Project Report

Time-series analysis for I-10 / Loop 202 traffic count

Apurba Kumar Saha

Introduction:

Sitting in bumper-to-bumper traffic during rush hour is an ubiquitous reality of modern life. But what if we could anticipate and prevent the worst bottlenecks before they happen? This is exactly what we set out to do with our machine learning models focused on the Arizona I-10 / Loop 202 interchange.

This major freeway interchange in Phoenix ties together Interstate 10 and State Loop 202 - two highways that cumulatively carry over 200,000 vehicles per day. Even minor disruptions can spiral into mile-long standstills. Our analysis targets the prediction of daily traffic volumes specifically at this interchange to identify future periods of exceptionally heavy congestion.

By forecasting traffic surges in advance, our models can empower transportation authorities to take preemptive actions like modifying ramp metering, adjusting speed limits or even issuing precautionary warnings. This could smooth traffic flows and drastically reduce congestion for thousands of daily commuters.

We formulate this as a time-series problem, with a rich temporal dataset tracing 730 days of traffic history. By harnessing machine learning and statistical analysis, our data-driven models aim to foresee traffic volumes many days into the future. The end goal is not merely accurate predictions, but actual alleviation of real-world congestion through preventative measures. In this project, we try to find the answer for the following question: “Will our interchange continue to frustrate drivers, or can machine learning offer a smoother ride?”

Literature review:

Traffic volume forecasting has been extensively studied using both statistical and machine learning techniques. On the statistical side, Autoregressive Integrated Moving Average (ARIMA) and its seasonal variant SARIMA are widely applied for modeling time series such as traffic flow (Williams and Hoel, 2003). Exponential smoothing methods like Holt-Winters have also shown success by accounting for trends and seasonal cycles (Smith et al., 2002).

Machine learning approaches like regression trees and ensembles of trees tend to outperform statistical methods, as demonstrated empirically in a recent comparison study (Lv et al., 2015). Tree-based approaches automatically model complex data relationships and temporal dependence. Random forest regression, incorporating bagging and feature randomness, yielded the most accurate forecasts across multiple traffic datasets in the study. Broadly, deep learning models like Long Short-Term Memory (LSTM) networks have delivered state-of-the-art results by directly learning traffic flow characteristics from raw input sequences (Ma et al., 2017).

In addition to daily and weekly seasonal cycles, incorporating explanatory variables like weather, holidays and local events can improve prediction accuracy (Ding et al., 2002). Features capturing historical lagged traffic can also help neural network models learn complex traffic dynamics (Wu et al., 2003).

In this project, we will explore statistical methods and machine learning models to predict traffic volume. Overall, our feature-rich dataset provides a strong foundation for evaluating both classical and modern traffic prediction techniques applicable to this I-10/Loop 202 interchange.

Data Description

The dataset for this project consists of daily traffic volume data at the Arizona I-10 / Loop 202 interchange for the years 2018 and 2019. The raw data comes from the Federal Highway Administration's (FHWA) Traffic Monitoring Analysis System (TMAS) and can be accessed on FHA website. The dataset contains traffic counts for both eastbound and westbound directions, providing a complete view of overall daily traffic flow through this major interchange. In total, the dataset spans 729 days over the two-year period from January 1, 2018 to December 31, 2019. Each row in the dataset corresponds to a single calendar date and contains two variables:

- Date: The calendar date in YYYY-MM-DD format
- Traffic Volume: The total vehicles per day traversing the interchange for that date.

In its raw form, the data consists of 1458 rows and 2 columns. In addition to the raw date and traffic volume variables from the TMAS dataset, several new indicator features have been constructed to capture relevant temporal and seasonal effects. Categorical features like day of week, week of month and month have been converted into dummy variables using one-hot encoding:

- Day of Week: Transformed into 6 binary columns indicating each day.
- Week of Month: Encoded as 3 binary columns indicating week 1, week 2 etc.
- Month: Converted into 11 binary columns for each month.

By dropping one level for each categorical feature during one-hot encoding, all the dummy variables introduced are independent of each other. The other engineered inputs include:

- *Trend:* A linear trend variable representing the overall directional increase or decrease in traffic.
- *Day of Week:* Categorical variable indicating the day - Monday, Tuesday etc.
- *Week of Month:* Numerical variable identifying 1st week, 2nd week etc. within each month.
- *Month:* Categorical indicator for the 12 months.
- *Thanksgiving Week:* Captures the peak travel week containing Thanksgiving.
- *New Year Season:* Binary indicator for late-December weeks with holiday traffic.
- *Holiday:* Binary indicator marking recognized public holidays along with 1 day lagging and 1-day leading holiday indicators.
- *Long Weekend:* Binary flag for holiday weekends expected to have exceptional traffic, including 1-day lag and lead of this flag.
- *Lagged traffic volume:* Traffic volume from prior 2 days included to model short-term autocorrelation.

These engineered inputs supplement the raw series with relevant seasonal, holiday, lag and lead information. Prior literature strongly supports their utility in improving predictive traffic models. In total, the final preprocessed dataset comprises 1458 rows and 44 features. The wide range of temporal effects captured by these features should provide a rich set of inputs to enable our machine learning models to accurately forecast daily traffic volumes, even for exceptional periods like holiday weekends or annual traffic surges.

Data exploration:

Data cleaning:

The raw TMAS dataset was first checked for invalid or erroneous records. All dates fell within the valid 2018-2019 timeframe and traffic values ranged between 60,000 and 120,000 vehicles per day. There were missing values as the traffic volume tracker at our target station was out of order. So, we used the data from

a backup station to fill in the missing values in our primary dataset. The backup station was located within half a mile of our target station and there was no major exit point between these stations. So, the traffic count should almost be the same for both stations. In cases when both stations were down, we used linear interpolation to fill in the missing data.

Visualization:

An annual and weekly time series plots were generated to analysis the annual and weekly pattern of the traffic volume (Figure 1). The annual pattern shows a uniform distribution of traffic flow throughout the year expect in March and April (due to tourism season) as well as in November and December (due to vacation period). The weekly pattern clearly shows less traffic volume on weekends, which indicates weekly seasonality. The large spikes at seasonal lags (lag 7, 14, 21, and 28) confirms the presence of such weekly seasonality.

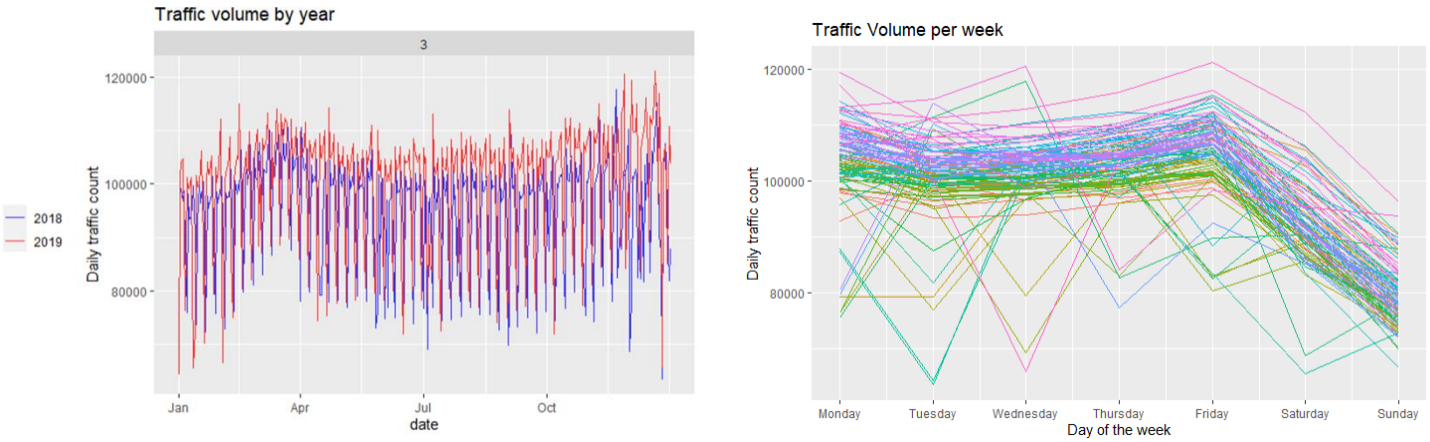


Figure 1: Annual and weekly pattern of traffic flow

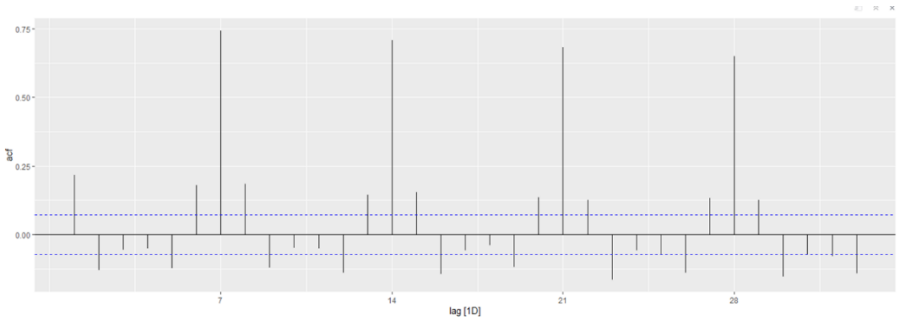


Figure 2: Auto-correlation plot for the traffic volume

Moreover, STL decomposition was used to break the time-series data down into three components: trend, weekly seasonality, and remainder. STL decomposition is a popular method for decomposing time series data into its constituent components. STL stands for "Seasonal and Trend decomposition using Loess". The STL method fits Loess curves to extract the trend and seasonal components separately. Figure 3 depicts this decomposition graphically.

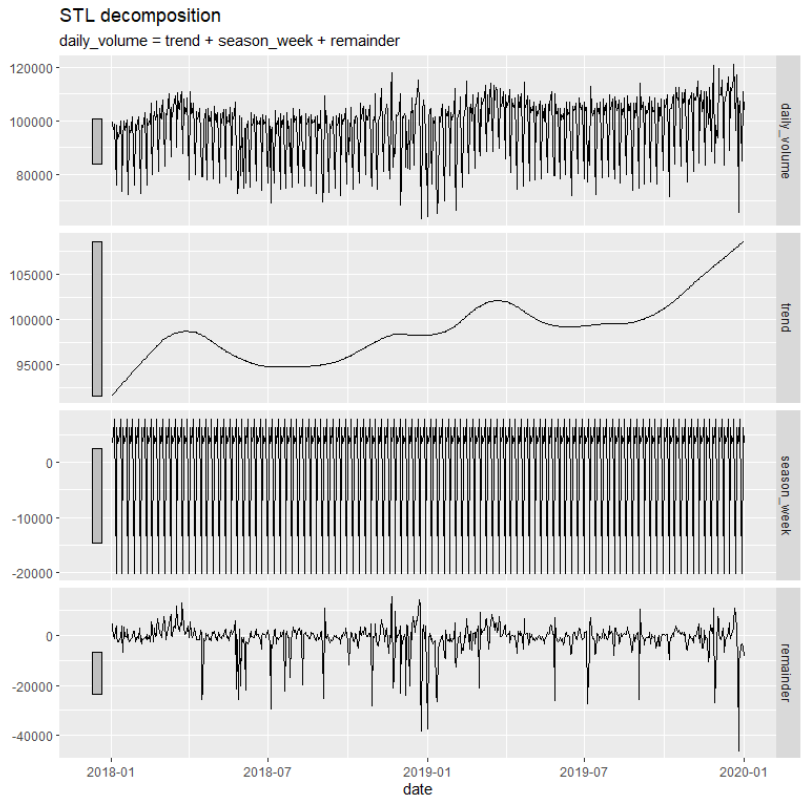


Figure 3: Decomposition of time-series data

As we can observe in Figure 3, the remainder component doesn't fluctuate randomly around zero. Also, there is significant noise present in the plot, which suggests the leak of important signal in the remainder component. Thus, we need to use exogenous variables along with trend and seasonality to explain the time-series data well.

Data Analysis:

The raw dataset was divided into training and test sets for modeling. The training data spans January 1, 2018, to November 30, 2019, while the test set comprises December 2019 to allow final model evaluation on the most recent unseen traffic figures. Next, we tried different methods to fit the training data and evaluate the performance. Each method is described below:

Baseline models:

At first, we evaluated four baseline models to set the benchmark for our time-series analysis. This methods include:

- Naive - Forecasts are the last observed value.
- Drift - Fit a linear trend line extrapolated into future.
- Mean - Predict historical mean traffic every day.
- Seasonal Naive - Forecasts are last week's observed value.

These simplistic baselines will provide comparative reference points to assess if the more advanced methods like SARIMA and machine learning can deliver meaningful improvements. Each model was trained on the full training dataset and used to evaluate the performance on test data using mean absolute percentage error (MAPE) and root mean scaled squared error (RMSSE). The performance scores are shown in Table 1.

Table 1: Performance scores for baseline models

.model <chr>	MAPE <dbl>	RMSSE <dbl>
Drift	13.33265	2.078533
Mean	12.06426	1.887269
Naïve	13.27572	2.071091
Seasonal_naïve_week	12.17263	2.205258

Among the baselines, Mean produced the best results with a MAPE of 12.06 and RMSSE of 1.88. So, while simplistic, the mean approach provides a competitive benchmark for forecasting traffic volume.

SARIMA:

Initial unit root tests using the Kwiatkowski–Phillips–Schmidt–Shin (KPSS) test revealed non-stationarity in the traffic volume time series. Applying one order of seasonal differencing at a lag of 7 days addressed this, yielding a stationary series suitable for further modeling.

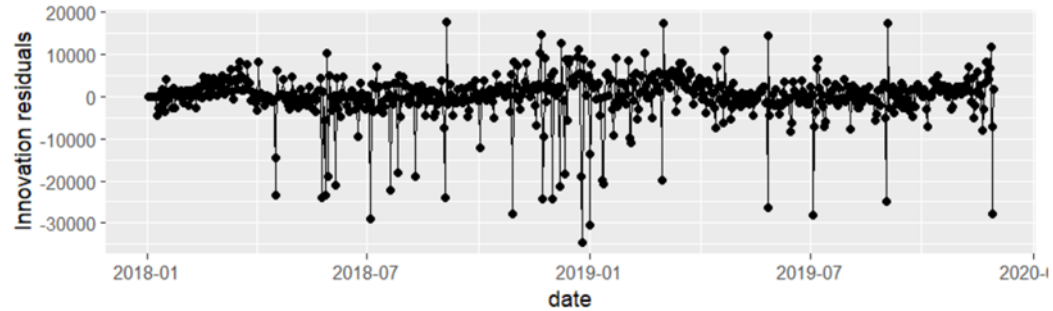
Seasonal ARIMA with a stationary input and grid search hyperparameters tuning identified the optimal seasonal ARIMA model to be ARIMA(5,0,0)(0,1,1)[7]. This contains:

- AR(5) - A 5th order autoregressive component
- I(0) - No differencing required
- MA(0) - No moving average component
- Seasonal AR(0) - No seasonal autoregression
- Seasonal I(1) - 1 order of seasonal differencing
- Seasonal MA(1) - 1st order seasonal moving average

So, in total, the chosen Seasonal ARIMA formulation for the I-10/Loop-202 traffic data is: SARIMA(5, 0, 0)(0, 1, 1)[7]. This model was fitted on the full training dataset and was evaluated on the test data. With a **MAPE** score of 8.02 and **RMSSE** score of 1.41, SARIMA strongly outperformed the baseline statistical approaches by better accounting for autocorrelation, trends and multiple seasonal cycles. However, when performing the residual analysis, we observe some significant issues. As we can observe in Figure 4, there is significant autocorrelation between the residuals as shown in the ACF subplot. This observation is supported by the p-value of the Ljung-box test (a p-value less than 0.05 rejects the null hypothesis that the observed residuals are not autocorrelated). Additionally, the distribution of the residuals is skewed, and the variance is not constant as shown in the first and third subplot, respectively. Such violation in the assumption will impact the estimation of confidence interval for the forecasted value.

Linear Regression:

In addition to SARIMA, a linear regression model was tested for predictive performance. Backwards stepwise feature selection identified an optimal subset of predictors. The Bayesian Information Criterion (BIC) resulted in a model with 13 predictor variables including trend, seasonality indicators and lags. The Akaike Information Criterion (AIC) selected 20 features, including additional interaction and squared terms at a higher model complexity. The selected features for both criterion and their estimated coefficients are shown in Figure 5. On the test set, the complex model with 20 features achieved a better RMSSE score of 1.24 and MAPE score of 7.10 compared to the simpler 13 feature BIC formulation (MAPE = 7.52 and RMSSE = 1.27). It suggests that we need more variables to explain the complexity of the time-series data.



Travel_Dir <int>	.model <chr>	lb_stat <dbl>	lb_pvalue <dbl>
3	auto	64.51565	1.393687e-06
7	auto	87.42903	2.079639e-10

Figure 4: Residual analysis for SARIMA

```
Series: daily_volume
Model: TSLM

Residuals:
     Min       1Q   Median       3Q      Max
-28607.7  -1285.5    347.1   1967.6  24664.2

Coefficients:
                      Estimate Std. Error t value Pr(>|t|)
(Intercept)           8.490e+04  2.580e+03  32.906  < 2e-16 ***
trend()               8.877e+00  9.709e-01   9.143  < 2e-16 ***
is_holiday           -2.516e+04  1.853e+03 -13.575  < 2e-16 ***
is_long_weekend       1.342e+04  2.288e+03   5.866 6.96e-09 ***
is_long_weekend_lead2 4.800e+03  1.331e+03   3.605 0.000335 ***
daily_volume_lag1     1.020e-01  2.729e-02   3.738 0.000201 ***
day_of_week.L         5.912e+03  6.616e+02   8.936  < 2e-16 ***
day_of_week.Q        -1.852e+04  5.025e+02 -36.860  < 2e-16 ***
day_of_week.C         4.475e+03  5.081e+02   8.808  < 2e-16 ***
`day_of_week^4`      -1.219e+04  5.730e+02 -21.269  < 2e-16 ***
`day_of_week^5`       3.110e+03  5.462e+02   5.694 1.84e-08 ***
`day_of_week^6`      -2.478e+03  5.044e+02  -4.912 1.13e-06 ***
month.C               3.633e+03  6.830e+02   5.319 1.42e-07 ***
`month^4`            -5.601e+03  6.767e+02  -8.276 6.66e-16 ***
`month^7`            -2.925e+03  6.289e+02  -4.651 3.97e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4833 on 683 degrees of freedom
Multiple R-squared: 0.8059,     Adjusted R-squared: 0.802
F-statistic: 202.6 on 14 and 683 DF, p-value: < 2.22e-16
```

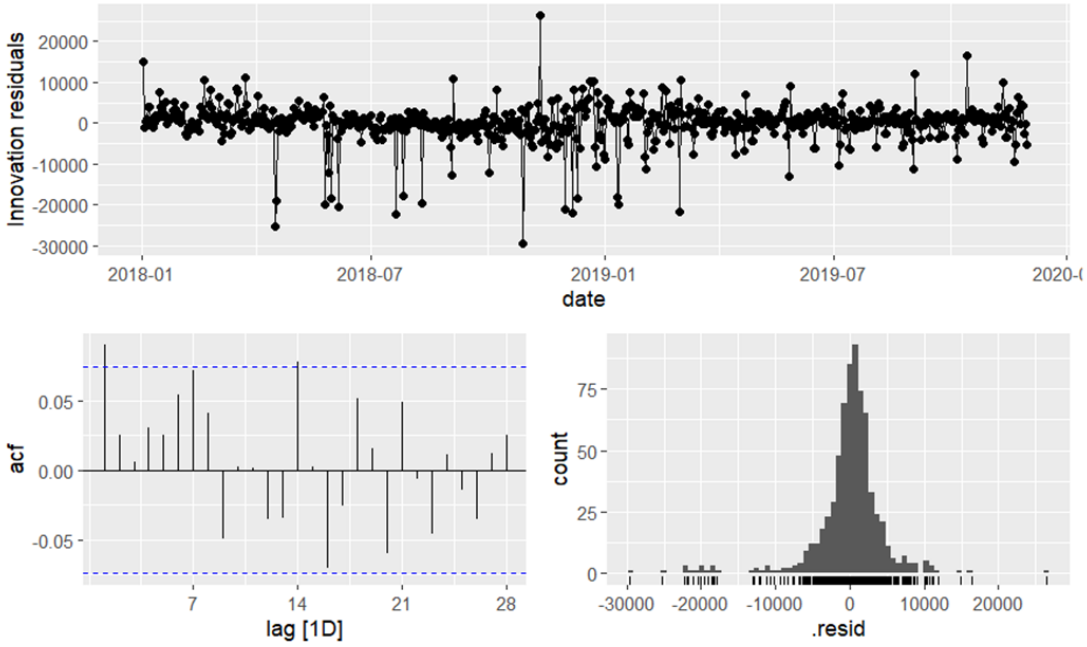
(a)

term <chr>	estimate <dbl>
ar1	-2.404178e-01
ar2	-1.059629e-01
ar3	-1.089662e-01
sar1	4.584445e-01
sma1	-4.292876e-01
sma2	1.126943e-01
trend()	4.652165e+00
is_holiday	-1.789463e+04
is_long_weekend	1.205319e+04
is_thanksgiving_week	2.207452e+03
is_new_year_season	-8.093390e+02
is_long_weekend_lag2	-2.795012e+03
is_long_weekend_lag1	-4.458062e+01
is_long_weekend_lead2	4.020250e+03
daily_volume_lag1	4.106567e-01
daily_volume_lag6	1.388573e-01
week_of_month1	-7.285968e+02
day_of_week.L	9.429064e+03
day_of_week.Q	-1.238641e+04
day_of_week.C	1.223215e+03
`day_of_week^4`	-7.703871e+03
`day_of_week^5`	-7.157400e+02
`day_of_week^6`	-2.109980e+03
month.C	1.918071e+03
`month^4`	-3.129339e+03
`month^7`	-1.164242e+03
`month^8`	2.410029e+02
intercept	4.096131e+04

(b)

Figure 5: a) Selected features for BIC criterion b) Selected features for AIC criterion.

However, significant correlation was still present between the residuals found from AIC formulation as supported by the p-values of Ljung-box test in Figure 6. It indicates there is still scope for improvement as important signals are still left in the residuals.



Travel_Dir <int>	.model <chr>	lb_stat <dbl>	lb_pvalue <dbl>
3	tslm	21.97060	0.07922053
7	tslm	24.27206	0.04247173

Figure 6: Residual analysis for linear regression

Dynamic regression:

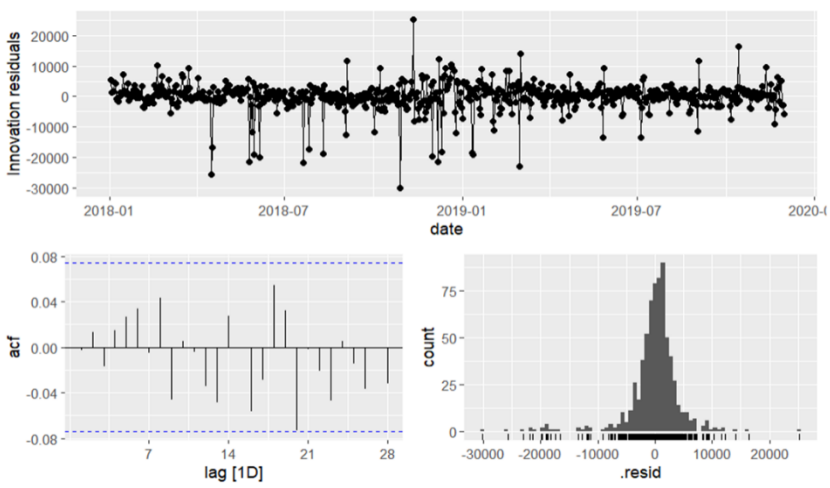
As an extension to standard linear regression, a Dynamic Regression formulation was evaluated. Standard regression models do not allow for the subtle time series dynamics that can be handled with ARIMA models. In dynamic regression, we allow the errors from a regression to contain autocorrelation. The regression error series, η_t is assumed to follow an ARIMA model. For example, if η_t follows an ARIMA(1,1,1) model, we can write:

$$y_t = \beta_0 + \beta_1 x_{1,t} + \cdots + \beta_k x_{k,t} + \eta_t,$$
$$(1 - \phi_1 B)(1 - B)\eta_t = (1 + \theta_1 B)\varepsilon_t,$$

where, ε_t is a white noise series. Notice that the model has two error terms here — the error from the regression model, which we denote by η_t, and the error from the ARIMA model, which we denote by ε_t. Only the ARIMA model errors are assumed to be white noise. Again, we used Akaike Information Criterion (AIC) chosen 20 variables for inclusion initially, similar to the linear regression case. And the optimal seasonal ARIMA order that was identified to model η_t was the ARIMA (3,0,0)(1,0,2)[7]. The estimated coefficients and residual plots for this dynamic regression model are shown in Figure 7. As we observe from the autocorrelation plot, all the lag values are bounded by the blue dotted lines, which suggests the residuals are white noise. Thus, the model captures all the critical information in training data well. However, when evaluating test data, the model performance deteriorates compared to SARIMA and standard linear regression models achieving an RMSSE score of 1.35 and MAPE score of 8.09. Thus, this model may be too complex for our time-series data.

term	estimate
<chr>	<dbl>
ar1	-2.404178e-01
ar2	-1.059629e-01
ar3	-1.089662e-01
sar1	4.584445e-01
sma1	-4.292876e-01
sma2	1.126943e-01
trend()	4.652165e+00
is_holiday	-1.789463e+04
is_long_weekend	1.205319e+04
is_thanksgiving_week	2.207452e+03
is_new_year_season	-8.093390e+02
is_long_weekend_lag2	-2.795012e+03
is_long_weekend_lag1	-4.458062e+01
is_long_weekend_lead2	4.020250e+03
daily_volume_lag1	4.106567e-01
daily_volume_lag6	1.388573e-01
week_of_month1	-7.285968e+02
day_of_week.L	9.429064e+03
day_of_week.Q	-1.238641e+04
day_of_week.C	1.223215e+03
`day_of_week^4`	-7.703871e+03
`day_of_week^5`	-7.157400e+02
`day_of_week^6`	-2.109980e+03
month.C	1.918071e+03
`month^4`	-3.129339e+03
`month^7`	-1.164242e+03
`month^8`	2.410029e+02
intercept	4.096131e+04

(a)



(b)

Figure 7: a) Estimated coefficients b) Residual analysis.

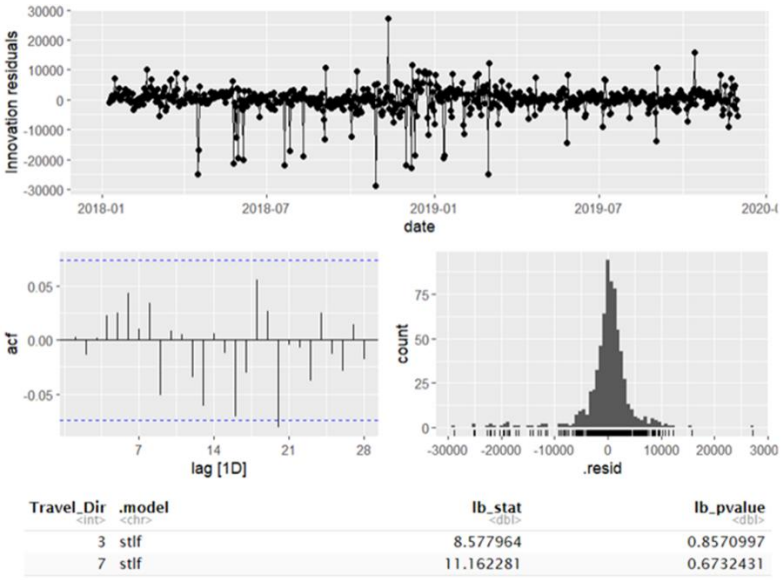
Decomposition method:

Assuming an additive decomposition, the decomposed time series can be written as $y_t = S_t + A_t$, where $A_t = T_t + R_t$ is the seasonally adjusted component. To forecast a decomposed time series, we forecast the seasonal component, S_t, and the seasonally adjusted component A_t, separately. It is usually assumed that the seasonal component is unchanging, or changing extremely slowly, so it is forecast by simply taking the last year of the estimated component. In other words, a seasonal naïve method is used for the seasonal component. To forecast the seasonally adjusted component, any non-seasonal forecasting method may be used. In this project, we used dynamic regression to forecast the seasonally adjusted component. The estimated coefficient for the dynamic regression model is shown in Figure 8.

```
Series: daily_volume
Model: STL decomposition model
Combination: season_adjust + season_week

========================================

Series: season_adjust
Model: LM w/ ARIMA(0,0,0)(1,0,1)[7] errors

Coefficients:
         sar1     sma1  trend()  is_holiday  is_long_weekend  is_thanksgiving_week  is_new_year_season  is_long_weekend_lag2
      0.8815  -0.8412   7.7987  -26919.639        15006.137              2539.881             35.4794               44.2939
s.e.  0.1221   0.1398   1.2321    1857.119         2185.928              1397.337           1822.2034             1337.9996
      is_long_weekend_lag1  is_long_weekend_lead2  lag(season_adjust, 1)  lag(season_adjust, 2)  lag(season_adjust, 3)    month.C
                 4689.844               4490.225                 0.1899                 0.0265                -0.0079  3277.3572
s.e.             1328.271               1241.775                 0.0332                 0.0341                 0.0325   735.3434
      `month^4`   `month^7`  `month^8`  `month^9`  intercept
      -4606.580  -2481.7941  1297.5434  -852.2998  74773.139
s.e.    750.049    628.4171   602.5244   593.6297   4885.812

sigma^2 estimated as 21622101:  log likelihood=-6847.16
AIC=13734.32   AICc=13735.56   BIC=13825.29

Series: season_week
Model: SNAIVE
```

Figure 8: Coefficients for dynamic regression method fitting seasonally adjusted data.

Diagnostic evaluation, as depicted in Figure 9, showed no major violations of normality and constant variance assumptions in this model. Additionally, there is no autocorrelation between the residuals. Thus, this method captures useful information in the training data. When evaluating on test data, this model achieved an RMSSE score of 1.20 and a MAPE score of 7.01, which is the best performance among all the models considered so far.



Travel_Dir <int>	.model <chr>	lb_stat <dbl>	lb_pvalue <dbl>
3	stlf	8.577964	0.8570997
7	stlf	11.162281	0.6732431

Figure 9: Residual analysis for decomposition method

Comparisons and Summary:

In Table 2, we summarize the performance of all models we have studied in this project:

Table 2: Comparison of models

MODEL	MAPE	RMSSE
Mean	12.06	1.88
Sarima	8.02	1.41
Linear Regression	7.10	1.24
Dynamic Regression	8.09	1.35
Decomposition	7.01	1.20

From Table 2, it is obvious that Decomposition method performed the best in testing environment. This model leads to a 42% improvement in MAPE score and 35% improvement in RMSSE score compared to the best baseline model. So, we applied this decomposition model to make the final forecast. A graphical illustration comparing the forecasted values with actual values is shown in Figure 10.

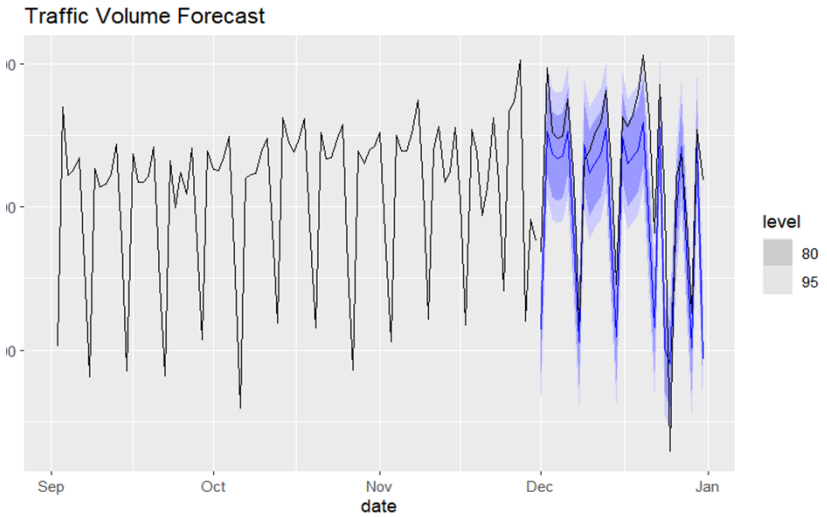


Figure 10: Forecast of traffic volume using decomposition method

In Figure 10, the forecasted values represented by the blue line can accurately capture both the seasonal cycles and directional trend evident in the actual traffic volume data traced by the black line. Moreover, the real test set traffic volumes fall within the 95% prediction intervals of our model over the entire December 2019 horizon. This close agreement between forecasts and outcomes indicates the decomposition modeling approach can reliably predict future traffic patterns, within reasonable quantification of uncertainty bounds.

Final Remarks:

We found that the decomposition method performed the best among all the models considered in this study. However, we haven't explored tree models and neural networks in this project. In future, those models can be applied to predict traffic volume. As these models are more complex than the models considered in this project, they can improve the evaluation metric scores further.

References:

Williams, B.M., Hoel, L.A., 2003. Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: Theoretical basis and empirical results. *Journal of Transportation Engineering* 129 (6), 664-672.

Smith, B.L., Williams, B.M., Oswald, R.K., 2002. Comparison of parametric and nonparametric models for traffic flow forecasting. *Transportation Research Part C: Emerging Technologies* 10(4), 303-321.

Lv, Y., Duan, Y., Kang, W., Li, Z., Wang, F.Y., 2015. Traffic Flow Prediction with Big Data: A Deep Learning Approach. *IEEE Transactions on Intelligent Transportation Systems* 16(2), 865-873.

Ding, A., Zhao, X., Jiao, L., 2002. Traffic flow time series prediction based on statistical learning theory. *IEEE 5th International Conference on Intelligent Transportation Systems*. pp. 727-730.

Wu, C.H., Ho, J.M., Lee, D.T., 2004. Travel-time prediction with support vector regression. *IEEE transactions on intelligent transportation systems* 5(4), 276-281.

Ma, X., Tao, Z., Wang, Y., Yu, H., Wang, Y., 2017. Long short-term memory neural network for traffic speed prediction using remote microwave sensor data. *Transportation Research Part C: Emerging Technologies* 54, 187-197.