# Time series analysis on hourly traffic congestion indicator

Sergio Verga

July 13, 2025


## Abstract

This work presents a comparative study of time series forecasting models applied to an hourly traffic congestion indicator for a major U.S. freeway. The analysis explores three modeling families: Seasonal ARIMA (SARIMAX), Unobserved Components Models (UCM), and Deep Learning architectures (LSTM/GRU). The dataset exhibits strong daily and weekly seasonality, which is addressed through global and per-hour modeling strategies, as well as the incorporation of calendar-based features. Results show that the best performance is achieved by a per-hour SARIMAX model with conditional seasonal differencing (MAE = 0.0120), followed closely by a per-hour UCM model enhanced with a holiday regressor (MAE = 0.0125). The study highlights the importance of seasonality-aware modeling and provides recommendations for further improvement through feature engineering, ensemble methods, and deeper hyperparameter optimization.


## Contents


**1 Introduction** 1

**2 Dataset exploration** 1
- 2.1 Stationarity and Seasonality . . . . . . . . 2

**3 Training pipeline** 2

**4 SARIMA models** 3
- 4.1 Baseline ARIMA (Non-seasonal) . . . . . 3
- 4.2 Seasonal Extension: SARIMAX . . . . . . 3
- 4.3 Per-Hour SARIMAX Modeling . . . . . . 4
- 4.4 Improved Per-Hour SARIMAX with Differencing . . . . . . . . . . . . . . . . . . . . 4

**5 UCM** 5
- 5.1 Model comparison: global vs per-hour UCM 5
- 5.2 Effect of Holiday Feature on Per-Hour UCM 5

**6 Deep Learning models** 6
- 6.1 Baseline LSTM Model . . . . . . . . . . . . 6
- 6.2 Benchmark Architectures and Model Selection . . . . . . . . . . . . . . . . . . . . . 6
- 6.3 Hyperparameter Optimization . . . . . . . 7
- 6.4 With and Without Calendar Features . . 7

**7 Results** 7
- 7.1 Validation MAE Comparison . . . . . . . 7

**8 Conclusions and future perspectives** 8


## 1 Introduction

The objective of this study is to analyze a dataset containing a hourly time series describing a traffic congestion indicator for a freeway in a large U.S. city. For this purpose, three families of forecasting models are developed:

- An Autoregressive Integrated Moving Average (**ARIMA**) model.
- An Unobserved Components Model (**UCM**).
- A **Deep Learning**-based model.

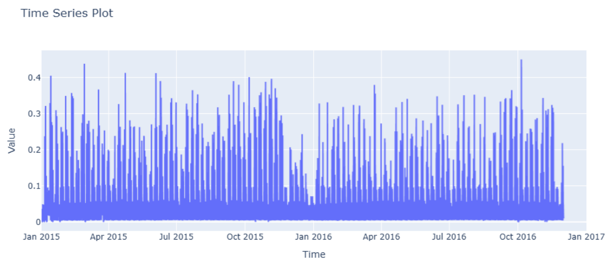
Each one is analyzed in the report. First, a section describes the results of the dataset exploration phase, also inspecting the seasonality and stationarity of the time series, as well as the training pipeline used in the theee families of models. Then, each family of model is studied in detail (ARIMA, UCM, and DL-based) and finally the results are evaluated in the conclusive section.

**Note:** the codebase is publicly available in Github.

## 2 Dataset exploration

The dataset contains an hourly time series, describing a traffic congestion indicator for a freeway in a large U.S. city. In particular, it is composed of the following features:

- **DateTime**: The timestamp of each observation [yyyy-mm-dd hh:mm:ss].
- **Date**: The corresponding date of the observation [yyyy-mm-dd].
- **Hour**: The hour of the day (0-23) [Integer].
- **X**: The target time series variable to be forecasted [Double].

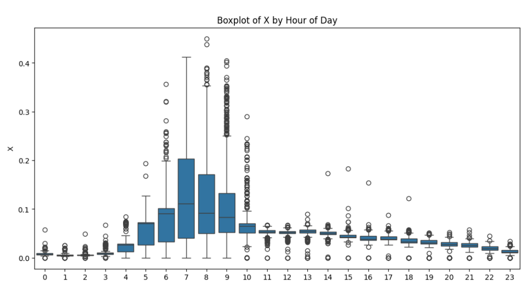



In total, the dataset has 17544 rows, with completeness in each temporal feature. The target variable X has 744 missing values, representing the time points to predict. A preliminary exploratory data analysis (EDA) was conducted to assess the structure and statistical properties of the dataset. Table 2 presents key summary statistics.

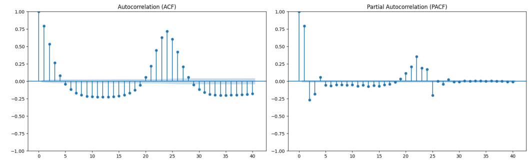
| | **Min** | **Mean** | **Median** | **Max** |
|---|---|---|---|---|
| X | 0.0000 | 0.0463 | 0.0368 | 0.4500 |

## 2.1 Stationarity and Seasonality

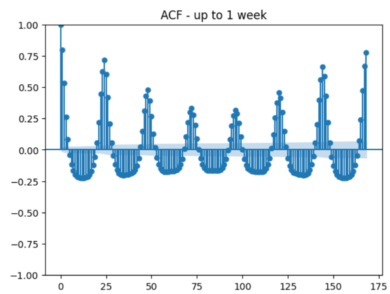
The stationarity of the time series was assessed using the Augmented Dickey-Fuller (ADF) test. The test yielded a statistic of $-15.711$ with a $p$-value effectively equal to zero, well below the 1% critical value ($-3.4307$). This strongly confirms that the series is stationary without requiring differencing. In terms of seasonality, clear daily patterns were identified, particularly in the distribution of `X` across hours of the day. As shown in Figure 2.1, peak activity consistently occurs between 6 AM and 10 AM, suggesting that incorporating hourly seasonal effects may improve model performance.




The autocorrelation and partial autocorrelation plots provide further evidence of temporal structure in the series. The ACF shows a slow decay with a prominent secondary wave around lag 24, while the PACF displays a sharp spike at lag 1 and smaller seasonal peaks near lag 24. These patterns suggest both a short-term autoregressive component and strong daily seasonality (24-hour periodicity). Such findings justify the inclusion of seasonal terms in subsequent time series models. Candidate structures include $\text{SARIMA}(1,0,0) \times (1,0,0)_{24}$ or a state-space specification using a Unobserved Components Model (UCM) with a stochastic seasonal component of period 24.




Also important is the weekly seasonality (s=168 hours) for the patterns identifiable in the different days of the week.




# 3 Training pipeline

The dataset is split as follows:

1. **Test**: last 744 data points
2. **Validation**: previous 744 data points
3. **Training**: all of the previous data points

For each of the families, the pipeline follows the logic:

- Use **training** and **validation** to select the optimal model configuration. The optimal model is defined optimizing MAE
- Retrain the optimal model configuration on the training dataset plus the validation dataset
- Forecast the test dataset.

### Evaluation metric: Mean Absolute Error

The Mean Absolute Error (MAE) is a commonly used metric to evaluate the accuracy of forecasts in time series analysis. It measures the average magnitude of the errors between predicted and actual values, without considering their direction. Formally, for a series of $n$ predictions $\hat{y}_t$ and actual values $y_t$, the MAE is defined as:

$$\text{MAE} = \frac{1}{n} \sum_{t=1}^{n} |y_t - \hat{y}_t| \, .$$

MAE is scale-dependent and provides an intuitive measure of average prediction error in the same unit as the data.

# 4 SARIMA models

The Autoregressive Integrated Moving Average (ARIMA) model is a general class of linear models for time series forecasting, characterized by three components:

- **Autoregressive (AR)**: The AR component captures the influence of past values on the current value. An AR($p$) model expresses the series as a linear combination of its $p$ previous values:

$$Y_t = \phi_1 Y_{t-1} + \cdots + \phi_p Y_{t-p} + \epsilon_t,$$

where $\epsilon_t$ is a white noise process.

- **Integrated (I)**: The I component accounts for non-stationarity by differencing the series $d$ times. A time series is said to be integrated of order $d$ (denoted $I(d)$) if it becomes stationary after $d$ differences:

$$Y'_t = \Delta^d Y_t.$$

- **Moving Average (MA)**: The MA component models the error term as a linear combination of past white noise terms. An MA($q$) model is given by:

$$Y_t = \epsilon_t + \theta_1 \epsilon_{t-1} + \cdots + \theta_q \epsilon_{t-q}.$$

The general ARIMA($p$, $d$, $q$) model thus combines these components:

$$\Delta^d Y_t = \phi_1 Y_{t-1} + \cdots + \phi_p Y_{t-p} + \epsilon_t + \theta_1 \epsilon_{t-1} + \cdots + \theta_q \epsilon_{t-q}.$$

Although ARIMA models are effective for a wide range of series, they do not natively incorporate seasonality. This limitation motivates the use of their seasonal extension, SARIMA, discussed later.

## 4.1 Baseline ARIMA (Non-seasonal)

As a baseline, ARIMA models were tested on the hourly congestion indicator series, despite the presence of evident daily and weekly seasonality. This choice serves to quantify the performance gain later achieved by explicitly modeling seasonal components. The training pipeline described in Section 3 was adopted, splitting the data into training (pre-validation), validation, and test segments. The ARIMA models were fit on the training data, and evaluated on the validation set using Mean Absolute Error (MAE). A grid search over a small range of parameters ($p, d, q \in [0, 1, 2]$) was conducted to identify the optimal configuration.

The best model configuration was ARIMA(2,0,1), achieving a validation MAE of 0.0313. This confirms that the series is already stationary ($d = 0$), consistent with the results from the Augmented Dickey-Fuller test. However, the model does not explicitly account for daily (24-hour) or weekly (168-hour) seasonal structure. In the subsequent section, this limitation is addressed using SARIMA models.

| **p** | **d** | **q** | **Val MAE** |
|---|---|---|---|
| 0 | 0 | 0 | 0.0314 |
| 0 | 0 | 1 | 0.0314 |
| 0 | 0 | 2 | 0.0314 |
| 0 | 1 | 0 | 0.0393 |
| 0 | 1 | 1 | 0.0398 |
| 0 | 1 | 2 | 0.0408 |
| 1 | 0 | 0 | 0.0314 |
| 1 | 0 | 1 | 0.0314 |
| 1 | 0 | 2 | 0.0313 |
| 1 | 1 | 0 | 0.0400 |
| 1 | 1 | 1 | 0.0400 |
| 1 | 1 | 2 | 0.0313 |
| 2 | 0 | 0 | 0.0313 |
| 2 | 0 | 1 | **0.0313** |
| 2 | 0 | 2 | 0.0313 |
| 2 | 1 | 0 | 0.0400 |
| 2 | 1 | 1 | 0.0403 |
| 2 | 1 | 2 | 0.0324 |

## 4.2 Seasonal Extension: SARIMAX

Given the suboptimal performance of the non-seasonal ARIMA model in the presence of evident daily and weekly periodicity, a seasonal extension was tested. The SARIMAX model (Seasonal AutoRegressive Integrated Moving Average with eXogenous regressors) generalizes ARIMA by incorporating seasonal autoregressive and moving average terms. It is parameterized as:

$$\text{SARIMAX}(p, d, q) \times (P, D, Q)_s,$$

where $(P, D, Q)$ defines the seasonal components and $s$ is the seasonal period.

An initial model with configuration SARIMAX(1,0,1) $\times$ $(1,0,1)_{24}$ was tested, incorporating a single seasonal AR and MA term with a daily frequency ($s = 24$). This configuration, without further tuning, already resulted in a substantial reduction in validation MAE:

$$\text{Validation MAE} = 0.0186$$

This represents a clear improvement over the best ARIMA configuration, whose MAE was 0.0313. The result confirmed the importance of explicitly modeling daily seasonality.

#### Seasonal Parameter Optimization

To further refine the model, alternative seasonal configurations were explored while keeping the non-seasonal part fixed as $(1, 0, 1)$. The following seasonal orders were tested:

- (1,0,1,24)
- (1,0,2,24)
- (2,0,1,24)

Weekly seasonalities (e.g., with period $s = 168$) were also considered but not evaluated in this round.

| Seasonal Order | Validation MAE |
|---|---|
| (1, 0, 1, 24) | **0.0186** |
| (1, 0, 2, 24) | 0.0186 |
| (2, 0, 1, 24) | 0.0186 |

The performance remained stable across all tested seasonal configurations. This suggests that the key factor contributing to the performance gain is the inclusion of daily seasonality itself, rather than the specific choice of seasonal AR or MA lag order. The configuration SARIMAX(1,0,1) $\times$ $(1,0,1)_{24}$ was therefore retained as the final model for this family.

## 4.3 Per-Hour SARIMAX Modeling

To simultaneously capture both daily and weekly seasonalities, an alternative strategy was explored by constructing a separate SARIMAX model for each hour of the day. This approach, referred to as the *Per-Hour* model, transforms the original time series into 24 independent sub-series, each corresponding to a fixed hour across days (e.g., a series of all 7:00 AM values). Each sub-series was modeled independently, allowing the model to capture hour-specific weekly seasonal dynamics.

| Hour | Stationary (ADF) |
|---|---|
| 00–18 | Yes |
| 19–23 | No |

In this initial configuration, no differencing was applied to any of the 24 hourly series. That is, both the non-seasonal differencing order $d$ and the seasonal differencing order $D$ were fixed to 0 across all models. While some sub-series displayed borderline non-stationary behavior in the Augmented Dickey-Fuller (ADF) test—especially in the late evening hours—this choice was intentional to keep the structure simple and consistent across all hours. The training and validation sets were segmented by hour. For each of the 24 sub-series, a SARIMAX(1,0,1) $\times$ $(1,0,1)_7$ model was trained and used to forecast the corresponding segment of the validation period. The forecasts were then recombined to produce the full validation prediction. The Mean Absolute Error (MAE) on the validation set for the Per-Hour model was 0.0246, outperforming the baseline ARIMA but underperforming the global SARIMAX:

| Model | Validation MAE |
|---|---|
| ARIMA(2,0,1) | 0.0313 |
| SARIMAX(1,0,1)$(1,0,1)_{24}$ | **0.0186** |
| Per-Hour SARIMAX | 0.0246 |

The Per-Hour strategy demonstrated that modeling weekly seasonal effects on a per-hour basis can yield improvements over non-seasonal baselines. However, it did not outperform the global SARIMAX model that explicitly captures daily seasonality through a seasonal period of 24. This performance gap is largely attributed to the absence of differencing terms and the smaller sample size available in each hourly sub-series, which may limit the capacity to capture underlying trends and seasonality effectively.

## 4.4 Improved Per-Hour SARIMAX with Differencing

To address the limitations of the previous per-hour approach—where neither trend nor seasonal differencing was applied—an improved configuration was implemented. This version conditionally includes seasonal differencing of order $D = 1$ for those hourly sub-series that exhibited non-stationarity in the earlier Augmented Dickey-Fuller (ADF) analysis (specifically, for hours $\geq$ 19). For all other hours, $D = 0$ was retained. No trend differencing ($d = 0$) was applied in any case. All models in this configuration use a fixed seasonal period of 7 (weekly seasonality).

#### Model Configuration Search

Two candidate configurations were tested for the autoregressive (AR) and moving average (MA) components, both at the non-seasonal and seasonal levels:

- **Config A:** SARIMAX(1,0,1) $\times$ $(1,D,1)_7$
- **Config B:** SARIMAX(2,0,2) $\times$ $(2,D,2)_7$

where $D = 1$ only for hours 19–23, and $D = 0$ otherwise.

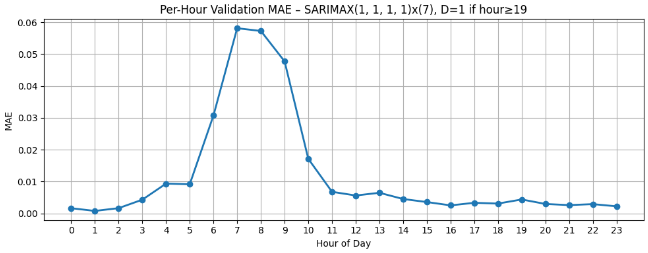
Each configuration was applied across all 24 hourly sub-series using this conditional differencing logic. The individual forecasts were then reassembled to compute the global validation error.

| Configuration | Validation MAE |
|---|---|
| (1,0,1) $\times$ $(1,D,1)_7$ | **0.0120** |
| (2,0,2) $\times$ $(2,D,2)_7$ | 0.0122 |

#### Hourly MAE Distribution

The best-performing configuration—SARIMAX(1,0,1) $\times$ $(1,D,1)_7$, with seasonal differencing applied only to hours 19 and above—achieved a global validation MAE of 0.0120. The hour-wise MAE distribution is reported in Figure 4.4, which shows a marked improvement compared to earlier models, particularly during peak morning hours.

Given the superior performance of this configuration, it was selected as the final per-hour forecasting model. It is used in the next stage to generate predictions on the held-out test set.




# 5 UCM

The Unobserved Components Model (UCM) expresses a time series as the sum of interpretable stochastic components such as trend, seasonality, cycle, and noise:

$$Y_t = \mu_t + \gamma_t + \psi_t + \epsilon_t,$$

where $\mu_t$ is the trend, $\gamma_t$ the seasonal component, $\psi_t$ the cycle, and $\epsilon_t \sim \mathcal{N}(0, \sigma_\epsilon^2)$ the irregular term. The trend is typically modeled as a local linear trend:

$$\mu_t = \mu_{t-1} + \beta_{t-1} + \eta_t, \quad \beta_t = \beta_{t-1} + \zeta_t,$$

with $\eta_t \sim \mathcal{N}(0, \sigma_\eta^2)$ and $\zeta_t \sim \mathcal{N}(0, \sigma_\zeta^2)$, allowing both the level and slope to evolve over time. Seasonality is represented as a stochastic process of known period $s$, constrained to have zero mean over each season:

$$\sum_{i=0}^{s-1} \gamma_{t-i} = 0.$$

Cyclical behavior can be captured using a damped stochastic oscillator. However, in this work, only trend and seasonal components are used. The UCM is estimated in state-space form via the Kalman filter, which enables efficient inference and forecasting.

## 5.1 Model comparison: global vs per-hour UCM

Two configurations of the Unobserved Components Model were evaluated using the validation pipeline described in Section 3. Both models rely on a local linear trend and a seasonal component with weekly periodicity.

- **Model A (global UCM)**: a single UCM is trained on the full hourly time series, capturing overall dynamics with no hour-specific segmentation.
- **Model B (per-hour UCM)**: a separate UCM is trained for each of the 24 hourly sub-series. Each model receives only the values corresponding to a fixed hour across different days, allowing hour-specific seasonal patterns to be learned.

Each model was trained on the combined training and validation set and evaluated on the validation set using Mean Absolute Error (MAE). The per-hour approach was designed to better capture inter-day variations specific to each hour, such as weekday/weekend effects or recurring hourly shifts.

| Model | Validation MAE |
|---|---|
| Global UCM | 0.0274 |
| Per-Hour UCM | **0.0251** |

The per-hour decomposition yields a consistent improvement in validation MAE, indicating that modeling hour-specific series independently allows the model to better exploit recurring intra-week structure. This strategy is retained for test forecasting in the final UCM pipeline.

## 5.2 Effect of Holiday Feature on Per-Hour UCM

To assess the impact of calendar effects, an additional experiment was conducted by enriching the per-hour UCM with a binary regressor indicating U.S. public holidays. For each hourly sub-series, a dummy variable was introduced to indicate whether the observation falls on a holiday. This feature was included as an exogenous regressor in the model specification:

$$Y_t = \mu_t + \gamma_t + \delta H_t + \epsilon_t,$$

where $H_t \in \{0, 1\}$ denotes the holiday indicator and $\delta$ is a fixed effect capturing the deviation in congestion on holidays.
Each of the 24 hourly UCMs was re-estimated with the additional holiday regressor, maintaining the same weekly seasonal structure and local linear trend. Validation performance was computed as before using the Mean Absolute Error (MAE) on the full validation period.

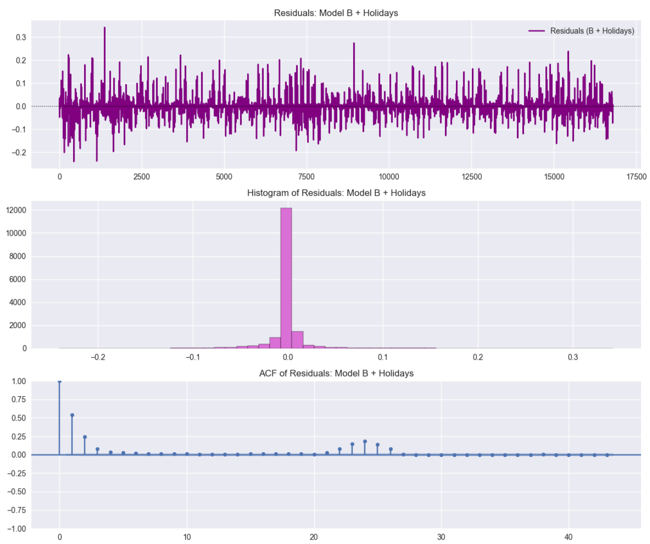
| Model | Validation MAE |
|---|---|
| Per-Hour UCM (no holidays) | 0.0251 |
| Per-Hour UCM (+ holidays) | **0.0125** |

The addition of the holiday indicator led to a substantial improvement in validation MAE, effectively halving the average forecasting error. This highlights the importance of explicitly modeling holiday-related effects in traffic-related time series, as they significantly impact congestion levels.

### Residual Diagnostics

Figure 5.2 displays residual diagnostics for the final per-hour UCM model with holiday adjustment. The top panel shows the residuals over time, which exhibit no visible long-term trends or structural breaks, suggesting stable model behavior. The histogram (middle panel) confirms that residuals are centered around zero with a sharp concentration, indicating low overall bias.

The bottom panel presents the autocorrelation function (ACF) of the residuals. Most correlations lie within the 95% confidence bounds, with no strong autocorrelation after the first few lags. This confirms that the model has effectively captured the main temporal structure, leaving behind mostly white noise.




# 6 Deep Learning models

This section describes the implementation and evaluation of deep learning approaches, specifically LSTM [?] and GRU architectures [?], for time series forecasting. Unlike the ARIMA and UCM models, the deep learning models pipeline has a preprocessing step of scaling (with MinMax in [0,1]). The target variable is scaled using statistics computed over the training set only, and these parameters are reused for transforming the validation and test sets (and then revert the transformation after the final prediction).

## 6.1 Baseline LSTM Model

The baseline model is a single-layer Long Short-Term Memory network with 64 hidden units (**LSTM 1x64**). This architecture was selected as an initial benchmark due to its simplicity and robustness in capturing temporal dependencies. The input consists of univariate sequences with a lookback window of 168 hours (one week), reshaped to 3D tensors for compatibility with Keras LSTM layers. The model ends with a fully connected layer to output a one-hour-ahead forecast. It is trained using the Mean Absolute Error (MAE) loss function and optimized with the Adam optimizer (learning rate $10^{-3}$). Early stopping is applied based on validation MAE, with a patience of 5 epochs. The model is trained over a maximum of 20 epochs with a batch size of 64. Input data is scaled using Min-Max normalization based on training statistics; test values are left untouched to prevent leakage.

| **Component** | **Specification** |
|---|---|
| Architecture | LSTM 1x64 |
| Lookback window | 168 hours (1 week) |
| Forecast horizon | 1 hour ahead |
| Input shape | (168, 1) |
| Output layer | Dense(1), linear activation |
| Loss function | Mean Absolute Error (MAE) |
| Optimizer | Adam (learning rate = $10^{-3}$) |
| Epochs | 20 |
| Batch size | 64 |
| Early stopping | Patience = 5 (val MAE) |
| Scaling | Min-Max (fitted on training set) |

## 6.2 Benchmark Architectures and Model Selection

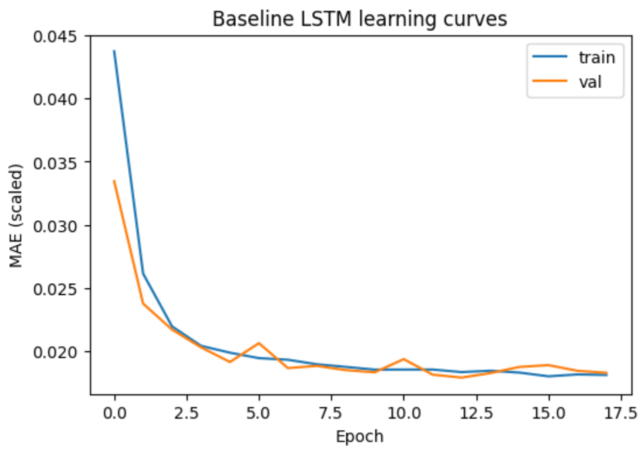
To assess the performance of the baseline model, we implemented and evaluated two additional deep learning architectures. Each model shares the same data preprocessing, input structure, and training configuration, differing only in architecture depth or recurrent cell type:

- **LSTM 1x64 (baseline)**: A single-layer LSTM with 64 units and no dropout. It serves as the simplest recurrent benchmark and provides strong generalization on the validation set.
- **LSTM 2x128 with dropout**: A deeper architecture composed of two stacked LSTM layers, each with 128 units. A dropout layer with rate 0.2 is applied between layers to mitigate overfitting. The final LSTM layer feeds into a dense output layer for one-step-ahead prediction.
- **GRU 2x64**: This model uses Gated Recurrent Units (GRUs) instead of LSTMs, with two layers of 64 units each. GRUs are computationally simpler than LSTMs and sometimes offer faster training with similar accuracy.

The results in Table 6.2 show that while deeper or alternative architectures can match the baseline performance in terms of validation MAE, they do not outperform it. The single-layer LSTM achieves the lowest validation error with fewer parameters, and thus is selected as the final model for test forecasting.

| **Model** | **Val MAE** |
|---|---|
| LSTM 1x64 | **0.01789** |
| LSTM 2x128 with dropout | 0.01794 |
| GRU 2x64 | 0.01801 |

The training curve for the LSTM 1x64 are as follows.




### 6.3 Hyperparameter Optimization

To assess the sensitivity of the baseline model to its recurrent capacity, we performed a grid search over the number of hidden units in a single-layer LSTM. Specifically, we tested `[16, 32, 64, 128]` units while keeping all other parameters fixed: no dropout, one recurrent layer, batch size 64, and a patience of 5 epochs. Each configuration was trained and evaluated using the same data splits and early stopping criteria. The goal was to identify whether a more compact or more expressive model could outperform the baseline in terms of validation MAE.

| Model | Units | Validation MAE (scaled) |
|---|---|---|
| LSTM 1x128 | 128 | **0.01795** |
| LSTM 1x64 | 64 | 0.01805 |
| LSTM 1x32 | 32 | 0.01855 |
| LSTM 1x16 | 16 | 0.01856 |

The results show that while increasing the number of units slightly improves validation MAE, the gains are marginal beyond 64 units. The best performance is achieved with 128 units, but the difference compared to 64 units is minimal ($< 0.0001$ MAE).

### 6.4 With and Without Calendar Features

In the final experiment, the impact of incorporating calendar-based features into the model input was evaluated. Up to this point, all architectures were trained using only lagged values of the target variable. Here, a multivariate setting was introduced by enriching the input with engineered features that encode temporal patterns known to influence traffic congestion.

The additional features include:

- **Hour of day**: encoded using sine and cosine transformations to capture daily periodicity.
- **Day of week**: encoded cyclically to represent weekly trends.
- **Month of year**: included to account for potential monthly seasonality.

All models were trained under the same conditions, using the previously selected single-layer LSTM with 128 units. The multivariate model receives, at each time step, the current value of the target variable and the six temporal features listed above.

| Model | Validation MAE (scaled) |
|---|---|
| LSTM | 0.01795 |
| LSTM + CF | **0.01776** |

The multivariate model exhibits improved performance on the validation set. While the reduction in MAE is relatively small, the inclusion of calendar features systematically benefits the forecasting accuracy. This outcome suggests that time-aware covariates offer valuable information that the model can leverage to capture seasonality and periodic behavior more effectively. As a result, the multivariate configuration is retained as the final model.

## 7 Results

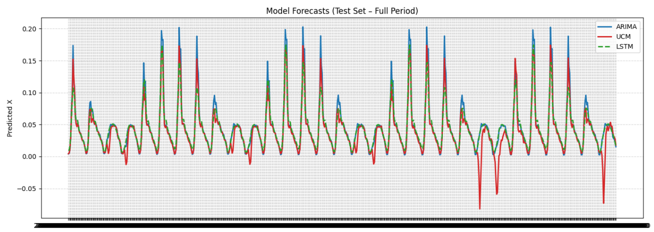
This section summarizes the forecasting accuracy of the selected models across the three families studied: SARIMAX, Unobserved Components Models (UCM), and Deep Learning architectures. For each family, the configuration with the best validation MAE was retained and compared using the same evaluation metric.

### 7.1 Validation MAE Comparison

The following table reports the validation MAE for each final model. The best performance is obtained by the per-hour SARIMAX with conditional differencing and weekly seasonality, followed closely by the per-hour UCM model enhanced with a holiday regressor. Deep learning models also achieve competitive results, with minor gains from the inclusion of calendar features.

| Family | Model | MAE |
|---|---|---|
| SARIMAX | SARIMAX $(1,0,1)(1,D,1)_7$ | **0.0120** |
| UCM | UCM + Hol. reg. | 0.0125 |
| DL | LSTM 1x128 + cal. feat. | 0.0178 |

In the next plot, it's possible to observe each family of model prediction on the test set.




# 8 Conclusions and future perspectives

This study presents a comprehensive comparison of three families of models—SARIMAX, Unobserved Components Models (UCM), and Deep Learning architectures—for the task of forecasting an hourly traffic congestion indicator. The dataset exhibits strong daily and weekly seasonal patterns, which were leveraged through model-specific strategies such as seasonal components, per-hour decompositions, and calendar features. The best overall performance was achieved by the per-hour SARIMAX model with conditional differencing and weekly seasonality (MAE = 0.0120). The UCM-based approach also demonstrated competitive accuracy (MAE = 0.0125) when enhanced with a holiday indicator, confirming the value of explicit calendar-based adjustments. Deep learning models, although not outperforming statistical methods, achieved solid results and showed improvement when enriched with temporal features.

**Limitations.** Despite the promising results, each modeling family presents limitations:

- **SARIMAX** models did not incorporate holiday effects or other exogenous variables, which may limit their capacity to fully adapt to calendar-driven variability. Including dummy variables—as done in the UCMs—could improve their flexibility.
- **UCM** experiments were limited to trend and weekly seasonal components. Further exploration of cyclical components, deterministic seasonality, or level shifts could enhance model interpretability and accuracy.
- **Deep Learning** models, while performant, require substantial tuning and are more data-hungry. They also lack native interpretability compared to traditional statistical models.

**Future perspectives.** Several avenues remain open for further development:

1. Perform more extensive hyperparameter searches, particularly for deep learning and SARIMAX seasonal configurations.
2. Explore ensemble strategies, such as averaging or weighted blending across model families, to combine their strengths.
3. Enrich the dataset with external covariates (e.g., weather, holidays, or local events) by studying the characteristics of the city where the data was collected.
4. Test more advanced deep learning architectures, such as attention-based models (Transformers), temporal convolutional networks (TCN), or hybrid models combining neural and statistical components.

Overall, the results confirm that combining seasonal modeling strategies, calendar-awareness, and hour-specific decomposition significantly improves forecasting accuracy for high-frequency traffic time series.