management.

Once we will have chosen the most appropriate model for this type of situation, we will try to take it further and refine the model, testing different methods to make it more reliable.

State of the art

2.1. Traffic state prediction

As cities grow, the urban mobility question becomes more and more complex, which means that the development of reliable traffic prediction tools is starting to be viewed as more of a necessity to create intelligent transport systems. Real-time traffic forecasts allow us to make decisions based on real-world data and to plan more efficiently. (1)

These predictions can be used for multiple purposes, whether to optimize traffic light cycles, schedule maintenance work, or provide accurate warnings of congested conditions, useful traffic prediction models are essential to maintain high functionality of urban road networks. However, the reality of urban traffic is that it changes constantly, it is very dynamic and is affected by a large web of complex interaction of many characteristics and factors, such as time of day, day of the week, weather conditions, public holidays, and accidents. All this combined is what creates traffic data that can have a strong periodicity, show abrupt changes and long-term trends, all of which the forecasting model must learn to identify and reproduce. (2) In addition, the data is not always clean and well organized: sensor failures, missing values, and noise, which are all common in real traffic data sets, must be accounted for.

To build models that reflect the real behavior of a road we generally try to predict three key macroscopic variables: traffic volume, traffic speed and traffic occupancy. These three factors combined give us an understanding of the area, and they allow us to develop accurate predictive models of a road's behavior. These variables are found in the data provided by the loop detectors.

There are a different approaches that can be used to forecast these variables. (1)

- *Classical statistical methods (Holt-Winters, ARIMA, SARIMA):*

Statistical models like Holt Winters, ARIMA or SARIMA are used for their simplicity, transparency and low data requirements. They have certain advantages, in particular when it comes to interpreting the results. They are usually a good choice when working with a data set that follows stable, time-based patterns and are generally not very sensitive to outliers. Models such as ARIMA and its seasonal variant, SARIMA (1), are especially useful when dealing with strong seasonal trends or consistent historical patterns. Another benefit is that these methods work well even with smaller datasets, which can be convenient when working with limited data.

That said, statistical models tend to struggle in more dynamic environments, where conditions change quickly and unpredictably (2), like it is the case for urban areas. This puts them at a disadvantage compared to machine learning methods. Today, they're mostly used as basis when testing the performance of more advanced models. (3)

- *Hybrid and machine learning models (Prophet, Random Forest, LSTM networks):*

Hybrid and machine learning models have been recently gaining more and more popularity in this field of study because they are able to interpret complex data and to detect non-linear relations in the data that statistical models easily miss. They have shown strong results in traffic prediction tasks in various papers. (4) However, they can be more challenging for the user as they need more data and a deep understanding of how to tune the model effectively.

Prophet is a hybrid model that combines machine learning and statistical methods. (5)

- *Models that incorporate Exogenous Variables (SARIMAX, VAR, Prophet):*

Models with exogenous variables take into account external factors such as weather, events… Some examples of this type of model are SARIMAX (a variation of ARIMA which includes external regressors) (6), VAR (Vector Autoregression), and also Prophet, using the "regressors" function.

They can be effective in some cases, giving a more realistic picture. The disadvantage is that they require a lot more processing of the data and extended knowledge about the local context.

The decision to use one of these prediction models depends on the characteristics of the data we will be working with and the goals of the traffic prediction task.

Criteria	Classical Statistical Models (Holt-Winters, ARIMA, SARIMA)	Hybrid & Machine Learning Models (Prophet, Random Forest, LSTM)	Models with Exogenous Variables (SARIMAX, Prophet+regressors, VAR)
User-friendliness	**High** Easy to implement and well-documented	**Medium** Prophet is easy, others like LSTM are complex	**Medium** Needs careful variable selection and prep
Data requirements	**Low** Works with small or incomplete datasets	**High** Requires large, clean datasets	**Medium to High** Depends on exogenous data availability
Interpretability	**High** Clear and explainable parameters	**Low** Often black-box models	**Medium** Interpretability depends on model and variables
Sensitivity to outliers	**Low sensitivity** Robust to moderate anomalies	**High sensitivity** Can be misled by noise	**Moderate** External variables can amplify/reduce sensitivity
Handling missing data	**Limited** Requires preprocessing	**High** Prophet and some ML models handle gaps well	**Moderate** Must handle missing exogenous data manually
Adaptability to non-linear patterns	**Low** Limited to linear patterns	**High** Captures nonlinearities effectively	**Moderate** Can model complex dynamics with regressors
Suitability for traffic data with seasonality	**Excellent** Designed for strong seasonal patterns	**Good** Especially Prophet handles seasonality well	**Good** Can enhance seasonal modeling with external variables

Table 1. Comparison between the different prediction model types.

2.2. Traffic data collection methods

In the last decades, the advances in technology have improved significantly the methods of traffic data collection. Today, cities rely on a variety of tools to collect accurate real-time data, from GPS systems to inductive loops. (7) This data is preprocessed and fed into the cities traffic management system, to then be used to better understand and adapt to the constant changing conditions of rad traffic.

Some of the most used traffic data collection tools are: (7)

- Traffic Counters: These are installed on roads to record the number of vehicles passing through a specific point over a given period of time.
- Traffic Cameras: the capture images or video of roads, these systems are used for vehicle counting, speed measurement, and general traffic analysis.
- Traffic Radars: These devices use radio waves to measure the speed of moving vehicles with high precision.
- Speed and Flow Sensors: Such as inductive loop sensors or microwave detectors, measure vehicle speed and traffic flow in real time.
- Global Positioning Systems (GPS): GPS data provides information on the location of vehicles or mobile phones inside vehicles, offering dynamic insights into traffic movement.

In this project we will focus on inductive loop detectors, which are commonly used in many European city, including Santander. These sensors are built into the roads, and they register each vehicle that passes through them, using magnetic inductance technology. work by detecting changes in inductance when a vehicle passes over them. (8)

Their precision is what makes them ideal for collecting traffic intensity data.

Some of the applications of loop detectors include:

- *Traffic Light Control:* Loop detectors can sense when a vehicle is waiting at an intersection and adjust the signal timing.
- *Vehicle Counting:* As we mentioned before, loops can register the number of vehicles that pass a specific point over time.
- *Average Speed Measurement:* When multiple loop detectors are installed along a road, they can be used to calculate the average speed of vehicles by measuring the time it takes to travel between detectors.

- *Congestion Monitoring:* By detecting vehicle presence and average speed in real time, loop detectors can help identify areas that are congested, allowing traffic management systems to respond more effectively.

Loop detectors are accurate, adaptable, and they can generate real-time data. (9) They can be installed in a variety of settings, intersections, road segments, or custom traffic configurations, and adjusted to suit specific analysis needs.

In our case, the data they generate allows us to develop more reliable prediction models.

Combined with artificial intelligence and machine learning tools, they play an essential role in improving traffic prediction tools and traffic management strategies.

Since the input comes from real-time observations, we can stop working with theorical data that is often estimated, (9) making them especially useful for forecasting.

2.3. Overview of forecasting models: classical and advanced approaches

Choosing the right forecasting model is one of the first and most important steps in traffic prediction. The decision depends on the style of the data, the goal of the analysis, if we want to do short-term or long-term predictions, or the future application of our results.

Classical models like ARIMA and Holt-Winters are widely used because of their simplicity, interpretability, and ability to perform well when the data is reliable, and it shows clear trends. Holt-Winters has shown to be effective in predicting urban traffic that follows daily or weekly patterns (10). Its ability to capture daily and weekly trends with low computational demands makes it especially suitable for predictions with limited historical data.

ARIMA, on the other hand, is a bit more demanding and more complex to calibrate. However, it can be very effective, especially for smoother or aggregated traffic flows. As noted by Tikunov and Nishimura (10) , ARIMA's main inconvenience is the need for human intervention for manual tuning, which makes it less practical for real-time sistems.

Prophet was designed to be more robust when working with missing data and be more flexible around holidays or irregular cycles. As shown by Aniszewski et al. (2022), Prophet performs well on complex traffic networks while remaining relatively intuitive and easy to adjust.

In contrast, advanced models, especially those involving Recurrent Neural Networks (RNNs) or hybrid architectures, have the capacity to capture more complex, non-linear patterns and irregularities in traffic. They tend to work better when the data has considerable noise or abrupt changes. For example, Madan and Mangipudi (2018) (11) combine ARIMA and RNNs to create a prediction model that is able to consider both the linear and non-linear aspects of traffic flow.

In a similar way, hybrid systems that link neural nets with residual-based ARIMA corrections (12) push the performance even further by compensating the weaknesses that the individual model presents. However, they are often harder to interpret and are computationally more complex.

Some researchers argue that the structure of traffic data, for example the correlations between nearby sensors or road segments, plays a bigger role in prediction performance. Li et al. (2015) explore how combining special and temporal models can improve short-term forecasts.

With all of that said, given the nature of our data and the objective of this study, we chose to focus on traditional forecasting models: ARIMA, Holt-Winters, and Prophet. Our dataset shows clear weekly periodicity and low noise, making it suited for classical time series approaches. These three methods are balanced between being lightweight, reliable and explainable.

In the future, incorporating deep learning could be a logical extension, but for now, the main objective is to develop a practical, well-understood tool that delivers good performance without overfitting.

In conclusion, traffic forecasting can be tackled from many angles, from simple statistical methods to machine learning and deep learning. Each model presents its own advantages and disadvantages, and there's no one model that fits in every context. In our case, since we have consistent and periodic data, models like Holt-Winters, ARIMA, and Prophet can give us enough flexibility to model the seasonal trends we observe, while still being interpretable enough to apply in a real world setting without overcomplicating things.

Advanced models may offer better performance in more complex or noisy scenarios, for this study, our focus is on reliability, simplicity, and a clear understanding of how each model behaves.

Methodology

3.1. Selection of the Study Area: Marqués de la Hermida, Santander, Spain

Before diving into the data analysis, it's important to start with some context. The area we have chosen for this study is the city of Santander, the capital of the Cantabria region in northern Spain. It is a coastal city known for its irregular terrain, maritime climate, and a hectic urban life that mixes tourism, commerce, and local daily activity. Santander offers a rich and complex environment for studying urban mobility.

Figure 1. Map of Spain with the localization of Santander

One of the key roads in the city is Marqués de la Hermida Avenue. It stands out as one of the city's main arteries, it is one the 3 main entrances to the city. This avenue stretches along the western part of the city, parallel to the port area, and acts as a major access point into Santander from the surrounding suburbs, highway and from the airport. While it's not a particularly pedestrian zone, it has one of the highest traffic volumes in the city.

Figure 2. Marques de la Hermida

We chose to focus on this street for several reasons. First, the high volume of vehicles using this route ensures a robust dataset, ideal setting for understanding real traffic behavior in an urban zone. Second, because it serves as a connection to both residential and commercial zones, it captures a wide variety of traffic scenarios: delivery trucks, city buses, commuters, and tourists all share this space on a daily basis.

Once we had identified Marqués de la Hermida as our study area, we reached out to Santander's Traffic Control Center, the institution responsible for monitoring and managing the traffic of the city. With their support, we were able to access loop detector data, collected through sensors all throughout the city. These sensors are able to record vehicle counts over time at specific locations.

We will be focusing on the data from five traffic loops: 1008, 1009, 1011, 1025, and 1031. These loops are distributed along Marqués de la Hermida and provide consistent traffic flow data. Together, they offer a solid starting point for our model and allow us to begin analyzing the traffic behavior of the area.

In the next image we can see the map of our area of study with all the traffic loops that are installed:

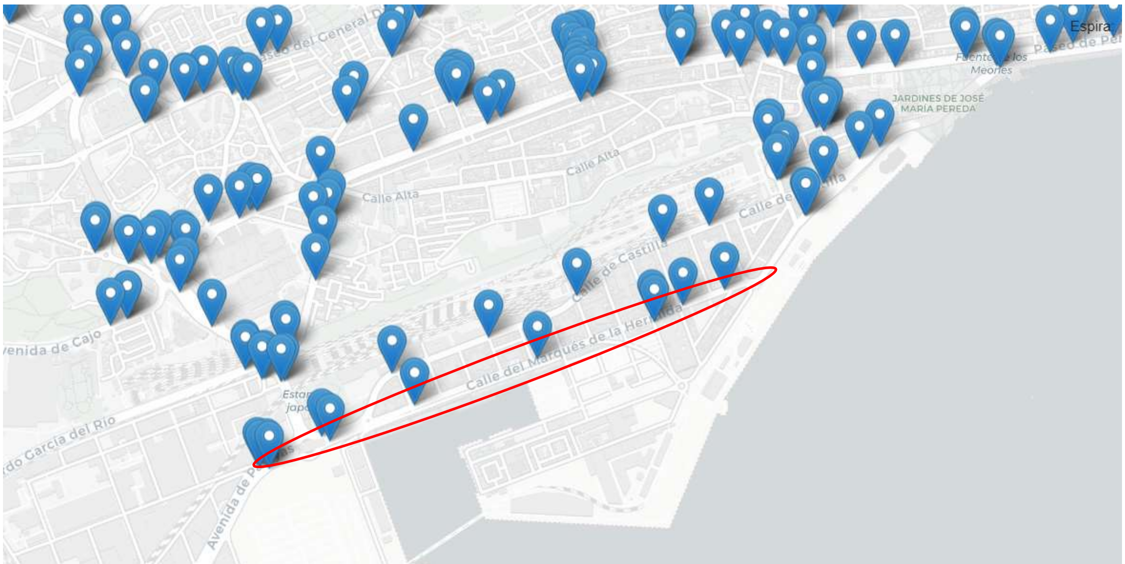


Figure 3. Map of the traffic loops we will be using to obtain our data and create our model.

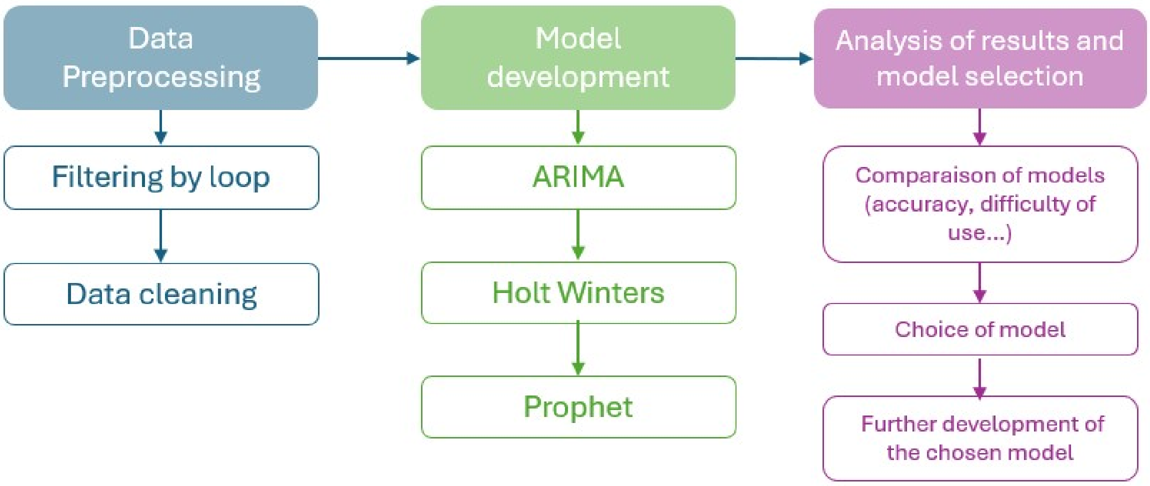


Figure 4. Methodological process

3.2. Data Import and Preprocessing

To start the development of our traffic prediction model, we begin by working with the data collected from the selected traffic loops along Marqués de la Hermida. The main variable of interest is the intensity, which represents the number of vehicles passing through a given loop within a specific time interval and which is typically recorded every 15 minutes.

Loop	Total of Days observed	Average of observations/day	Start date	End date	% Missing Values
1008	71	91,5	01/04/2024	14/06/2024	0
1009	71	91,5	01/04/2024	14/06/2024	0
1011	12	6,91	06/04/2024	14/06/2024	98%
1012	71	91,5	01/04/2024	14/06/2024	0
1025	71	91	01/04/2024	14/06/2024	0,55%
1031	71	91,5	01/04/2024	14/06/2024	0

Table 2. Average of observations per loop, data that we used and % of missing values

To guarantee a correct analysis, the first step after importing the raw data is to convert the date and time column into a standard datetime format. This allows us to carry out time-based operations more efficiently, (grouping data by hour, day, or week), and makes it easier to merge or compare data from different loops. Once the datetime formatting is in place, we sort the data chronologically for each loop to ensure that the time series is clean and ready for sequential analysis.

With the data now in a structured format, we move into the preprocessing phase, where we will apply several cleaning and filtering techniques to improve its overall quality and reliability. The main goal here is to remove noise (non-relevant data), take care of missing or wrong values, and prepare the dataset for accurate modeling.

In our case, we apply two primary filters that aim to correct different types of data irregularities:

a) *Zero-intensity filtering*

A recurring issue in traffic loop data is the presence of long stretches of intensity values stuck at zero. While isolated zeros may indicate periods of no traffic, such as at night or during holidays, extended periods of zero often mean that the sensor is not working properly.

To deal with this, we will implement a filter that automatically detects and removes segments where the intensity remains at zero for more than three consecutive time intervals, which in our case means over 45 minutes. This helps us keep valid zero values while getting incorrect data. This process is done loop by loop.

b) Flatline detection

Another form of error appears when the average intensity remains nearly identical from one day to the next, suggesting a possible failure or miscalibration in the sensor. These errors, a bit more subtle than the previous one, are represented in the form of “flatlines" indicate that the loop is giving repeated values without taking into account the actual traffic conditions.

To detect this, we compute the daily average intensity for each loop and examine for consecutive days with no significant variation. If two or more days in a row show the same or nearly the same average intensity, we review them. In general, these days are excluded or treated as unreliable during model training.

In addition to these main filters, we also incorporate some common preprocessing techniques used in time series modeling:

c) Missing values

Gaps in the data are relatively common and can be due to communication errors or sensor malfunctions. For small interruptions, we will apply forward fill or linear interpolation to keep the continuity. For longer gaps, we exclude them from the data to avoid training the model with unnatural patterns.

d) Outlier detection

Sometimes, sensors can report unrealistically high or low values due to glitches. We will calculate a rolling median with a window of several hours to identify and remove these outliers. If a value deviates significantly from the local median (ex. more than 3 standard deviations), it is replaced or removed.

e) Smoothing and aggregation

Creating smoothed versions of the intensity data using a moving average filter (ex. a 3-hour rolling mean) is used to better visualize trends and reduce noise. Smoothed data is easier to interpret and it gives us a better picture of the trends.

Once the data has been cleaned and filtered, we will generate initial graphs for each loop that will show the variation in traffic intensity over time. These graphs give us an early sense of the data's structure and help identify any problematic patterns that our preprocessing might have not taken care of.

For the first phase of our analysis we will focus on the period between April and June 2024. These three months provide a useful representation of seasonal variation during the spring and early summer.

By examining these plots, we are able to study traffic fluctuations and detect possible issues. We look for things like unusual peaks, sudden drops, or long periods of stable values, which might indicate malfunctions or inconsistencies in the data. This visual inspection is an essential part of the process, as it helps confirm that the data is reliable and of quality to continue with the forecasting.

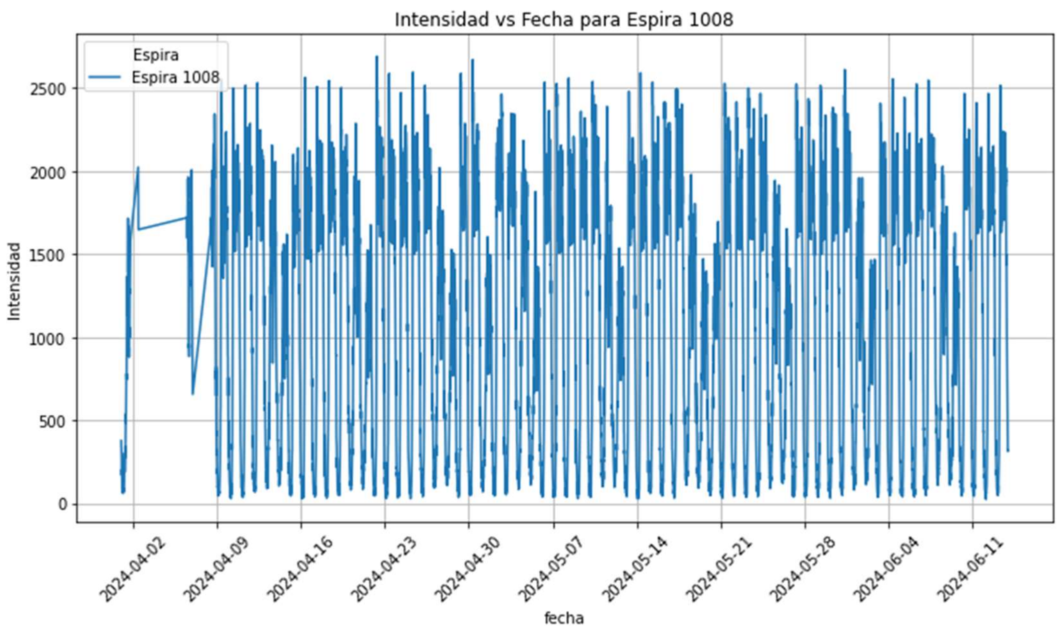


Figure 5. Traffic intensity for Lop 1008 (April-June 2024)

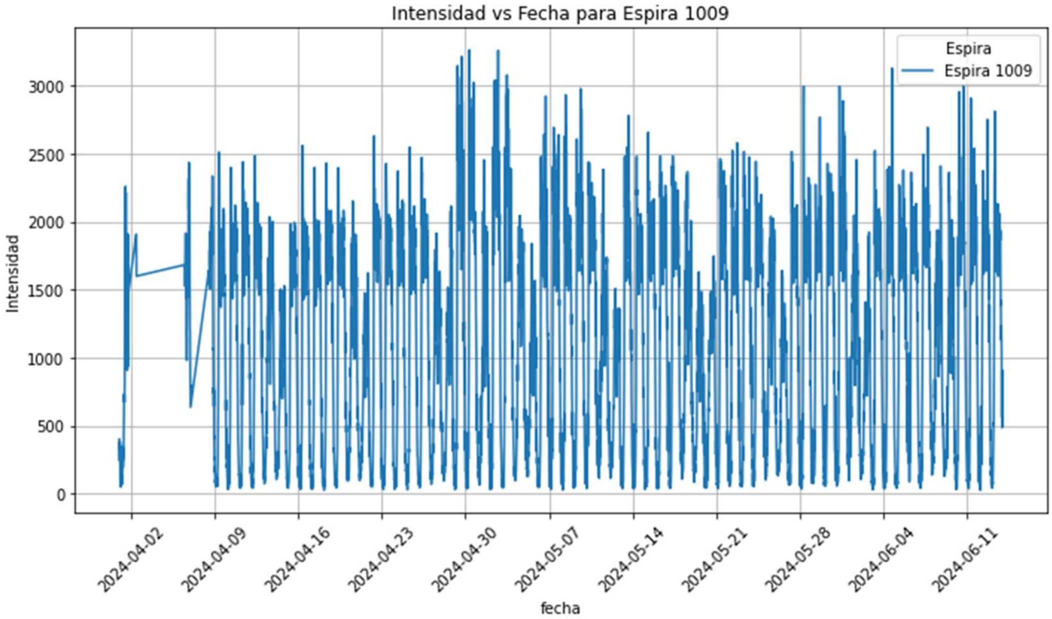


Figure 6. Traffic intensity for Loop 1009 (April-June 2024)

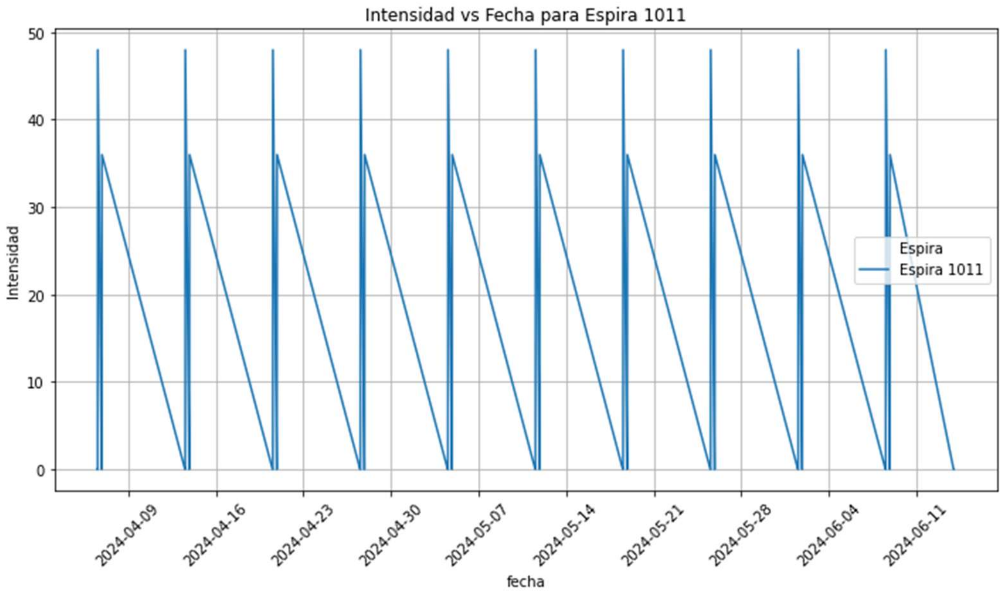


Figure 7. Traffic intensity for Loop 1011 (April-June 2024)

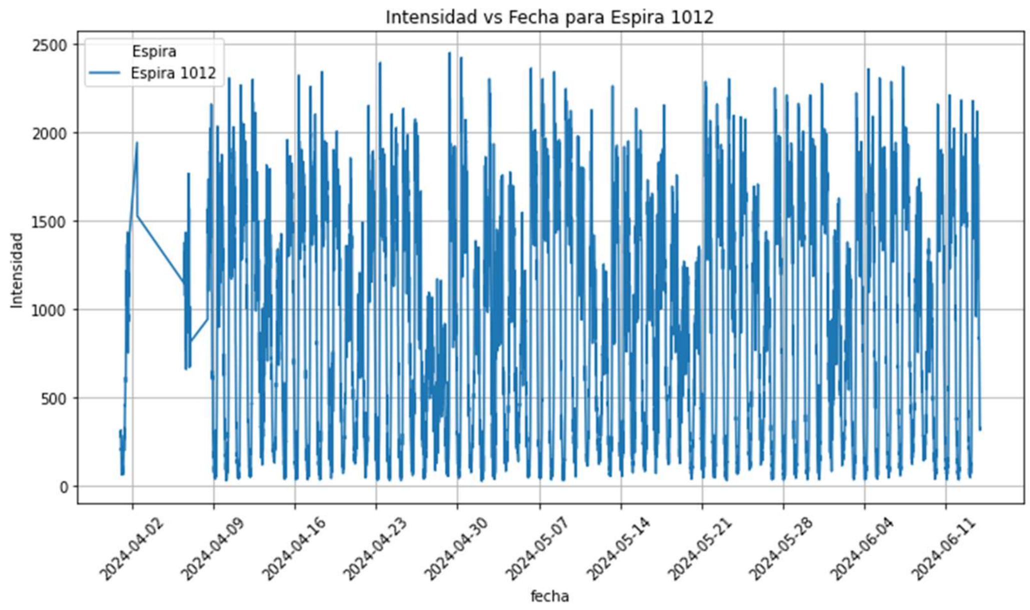


Figure 8. Traffic intensity for Lop 1012 (April-June 2024)

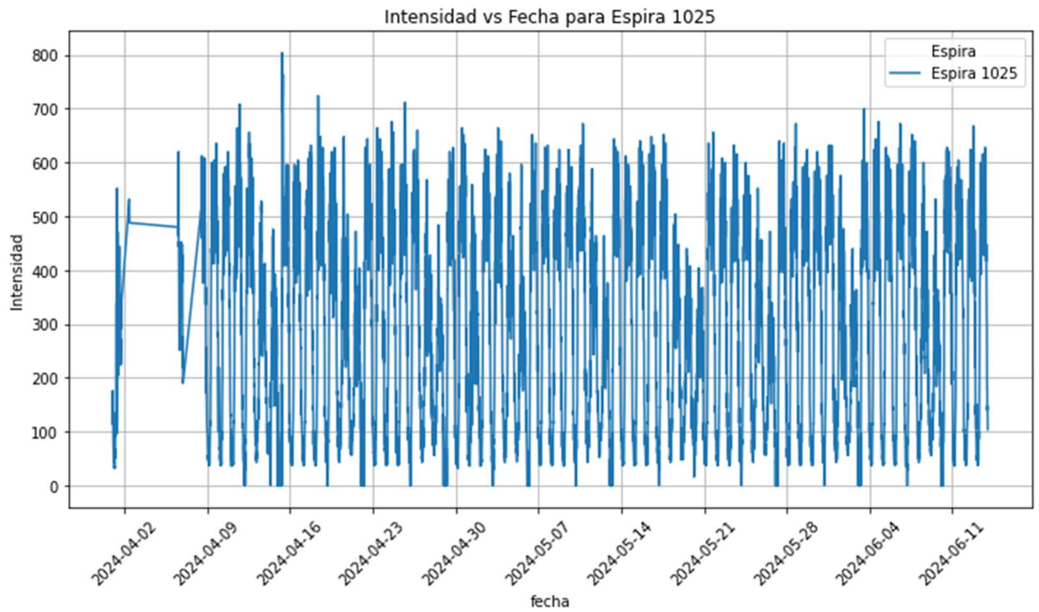


Figure 9. Traffic intensity for Loop 1025 (April-June 2024)

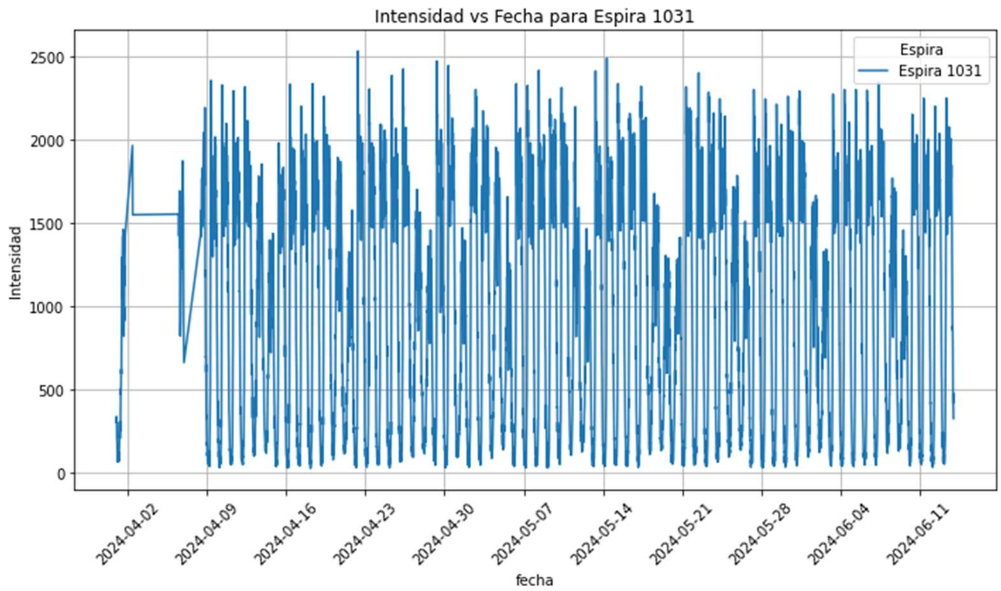


Figure 10. Traffic intensity for Loop 1031 (April-June 2024)

The initial inspection of the intensity plots, for each loop, already helps us identify several issues that will influence the next stages of out analysis.

The first and most striking conclusion concerns loop number 1011 (Figure 6), which appears to be unreliable. On the rare occasion that the values are registered, they follow a regular and repetitive pattern, which is not consistent with the expected tendencies of real traffic. This suggest that either the loop is not be functioning correctly or that it is completely disconnected from the network. For this reason, we have decided to exclude loop 1011 from further modeling to avoid distorting the results of our analysis.

We also noted a specific anomaly that affects all loops between September 2nd and 4th. During these three days, the intensity drops or behaves abnormally in all series, which does not match with known traffic events or public holidays in the city. Since the anomaly is not linked to only one loop, it is reasonable to assume that the issue comes from a recording error from the system. The most cautious decision is to exclude these three days from the dataset, to prevent them from biasing the training and calibration of the prediction model.

These two findings highlight the importance of visual inspection before continuing to the more complex modeling stages. Identifying and addressing errors, inconsistencies, or suspect patterns early in the process ensures that our forecast is based on a reliable foundation.

As we move forward with our study, we will continue to monitor the dataset for similar issues and refine our preprocessing as needed.

3.3. Building and evaluating traffic prediction models: A comparative analysis of ARIMA, Holt-Winters and Prophet.

With the dataset cleaned and validated, we are ready to move into the next phase of the study: the construction of our traffic prediction models. These forecasts will serve not only to uncover underlying traffic patterns but also to support urban mobility planning in cities like Santander.

To do this, three well known and tested time series forecasting models have been chosen: ARIMA, Holt-Winters, and Prophet. Each of these models has been used in the time series analysis literature. The decision to focus on these particular models is supported by previous work conducted at the University of Cantabria, along with findings from other academic case studies.

- The ARIMA model is a classic statistical forecasting method known for its ability to model temporal patterns using historical data. It is particularly effective when dealing with datasets with a strong autoregressive structure and no seasonal variation.

- The Holt-Winters is well-suited to time series with clear seasonal trends, such as the daily or weekly patterns that we often see in traffic data.

- Prophet, a more recent forecasting tool that is designed for simplicity, scalability, and flexibility, and has gained popularity due to its robust handling of missing data, outliers, and strong seasonal behavior.

Each of these models offer different strengths. The goal is to evaluate and compare their performance.

To assess the performance, we will train each model on the same cleaned traffic data and evaluate them using Mean Absolute Error (MAE) and Mean Squared Error (MSE).

- MAE (Mean Absolute Error) calculates the average of the absolute differences in between the real and predicted values. Its mathematical definition is:

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

- MSE (Mean squared error) measures the average of the squared differences between the initial data and the predicted values. Its mathematical definition is the following:

$$MSE = \frac{1}{n}\sum_{i=1}^{n} \left(y_i - y'_i\right)^2$$

Where yi is the actual value and y'I is the predicted value, with n being the number of observations.

A MAE that is smaller than the average intensity values from the real data we will be able to say that the model is making accurate predictions. On the contrary, if they are higher than the average, it will mean that the model is performing poorly.

In the case of the MSE, we will be able to say that the model is accurate when this value is lower. If we obtain large MSE results it will mean that the predictions present notable deviations.

3.3.1. ARIMA

ARIMA (Autoregressive Integrated Moving Average) is one of the widely used statistical prediction methods. It is an efficient linear model that builds on the idea that future values can be predicted using historical values. (11).

The mathematical expression for ARIMA (p, d, q) is given by the following equation:

$$\phi(B)(1-B)^d X_t = \theta(B)e_t$$

In this expression, X_t represents the time series variable, an observation at a specific moment; $\phi(B)$ y $\theta(B)$ are polynomials of degrees p, q, and d; e_t is the white noise at time t; and d is the order of differencing. (11)

Before fitting the ARIMA model, it is essential to check if the time series we are using is stationary, which means that its mean and variance are constant over time. To do this we tested the series using the Augmented Dickey-Fuller test. If the test gives us a value that is greater than 0.05, the series is considered non-stationary.

To correct this, we apply differencing until the series is stationary. We can repeat this process multiple times (second differential, third differential… and so on). The reason why this is important is because the number of times that we differentiate to achieve stationarity will determine the value that we will give our "d" value in the ARIMA model.

In our case, one single difference was necessary to make our series stationary, and so the d value that we used was 1.

Moving on to the p and q values, our autoregressive and moving average components.

P represents how many pas values the model uses and q represents how many past errors the model uses. In general they consider how much past information the model will be using to predict what is to come.

To choose these values, we use the Autocorrelation Function and the Partial Autocorrelation Function plots. ACF shows us how our time series is correlated with its past values, it helps us choose a "q" value to begin with, it calculates at what lag the correlation deviates from the confidence values ($\pm\frac{1.96}{\sqrt{n}}$) and gives the value of that lag to q.

For the “p” value we use a similar approach but removing the effect of shorter lags.

The values that this gives us are not the optimal ones but they give us a good place to start.

Now with these starting values we go on to plot the errors MAE and MSE as a function of p and q, with a value for d that is fixed.

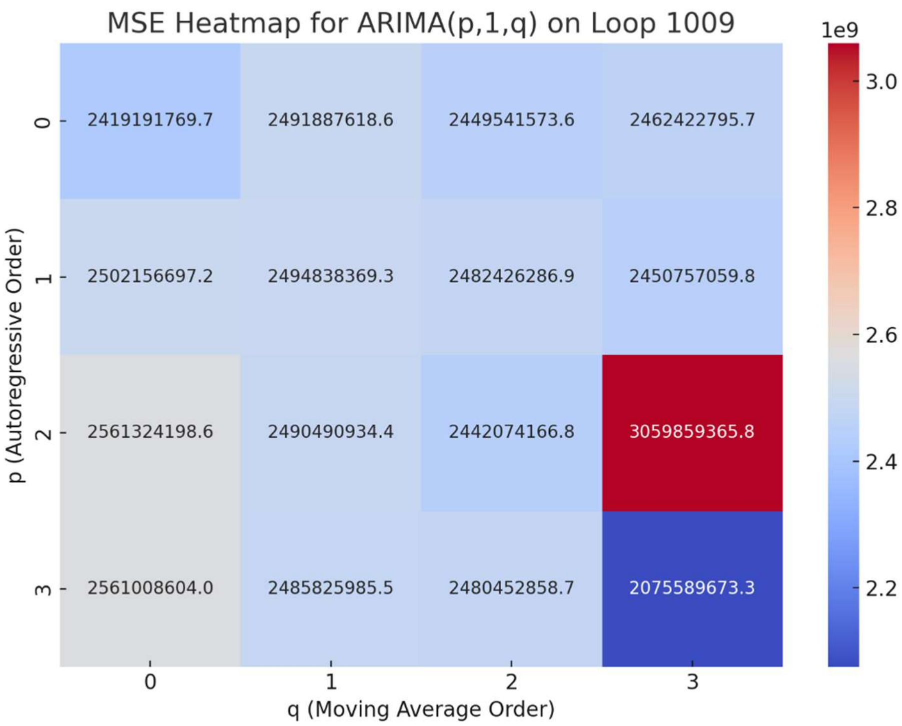


Figure 11. Example of a calibration heatmap for Loop 1009.

As we can see in our example for Loop 1009 (Figure 11), the best calibration for ARIMA (p, d, q) for this Loop is ARIMA (3,1,3).

Now that our model is calibrated, we will go on to the training phase. The model will extract 60 days of data input and set aside the next 7 days for testing. This model is trained using the existing data and historical patterns in the training set to capture trends and fluctuations in traffic intensity. Once the model is trained, it can generate a 7-day forecast, which we can compare to the actual observed values that we know.

3.3.2. Holt-Winters

The Holt-Winters method, also called triple exponential smoothing, is a time series forecasting technique that adapts well to both trend and seasonality, characteristics that are often found in traffic intensity data. It works by using three dynamic components: the level (the smoothed estimate of the series at the current time), the trend (the direction and rate of change over time), and the seasonality (systematic and repeating variations). Depending on how the seasonal component behaves over time we can choose between two types of models: additive, where seasonal fluctuations are relatively constant over time, and multiplicative, where seasonal effects are scaled proportionally with the level of the series. This makes Holt-Winters particularly well-suited for real problems like traffic forecasting, where daily and weekly patterns are present but long-term usage may still change due to infrastructure, population, or behavioral trends. (10)

Mathematically, the method relies on recursive equations that update at each time step. In the additive model, the equations are:

- Level: $St = \alpha * \frac{Yt}{I_{t-L}} + (1-\alpha)(S_{t-1} + b_{t-1})$
- Trend: $b_t = \gamma(S_t - S_{t-1}) + (1-\gamma)b_{t-1}$
- Seasonality: $I_t = \beta\frac{Y_t}{S_t} + (1-\beta)I_{t-L}$
- Forecast: $F_{t+m} = (S_t + mb_t) * I_{t-L+m}$ for m steps ahead

Here, Y_t is the actual observation at time t, S_t is the smoothed level, b_t is the estimated trend, and I_t is the seasonal component. α, β and γ are smoothing parameters between 0 and 1 that control the weight assigned to recent observations for each component. These parameters are often optimized to minimize forecasting error metrics like RMSE (Root Mean Squared Error).

Some studies show that the Holt-Winters method was successfully applied to historical usage data from base station cells, demonstrating that it could forecast short-term variations. (10) Importantly, the method has proved to be computationally efficient and easy to implement, requiring minimal manual tuning and no reliance on complex model selection procedures like ARIMA.

In addition to all of that, it performs well even with moderate volumes of historical data and does not need expert-level statistical intervention.

Given the traffic data we are working with, we have chosen the additive seasonal type.

Once we have chosen the type of model that we will be creating, we can start with the calibration. To calibrate a Holt-Winters model we use three parameters α, β and γ. They represent consequently the smoothing of the level, the smoothing of the trends and the smoothing of the seasonality. We assign each of these parameters a value between 0 and 1, the closer to 0. We can see in the next table what these values can mean.

Parameter	What it controls	Value close to 1	Value close to 0
α	Level	The model adapts quickly and efficiently to recent changes, it can become sensitive to new data (Spikes of traffic)	The level evolves slowly and it doesn't give that much importance to new data, it smoothes using the history of data available
β	Trend	It adjust quickly to changes (ex. closed roads)	The model tends to keep the trend stable and change more gradually
γ	Seasonality	The seasonal pattern can change over time	The seasonal patterns are assumed to be constant over time

Table 3. Characteristics of each calibration parameter for Holt-Winters

For this model we will use an automatic calibration tool, Python's "Exponential smoothing". This automatically tests different combinations of parameters and using an optimization algorithm it chooses the ones that result in the lower Sum of Squared Errors between the known data and the predictions. This contributes as well to the user-friendliness of the model.

For example, the parameters that we used for loop 1009 are: $\alpha = 0.111$, $\beta = 0.086$ and $\gamma = 0.0001$.

3.3.3. Prophet

Prophet is a time forecasting model created by Meta (formerly Facebook) designed to process real data that displays complex patterns. It is particularly helpful when working with datasets that show strong seasonality, clear trends, and occasional abrupt changes. One of Prophet's main strengths is that it model data without needing lots of fine tuning, which sets it apart from the traditional statistical models. (13)

Prophet decomposes time series data into three main components: trend, seasonality, and known holidays or events. The model uses linear or logistic functions to represent the trend, while the seasonal effects are captured through the Fourier series. Prophet also accounts for random variations that other models might miss because it includes a residual term to handle the noise and the unpredictable fluctuations.

At its core, Prophet models time series as a simple addition of components:

$$y(t) = g(t) + s(t) + h(t) + \epsilon_t$$

where:

- $g(t)$ models the **trend** (ex. long-term increase in traffic),
- $s(t)$ models **seasonality** (ex. weekly traffic patterns),
- $h(t)$ accounts for **holiday effects** or known events,
- ϵ_t is the **error term** (random noise).

In the context of traffic prediction, Prophet offers a balance between performance and usability. According to findings from a study that applied Prophet to predict network traffic in backbone optical network (5), the model performed well in scenarios where traffic patterns varied over time but still followed recognizable cycles. Parameters like the seasonality scale were found to influence accuracy, though the model typically performed reliably with minimal manual adjustments.

What makes Prophet particularly well-suited for our case is its adaptability to short and medium-term traffic forecasting without requiring large volumes of training data or complex preprocessing. Its transparent modeling structure allows for clear interpretation of trends and seasonal effects, making it not only a practical tool for predictive analytics but also a valuable aid in understanding the traffic dynamics. (14)

Unlike the other models, the calibration process for Prophet is entirely automated (15). It calibrates the trend component using a linear growth function, the seasonality component with the Fourier Series and it counts with a variance term that takes care of the noise.

Results

4.1. Graphical results for ARIMA, Holt-Winters and Prophet for loops 1008, 1009, 1012, 1025, 1031

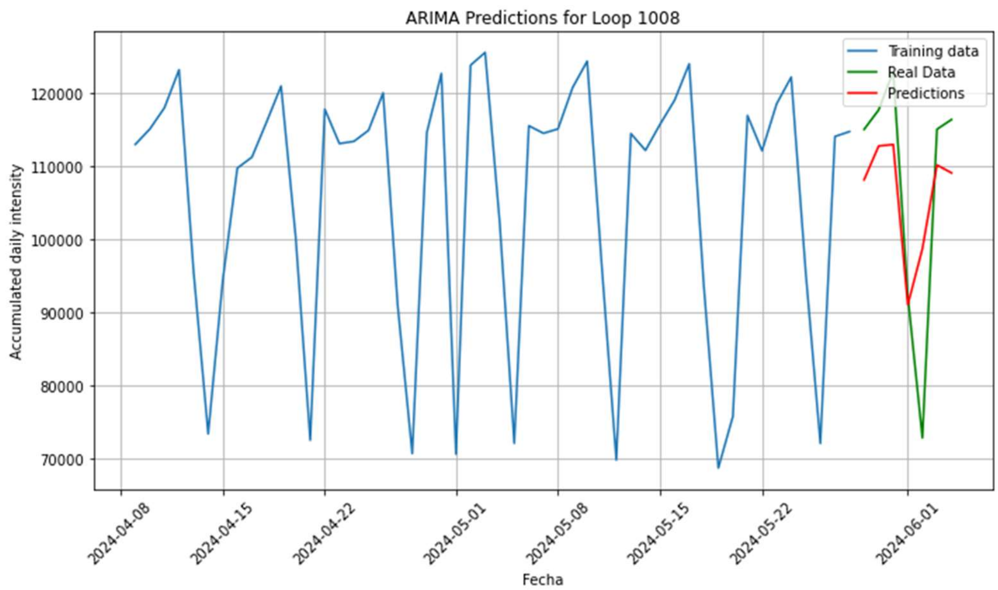


Figure 12. Intensity ARIMA predictions for Loop 1008

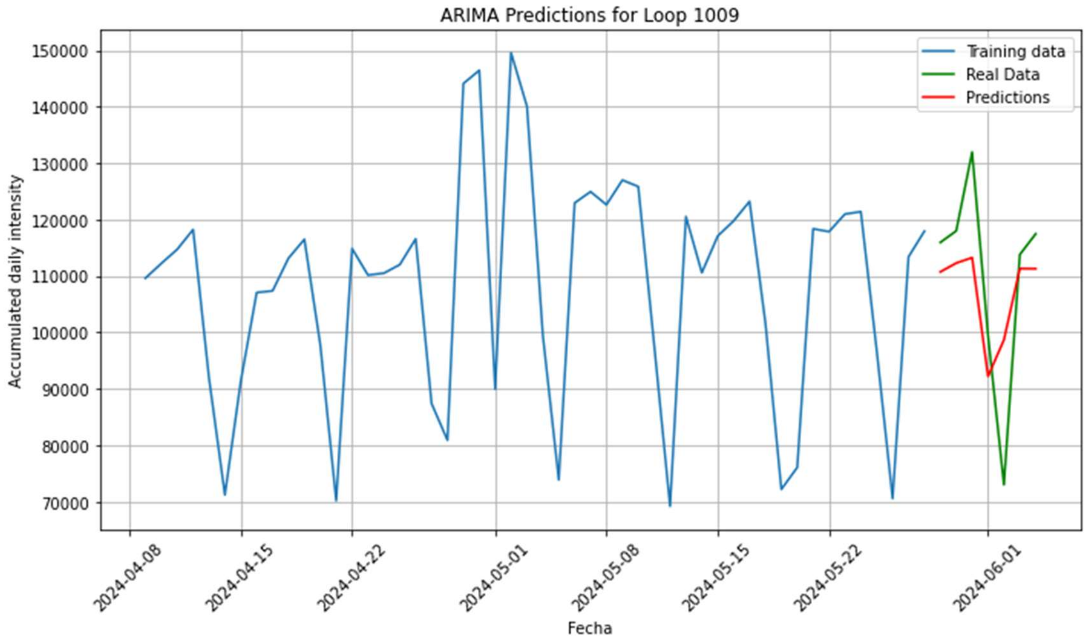


Figure 13. Intensity ARIMA predictions for Loop 1009

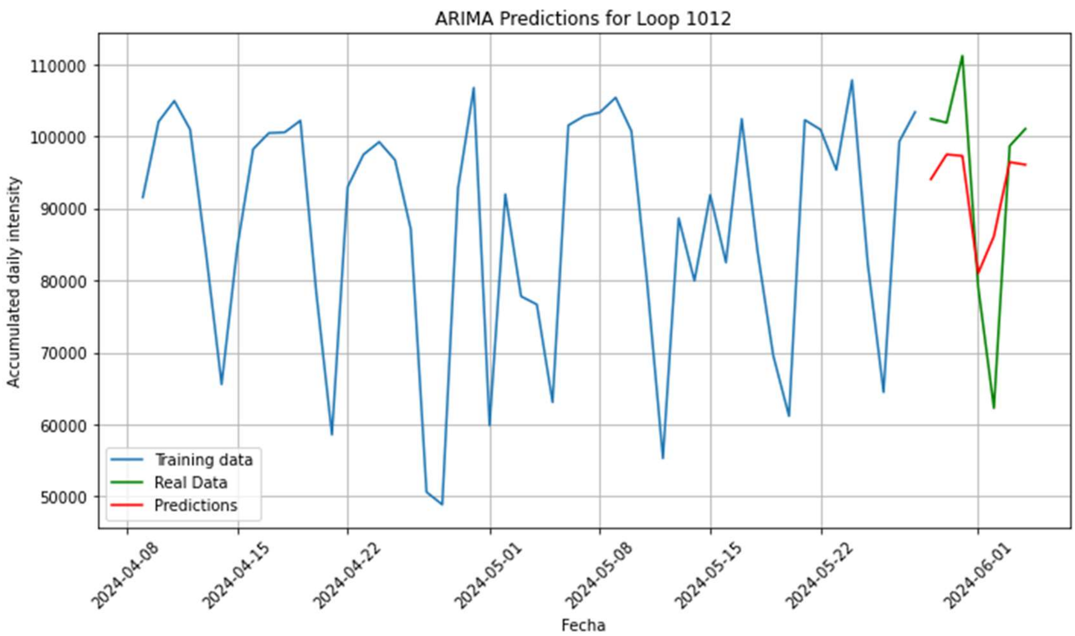


Figure 14. Intensity ARIMA predictions for Loop 1012

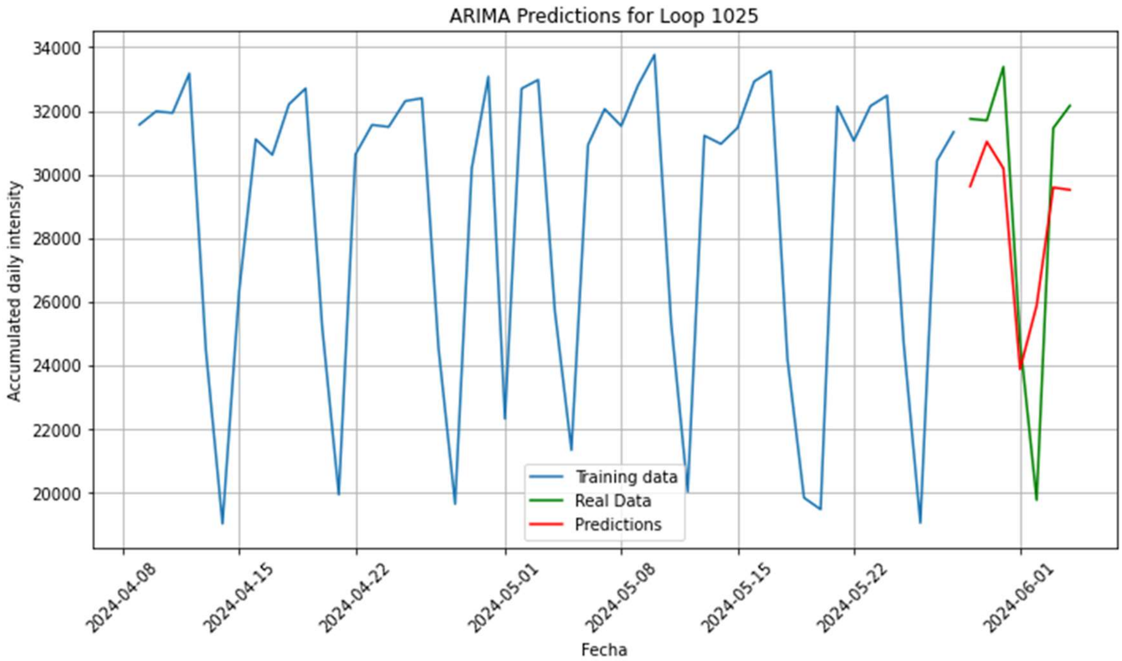


Figure 15. Intensity ARIMA predictions for Loop 1025

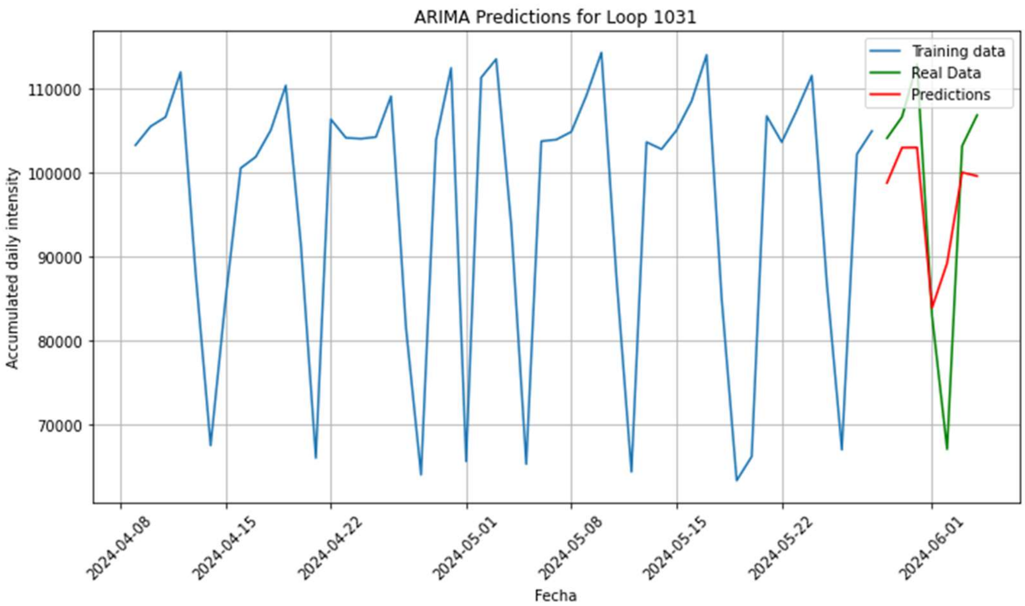


Figure 16. Intensity ARIMA predictions for Loop 1031

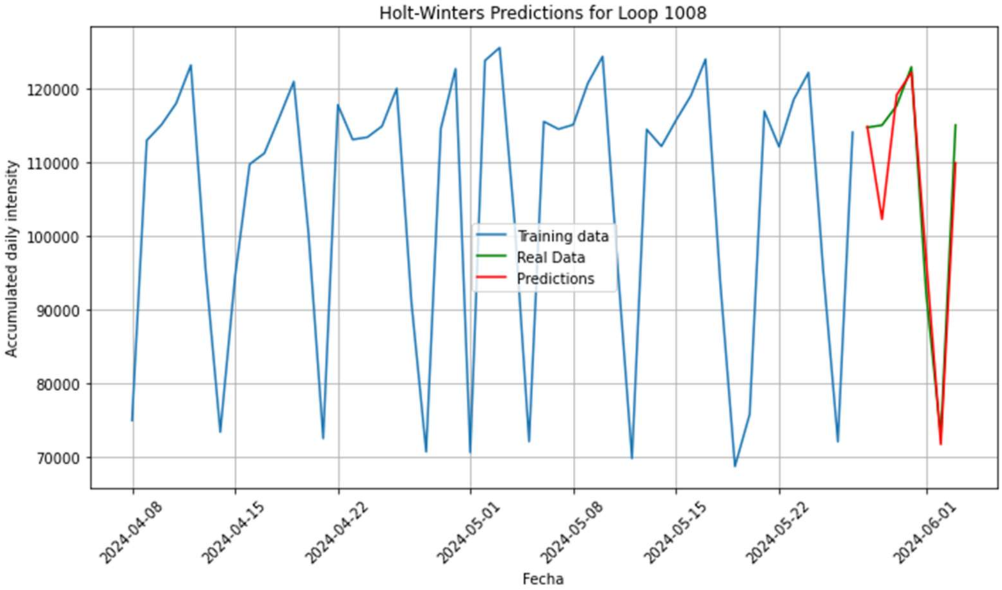


Figure 17. Intensity Holt Winters predictions for Loop 1008

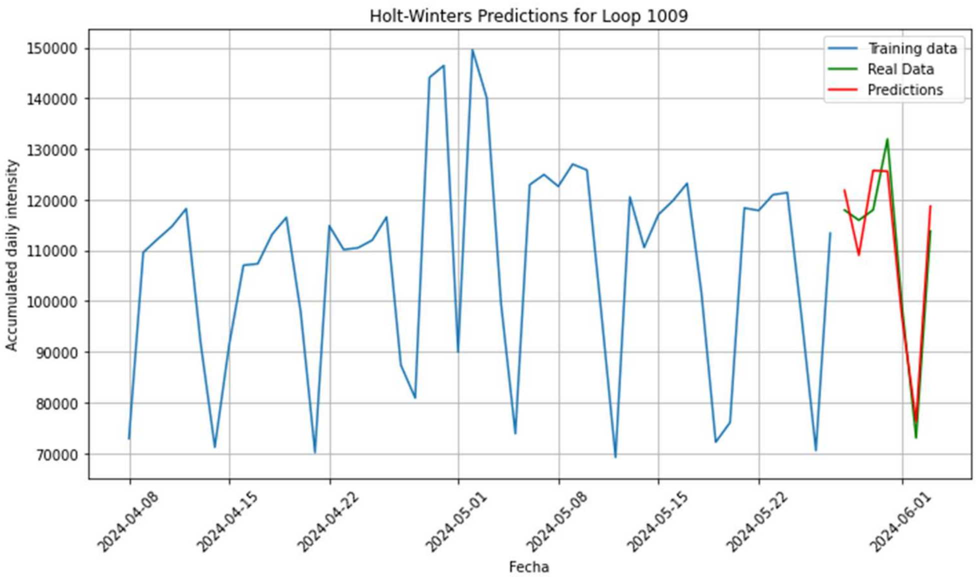


Figure 18. Intensity Holt Winters predictions for Loop 1009

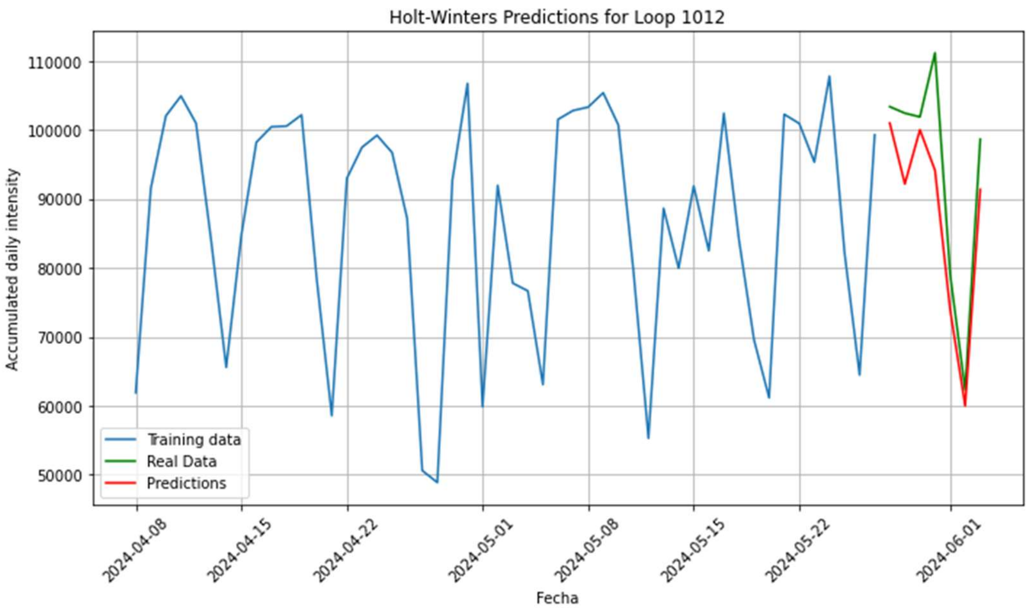


Figure 19. Intensity Holt Winters predictions for Loop 1012

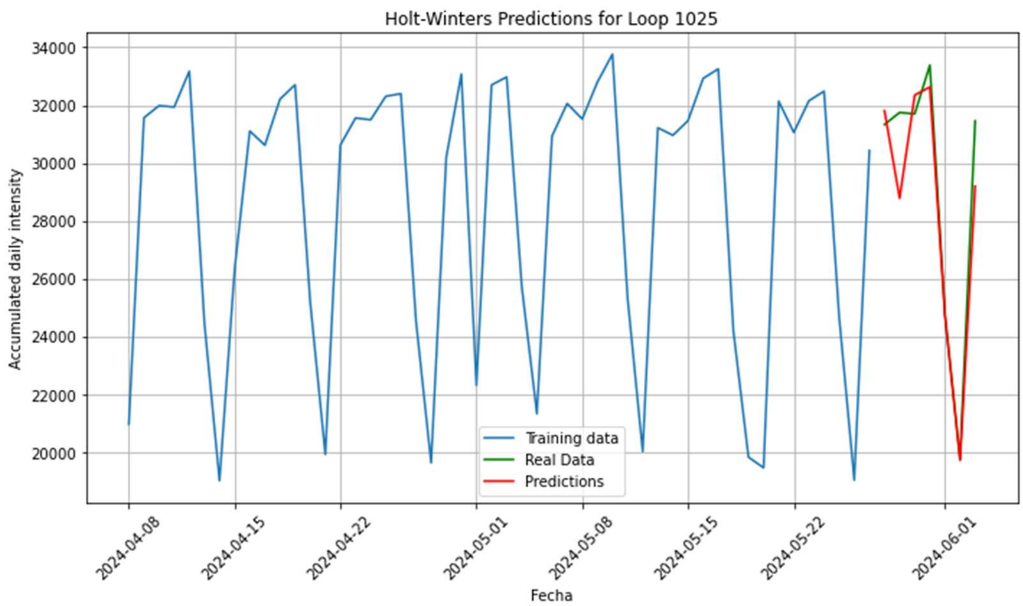


Figure 20. Intensity Holt Winters predictions for Loop 1025

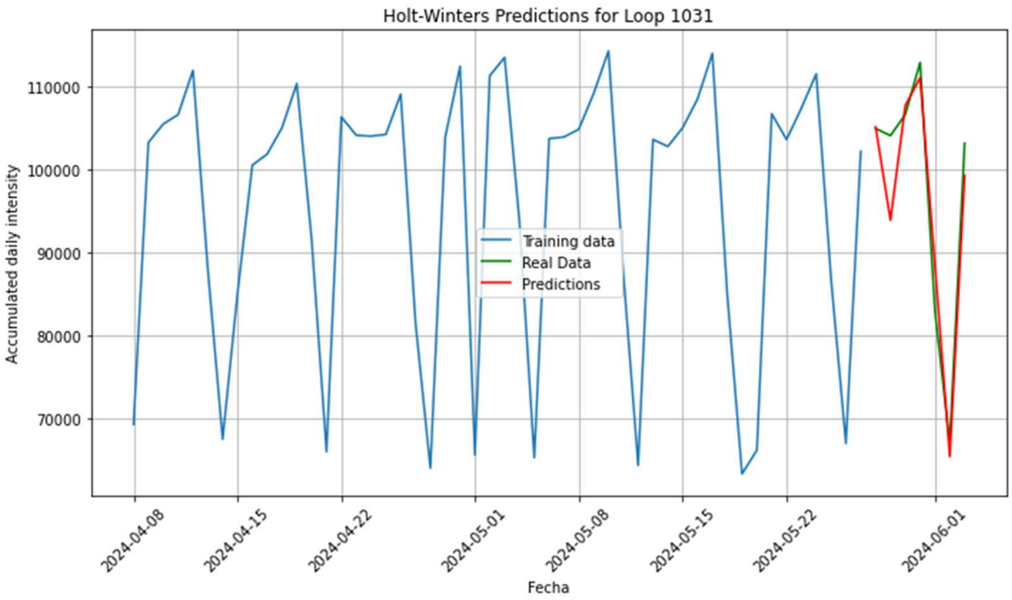


Figure 21. Intensity Holt Winters predictions for Loop 1031

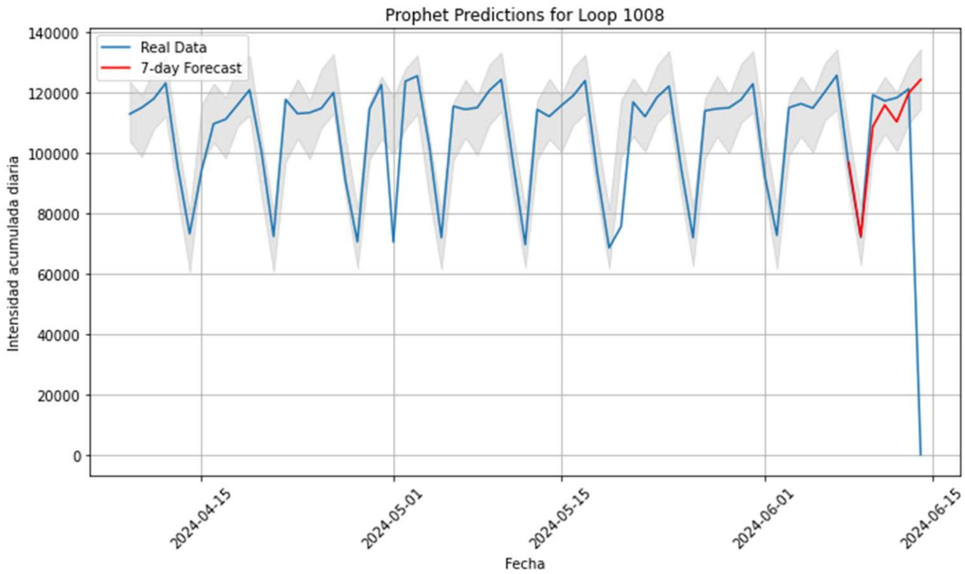


Figure 22. Intensity Prophet predictions for Loop 1008

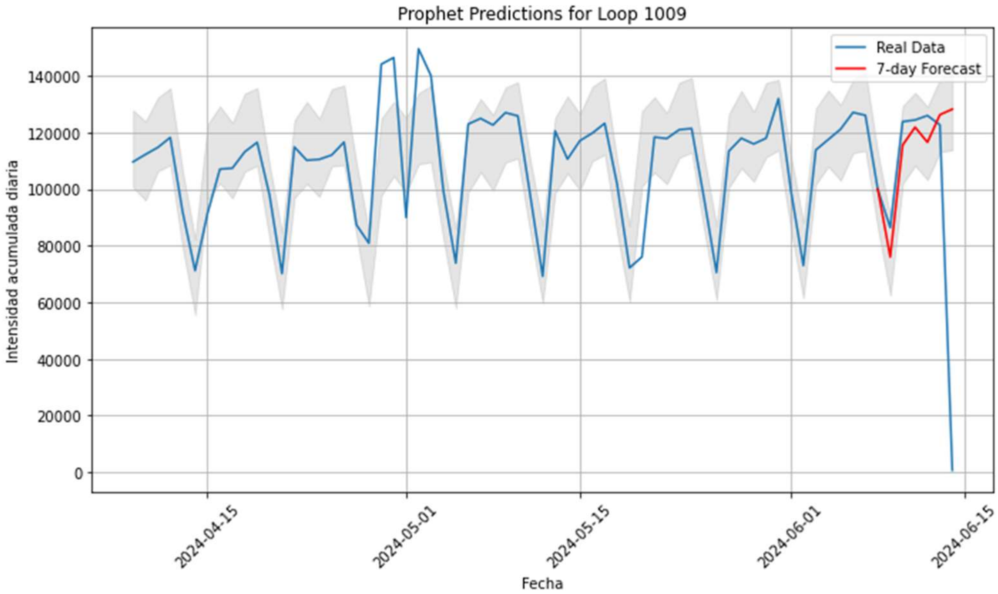


Figure 23. Intensity Prophet predictions for Loop 1009

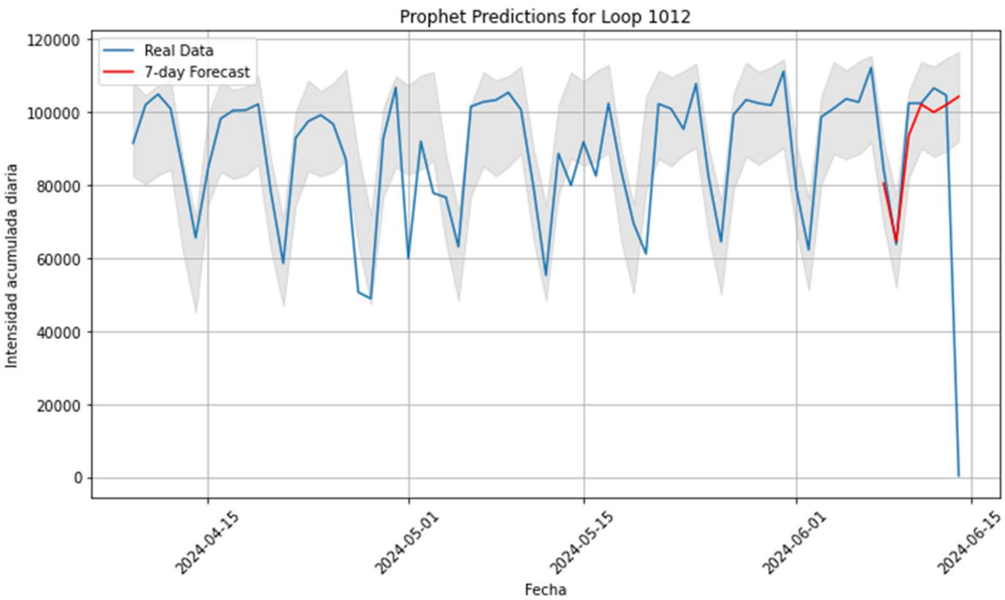


Figure 24. Intensity Prophet predictions for Loop 1012

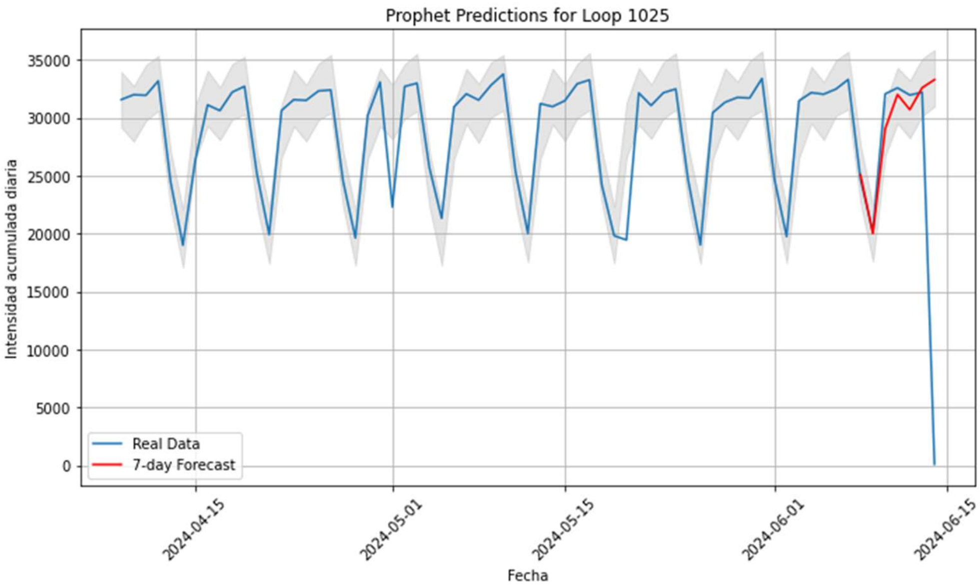


Figure 25. Intensity Prophet predictions for Loop 1025

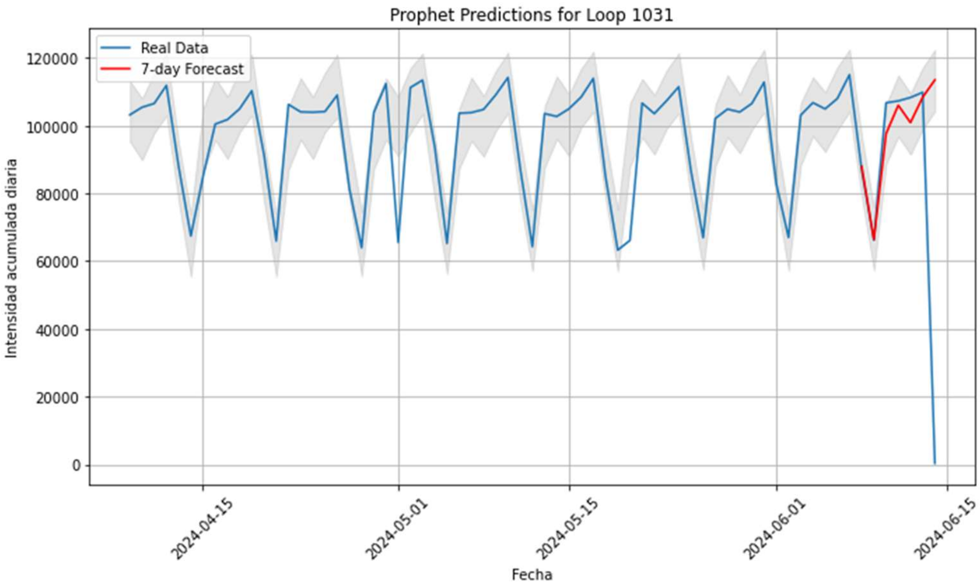


Figure 26. Intensity Prophet predictions for Loop 1031

4.2. Analysis of the results

The metrical error results we obtained from all three models were the following:

	ARIMA		Holt-Winters		Prophet	
Loop	MAE	MSE	MAE	MSE	MAE	MSE
1008	8724.67	131353327.14	3712	30680379	21015.81	2225782280.26
1009	10173.28	166521679.26	2064	29175916	1563845731.59	2364960862.09
1012	8519.11	126426436.61	6604	69886999	18126.16	23128.73
1025	2503.14	9143388.96	1024	2165048	5561.46	158831078.67
1031	7491.11	99229489.92	3474	22345760	19093.38	1851431390.72

Table 4. Comparison of the results of all three models.

In order to compare and evaluate the models, we have used the Mean Squared Error and the Mean Absolute Error for each loop. Knowing that the average intensity values range between 70000 and 120000 vehicles, we can compare the values of the errors to the real values, therefore being able to determine the accuracy of the models.

Generally speaking, Holt-Winters shows better results across all loops, it clearly outperformed the other models not only in the error metrics but in the visual fit as well. Visually, we can tell that the predicted curve adapts well to the trend shifts. This method is especially strong when we apply it with a seasonal period, in this case 7 days, which strengthens its suitability for this case. The high overall MSE tells us that the model would need some more fine tuning to give more reliable results, but we could say it has the potential to be a good fit for our case.

ARIMA was able to give us good results in some of the loops but it clearly failed in some of the others (loop 1009 for example). Prophet on the other hand, showed us the highest error values for all loops.

We obtained the best results for loop 1025, Holt-Winters presents its best MAE, 1024. ARIMA performed well, but with a slightly higher error, and finally Prophet gave us higher MSE and MAE, probably dur to the fact that it tends to over smooth, one of the things to consider when using this model. It is interesting because loop 1025 is the one that has the most missing data (apart from 1011), showing us that the Holt-Winters model is able to handle missing information, at least in this case where the patterns stay strong.

The worst results were for Loop 1009, all models struggled with the more irregular fluctuations. Holt-Winters was able to adapt slightly better and detect the trends, but less accurately than for the other Loops.

The other loops 1008, 1012, and 1031, are somewhere in the middle, ARIMA obtains MAE values going from 7,500 and 8,700. These results indicate moderate prediction accuracy, where the model performs reasonably well but still leaves room for improvement. Holt-Winters provided somewhat balanced predictions. Prophet showed the same troubles as on the other loops, making us think that it is not the best model for high-frequency traffic.

In a few loops, specially for 1008 and 1012, ARIMA tends to predict the peaks a few days too late, it doesn't react quickly to the sudden changes. Holt-Winters is able to better follow the shape of the real-world data, aligning almost perfectly with the weekly patter. As we said before, Prophet tends to smooth out too much, missing completely the sudden peaks.

For the amplitude of the fluctuations, Holt-Winters is once again the model that gives us the best results, it is able to predict the real amplitudes, unlike the other two models.

Interpretability wise, we could say that Prophets line's look the cleanest, but the predicted values are too far from the reality to consider it accurate. ARIMA can be a bit chaotic, all depending on how it is calibrated. Finally, Holt-Winters shows the best readability, which is very important when studying the models.

4.3. Conclusion

After testing three different forecasting models, (ARIMA, Holt-Winters, and Prophet) Holt-Winters clearly stood out as the most reliable and accurate for predicting traffic intensities with our dataset. Across all loops, it consistently produced the lowest error values, both in terms of Mean Squared Error (MSE) and Mean Absolute Error (MAE). This indicates that the model's predictions were not only closer to the real values on average, but also avoided important prediction mistakes that could significantly affect the reliability of the forecast. In contrast, while ARIMA and Prophet had some strong points, they struggled to match Holt-Winters' performance, especially in capturing the weekly seasonality and short-term fluctuations that are common in traffic data.

What makes Holt-Winters particularly well-suited for this case is its ability to model data with strong, regular seasonal patterns, exactly the kind of patterns we see in daily and weekly traffic flows. The method takes into account both trend and seasonality in an additive way, allowing it to adapt to the cyclical nature of the data without overfitting or over-smoothing. Unlike Prophet, which tended to generalize too broadly, or ARIMA, which can be more rigid without explicitly handling seasonality, Holt-Winters offers a good balance between flexibility and precision for this type of time series.

Because of these advantages, we've decided to continue exploring and developing the Holt-Winters model in greater detail. The next steps will involve fine-tuning its parameters, extending the prediction horizon, and potentially integrating it with anomaly detection techniques to improve real-time traffic monitoring.

Further development of the Holt Winters model

During the following stages, we explored a few strategies aiming at refining the accuracy and the reliability of our Holt-Winters model. Once we had a version that performed well with short-term forecasts, we started to stretch its limits a bit, testing it with longer chunks of test data and aiming to predict further into the future. The idea was to see how well the model could hold up over time, not just over a week or two, but across a full month or more.

The original code followed a solid structure: it aggregated daily traffic counts from raw loop readings, cleaned up suspicious zero values, and applied a basic Holt-Winters model with additive trend and seasonality. While this model gave fairly acceptable predictions to start with, the goal was to strengthen the performance and the model's ability to generalize well across all loops.

Our first major experiment involved changing the seasonal component from additive to multiplicative. The idea behind this is simple: if the magnitude of seasonal fluctuations grew alongside overall traffic levels (ex. busier weeks showing stronger weekly patterns) a multiplicative setup could better capture that effect. However, this didn't work out as expected. This ended up inflating the errors on almost every loop, especially on those with lower or more stable counts. This indicated that our seasonal effects remained relatively steady regardless of the level, making additive the more appropriate choice.

Another method that we tried was applying a Box-Cox transformation, which is often used to stabilize variance in a time series where fluctuations increase with magnitude. However, applying Box-Cox in this context led to distortions on the results and less interpretable forecasts. It turned out that the variance in our data was already fairly stable. The transformation did more harm than good.

We also evaluated for different seasonal cycle length. We tried cycles of 5, 7, and 14 days, expecting the results to vary depending on how regular the weekly patterns were. Not surprisingly, the 7-day period consistently gave the most accurate forecasts across all loops. This makes a lot of sense, given that we're dealing with traffic data from a city street, where activity tends to follow a predictable weekly rhythm: heavier flows during weekdays and lighter volumes over the weekend. The results confirmed what we intuitively expected, urban traffic behaves on a weekly cycle, and the model performs best when it reflects that.

Ultimately, what worked best was sticking to the basics, using an additive trend, additive seasonality, and avoiding any transformations.

We also added defensive programming elements like try-except blocks to gracefully handle problematic loops and improve robustness. Small improvements like indexing by date before model fitting, trimming test data to match the forecast window, and cleaning up anomalies in the input data helped solidify the model pipeline.

Through trial and error, we saw clearly how not all recommended techniques apply equally to all data types.

5.1. Results and analysis

After all the testing we obtained the following results:

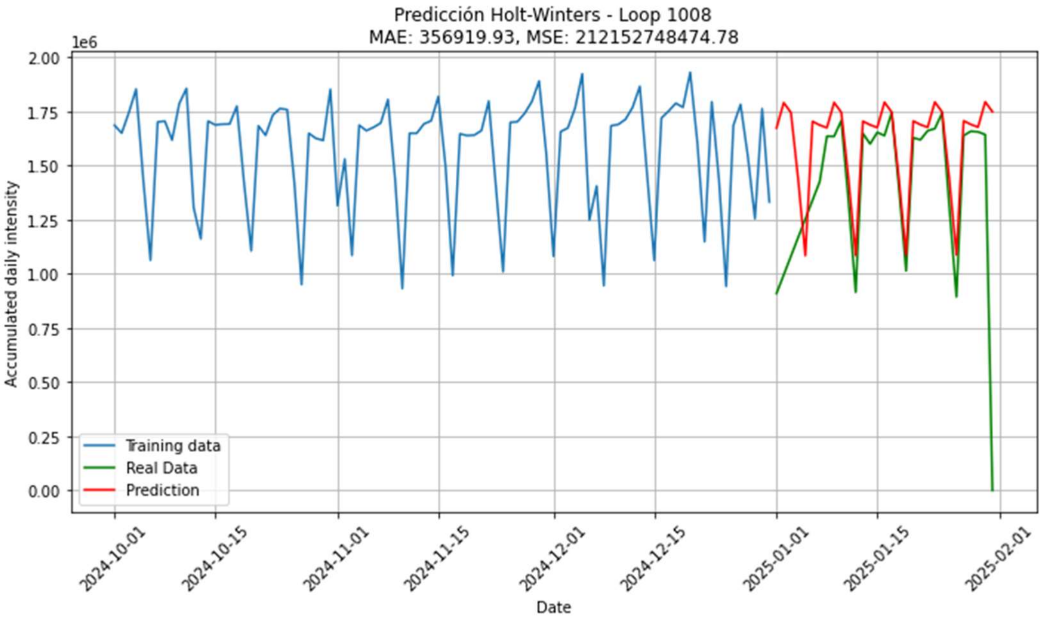


Figure 27. Holt Winters intensity predictions for a period of a month (Loop 1008)

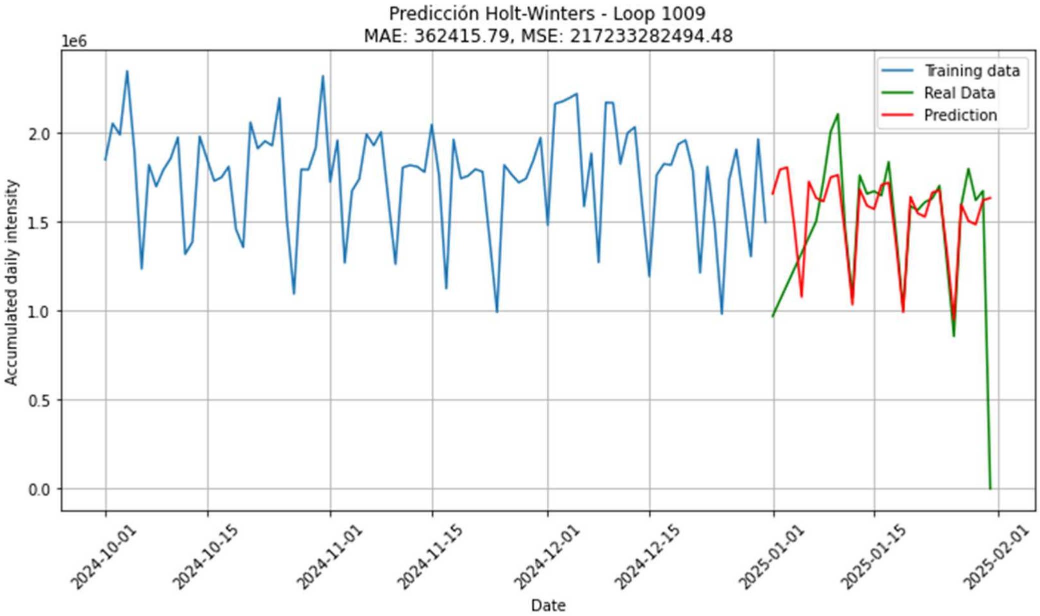


Figure 28. Holt Winters intensity predictions for a period of a month (Loop 1009)

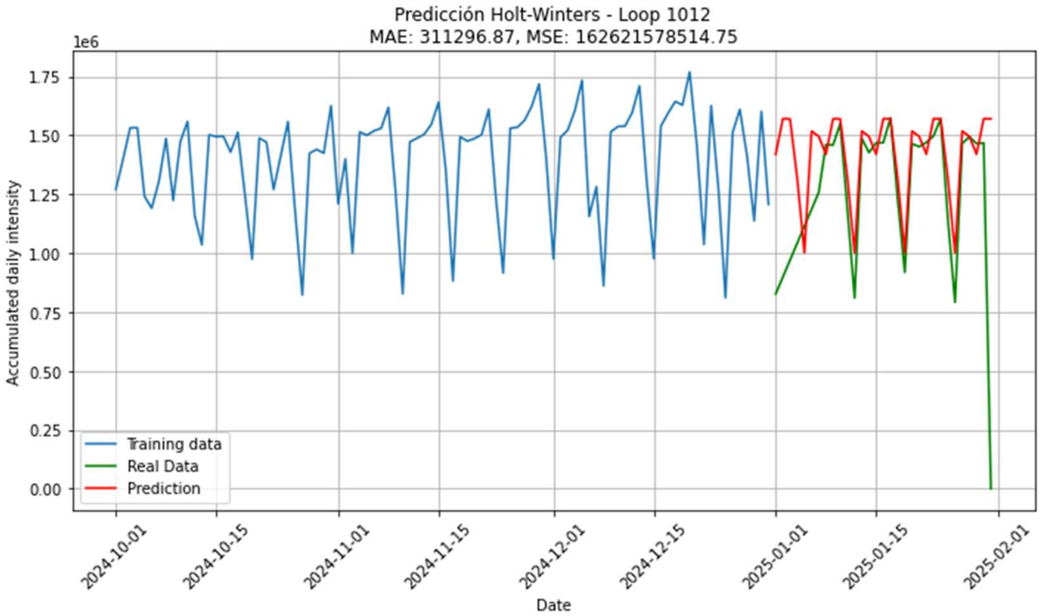


Figure 29. Holt Winters intensity predictions for a period of a month (Loop 1012)

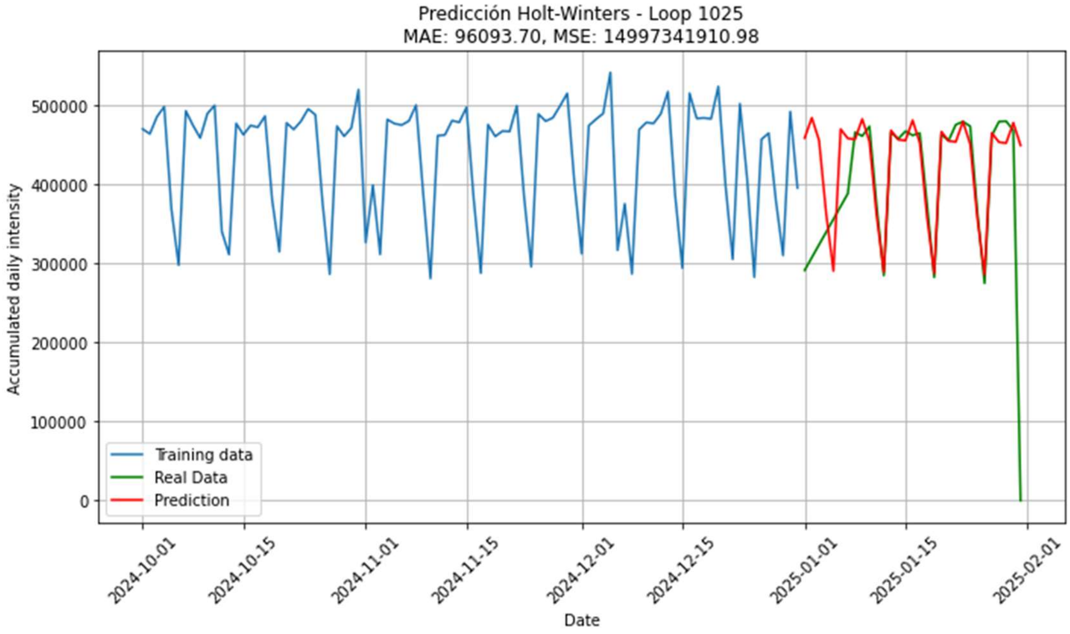


Figure 30. Holt Winters intensity predictions for a period of a month (Loop 1025)

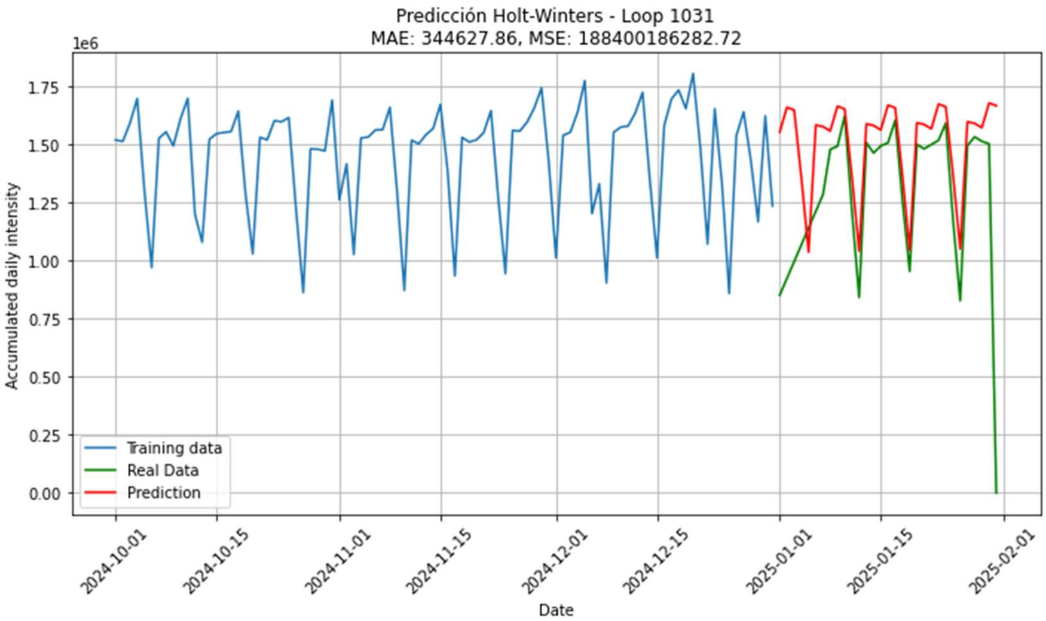


Figure 31. Holt Winters intensity predictions for a period of a month (Loop 1031)

We will now evaluate thoroughly how well our model performs. Our focus was on two commonly used forecasting metrics: Mean Absolute Error (MAE) and Mean Squared Error (MSE). These give us a clear sense of how far the model's predictions are from the actual traffic data. In different loops, we observed that the MAE generally stayed between 310,000 and 360,000, except for one loop, which doesn't bother us because it naturally had lower traffic levels. Given that the daily accumulated traffic intensities often reach 1.5 to 2 million units, these errors represent about 15% to 20% of the actual values on average. This might not sound all that precise, but in the context of time series forecasting, especially with no external variables involved, this level of error is quite acceptable and even expected.

Loop 1025 was an interesting case because unlike the others, it consistently registered lower traffic intensity values, which the model successfully captured. The MAE for this loop was around 96,000, which is much lower in absolute terms, but more importantly it reflects that the model is responsive to different traffic volumes and isn't rigid across scales. The fact that it can adapt and remain stable across such different contexts gives us confidence in its structure. The MSE values, though large, are naturally inflated due to the size of the data (since they square the error). Rather than being discouraged by those numbers, we focused more on the consistency of the results across loops and the shape of the forecasts themselves, and both turned out well.

It's also worth noting that we did consider the main public holidays during the modeling process, but we didn't see a significant impact on the results. This was likely because the model was already performing well on its own. The Holt-Winters method captured the weekly patterns and seasonal trends effectively, without the need for many additional adjustments. That said, incorporating holiday or weather effects could become more important when forecasting over longer time periods (ex. If we had years of training data), where deviations from usual patterns could have a greater influence. In the context of our current forecast window, the model's predictions already aligned closely with the observed data, we didn't feel the need to add these kinds of filters that weren't necessarily improving the model's performance.

To push the model further, we also experimented with different seasonal periods, testing 5-, 7-, and 14-day cycles, to see which of those would produce the most accurate forecasts. Unsurprisingly, the 7-day period came out on top. This aligns with what we already know about city traffic: it follows a clear weekly rhythm, between weekdays and weekends.

In general, the model performs well. It runs efficiently, and it produces results that are consistent, interpretable, and fairly accurate. Whether for monitoring expected traffic loads or serving as a base for more advanced models down the line, this Holt-Winters implementation holds its ground. In short, it's a solid forecasting tool.

For improving of our model in the future we could see a few main lines of work. The first one would be to include exogenous variables (for example with Sarimax or with the Prophet functionality that adds regressors)

If we look at it from the perspective of its limitations, this model is fairly simple. When used for more complex cases or longer periods of time it could be beneficial to take into account any external (or exogenous) variables that can be affecting our data. These kinds of variables are often the reason for unpredictable changes in the patterns that our models were having a hard time anticipating.

The second line of work would be to start using Machine learning models. This idea comes from the fact that the classical statistical models that we used in our work rely on the trends and the seasonality of the data. As we know, traffic data can often be chaotic and non-linear. The machine learning models, such as LSTM neural networks, can achieve a better understanding of the irregular ways in which traffic behaves and create logical relations between the external factors (weather conditions, events…) and the base data. They are often more capable of handling noise and missing data, which for longer periods of time is almost inevitable.

Lastly, one of the other options that we could consider is creating a hybrid model. These types of models stack different methods to create predictions. This allows us to improve the robustness of the model, combining all the strengths of different models to try reduce the error that the weaknesses of each model cause.

Ultimately, the current goal of predicting traffic is to achieve models that are more flexible and that can consider and adapt to the real context. This is what will allow us to make informed real-time decisions that are able to better manage the complex traffic flow in urban areas.

Conclusions and policy implications

6.1. Summary of conclusions

In this study, we explored three classical time series models, ARIMA, Holt-Winters and Prophet in the context of short-term traffic prediction, using the data from loop detectors in one of the main arteries of the city of Santander. With the obtained results, we can now go back and answer the questions that we set at the beginning of the study:

- *Which of the three models can give us the most accuracy in terms of short-term traffic forecasting?*

Holt-Winters constantly delivered the most accuracy in its results, especially when we used the additive seasonality. It was able to capture the main trends without lots of fine tuning. The other two models clearly underperformed.

- *How are these models different in their complexity, and difficulty of use?*

In terms of difficulty of use, Holt-Winters is the most user-friendly. Prophet, with its ability to automatically handle external factors and missing data, makes it accessible for users as well. ARIMA on the other hand was quite complex to calibrate an it is not suited for casual use.

- *How robust is each model when the data contains noise, is incomplete, or the traffic data is irregular?*

Prophet was the one able to handle irregularity the best. Its design is made to do this automatically, unlike the other models. Holt-Winters showed impressive resilience as well, although it has a harder time handling sudden changes. Lastly, ARIMA was considerably sensitive to irregularities or noise and needs a really good preparation of the data in order to perform.

This study found that Holt-Winters offered the best balance between accuracy, simplicity, and adaptability for short-term traffic forecasting. While Prophet handled irregularities and missing data well, ARIMA required more careful tuning and was less robust to noise. Overall, classical models remain effective tools, especially when working with clean, regular traffic patterns.

6.2. Policy implications

Beyond its technical contributions, this study opens the door to practical applications in urban mobility management. Predictive traffic modeling, even at a short time horizon, offers a real opportunity for city planners to move from reactive measures to proactive strategies.

In cities where congestion is a recurrent issue, anticipating traffic conditions a few days in advance can make a significant difference. For example, forecasts could help fine-tune traffic signal schedules during rush hour, schedule maintenance during low intensity windows, or better coordinate public transportation services. Similarly, city officials might use forecasted data to prepare for traffic spikes around planned events or emergencies.

The models that we studied in this paper are scalable. This framework can be adopted by any municipality with access to the loop detector data.

From a broader perspective, this work advocates for more integrated use of data prediction tools in traffic management. While more complex solutions, such as deep learning or spatial-temporal networks, could enhance accuracy, they may not be feasible or necessary for every urban setting.

To sum up, this thesis demonstrates that classical forecasting tools can still provide high-impact insights when applied with care and adapted to local traffic dynamics. Their potential to inform and support urban mobility policies is significant, especially in relatively small cities where data might be available, but the technical infrastructure may not be.

Bibliography

1. *Modeling and Forecasting Vehicular Traffic Flow as a Seasonal ARIMA Process: Theoretical Basis and Empirical Results.* **Williams, Billy M. and Hoel, Lester A.** 2003.

2. *Application of Subset Autoregressive Integrated Moving Average Model for Short-Term Freeway Traffic Volume Forecasting.* **Lee, Sangsoo and Fambro, Daniel.** 1999.

3. *Comparison of parametric and nonparametric models for traffic flow forecasting.* **Smith, Brian L., Williams, Billy M. and Oswald, R. Keith.** 2002.

4. *A Hybrid Prediction Method for Realistic Network Traffic With Temporal Convolutional Network and LSTM.* **Bi, Jing, et al.**

5. *Forecasting the network traffic with PROPHET.* **Aniszewski, Jakub, Walkowiak, Krzysztof and Knapińska, Aleksandra.**

6. *Investigating the Variability in Daily Traffic Counts through use of ARIMAX and SARIMAX Models: Assessing the Effect of Holidays on Two Site Locations.* **Cools, Mario and Wets, Geert.**

7. **Leduc, Guillaume.** *Road Traffic Data: Collection Methods and applications.* 2008.

8. **Federal Higway Administration.** *Traffic Detector Handbook, Chapter 2.* 2006.

9. *Evaluation and Improvement of Inductive Loop Traffic Detectors.* **Bikowitz, Edward W. and Ross, Scott P.**

10. *Traffic prediction for mobile network using Holt-Wintrer's Exponential smoothing.* **Tikunov, Denis and Nishimura, Toshikazu.**

11. *Predictin Computer Network Traffic: A Time Series Forecasting Approach using DWT, ARIMA and RNN.* **Madan , Rishabah and Mangipudi, Partha Sarathi.** Noida, India : IEEE, 2018.

12. *Hybrid machine learning algorithm and statistical time series model for network-wide traffic forecast.* **Ma, Tao, Antoniou, Constantinos and Toledo, Tomer.** 2019.

13. **Prophet - Forecasting at scale. *https://facebook.github.io/prophet/.* [Online]**

14. *Short-Term Traffic Flow Forecasting Using Facebook Prophet Model.* Alsbou, A. and Alsbou, N.

15. *Forecasting at scale.* Taylor, S.J. and Letham, B.

16. *Effects of traffic control regulation on Network Macroscopic.* Alonso , Borja, et al. 2019, Elsevier.

17. *Estimating fundamental diagram for multi-modal signalized.* Ruyang, Yin, Nan , Zheng and Zhiyuan, Liu. 2022, Elsevier.

18. *Traffic dynamics: Its impact on the Macroscopic Fundamental Diagram.* Knoop, Victor L., van Lint, Hans and Hoogendoorn, Serge P. . 2015, Elsevier.

19. *Network Fundamental Diagram (NFD) and traffic signal control: first empirical evidences from the city of Santander* . Alonso, Borja, et al. 2017, Elsevier.

20. *Electromagnetic Loop Vehicle Detectors* . Anderson, Robert L. 1970.

21. *Introduction to Time Series and Forecasting* . Brockwell, J.P. New York : Springer Science + Business Media , 2016.

22. ***An NN approach for MPEG video traffic prediction.*** **Dharmadhikari, Vidhyadhar B. and Gavade, J.D. 2010.**

23. ***Evaluation of neural network architectures for MPEG-4 video traffic prediction.*** **Abdennour, Abdel. 2006.**

24. Haykin, Simon. ***Neural Networks, 2nd Edition.*** **1994.**

25. Charniak, Eugene. ***Introduction to Deep Learning.*** **2021.**

26. Zhang, G. Peter. ***Time series forecasting using a hybrid ARIMA and neural network model.*** **2003.**

27. Amidi , Afshine and Amidi, Shervine. Pense-bête de réseaux de neurones récurrents. ***Stanford.edu.*** **[Online] [Cited:] https://stanford.edu/~shervine/l/fr/teaching/cs-230/pense-bete-reseaux-neurones-recurrents.**

29. Box, G.E.P., Reinsel, G.C. and Ljung, G.M. ***Time Series Analysis: Forecasting and Control (5th ed.).*** **2015.**

Comparative analysis of time series models for short-term urban traffic forecasting

UNIVERSIDAD DE CANTABRIA

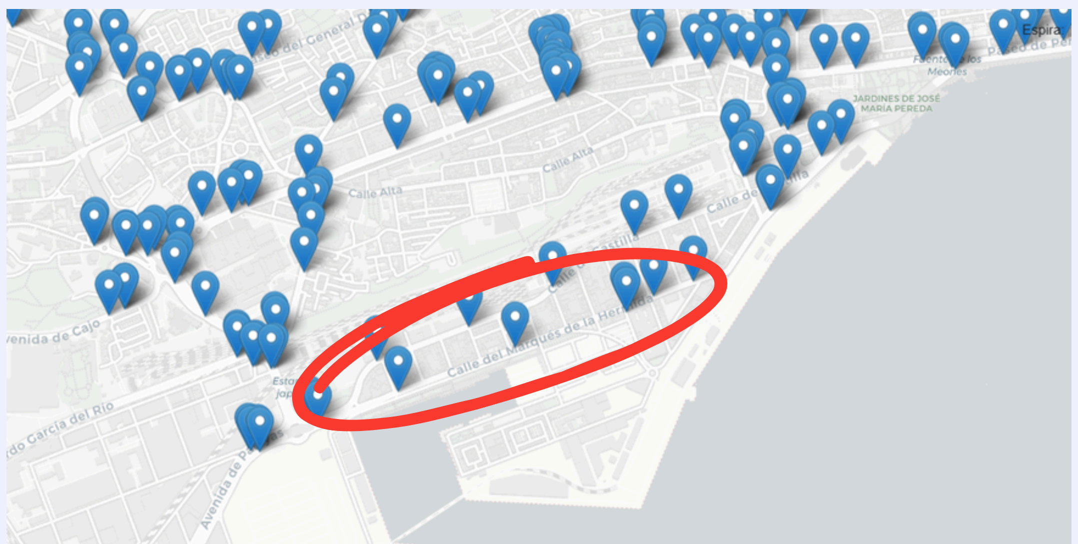
ESCUELA DE CAMINOS UNIVERSIDAD DE CANTABRIA

Criteria	Classical Statistical Models (Holt-Winters, ARIMA, SARIMA)	Hybrid & Machine Learning Models (Prophet, Random Forest, LSTM)	Models with Exogenous Variables (SARIMAX, Prophet+regressors, VAR)
User-friendliness	**High** Easy to implement and well-documented	**Medium** Prophet is easy, others like LSTM are complex	**Medium** Needs careful variable selection and prep
Data requirements	**Low** Works with small or incomplete datasets	**High** Requires large, clean datasets	**Medium to High** Depends on exogenous data availability
Interpretability	**High** Clear and explainable parameters	**Low** Often black-box models	**Medium** Interpretability depends on model and variables
Sensitivity to outliers	**Low sensitivity** Robust to moderate anomalies	**High sensitivity** Can be misled by noise	**Moderate** External variables can amplify/reduce sensitivity
Handling missing data	**Limited** Requires preprocessing	**High** Prophet and some ML models handle gaps well	**Moderate** Must handle missing exogenous data manually
Adaptability to non-linear patterns	**Low** Limited to linear patterns	**High** Captures nonlinearities effectively	**Moderate** Can model complex dynamics with regressors
Suitability for traffic data with seasonality	**Excellent** Designed for strong seasonal patterns	**Good** Especially Prophet handles seasonality well	**Good** Can enhance seasonal modeling with external variables

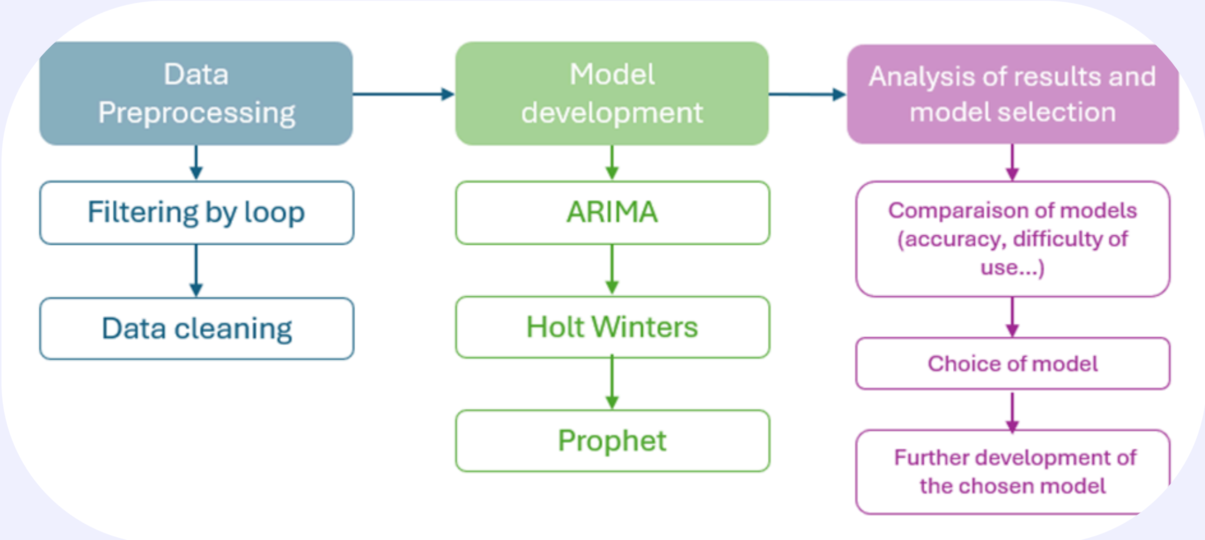
Objective → Answering these questions

- Which of the three models can give us the most accuracy in terms of short-term traffic forecasting?
- How are these models different in their complexity, and difficulty of use?
- How robust is each model when the data contains noise, is incomplete, or the traffic data is irregular?

Area of study - Marques de la Hermida (Santander)



Followed methodology



ARIMA

ARIMA is a classic **statistical model** used to forecast time series data by combining past values and errors in the series.

It's particularly useful when the data doesn't have a clear seasonal cycle but still shows strong temporal patterns.

Although it requires **careful parameter tuning**, it remains a solid baseline for many forecasting problems.

Holt-Winters

The Holt-Winters method is a time series forecasting technique that accounts for both **trends and repeating seasonal patterns**, like those seen in weekly traffic data.

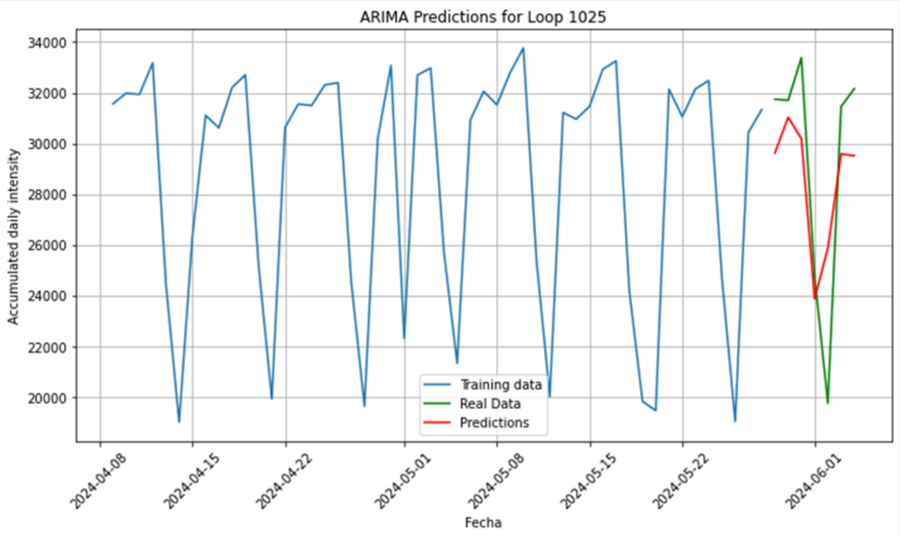
It works by smoothing the data over time and adjusting dynamically as new values appear. Because of its ability to model seasonality explicitly, it's **often a good fit for structured, recurring behaviors** like urban traffic flows.

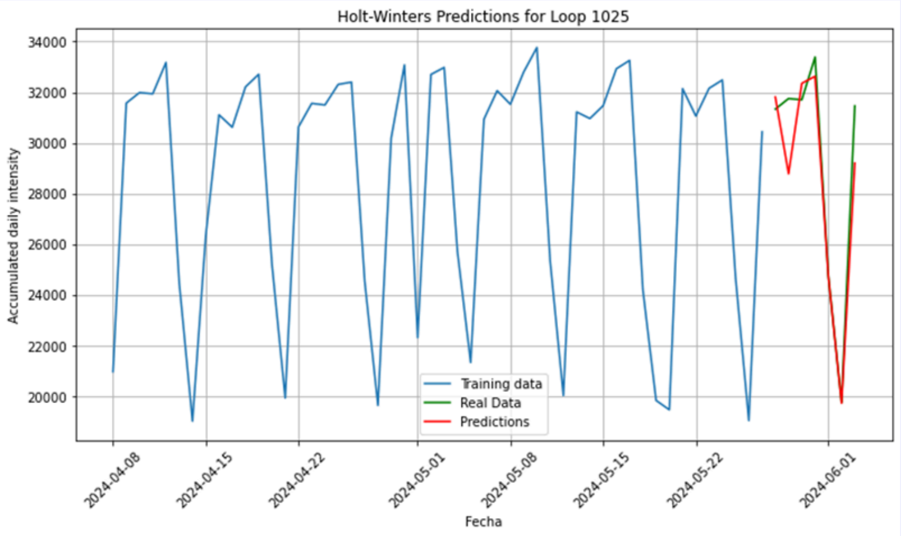
Prophet

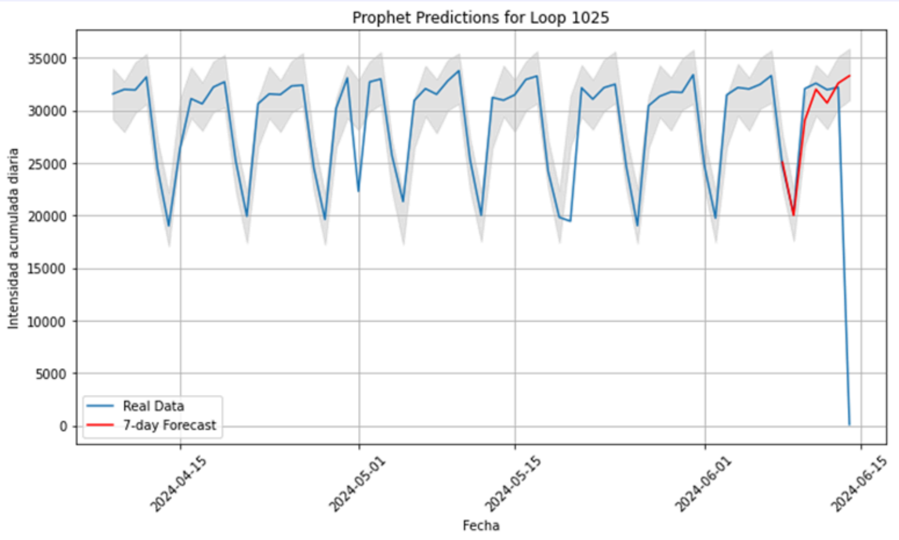
Prophet is a forecasting tool developed by Meta (Facebook) that's designed to be **flexible and easy to use**, even for non-experts.

It breaks down the time series into trend and seasonality.

Its main strength is how intuitively it can model complex patterns without needing extensive manual tuning, making it a popular choice for real-world applications with noisy or irregular data.







Results

	ARIMA		Holt-Winters		Prophet	
Loop	MAE	MSE	MAE	MSE	MAE	MSE
1008	8724.67	131353327.14	3712	30680379	21015.81	2225782280.26
1009	10173.28	166521679.26	2064	29175916	1563845731.59	2364960862.09
1012	8519.11	126426436.61	6604	69886999	18126.16	23128.73
1025	2503.14	9143388.96	1024	2165048	5561.46	158831078.67
1031	7491.11	99229489.92	3474	22345760	19093.38	1851431390.72

- **Holt-Winters constantly delivered the most accuracy** in its results, especially when we used the additive seasonality. It was able to capture the main trends without lots of fine tuning. The other two models clearly underperformed.
- In terms of difficulty of use, **Holt-Winters is the most user-friendly**. Prophet, with its ability to automatically handle external factors and missing data, makes it accessible for users as well. ARIMA on the other hand was quite complex to calibrate an it is not suited for casual use.
- **Prophet was the one able to handle irregularity the best**. Its design is made to do this automatically, unlike the other models. Holt-Winters showed impressive resilience as well, although it has a harder time handling sudden changes. Lastly, ARIMA was considerably sensitive to irregularities or noise and needs a really good preparation of the data in order to perform.

Conclusion

- **For short-term traffic prediction in urban environments, classical models like Holt-Winters remain competitive when seasonality is present.**
- **Future improvements could involve adding external variables (like weather or events) or exploring machine learning models (e.g., LSTM) to capture more complex dynamics.**
- **A balance between accuracy, interpretability, and practicality is key for real-world applications like urban traffic management.**