Analyzing Traffic congestion through Time series analysis

Jay Rathod
Time series analysis (SC475), DAU
Gandhinagar, India
202201255@daiict.ac.in

Karnik Rathva
Time series analysis (SC475), DAU
Gandhinagar, India
202201266@daiict.ac.in

***Abstract*—Traffic jams are a common phenomenon all over the world, especially in densely populated countries. Hence the traffic related research is a hot topic now a days which will be quite beneficial for all people living in congested cities. Traffic time-series analysis is important because it's the only method that deals with both predicting traffic and controlling it. Subsequently, time series decomposition is performed to extract and visualize the trend, seasonal, and residual components. This is followed by comprehensive statistical analysis, including rolling mean and variance evaluation, autocovariance structure examination, and formal stationarity testing. Additionally, residual analysis is carried out to ensure the residual component is non-deterministic and suitable for further modeling and forecasting.**

I. Introduction

Urban traffic congestion is a growing global concern, driven by increased population density, limited infrastructure, and uncoordinated traffic systems. To address this, data-driven forecasting methods are essential for effective traffic management and planning.

This work highlights the potential of time series analysis in developing smarter, data-assisted urban traffic management systems.

II. Traffic Dataset Description

A. Description

This dataset comprises four features: datetime, number of vehicles, junction, and ID, collected over the period from November 1, 2015, to June 30, 2017. It contains a total of 48,120 hourly observations of vehicle counts recorded across four different junctions. For the purpose of this project, we focus exclusively on Junction 1. Since the original data are recorded on an hourly basis, it tends to be noisy and may obscured underlying trends and patterns. Therefore, we aggregated the data to a daily level to enable clearer visualization and more robust time series analysis.

B. Distribution

Compare how well the Normal vs. Cauchy distributions fit our data. Normal assumes symmetry and thin tails; Cauchy handles heavy-tailed distributions better.

Observations from figure 2.

The histogram is not symmetric and seems right-skewed. This suggests that data is not normally distributed.

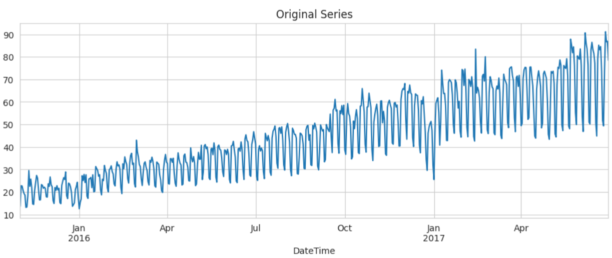

Fig. 1. Original data plot

If the data were normally distributed, the points would lie on the red line. Here, the points curve away from the red line at both ends: The data has heavier tails than a normal distribution likely skewed and not Gaussian. In Figure 2, The distribution isn't perfectly symmetric.

It may have mild skewness and multiple peaks (multimodality).

This tells that simple models assuming normality may not fit well without transformation or differencing.

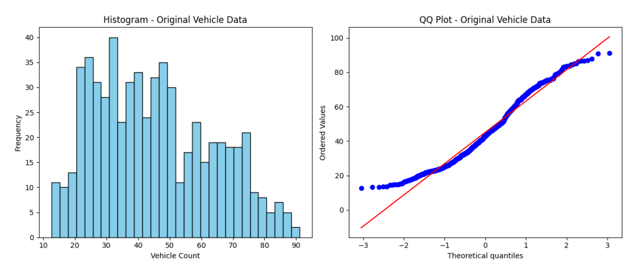

Fig. 2. Histogram and Q_Q plot

Q-Q plot compares the quantiles of the data with the theoretical quantiles of a normal distribution.

In QQ Plot,If the data were perfectly normally distributed, the blue dots would fall along the red diagonal line.But it shows deviations at both ends : heavy tails and some skew,so data is not normally distributed.

C.

Figure 3 illustrates the distribution of traffic number using a histogram overlaid with a kernel density estimate. Two theoretical probability distributions—Normal and Cauchy—are

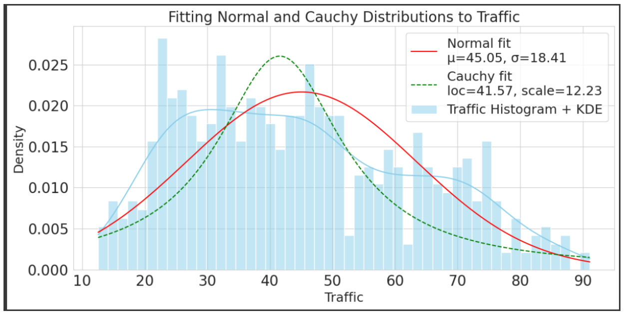

Fig. 3. Normal and Cauchy distribution fitting

fitted to the data to assess the underlying shape. The Normal distribution captures the central bulk of the data but fails to adequately represent the heavier tails. In contrast, the Cauchy distribution provides a better fit in the tails, indicating the presence of extreme traffic values. This suggests that the traffic data exhibits heavier-tailed characteristics and deviates from a perfectly Gaussian pattern, likely due to real-world variability and sporadic pollution spikes.

D. Impact of Holidays on Traffic Volume

The traffic patterns differ between holidays and non-holidays, in Figure 4 the daily vehicle count over time. The blue line represents overall traffic trends, while the red dots highlight traffic volumes on holiday dates.

From the figure, we observe that most red dots, representing holidays, tend to lie at the lower end of the weekly traffic cycles, indicating a noticeable drop in vehicle count on those days. This suggests that traffic volume is generally reduced during holidays compared to normal weekdays.

A quantitative comparison further supports this observation:

- **Average vehicle count on holidays:** 33.98
- **Average vehicle count on non-holidays:** 47.34
- **Percentage difference: -28.21%**

This significant decrease in traffic volume (over 28%) on holidays indicates that public holidays have a marked impact on vehicle movement, likely due to reduced commuting and business activities.

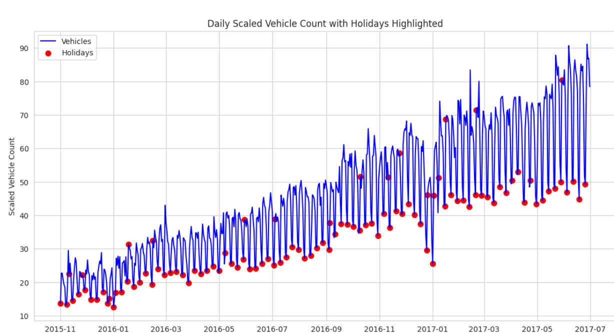

Fig. 4. Daily Scaled Vehicle Count with Holidays Highlighted

III. Rolling Mean and variance of original data

A stationary time series has constant mean and variance over time. Below is the visualization of rolling mean and variance.

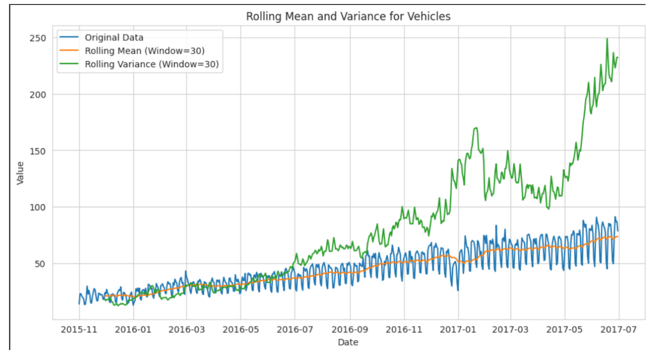

Fig. 5. Rolling Mean and Variance (Window=30) over the Original vehile data

From figure 5, mean and variance are not constant. It depends on time, it is a function of time. To make it stationary time series we need to tackle this problem.

- The **rolling mean of window=30** closely follows the trend, smoothing short-term noise and highlighting long-term behavior.
- The **rolling variance** (orange line) grows with time, confirming the increasing variability in the data.
- These patterns indicate that the series is **non-stationary**, as both mean and variance change over time.

IV. Different transformations on data to stabilize variance

To stabilize variance,we need different transformation method.

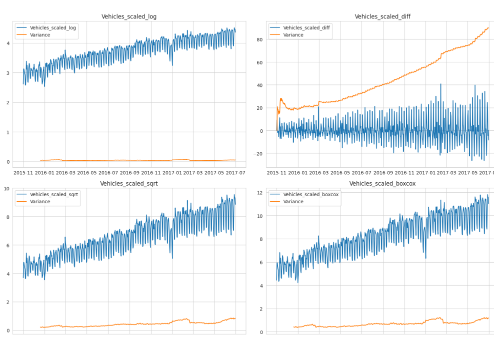

Fig. 6. Various transformation on data and it's variance

- **Log:** Variance (orange line) is mostly stable, indicating good variance stabilization.
- **Difference :**Trend appears removed, but variance is still increasing, especially toward the end.
- **Square root:**Variance (orange line) is lower, but not flat — some increasing pattern remains.
- **boxcox:**Variance line is much smoother and more stable compared to others.

V. Comparison between transformations

- Log Transformation is used when variance increases with larger values.Log Transformation (Blue) yields the lowest and most stable variance.
- Square Root Transformation Works for moderate variance fluctuations.Square Root Transformation (Orange) provides moderate variance stabilization , not as effective as log, but better than raw data.

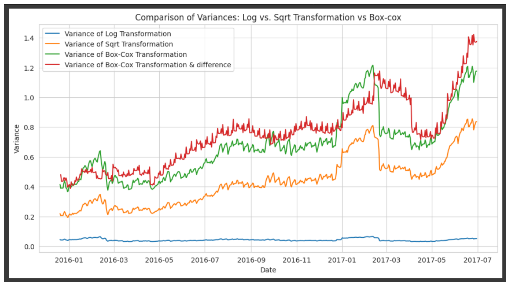

Fig. 7. Comparison of variance of different Transformation techniques

- Box-Cox Transformation is More flexible than log or square root. Requires positive values.Box-Cox Transformation (Green) offers better variance reduction than sqrt but not as good as log.
- Box-Cox + Differencing, it's usually necessary to difference after Box-Cox if your data still has a trend or is not stationary in mean. To remove trends or seasonality, so your mean may still not be stationary. It(Red) shows the highest variance overall, but it captures trends and seasonality changes better (hence more fluctuation). It might be better for models that require stationarity rather than just low variance.

Conclusion: This comparison reveals that the rolling variances for both transformations exhibit similar patterns, with Box-Cox showing a slightly different behavior. These transformations can help address potential non-stationarity, but further analysis may be needed to determine the most effective approach.

VI. Decomposition

First we have to choose between additive or multiplicative decomposition. If the seasonal variation increases or decreases with the trend , a multiplicative model is more suitable.As we seen earlier of plot we could see that seasonality is increasing over time, and trend is also increasing in long term

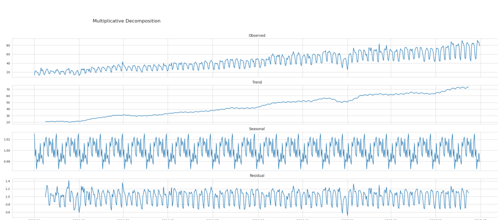

Fig. 8. Multiplicative decomposition of original

- **Observed Data :** The seasonal fluctuations increase as the trend increases.
- **Trend Component :**Shows a clear upward trend.
- **Seasonal Component :** The magnitude of seasonal variations increases as the trend increases, indicating a multiplicative relationship.
- **Residual Component :** The residuals in multiplicative decomposition appear more stable in relative magnitude.

A. Detreding

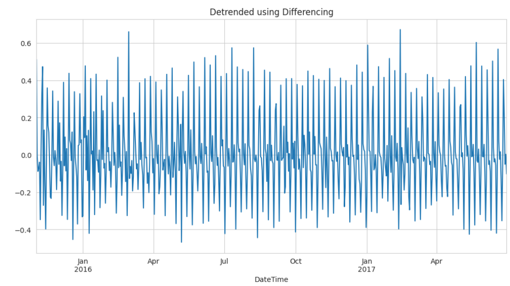

Fig. 9. Detrending using Differencing

Differencing removes trend by subtracting the previous observation from the current one. The series now oscillates around zero, which means trend has been effectively removed.

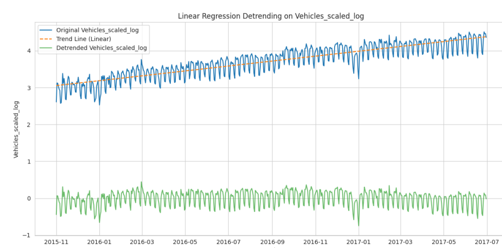

Fig. 10. Linear Regression Detrending

The trend in the original series was mostly linear, making linear regression a good choice for detrending. After detrending (green line), the fluctuations are centered around zero, which is expected. The periodic peaks and valleys suggest strong seasonality, which can now be studied more easily without the trend interfering.

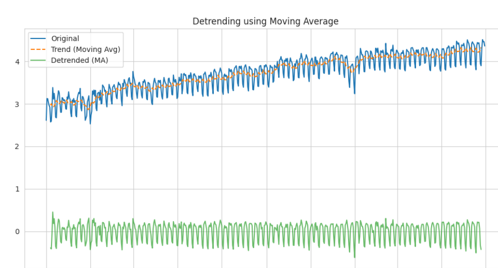

Fig. 11. Detrending using MA

Figure shows the trend-dependent implementation of the MA process. The green line (detrended series) shows the deviations from this trend — fluctuations around zero. The green line still exhibits regular periodic spikes, indicating that seasonal patterns remain in the data.Moving average removes trend, not seasonality.

B. *Deseasonalized*

A traditional and intuitive method for deseasonalization based on smoothing the time series using a centered moving average. The moving average is used to estimate the seasonal component by smoothing out high-frequency noise.

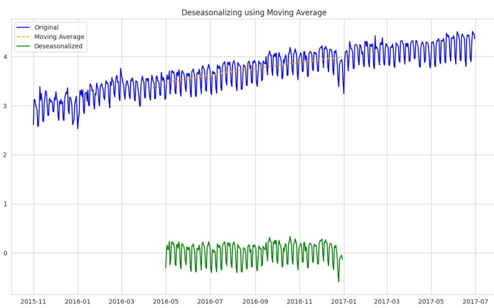

Fig. 12. Deseasonalizing using MA

The Local Trend Method relies on a decomposition technique (like STL or classical decomposition) to isolate the trend, seasonal, and residual components from a time series.

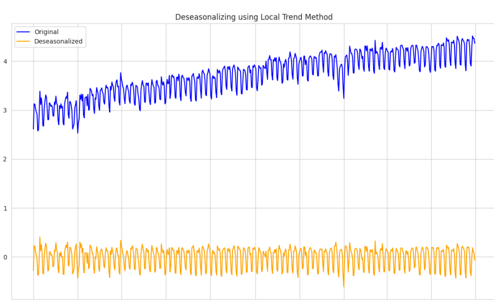

Fig. 13. Deseasonalizing using Local Trend Method

This method uses differencing with a lag equal to the seasonal period to eliminate seasonal cycles.

Fig. 14. Detrending using Lag operator method

VII. Mean, Variance and ACF Plots for Stationary Check

After removing deterministic parts like Trend and Seasonal, we get our residual.

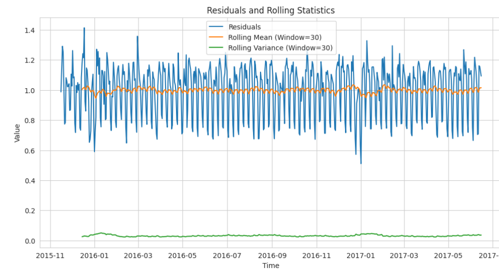

Fig. 15. Rolling mean and rolling variance of Residual with period=30 (monthly)

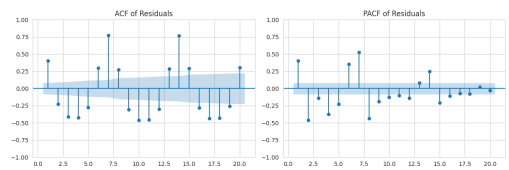

Fig. 16. ACF AND PACF Plot of Residual with period=30 (monthly)

A. . *Decomposition Analysis:*

• Mean: Upon decomposition, the mean of the residuals almost remains constant which indicates that there is no continuous increasing or decreasing pattern.

• Variance: The variance is also found to be constant at almost entire time period which suggests residual might be stationary since both these statistics are constant.

B. *Problem in ACF and PACF plot*

This pattern is not typical white noise — even though our residuals appear stationary (constant mean/variance), they still contain autocorrelation. Model hasn't captured all patterns:

This usually means our model is underfitting — it's not fully explaining the structure in the data.The residuals should look like white noise (i.e., no pattern), but in our case, they still have predictable structure(sinusoidal).

C. *Counter residual problem*

We need to make sure the seasonal component is correctly captured. Consider adjusting the seasonal window or using an alternative method like STL or seasonal_decompose with different parameters.Period is one of the most critical parameter. that defines the number of observations that make up one complete seasonal cycle. Before we were taking period=30 for seasonal_decompose, now if we take period=7(weekly).

To verify that your time series model/decomposition is determinable, residuals should behave like white noise, meaning:

No significant spikes in ACF/PACF (i.e., residuals are uncorrelated) Mean and variance are constant (which you've

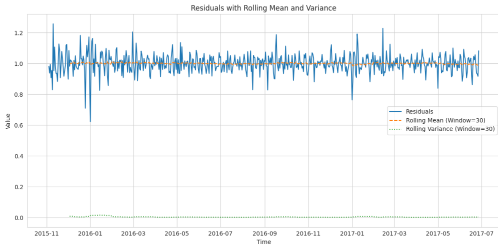

Fig. 17. Rolling mean and rolling variance of Residual with period=7 (weekly)

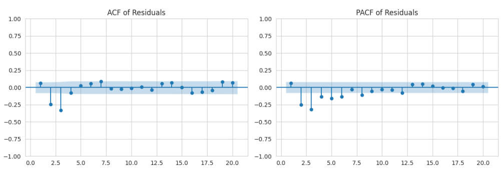

Fig. 18. ACF AND PACF Plot of Residual with period=7 (weekly)

already checked) ACF/PACF should lie within the blue confidence band (usually 95)

After performing seasonal decomposition with a period of 7 (weekly), the residuals were analyzed to verify if they approximate white noise. The following observations were made from the Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) plots:

- **Stationarity of Residuals:** The rolling mean and rolling variance of the residuals were found to be approximately constant, indicating stationarity in the residual component.
- **ACF Behavior:** The ACF plot shows that most of the autocorrelation coefficients lie within the 95% confidence bounds. A few minor spikes are observed at lag 2 and lag 3, but they are not strongly significant. This indicates a lack of strong autocorrelation in the residuals.
- **PACF Behavior:** The PACF plot similarly shows no significant partial autocorrelations beyond lag 2. All values after lag 3 remain within the confidence bounds, suggesting that any remaining structure is negligible.
- **Lack of Periodic Patterns:** No clear sinusoidal or periodic patterns were observed in either plot, confirming that the seasonal component has been adequately captured and removed.
- **White Noise Confirmation:** The behavior of both ACF and PACF plots suggests that the residuals approximate white noise. This implies that the time-dependent structure has been effectively modeled, and the remaining residuals are random.

These observations validate the effectiveness of the decomposition model with a period of 7 and confirm the suitability of the residuals for further modeling or forecasting.

VIII. Conclusion

This report presents a thorough analysis of traffic data using time series techniques. The data was aggregated to daily frequency, and seasonal decomposition revealed clear weekly patterns. Deseasonalization was performed using multiple methods, and stationarity of residuals was verified through rolling statistics and ACF/PACF plots. Overall, the study provided meaningful insights into traffic behavior and established a robust pipeline for time series modeling.

IX. References

References

[1] Peixeiro, Marco. Time Series Forecasting in Python. Simon and Schuster, 2022.

[2] Hyndman, R.J., Athanasopoulos, G. (2018). Forecasting: Principles and Practice.

[3] Hamilton, J.D. (1994). Time Series Analysis. Princeton University Press.