LONG-TERM MOBILE TRAFFIC FORECASTING USING DEEP SPATIO-TEMPORAL NEURAL NETWORKS

CIE 597

Dr.Kareem Abdullah

MAHMOUD THABET 202000328

MARWAN AHMED 202101214

MUHAMMAD ESSAM 202001411

MOHAMED ALAA 202100905

EYAD HANY 202101225

CONTENT OF TABLE


1.DATA
PREPROCESSING
5G

Fetching The Dataset

Challenge:

Fragmented raw data

- 62 daily “.txt” files, each covering one day of cell-level traffic, fetched from Harvard Dataverse

Heterogeneous file naming & content

- Timestamps in milliseconds, no column names, missing values

Verification complexity

- Ensuring no rows lost or duplicated when aggregating into larger files

Fetching The Dataset

Insights

ncremental processing

- Appending one day at a time into a weekly CSV keeps memory use bounded

Metadata-driven orchestration

- Dataverse API lets us dynamically list & pull the files we need.

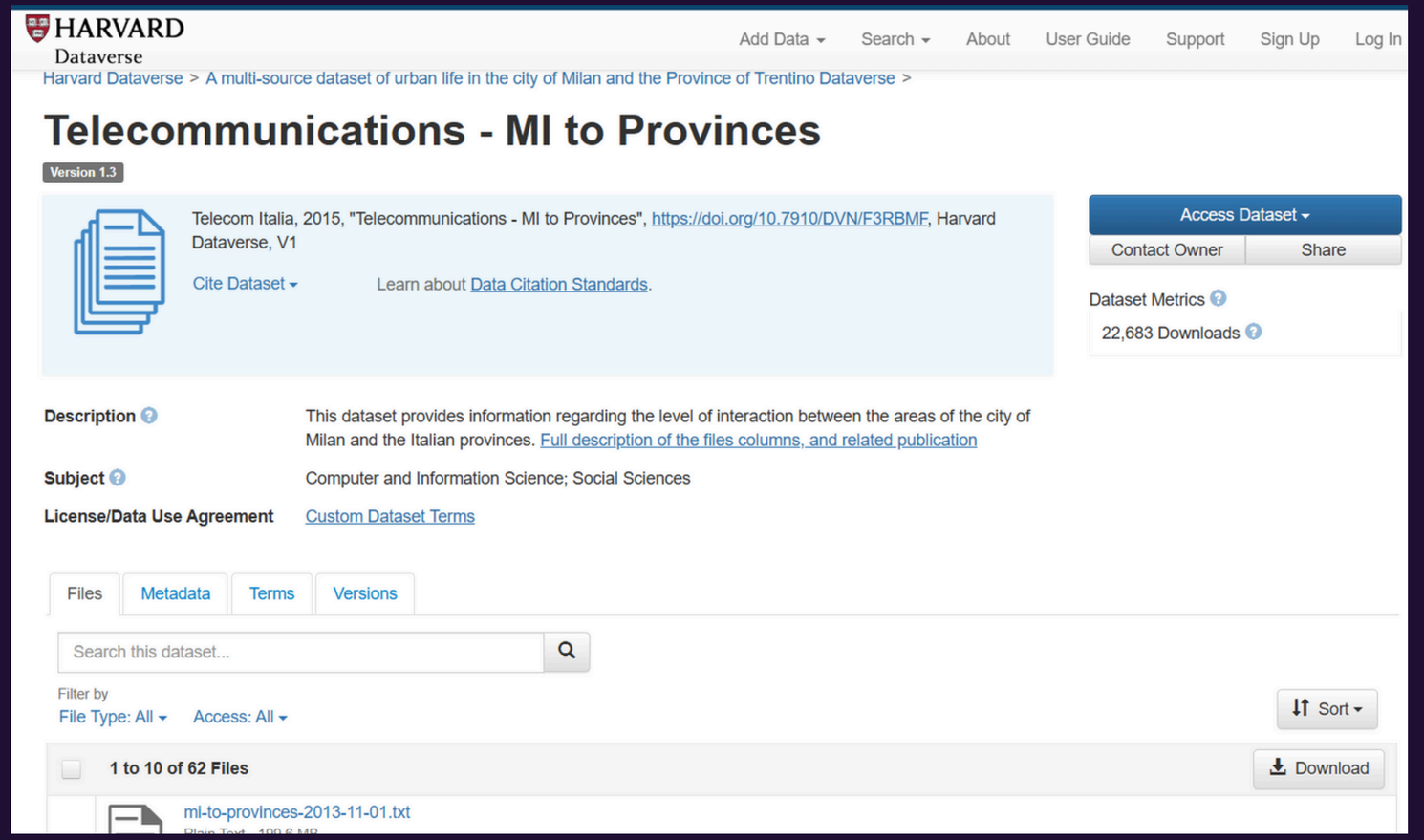
Row-count auditing

- Tracking raw line counts vs. CSV rows provides an end-to-end integrity check

Fetching The Dataset

Actions

1. Fetched the .txt files corresponding to each target week.
2. Read each day with pandas.read_csv() and append to week_{n}.csv, writing headers only once.
3. Tracked raw line counts per file, then compare against CSV row totals to verify integrity.



2.DATA OVERVIEW & EDA

OBJECTIVES

- UNDERSTAND DATASET STRUCTURE, SIZE, AND QUALITY
- UNCOVER PATTERNS, DISTRIBUTIONS, AND RELATIONSHIPS
- IDENTIFY AND HANDLE ANOMALOUS (OUTLIER) VALUES FOR CLEANER MODELING

Data Overview & EDA

- **Challenge: Multiple activity columns (sms-in, sms-out, call-in, call-out, internet) recorded unevenly, with many NaNs. In addition to different datapoints/timestamp.**
- **Insight**: Without a single volume metric, comparisons across time and cells are muddled.
- **Solution**:

1. Compute a total traffic volume (tot_vol) by summing all five activity columns into one consolidated metric.
2. Aggregate the datapoints that having the same timestamp

	CellID	datetime	countrycode	smsin	smsout	callin	callout	internet
0	1	138326040000	0	0.081363	NaN	NaN	NaN	NaN
1	1	138326040000	39	0.141864	0.156787	0.160938	0.052275	11.028366
2	1	138326100000	0	0.136588	NaN	NaN	0.0273	NaN
3	1	138326100000	33	NaN	NaN	NaN	NaN	0.026137
4	1	138326100000	39	0.278452	0.119926	0.188777	0.133637	11.100963

Data Overview & EDA

- **Challenge: Time stamps are irregular—some intervals missing entirely for many CellIDs.**
- **Insight**: A proper time series model demands a uniform 10-minute grid so gaps don’t bias trends.
- **Solution**:
    1. Define the complete time range at 10-minute steps over the dataset’s span.
    2. Build the Cartesian product of that time index × all CellIDs to force every possible slot to exist.

	CellID	datetime	countrycode	smsin	smsout	callin	callout	internet
0	1	138326040000	0	0.081363	NaN	NaN	NaN	NaN
1	1	138326040000	39	0.141864	0.156787	0.160938	0.052275	11.028366
2	1	138326100000	0	0.136588	NaN	NaN	0.0273	NaN
3	1	138326100000	33	NaN	NaN	NaN	NaN	0.026137
4	1	138326100000	39	0.278452	0.119926	0.188777	0.133637	11.100963

Data Overview & EDA

- **Challenge: Even after gridding, many (cell, time) points remain empty (NaN), breaking continuity.**
- **Insight**: Missing values interrupt pattern detection and degrade model inputs.
- **Solution**:
    1. (Again) Group by CellID & timestamp to aggregate our tot_vol.
    2. (Again) Reindex onto the full grid, which naturally inserts NaNs where data was absent.
    3. Apply linear interpolation per CellID to fill those gaps—yielding a smooth, complete 10-minute time series.

	CellID	datetime	datetime	datetime	datetime	datetime	datetime	datetime
0	1	138326040000	0	0.081363	NaN	NaN	NaN	NaN
1	1	138326040000	39	0.141864	0.156787	0.160938	0.052275	11.028366
2	1	138326100000	0	0.136588	NaN	NaN	0.0273	NaN
3	1	138326100000	33	NaN	NaN	NaN	NaN	0.026137
4	1	138326100000	39	0.278452	0.119926	0.188777	0.133637	11.100963

Week_1 data (37,622,898 * 8)

Data Overview & EDA

Cleaned Data (so far):

- Fully continuous time series at 10-minute intervals for every CellID.
- One row per (CellID, Timestamp) with **no gaps or missing slots.**
- Single metric tot_vol summing all traffic types (SMS, calls, internet).
- Zero NaNs after linear interpolation—clean and gap-free.
- Uniform structure (CellID, Timestamp, tot_vol) across entire period (9 weeks).

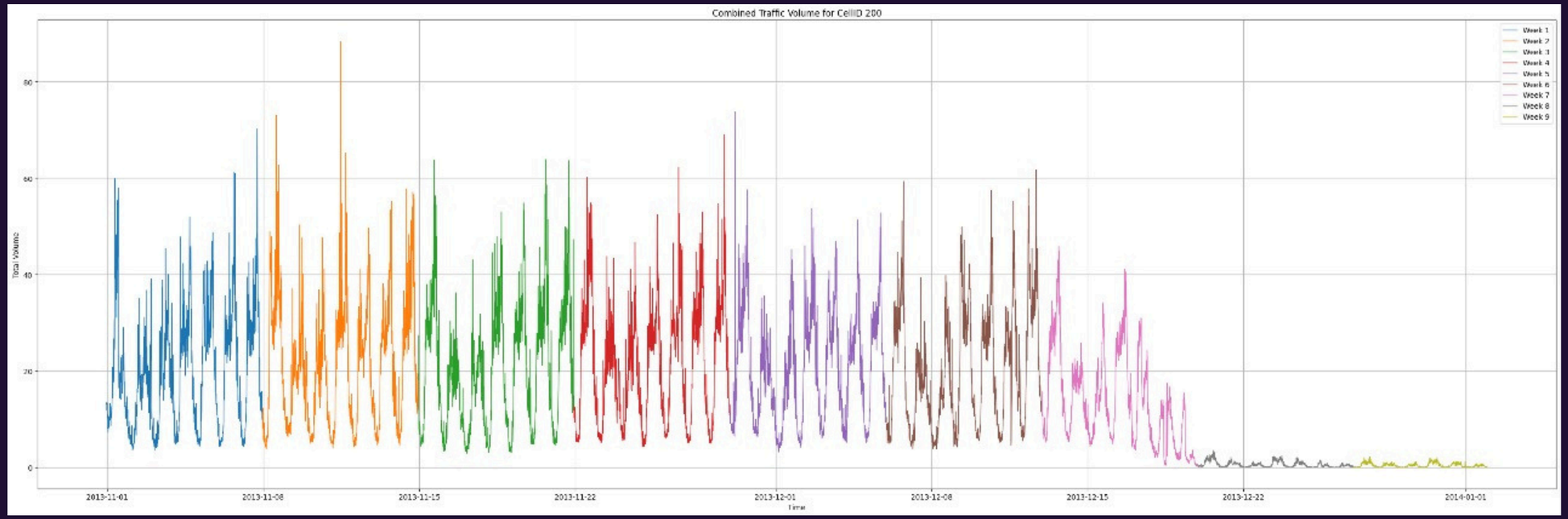
	CellID	time	tot_vol
	1	2013-12-05 23:00:0(	10.383293
	1	2013-12-05 23:10:0(	8.905027
	1	2013-12-05 23:20:0(	8.985766
	1	2013-12-05 23:30:0(	9.459189
	1	2013-12-05 23:40:0(	10.788013
...	...	...	...
	10000	2013-12-12 22:10:00	21.96578
	10000	2013-12-12 22:20:00	17.47666
	10000	2013-12-12 22:30:00	15.648683
	10000	2013-12-12 22:40:00	16.902803
	10000	2013-12-12 22:50:00	19.973341

Week_1 after cleaning (10,080,000 * 3)

Data Overview & EDA

Exploring a Random Cell Total Volume (CellID=200):

- The last 3 weeks has different behavior than the first 6 due to Christmas time.
- In this cell Christmas times results in reduction in cell total traffic volume; maybe, people travel to a different city in Christmas.
- We can see some little outlier traffic volumes at the same time across the weeks.



Cell 200 total volume (all weeks)

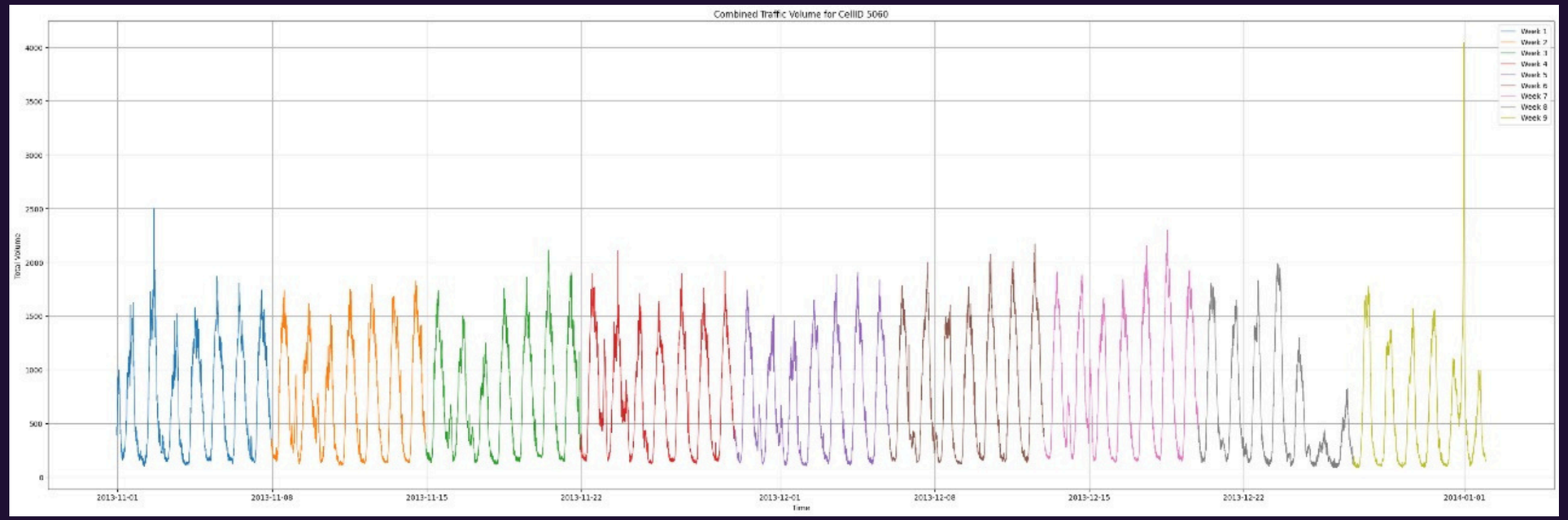
Data Overview & EDA

Exploring Cells Total Volume:

- The same pattern repeats almost for all cell except for some exceptional cases.

Decisions:

- We decided to drop the last 3 weeks as they have different pattern of traffic.
- Drop the outliers for robustness of the model.

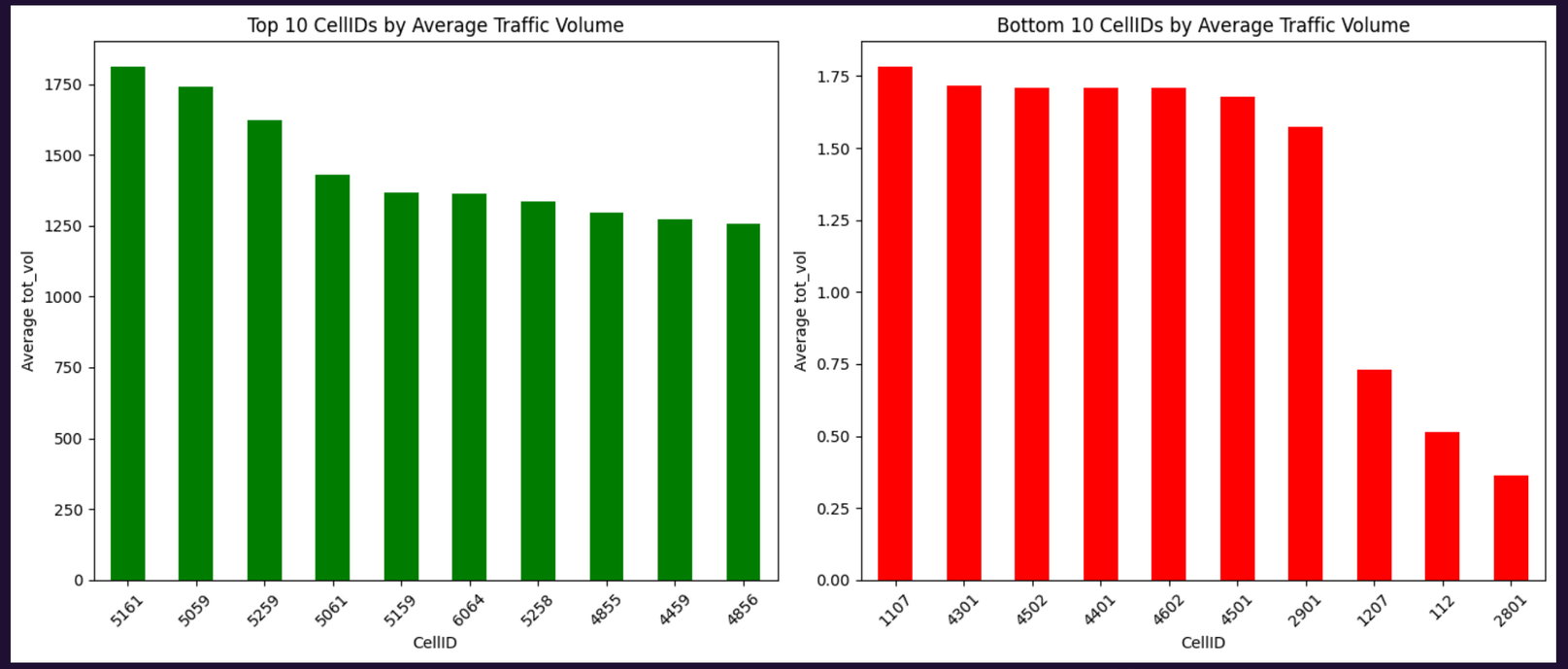


Duomo, the city centre of Milan (Square id: 5060)

Data Overview & EDA

Exploring Cells Total Volume Average:

- The top 10 Cells’ average traffic volume ranges from 4,856 to 5,161.
- The bttom 10 Cells’ average traffic volume ranges from 2,801 to 1,107.



Top and bottom 10 Cells by average traffic volume.

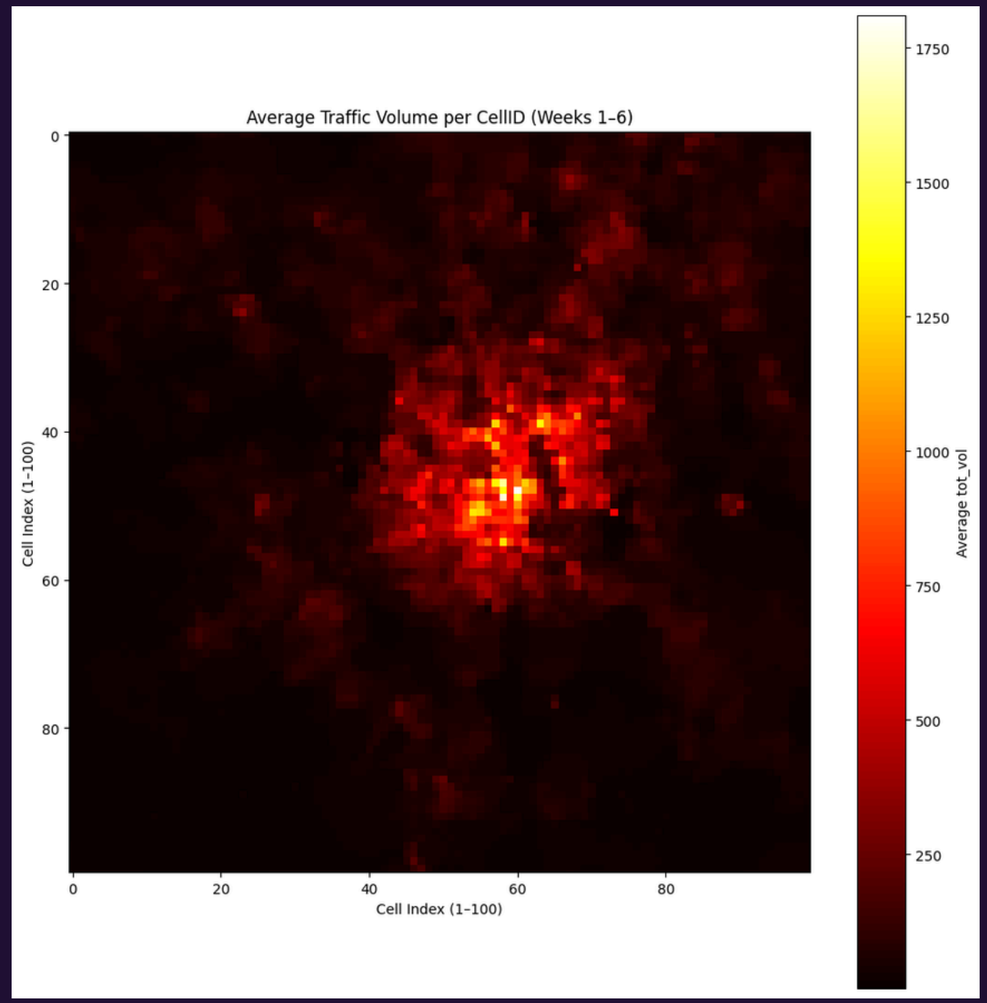
Data Overview & EDA

Average Total Volume per Cell

The average total volume per cell across all weeks shows that the center of Milano has the highest average traffic volume.

Decisions:

- We decided to model the traffic volume for the center of Milano due to its high traffic volume .



Heatmap of average total volume per cell across all weeks.

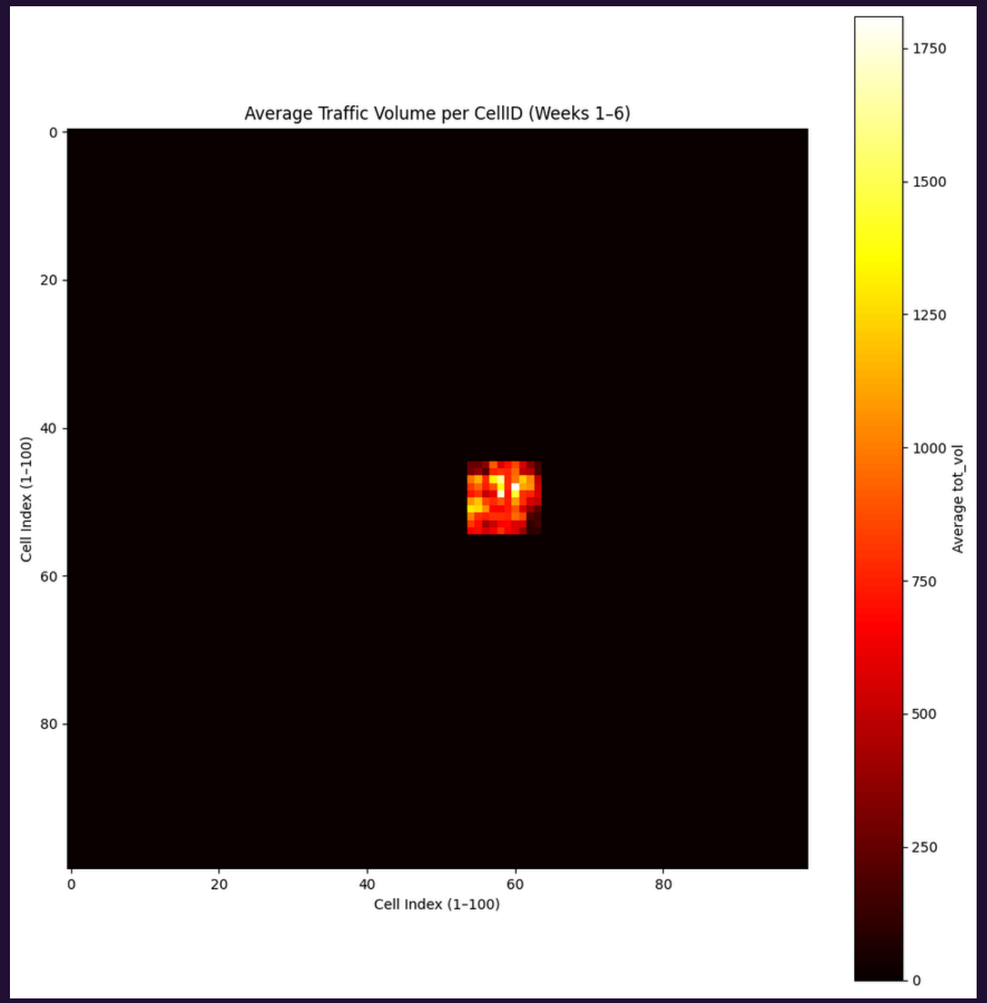
Data Overview & EDA

Average Total Volume per Cell

The average total volume per cell across all weeks shows that the center of Milano has the highest average traffic volume.

Decisions:

- We decided to model the traffic volume for the center of Milano due to its high traffic volume .



Heatmap of average total volume per cell across all weeks (Center of Milano).

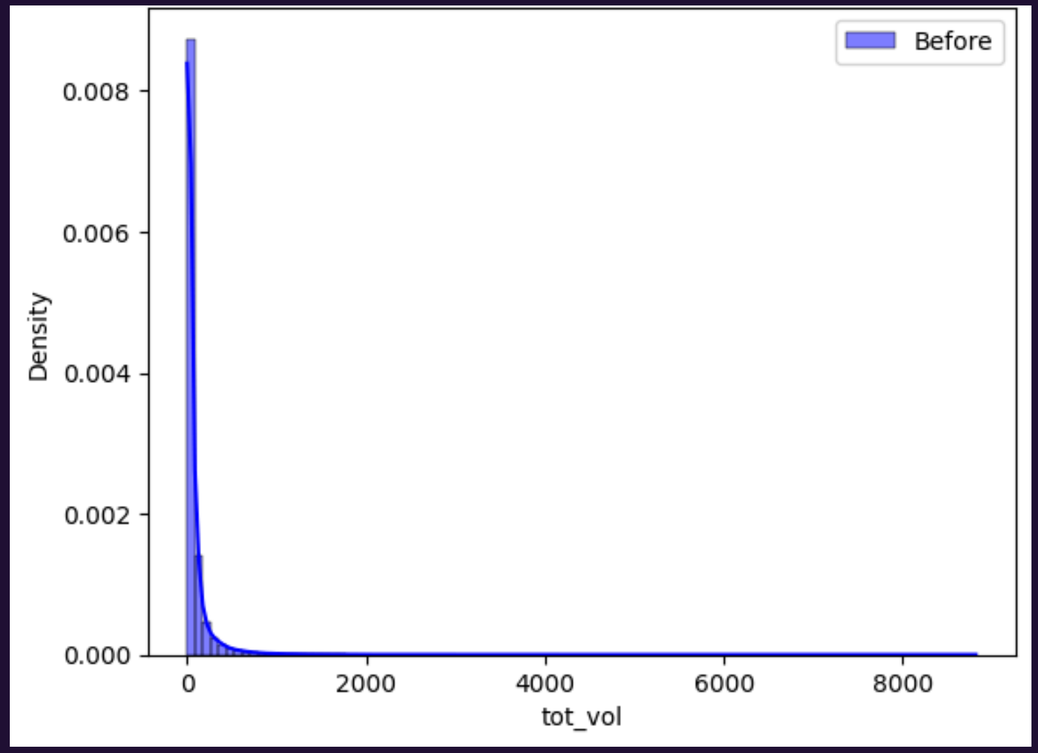
Data Overview & EDA

Data Distribution

The total volume is highly skewed to the right having the the low total traffic volume most frequent ones (across all selected cells and weeks).

Decisions:

- We decided to use IQR approach for outlier removal.

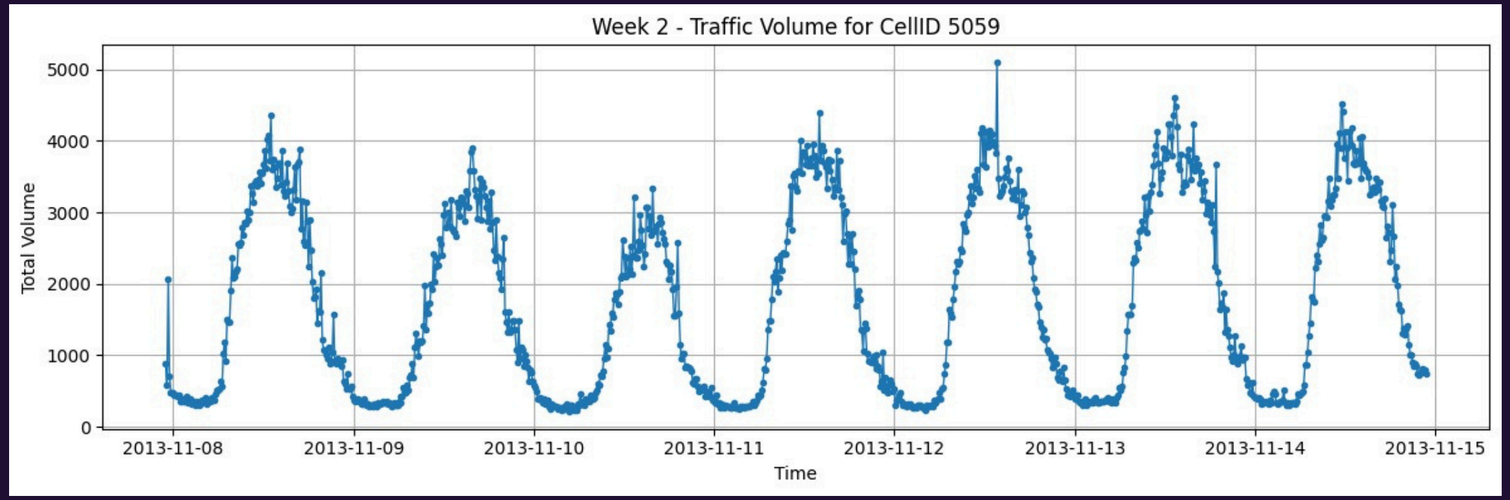


Total volume (all weeks) distribution before outlier removal.

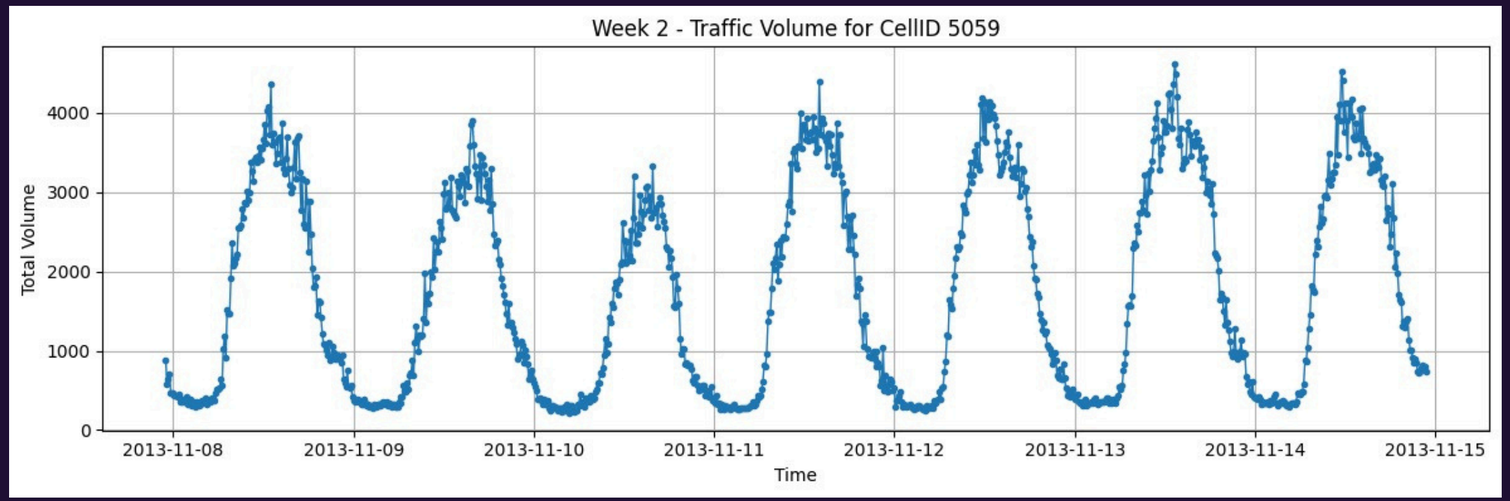
Data Overview & EDA

Outlier Removal

Cell 5059 total volume
(week 2) before outlier removal.



Cell 5059 total volume
(week 2) after outlier removal.

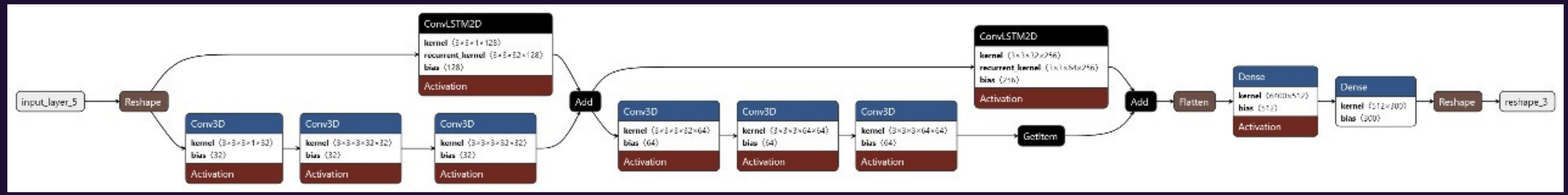



3. MODEL
5G
75,000
60,000
45,000
30,000
15,000
0
Jan
Feb
Mar
Sep
Aug

Proposed STN Architecture

- Uses an encoder-Decoder Arcitecture with two branches in the encoder:
  - convLstm for long-term-spatio-temporal features.
  - 3D Convolutional Network for short term patterns.

- Output from both branches are then fused.
- A multi-layer perceptron(MLP) decodes the fused features into the final traffic prediction.



Proposed STN Architecture

Proposed model overview

Bench marked against:

- **A**RIMA,LTSM,ConvLSTM,GRU
- Metrics:MAE and RMSE over hourly horizons
- STN performs better in baselines in both short and mid term
- D-STN outperforms in long term due the integrated traffic periodicity

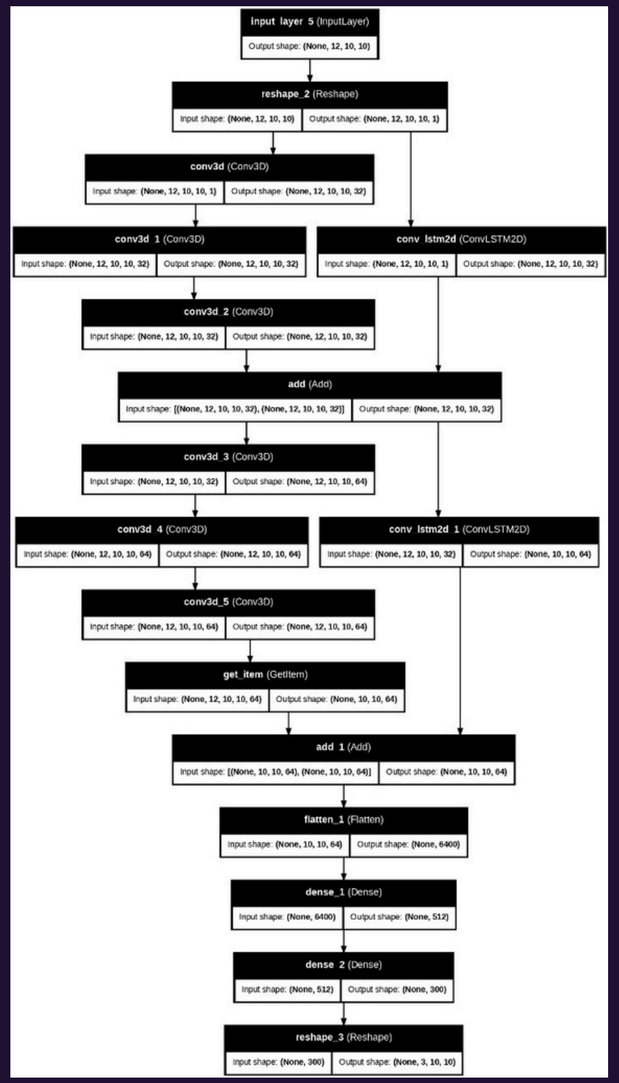
Proposed STN Architecture

1- ConvLSTM ENCODER

- Applies convolution inside LSTM units to retain spatial locality.
- Handles long sequences while modeling the spatial dependencies.
- stacked convlstm layers deepen the temporal modeling capacity.

2- 3d-convnet ENCODER

- learns local spatio-temporal interactions over short time intervals
- treats the input tensor as a volume and applies 3d convolutions
- its behavioer complements the convltsm; by focusing on recent and more rapid changing patterns.



Proposed STN Architecture

Proposed STN Architecture

3-Feature Fusion and Decoder

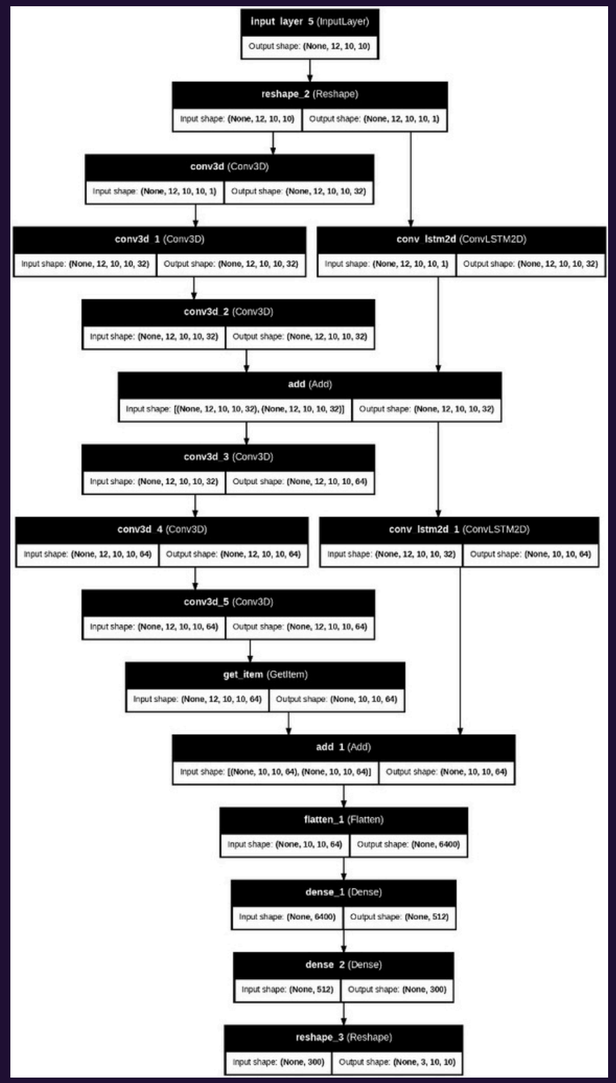
- Fusion is done through element-wise addition of the encoded features
- Fused Features from all the timestpes (‘s’) are flattend and passed to the fully connected MLP
- MLP acts as a decoder as it regresses to a scalar traffic value
- stacked convlstm layers deepen the temporal modeling capacity

4-Hyperparameters

- **Obj Fucntion: Mean Squared Error(MSE):**

$$L(\Theta) = \frac{1}{T \cdot X \cdot Y} \sum_{t=1}^{T} \sum_{x=1}^{X} \sum_{y=1}^{Y} \| M(\Theta; F_t^{(x,y)}) - d_t^{(x,y)} \|^2.$$

- Learning rate=0.005(initial)
- Optimizer=Adam
- Traffic Matrices is to be normalized during the Preprocessing



Proposed STN Architecture

Results

- Accurate Zone Identification:

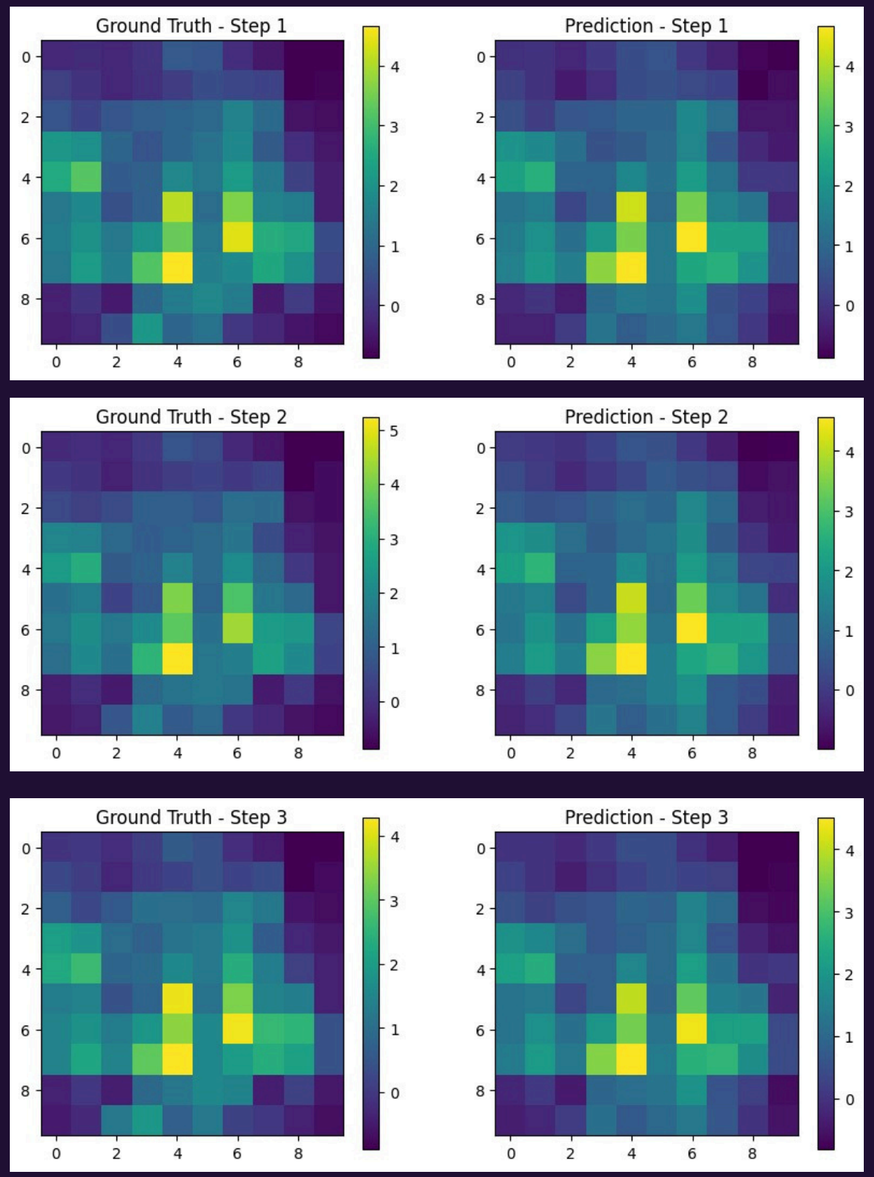
Pinpoints high-traffic areas that closely match observed data.

- Coherent Multi-Step Forecasts:

Maintains realistic spatial patterns over several future time steps.

- Stable Intensity Scaling:

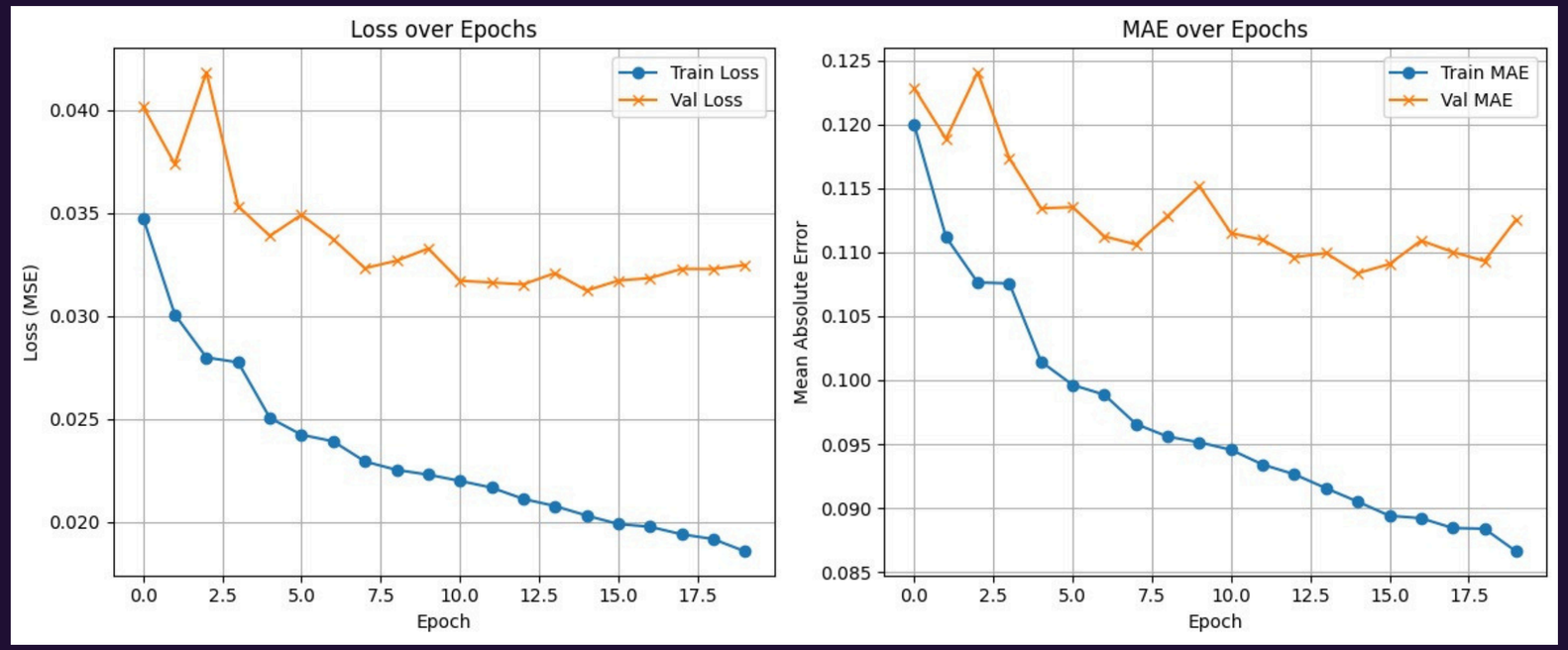
Produces traffic intensity values that stay within expected numerical ranges and align with real measurements.



10*10 (chosen) Ground Truth vs Forecasting..

Results

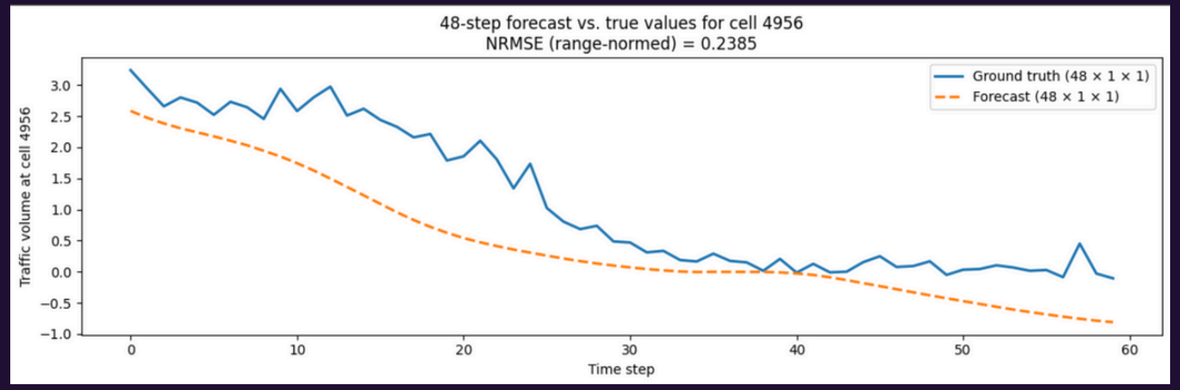
- Both Mean Squared Error (MSE) and Mean Absolute Error (MAE) decrease steadily, indicating effective learning.
- The rate of improvement slows around epoch 18, suggesting the model is approaching convergence.
- However, in validation, metrics are higher than training metrics, which is expected.

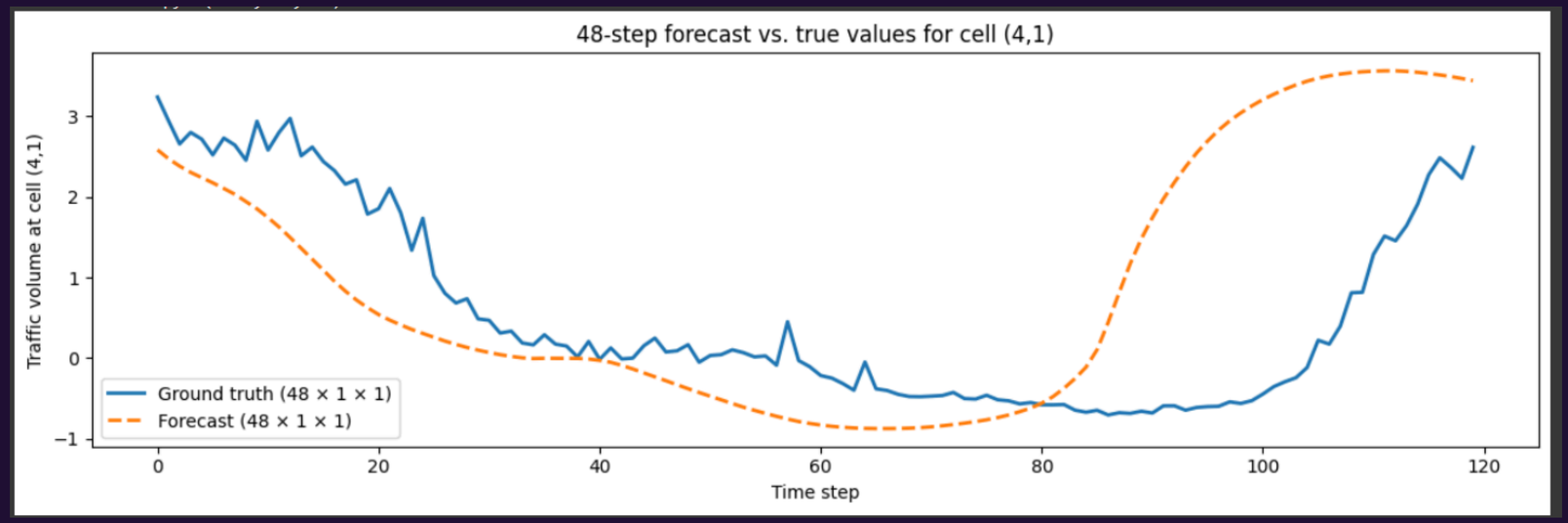


Loss and MAE accros different epochs.

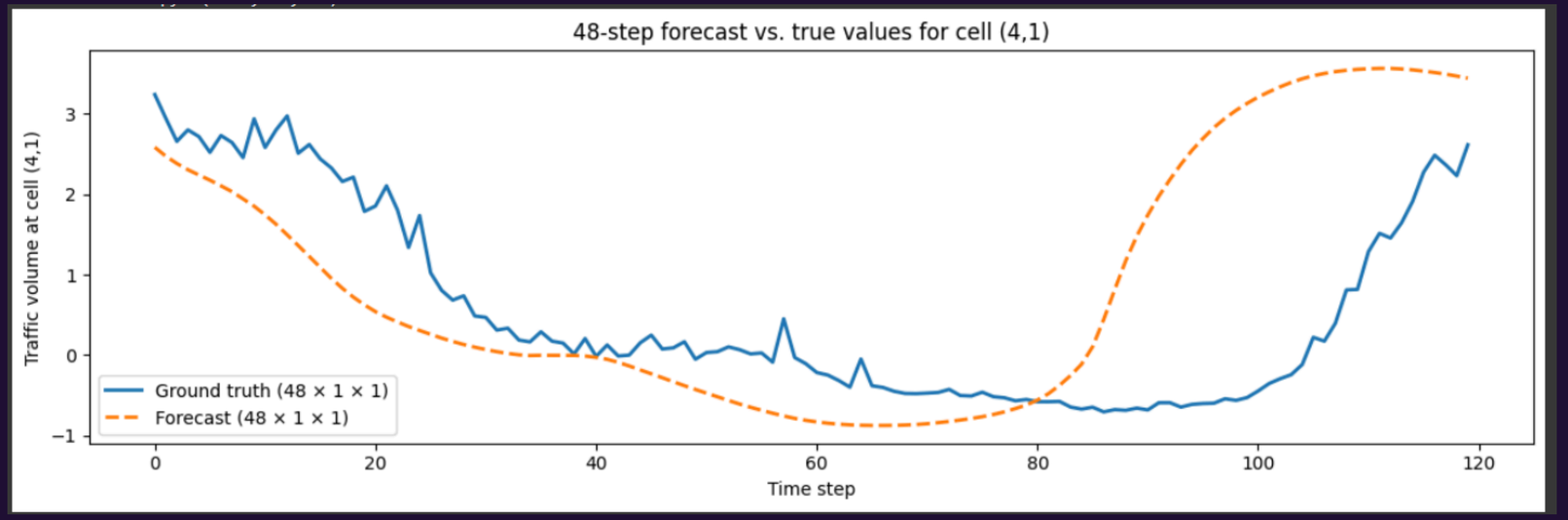
Results

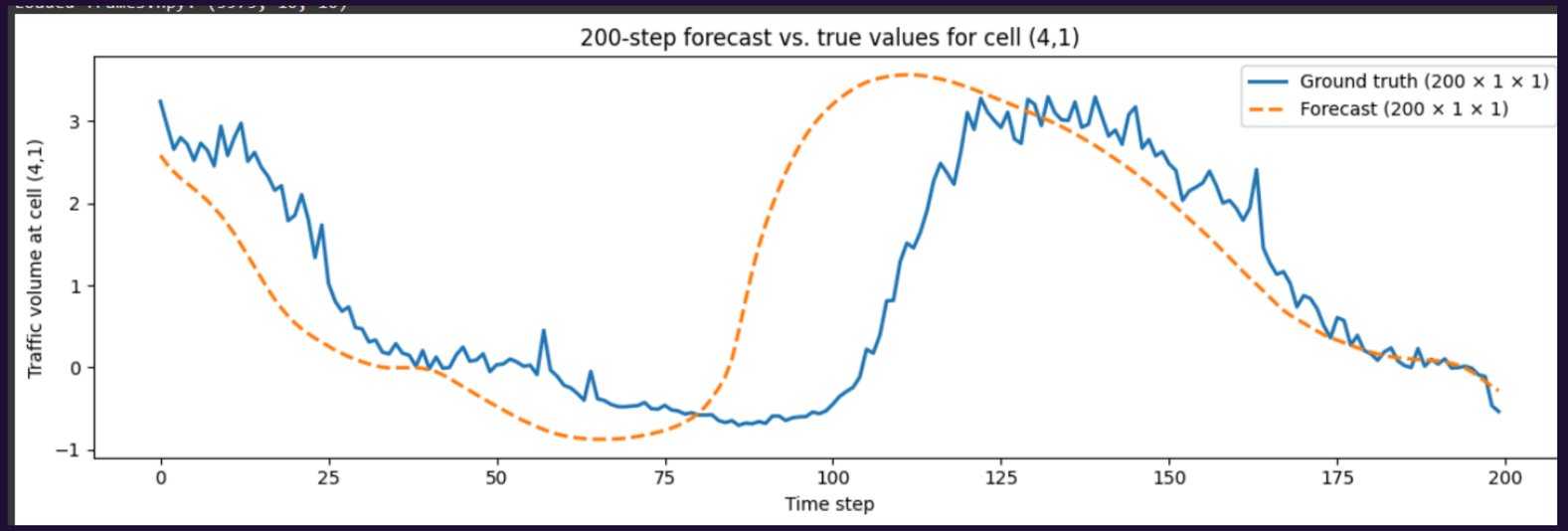
- the forecast of DSTN of the cell of 4956 of size of 60 timestamps (10 hr)
- with NRMSE = 0.2385





Results


48-step forecast vs. true values for cell (4,1)
Traffic volume at cell (4,1)
Time step
Ground truth (48 × 1 × 1)
Forecast (48 × 1 × 1)
3
2
1
0
−1
0
20
40
60
80
100
120


200-step forecast vs. true values for cell (4,1)
Traffic volume at cell (4,1)
Time step
Ground truth (200 × 1 × 1)
Forecast (200 × 1 × 1)
3
2
1
0
−1
0
25
50
75
100
125
150
175
200

ARIMA-Based Models

1. AUTO-ARIMA

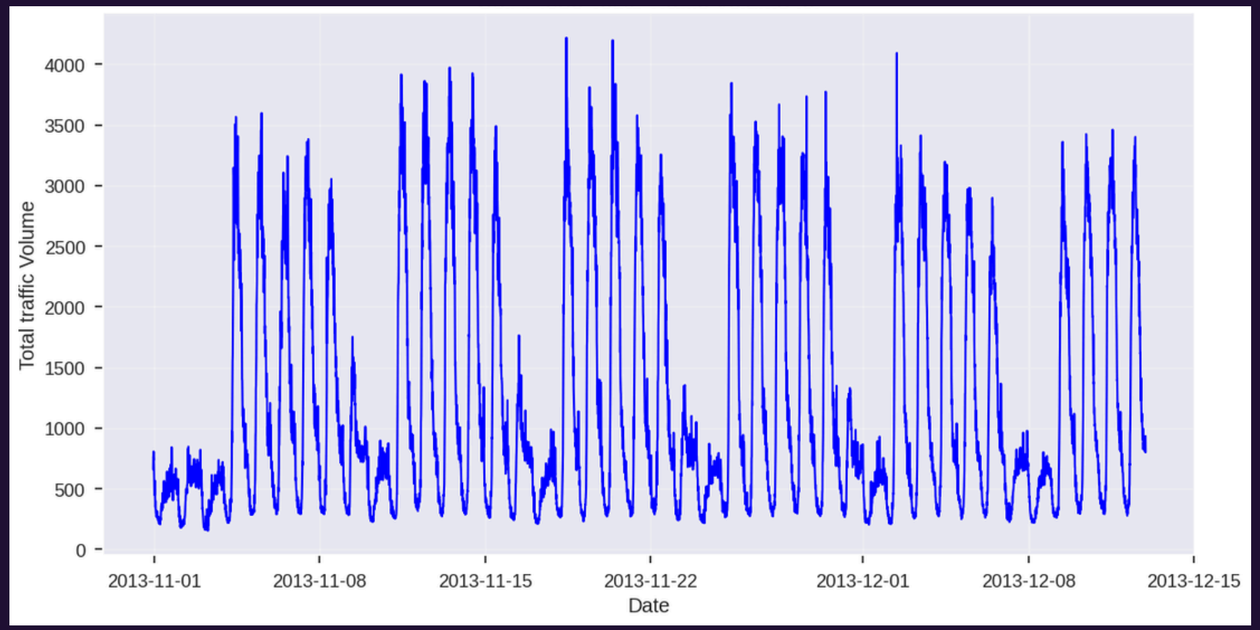
Cell #4956

Observations

The time series appears seasonal with:

- Primary seasonality: Daily
- Secondary pattern: Weekly layered on top of the daily pattern.

Drop Periods: correspond to weekends (consistent low traffic every 7 days).



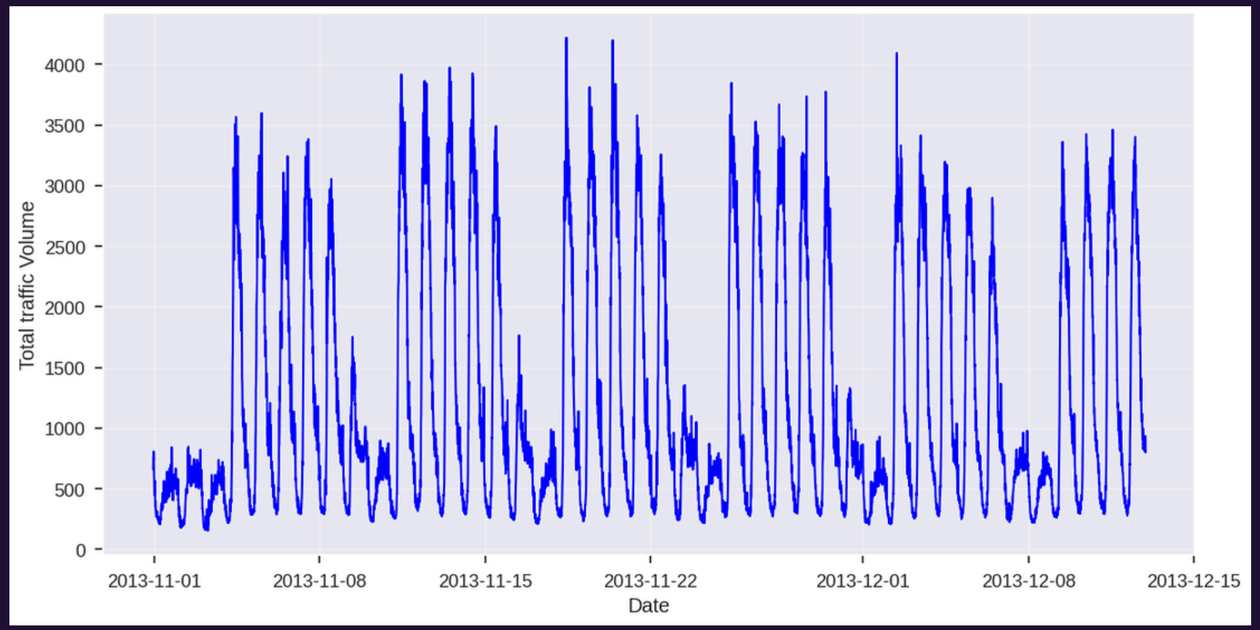
Cell #4956

Confirm Seasonality Statistically

1. We use STL with period 24 as:
    - roubust against outliers.
    - handle non-stationary seasonality.

2. ACF to confrirm the 24-hrs lag cycle.

3. Kruskal-Wallis to statistically confirm traffic at different times (hours of the day) is significantly different.



Daily Seasonality with STL

- Strong daily patterns.

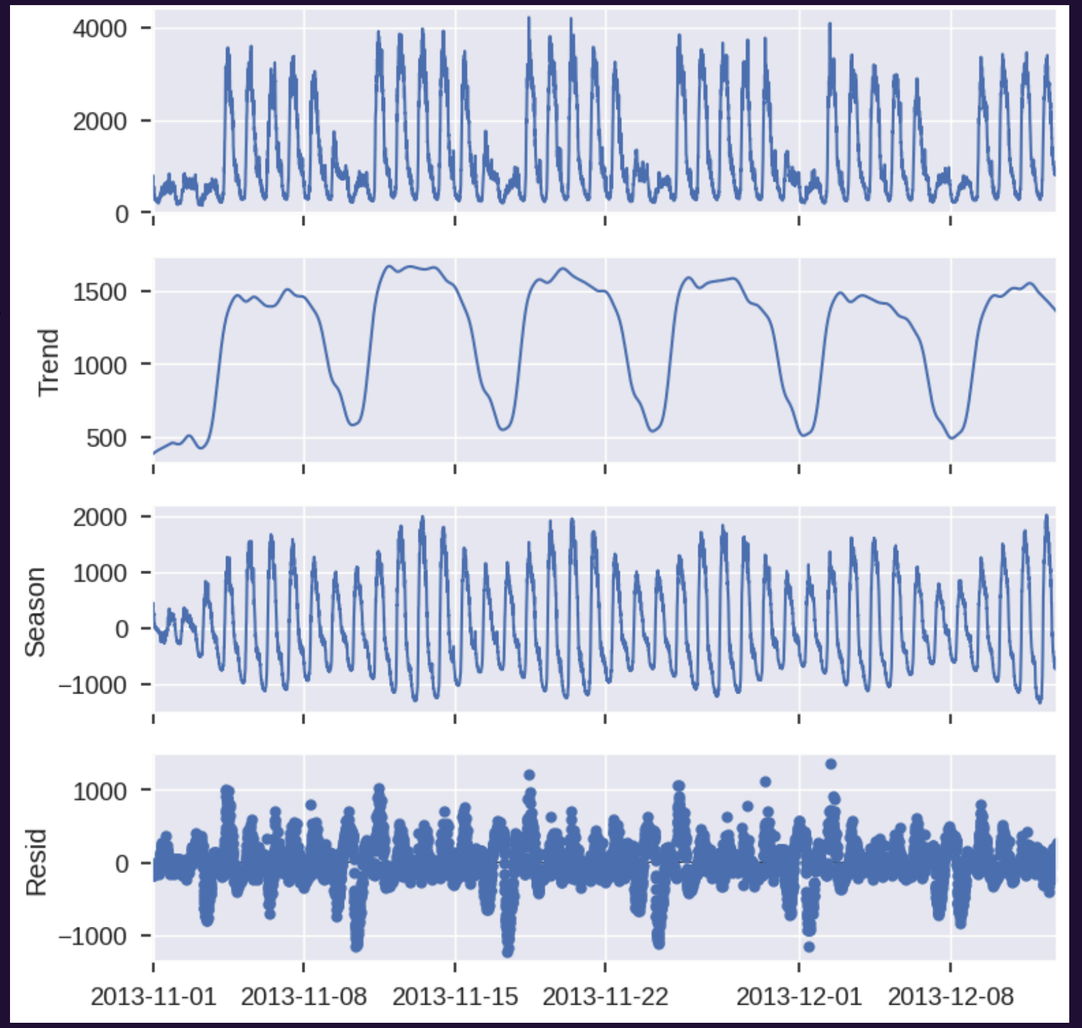
Trend:

- shows weekly-like dips and recoveries:
- reflecting weekend effects (lower traffic on weekends).
- separating low-frequency trend variation from high-frequency daily cycles.

Seasonality:

- captures the core daily shape of the signal:
  - High peaks during active hours.
  - Lows during night/inactive hours.
  - The amplitude of the seasonality seems mostly stable, with slight variation.

shows a very strong seasonality of 89%.



Weekly Seasonality with STL

- Strong repeating cycles, confirming regular daily behavior.

Trend:

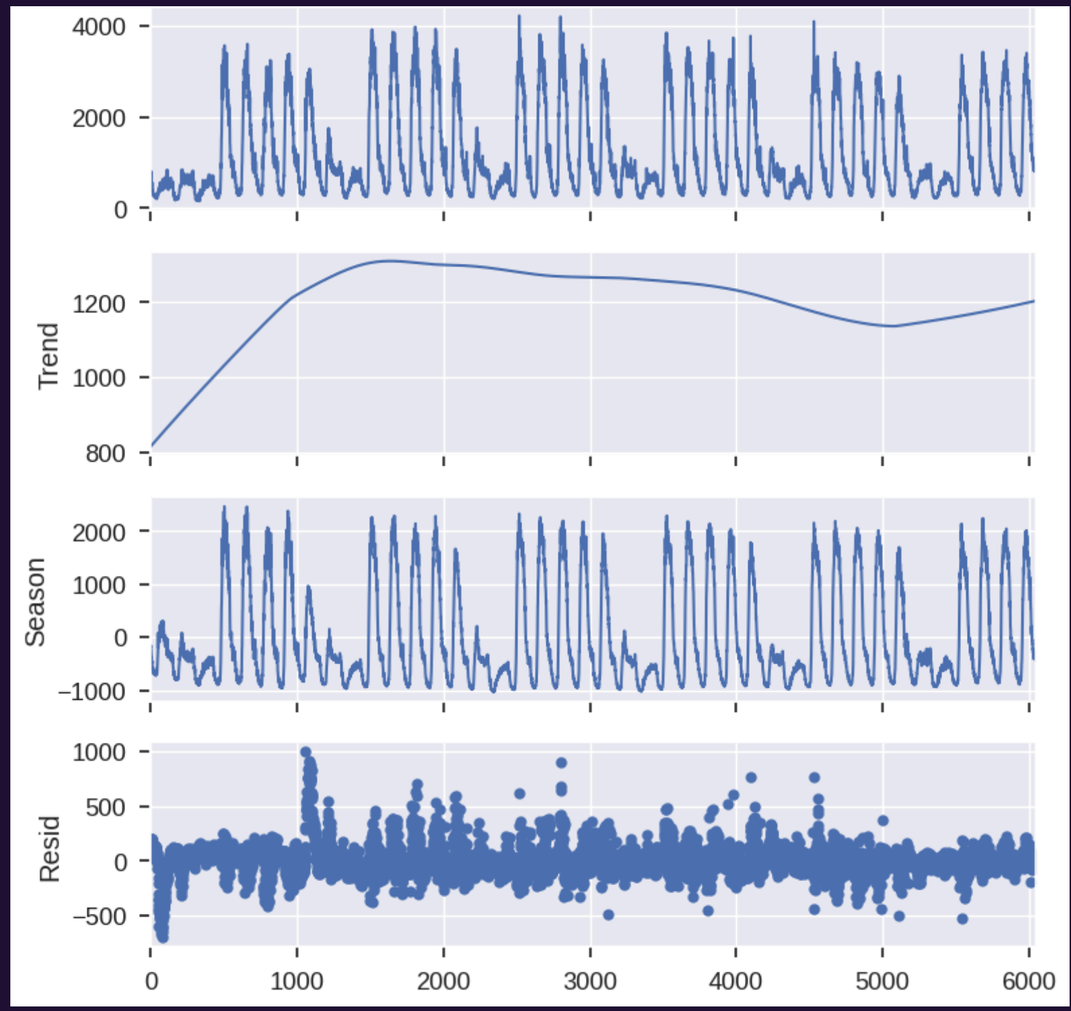
- The trend component is much smoother now, since daily variations were absorbed by the seasonal component.
- Captures long-term drift.

Seasonality:

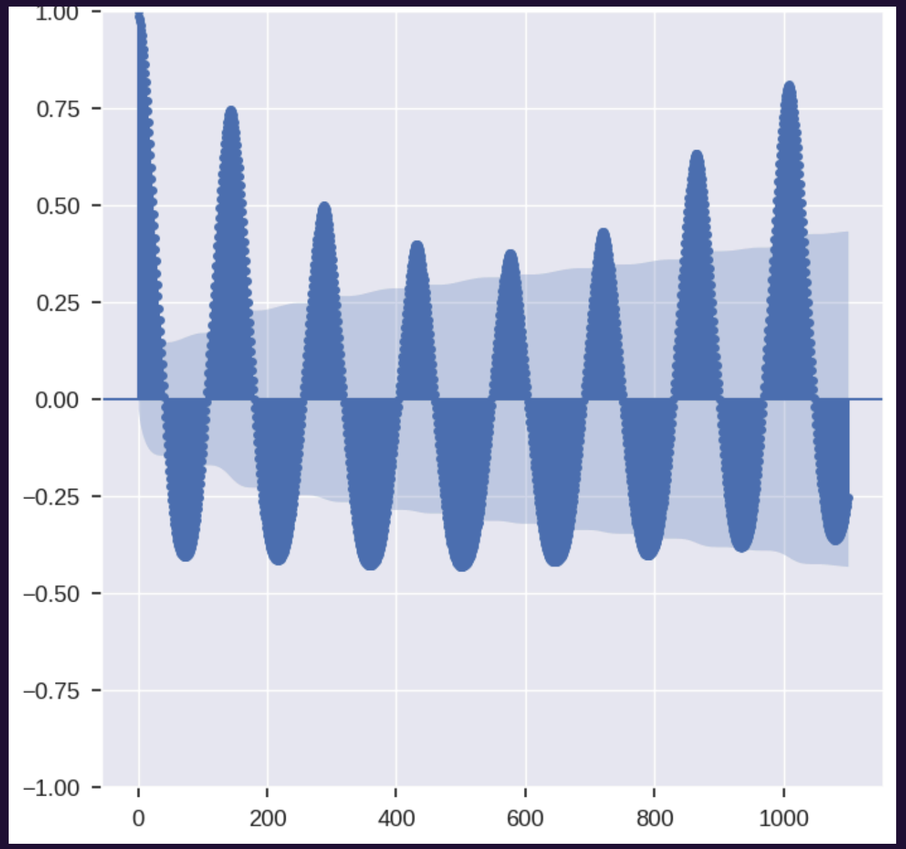
- captures weekly periodicity:

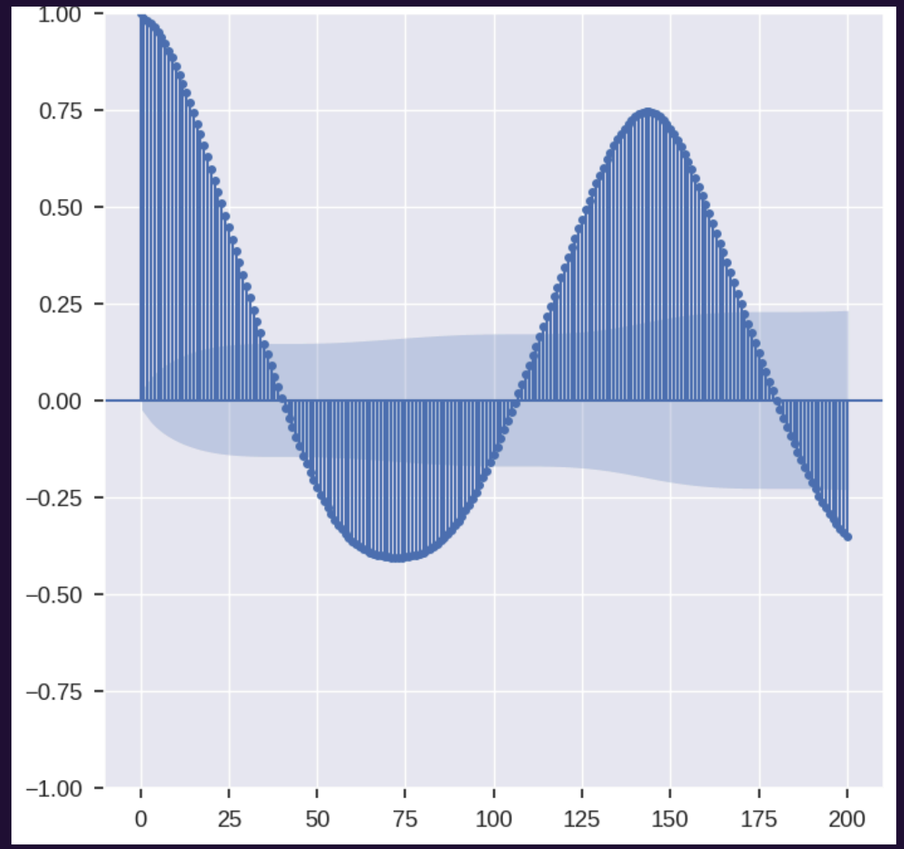
- strong weekly seasonal structure, on top of daily cycles.
- daily seasonality has been absorbed into the residuals.

shows a very strong seasonality of 98%.



ACF





- First dip at lag 72 (~144/2): inverse correlation with data half a day apart.
- Second strong positive peak at lag 144: confirms daily seasonality.
- The repeating cosine-like pattern indicates regular, stable cycles.
- Strong, periodic spikes in autocorrelation: At lags: 0, 144, 288, 432, ..., up to 1008
- The pattern repeats cleanly, with very high autocorrelation near lag 1008.

Kruskal-Wallis Test for Seasonality

- To check if the distribution of values differs significantly between across hours of the day.

Kruskal-Wallis statistic (Daily): 4148.50, p-value: 0.0000e+00
Kruskal-Wallis statistic (Weekly): 909.11, p-value: 4.0276e-193

- There are significant differences in total traffic volume across different times of day.
- $p < 0.05$: we reject the null hypothesis that all time intervals have the same distribution.
- Combined with STL and ACF, this strongly confirms daily seasonality from both:
  - Statistical perspective.
  - Visual perspective.

PACF

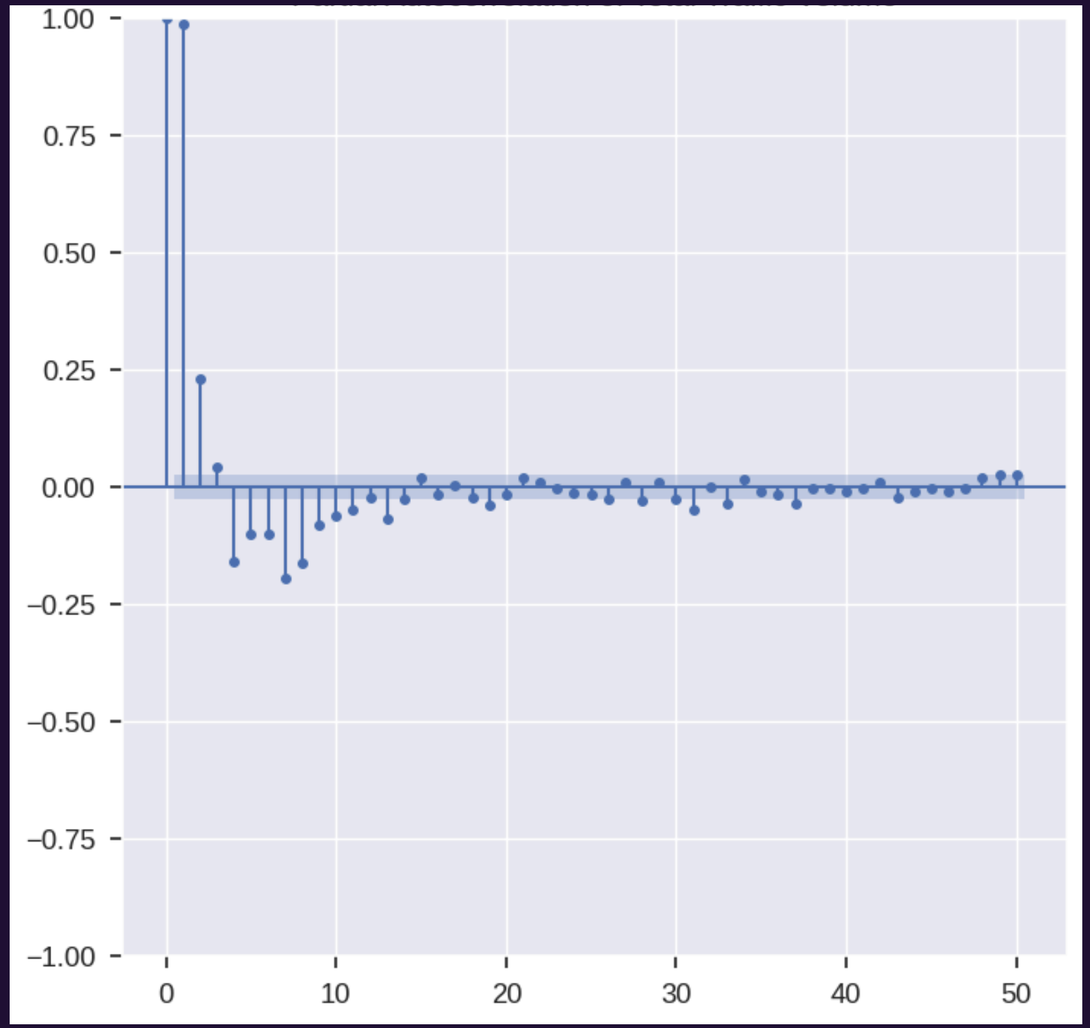
- Lag 1 and 2: Strong and significant spikes → high partial autocorrelation

- Lag 3: Still slightly above confidence band

- Beyond lag 4: Rapid decay, values mostly within confidence bounds

Interpretation:

- There's significant short-term dependency up to lag 2-3
- AR component p = 2 or 3.



Stationary Test

ADF Test for Non-Stationary Series:

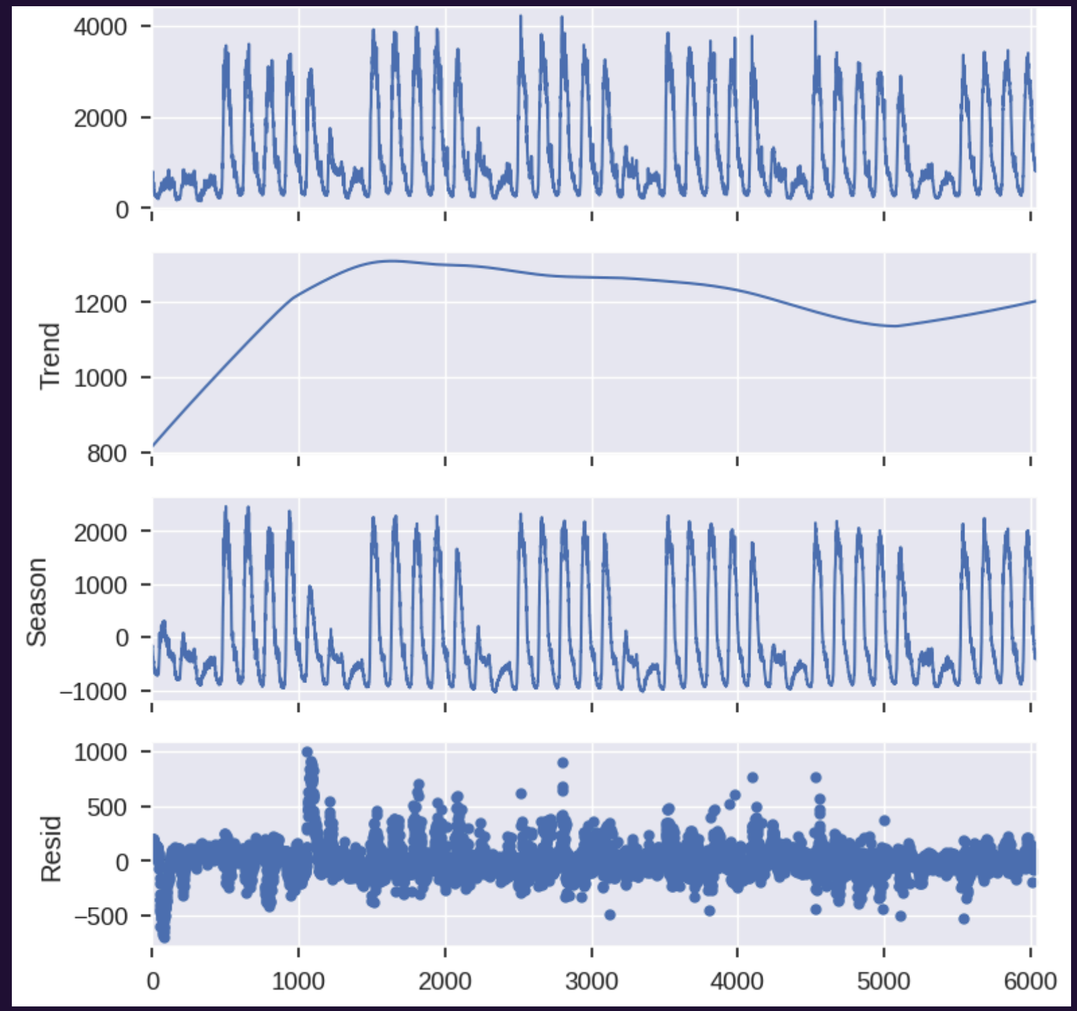
- ADF Statistic: -12.3021
- p-value: 0.0000

Series is stationary (reject null hypothesis)

- series is likely trend-stationary with deterministic seasonal cycles.

- To further confirm it, we apply stationarity test after removing daily seasonality.

ADF on residuals: stat=-13.32, p=6.4745e-25

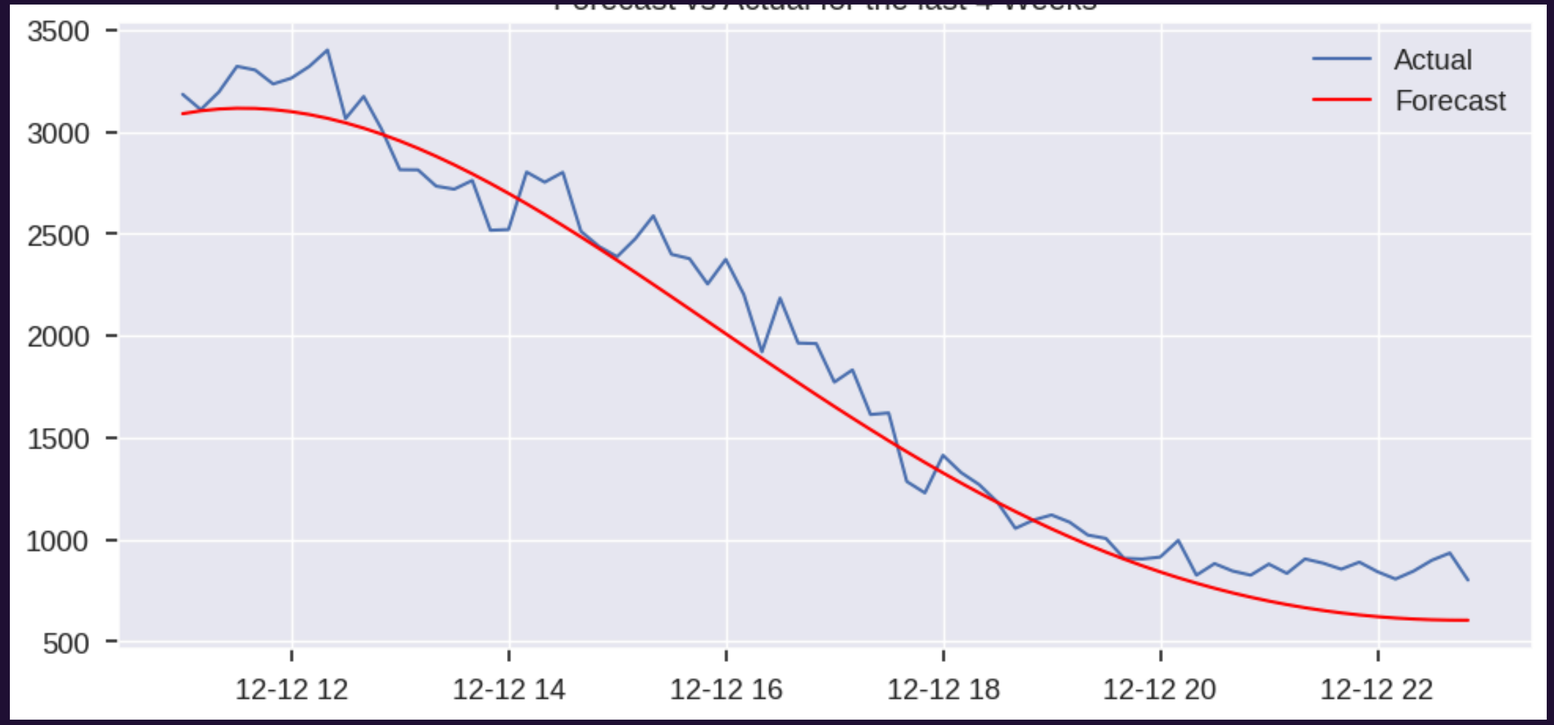


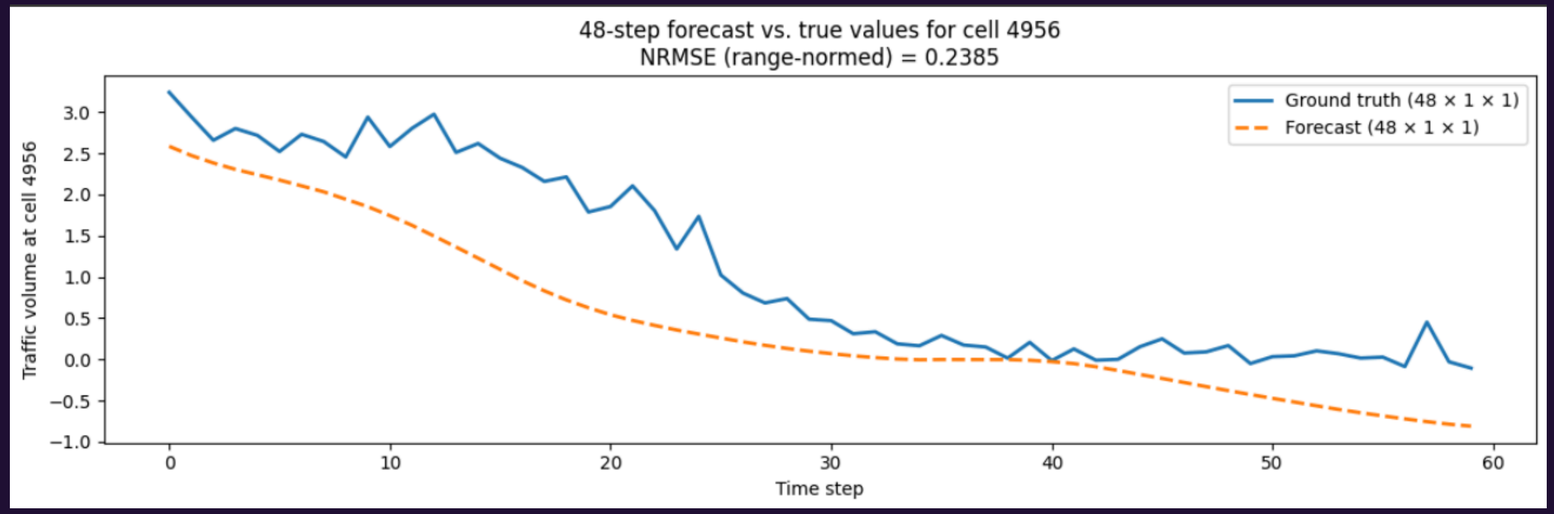
Auto ARIMA

- Test size: 12 hours.
- ARIMA with grid search to choose the best parameters.
  - Best ARIMA order (p,d,q): (2, 0, 2)
  - AIC: 76353.81

- NRMSE: 0.0926

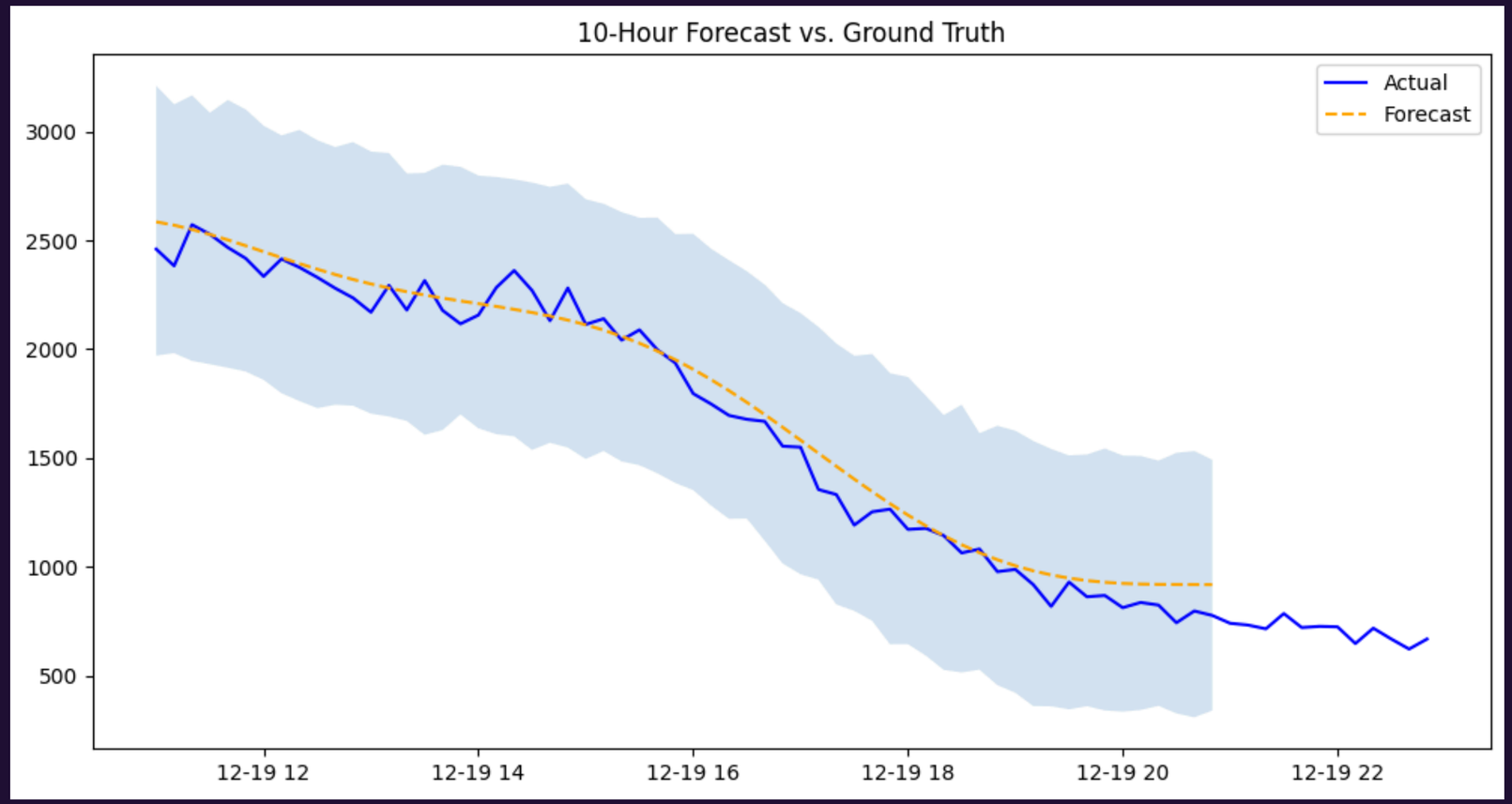
In Comparison to DSTN
Test size: 10 hours.
NMRSE =0.2385.





baseline forecast

MAE: 73.82
MAPE: 5.36%
RMSE: 80.20

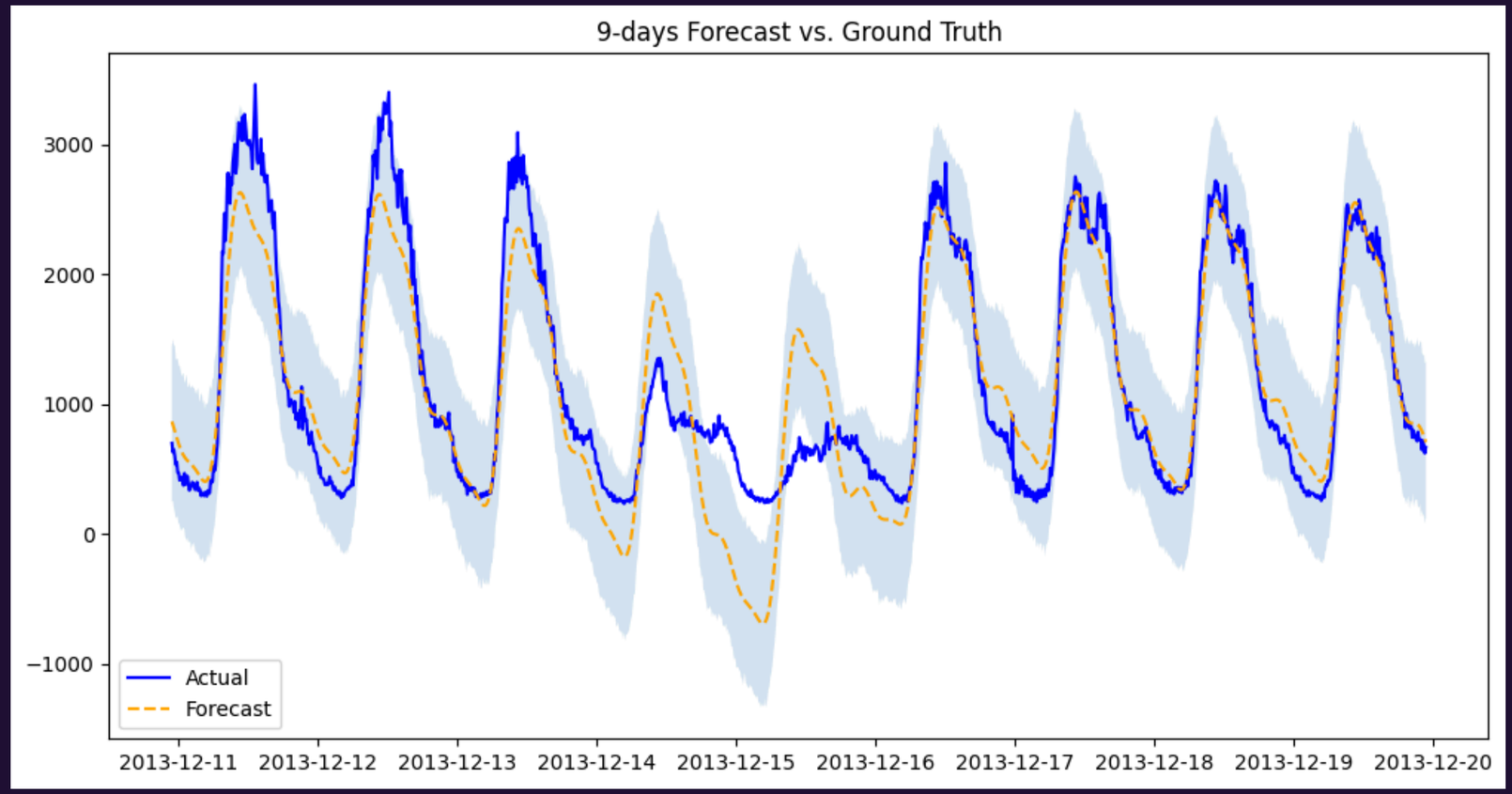


Long term forecast

MAE: 276.55

MAPE: 41.83%

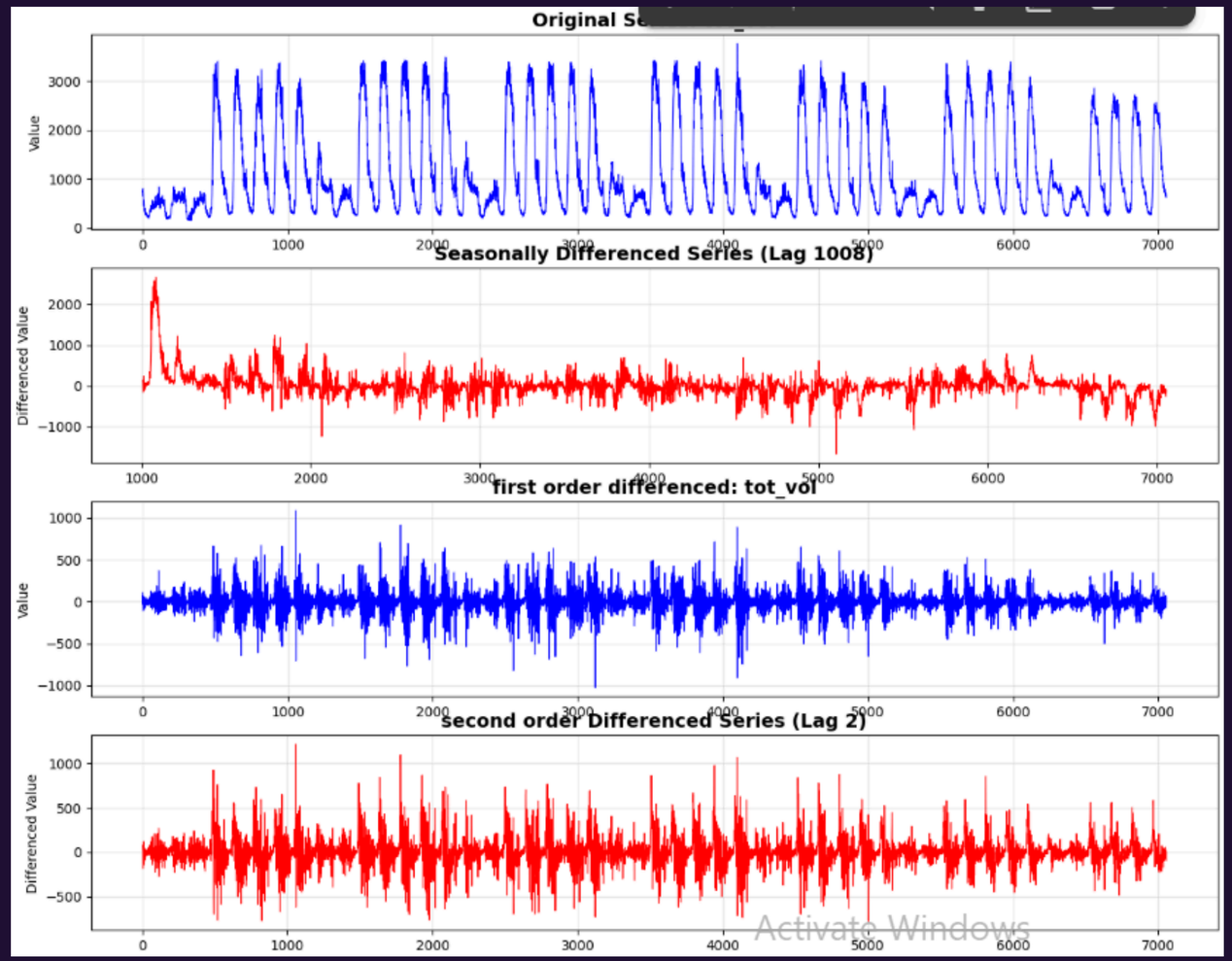
RMSE: 368.08



ARIMA

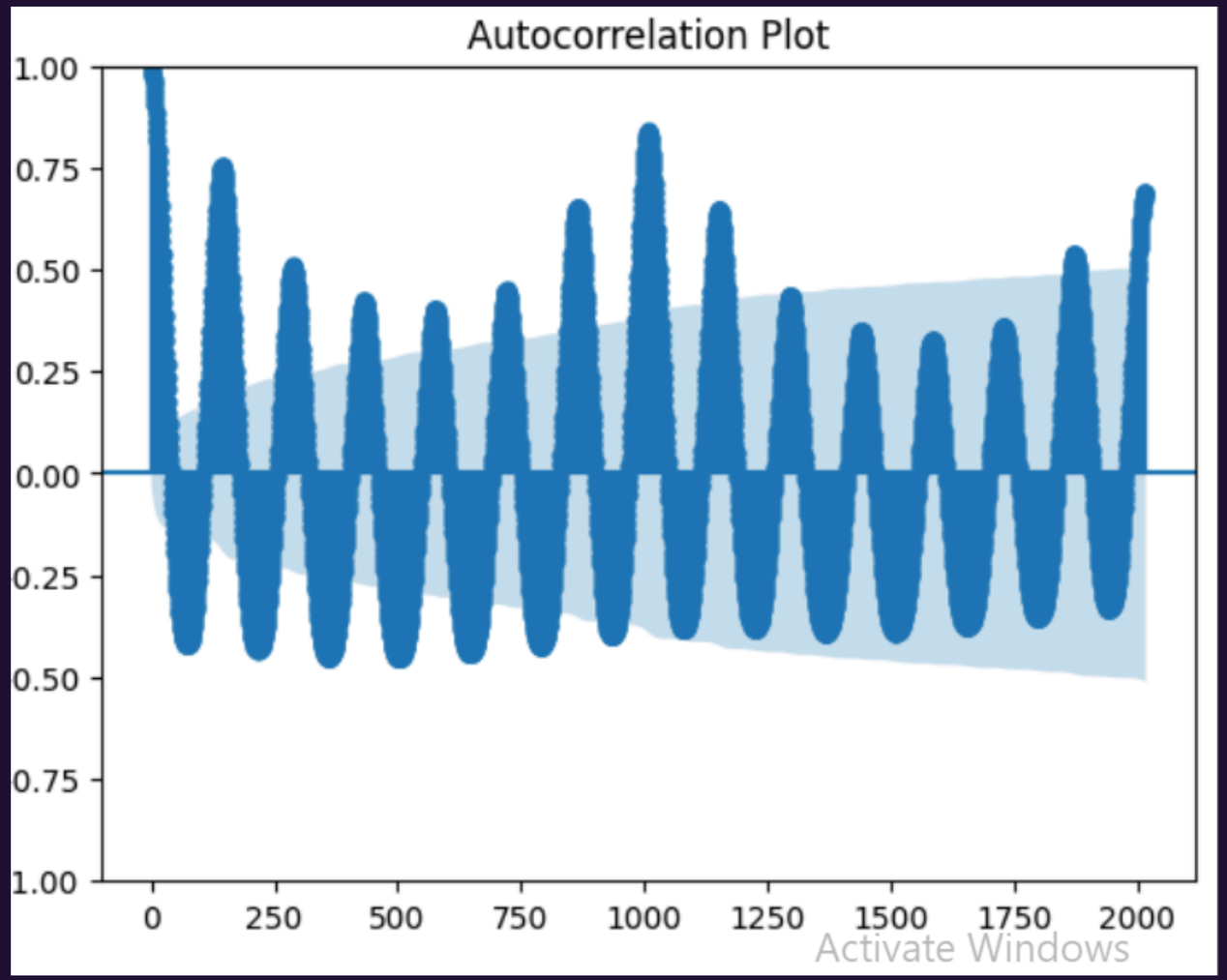
-Seasonal differencing

-1st & 2nd order differencing



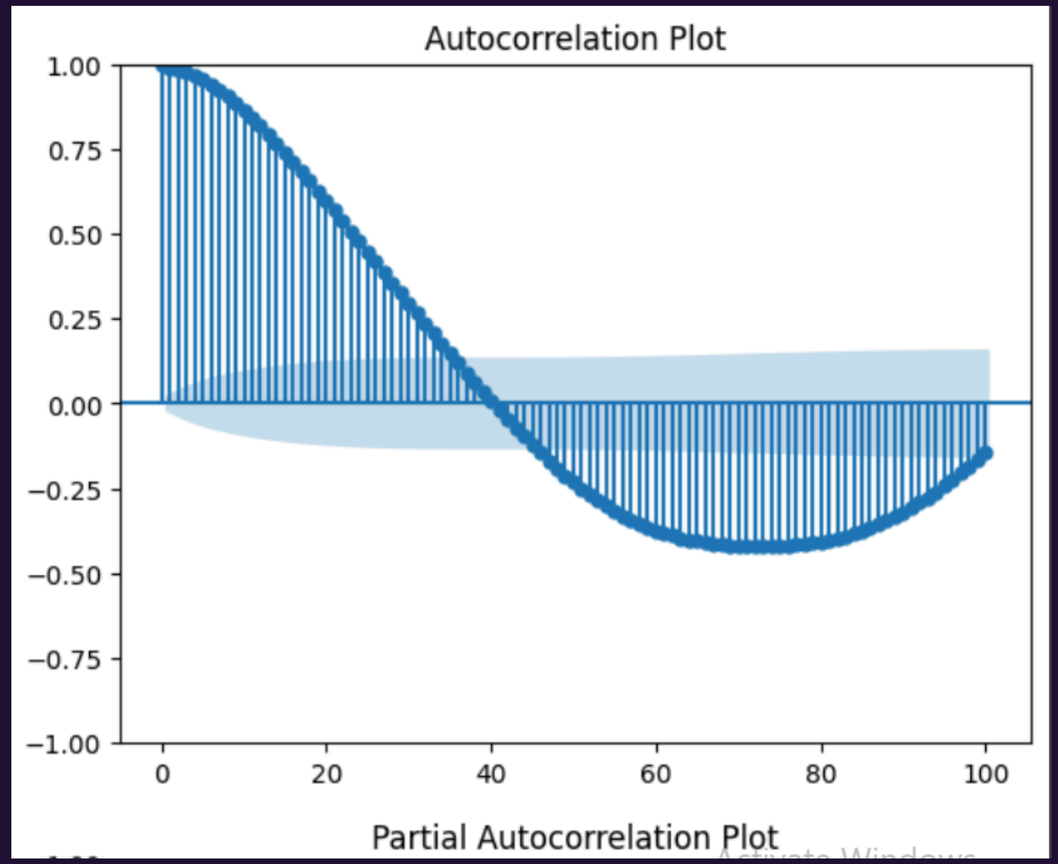
ARIMA

ACF function showing weekly seasonality every 1008 cell



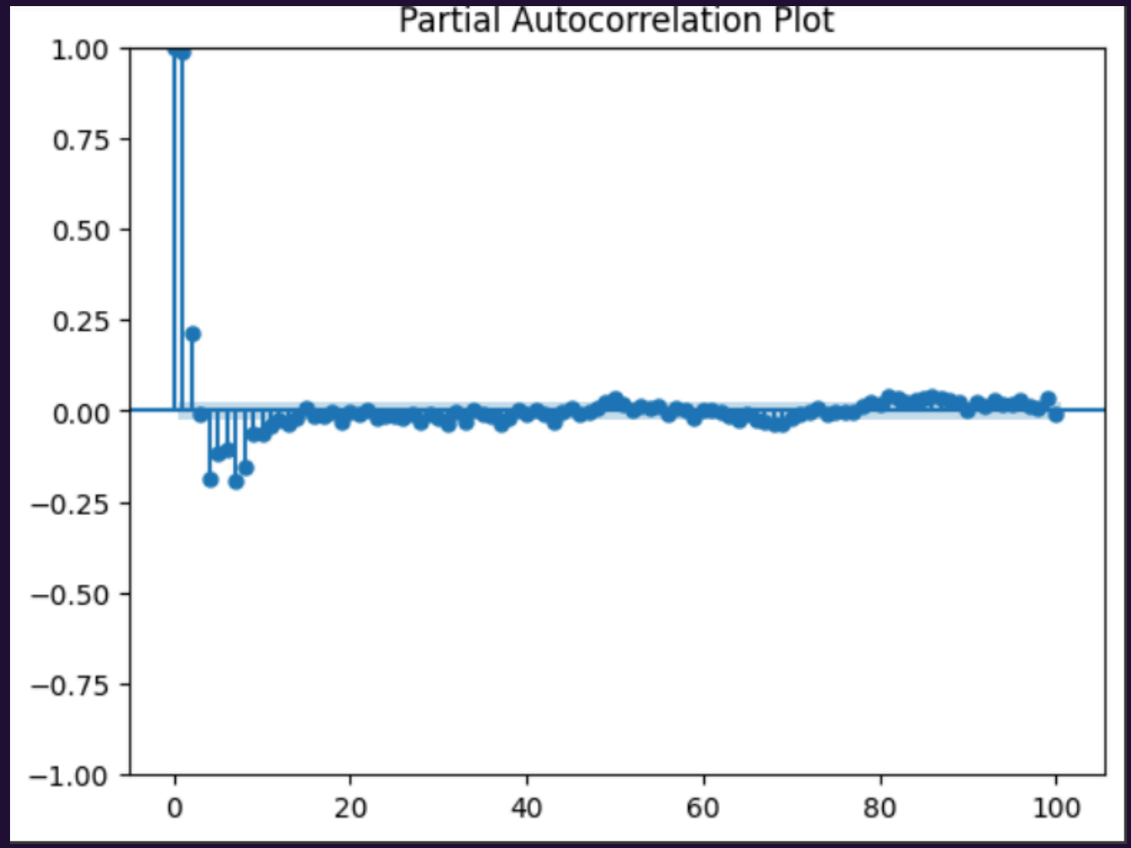
ARIMA

ACF plot:
less lags shows the sinusoidal pattern associated with the presence of a seasonality



ARIMA

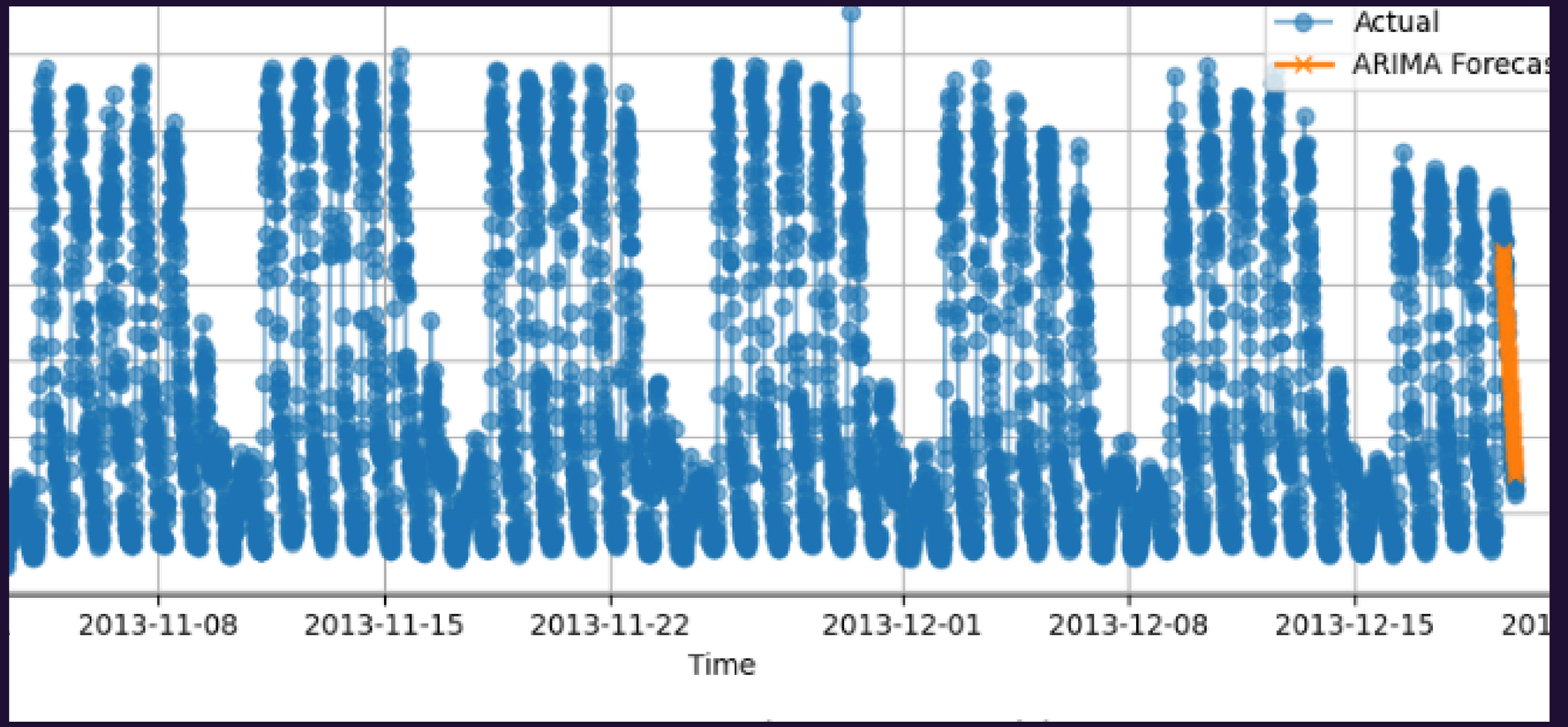
PACF plot: correlation decays fast after the 20th lag



ARIMA

Arima Forecasting

- ARIMA Forecast:
  - MAE = 212.87
  - RMSE = 242.71
  - MAPE = 20.55%




THANK
YOU