Internet Traffic Prediction Using Time Series Forecasting Models SARIMA and LSTM Models

Zehra KOLAT 206001007

Ayşe Lara GÜNEŞ 206001008

Enes KALECİ 206001021

ISE043 Time Series Analysis and Forecasting

Instructor: Nadi Serhan AYDIN

Istinye University

Faculty of Engineering and Natural Sciences

Mathematics Department

January 2025

Internet Traffic Prediction Using Time Series Forecasting Models SARIMA and LSTM Models

Zehra KOLAT 206001007
Ayşe Lara GÜNEŞ 206001008
Enes KALECİ 206001021

25/01/2025

Abstract

This study aims to predict daily vehicle traffic using SARIMA and LSTM models based on two-year dataset. Multiple parameters were used to determine the most suitable SARIMA model. The analysis showed that the SARIMA model successfully identified both seasonal and non-seasonal patterns in the data. On the other hand, the LSTM model achieved better results in more complex data. The performance of the model was evaluated using RMSE. The generated predictions aim to provide information for traffic prediction in the study area.

Acknowledgement

We would like to thank our Instructor, Dr. Nadi Serhan Aydın , for the guidance and support he provided throughout this study.

Contents

1 Introduction **3**

2 Literature review **6**

3 Data and Time-Series Analysis **8**

3.1 SARIMA Model 8

4 Model Forecasting **9**

5 Forecasting Results **16**

6 Conclusion **22**

1 Introduction

Today, traffic congestion not only causes time loss and stress for individuals, but also brings about large-scale problems such as environmental impacts and economic losses. Traffic increases the pressure on the environment by causing air pollution, noise pollution and unnecessary fossil fuel use. Prolonged idling (in motor vehicles, setting the engine to run smoothly with minimal fuel while the vehicle is stationary) and stop-and-go traffic patterns increase greenhouse gas emissions and worsen climate change. In addition, the excessive use of non-renewable energy sources increases the pressure on natural resources, undermining global efforts to promote sustainability. Given these impacts, accurately predicting traffic flow has become an essential tool for urban transportation planning. Effective traffic forecasting allows cities to optimize transportation infrastructure, reduce congestion and increase the overall efficiency of the transportation system. From an economic perspective, traffic congestion can cause economic losses by causing labor loss. Traffic forecasting informs in real time to improve daily driving experiences and helps make better decisions about routes and planning. Predicting traffic flow effectively is a critical step for properly planning traffic regulations and making journeys more efficient, and it is also a cornerstone of sustainable urban development. As a result, traffic forecasting not only makes daily life easier, but is also an important tool for supporting cities' environmental sustainability, increasing economic efficiency, and improving quality of life.Today, traffic congestion not only causes time loss and stress for individuals, but also brings about large-scale problems such as environmental impacts and economic losses. Traffic increases the pressure on the environment by causing air pollution, noise pollution and unnecessary fossil fuel use.

Long-term vehicle idling (in motor vehicles, setting the engine to work properly with minimal fuel while the vehicle is stationary) and stop-and-go traffic patterns contribute to increased greenhouse gasemissions, worsening climate change. Effective traffic forecasting allows cities to optimize transportation infrastructure, reduce congestion and increase the overall efficiency of the transportation system. From an economic perspective, traffic congestion can cause economic losses by causing labor loss. Additionally, traffic forecasting improves drivers' daily experiences by providing drivers with real-time information about traffic conditions and helps them make better route and scheduling decisions. For these reasons, effectively predicting traffic flow is a critical step in optimizing cities' transportation infrastructure, planning traffic regulations accurately, and making drivers' journeys more efficient, as well as being a cornerstone of sustainable urban development. Traffic forecasting enables the development of emergency response and traffic flow relief strategies in the short term, while shaping sustainable transportation policies in cities in the long term. With effective forecasting methods and strategies, cities can pave the way for greener, smarter, and more livable urban environments that benefit both current and future generations. As a result, traffic forecasting not only makes daily life easier, but is also an important tool for supporting cities' environmental sustainability, increasing economic efficiency, and improving quality of life.Today, traffic congestion not only causes time loss and stress for individuals, but also brings about large-scale problems such as environmental impacts and economic losses. Traffic increases the pressure on the environment by causing air pollution, noise pollution and unnecessary fossil fuel use. Long-term vehicle idling and stop-and-go traffic patterns contribute to increased greenhouse gas emissions, worsening climate change. Noise pollution from congested traffic further reduces urban livability, affecting public health and well-being. In addition, the excessive use of non-renewable energy sources increases the pressure on natural resources, undermining global efforts to promote sustainability. Given these impacts, accurately predicting traffic flow has become an essential tool for urban transportation planning. Effective traffic forecasting allows cities to optimize transportation infrastructure, reduce congestion and increase the overall efficiency of the transportation system. From an economic perspective, traffic congestion causes loss of labor, which can lead to economic losses. In addition, traffic forecasting helps drivers make better route and timing decisions by providing real-time information. For these reasons, effectively predicting traffic flow is a critical step in optimizing cities' transportation infrastructure, planning traffic regulations accurately, and making drivers' journeys more efficient, as well as being a cornerstone of sustainable urban development. Traffic increases the pressure on the environment by causing air pollution, noise pollution and unnecessary usage of fossil fuel. Prolonged vehicle idling and stop-and-go traffic patterns result in increased greenhouse gas emissions, worsening climate change. Noise pollution arising from congested traffic further diminishes urban livability, affecting public health and well-being. Additionally, the overuse of non-renewable energy sources exacerbates the strain on natural resources, undermining global efforts to promote sustainability.

2 Literature review

Wang the article of takes into account the SARIMA-NAR model for traffic flow forecasting and focuses on seasonality. This article examines the SARIMA model for traffic flow prediction. Wang and others showed that hourly data are examined to increase traffic prediction accuracy during periods of significant fluctuations caused by external factors. it effectively integrated linear and nonlinear components to achieve higher prediction accuracy. Even with limited data, SARIMA has shown successful results, (especially by developing a model that can make a 24-hour forecast using only three days of data). The results show that the MAPE values are within acceptable limits for most applications. [1]

The study by Kumar and Vanajakshi investigated how SARIMA model can be applied for traffic flow prediction. The study examined a three-lane road in India and the model was developed with limited traffic data from three consecutive days. While developing the SARIMA model, the researchers determined the appropriate model parameters by applying the necessary processes to stationarize the data. [2]

The article Traffic Congestion Prediction Using Machine Learning Techniques" by Moumita Asad. They used a machine learning model that takes into account factors such as weather, day, and time to predict traffic congestion. This model provided successful results with an average RMSE value of 1.12 for one-week forecasts. This study highlights the importance of environmental factors such as weather conditions in traffic predictions. [3]

In Deep Learning with Applications Using Python by Navin Kumar Manaswi, Rnn and Lstm introduced as a key architecture. Rnn is the artificial neural network is created as sequential data by maintaining a hidden state. It depends on previous step so this feedback provide temporal dependencies in data. However, traditional Rnns is not good at long term because of vanishing gradient problem. Lstm is formed for solving the this problem. It has cells and gate to arrange the flow of information. There are 2 parts: remember for important information in long periods and forgetting for irrelevant details. It is used effectively in time series forecasting, speech recognition and text processing. [4]

RNNs are branch of artificial neural networks. Early RNNs suffered from the vanishing gradient problem. They have limitations to learn long-range dependencies. This was solved by the long short term memory (LSTM) variant in 1997. As a history, in model before, in 1980s, with foundational work by John Hopfield and David Rumelhart. To solve this vanishing gradient problem, advanced architectures is designed such as Long Short-Term Memory (LSTM) networks and Gated Recurrent Units (GRUs). [5]

In machine learning the vanishing gradient problem is encountered when training neural networks with gradient-based learning methods and backpropogation. In such methods, during each training iteration, each neural network weight receives an update proportional to the partial derivative of the loss with respect to the current weight. The problem is that as the network depth increases, the gradient magnitude typically is expected to decrease. [6]

3 Data and Time-Series Analysis

The aim of this study is to apply the most appropriate model to predict the traffic density more accurately in the city. Traffic data contains complex models that show seasonal changes and trends over time. In our data set is consisting of DateTime, Junction, Vehicles and ID.

Traffic Data Table

Date	Vehicles
2015-11-01	596
2015-11-02	999
2015-11-03	911
2015-11-04	818
2015-11-05	770
2015-11-06	746
2015-11-07	569
2015-11-08	591
2015-11-09	772
2015-11-10	1046
2015-11-11	943
2015-11-12	839
2015-11-13	606
2015-11-14	672
2015-11-15	825
2015-11-16	976
2015-11-17	1260
2015-11-18	1281
2015-11-19	1196
2015-11-20	847
2015-11-21	854
2015-11-22	1031
2015-11-23	1041
2015-11-24	930
2015-11-25	840

3.1 SARIMA Model

The SARIMA model predicts future trends based on historical data of traffic density in a given time period, taking into account seasonality in time series data. The data usually show seasonal changes due to various factors affecting traffic flow.

SARIMA Model Components

- **AutoRegressive (AR):** Estimates the current value using values from p

past time steps. Here, p represents the autoregression order.

- **Integrated (I):** Makes the time series stationary by using the differencing process. The parameter d represents the order of differencing.
- **Moving Average (MA):** Accounts for the effects of errors in the model, representing them as a series of lagged error terms. The parameter q represents the order of the moving average.
- **S:** Adds seasonality to the ARIMA model.
  - **P:** Number of seasonal autoregressive terms.
  - **D:** Degree of seasonal differencing.
  - **Q:** Number of seasonal moving average terms.
  - **S:** Refers to the length of the seasonal cycle.

SARIMA Model Order Selection For order selection, we need to check the stationarity of the time series. So we first check the stationarity, make the series stationary with ADF test and differencing operations and as a result we find (d, D). We then analyze the ACF and PACF graphs to find the p and q values: ACF is related between the different lags and from this graph we analyze the q value; PACF is related to the autocorrelation of a particular lag and from this graph we analyze the p value. It is necessary to determine the seasonal cycle length for s.

4 Model Forecasting

With the SARIMA model, forward-looking forecasts can be made based on historical data. When the correct sequence selection is made, the SARIMA model gives effective results in predictions. Therefore, Seasonal Autoregressive Integrated Moving Average (SARIMA) model was selected to accurately model the seasonal and non-seasonal components of the data. In the modeling process, firstly, seasonality, trends and random fluctuations in the data were analyzed. Before applying the SARIMA model, the autocorrelation and seasonal order properties of the data were examined and the parameters of the model were determined. In the modeling phase, some stationarity tests (such as ADF test)were performed on the data. The data were made stationary. Finally, the prediction performance of the model was evaluated with metrics such as RMSE and MAE and the model that gave the best performance was determined. This study applies time series analysis methods to traffic prediction problems.

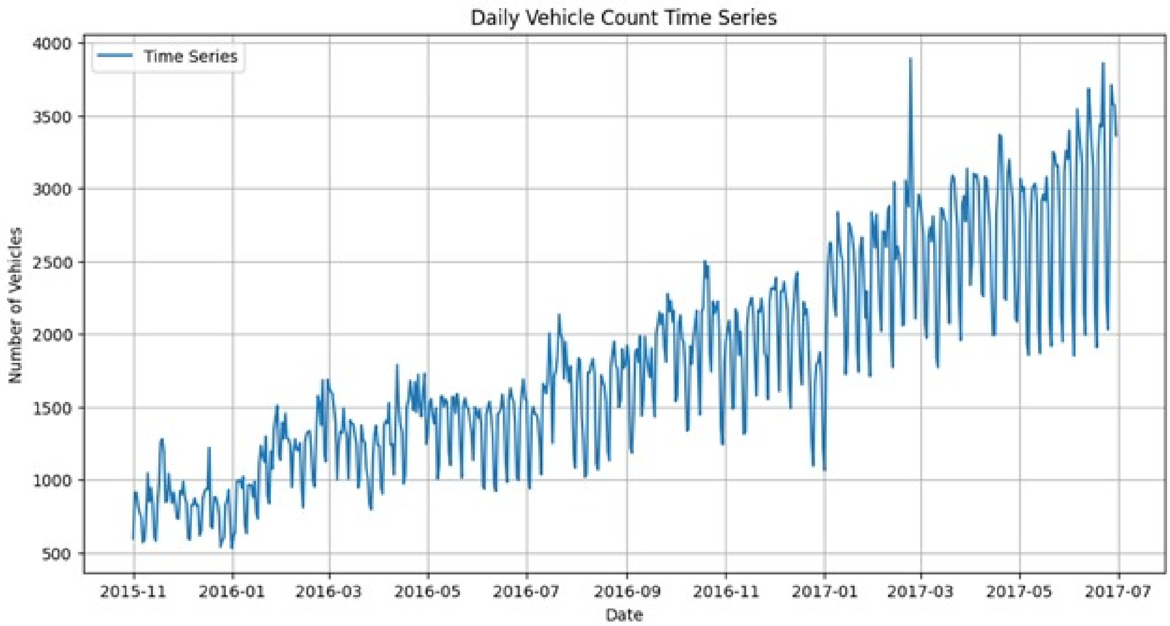

Figure 1: A time series on daily traffic from 2015 to 2017.

Separating the time series into its different components helps to better understand the time series. In this step, the time series is separated into trend, seasonality and residual components. The seasonal decompose function was used for this separation:

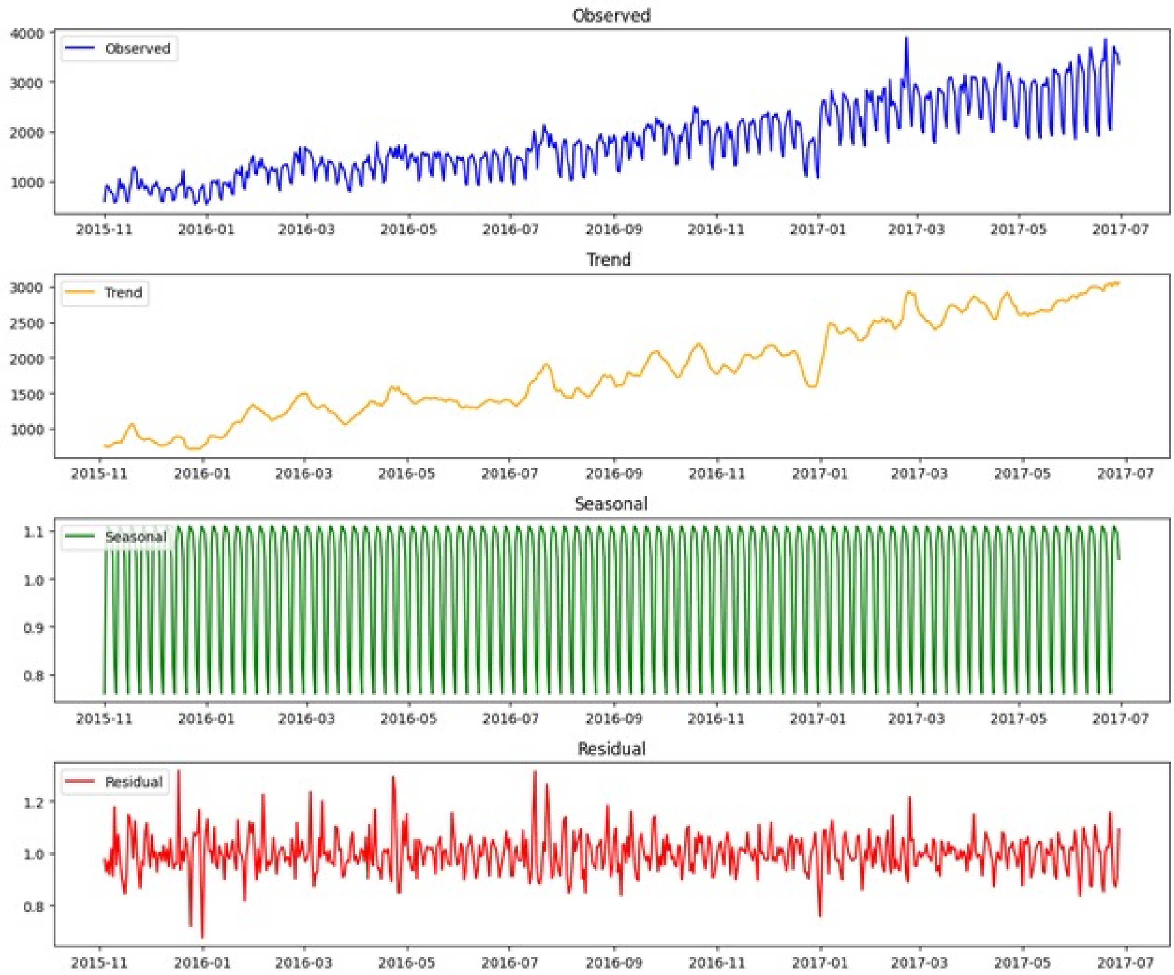

Figure 2:

Trend: A general trend that increases or decreases over time. Seasonality: Repeated fluctuations during certain periods of the year (for example, differences in precipitation in winter and summer). Residual: A component that changes randomly, regardless of trend and seasonality. Determining whether the time series is stationary is important for prediction. ADF (Augmented Dickey-Fuller) test is used to test the stationary of time series. If the P value is less than 0.05, the time series is constant. Otherwise, the time series are not stationary and need to be transformed.

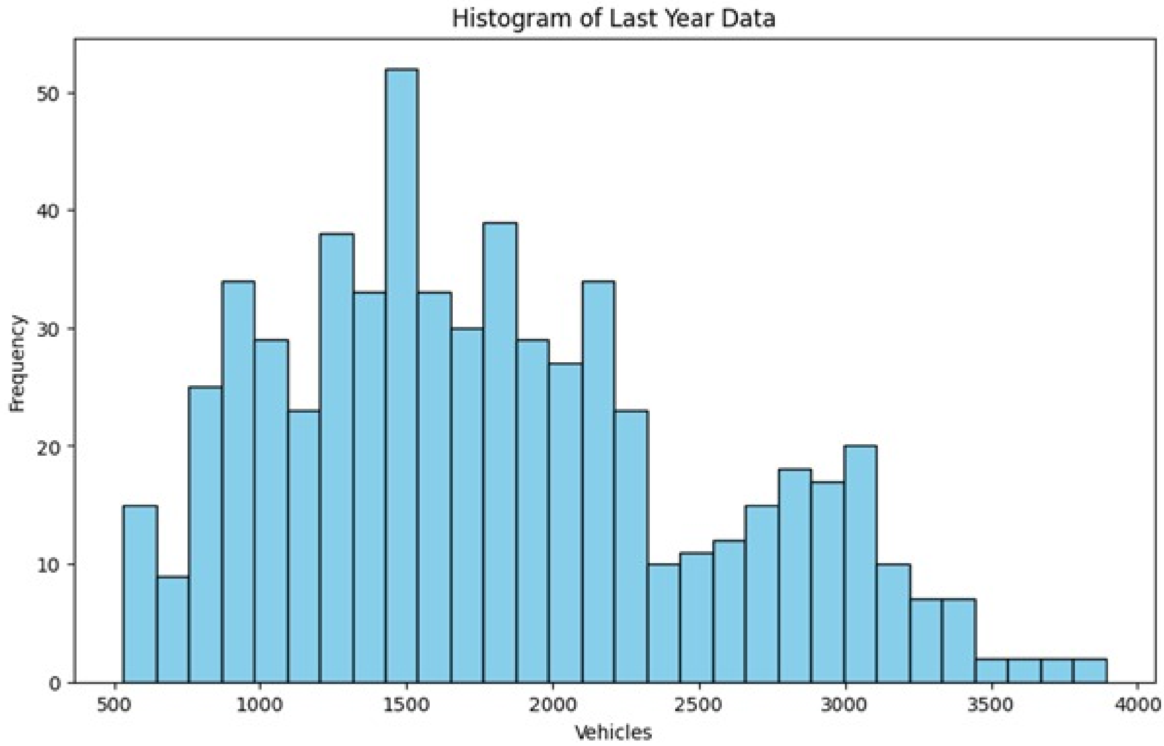

Figure 3:

ADF test is performed as follows:

- **p-value**: 0.9504831043579367. this means, the time series is not stationary (p $\geq$ 0.05).

The mean and variance values of each chunk are as follows:

Chunk	1	2	3	4	5	6	7	8	9	10
Mean	828.72	1131.48	1320.92	1388.18	1576.51	1879.59	1942.48	2470.75	2654.57	2876.48
Variance	28462.07	71836.75	54130.11	46187.18	90384.29	100339.91	109907.32	204597.26	171389.27	330827.54

Table 1: Mean and Variance Values According to Chunks

Our time series does not have a stationary structure and various tests are used to detect this. Histogram plot graphics and ADF (Augmented Dickey-Fuller) tests clearly show the status of our data. It is very clear that they do not have constant mean and constant variance values in the chunk analysis applied on the histogram plot. For this reason, we first need to apply log transformation to our time series, this transformation helps us convert the non-constant variance in our time series into a constant variance. The next method is the difference method, thanks to this method, it helps us to make our time series stationary by eliminating the trend and seasonal components in our time series. After these operations, we test the stationarity of our time series by applying ADF test and histogram chunk analysis to our time series again. If it is still not stable, we try to make our time series stationary with different methods. If our time series is stable, we can continue the modeling process. Stability results of log transformed time series:

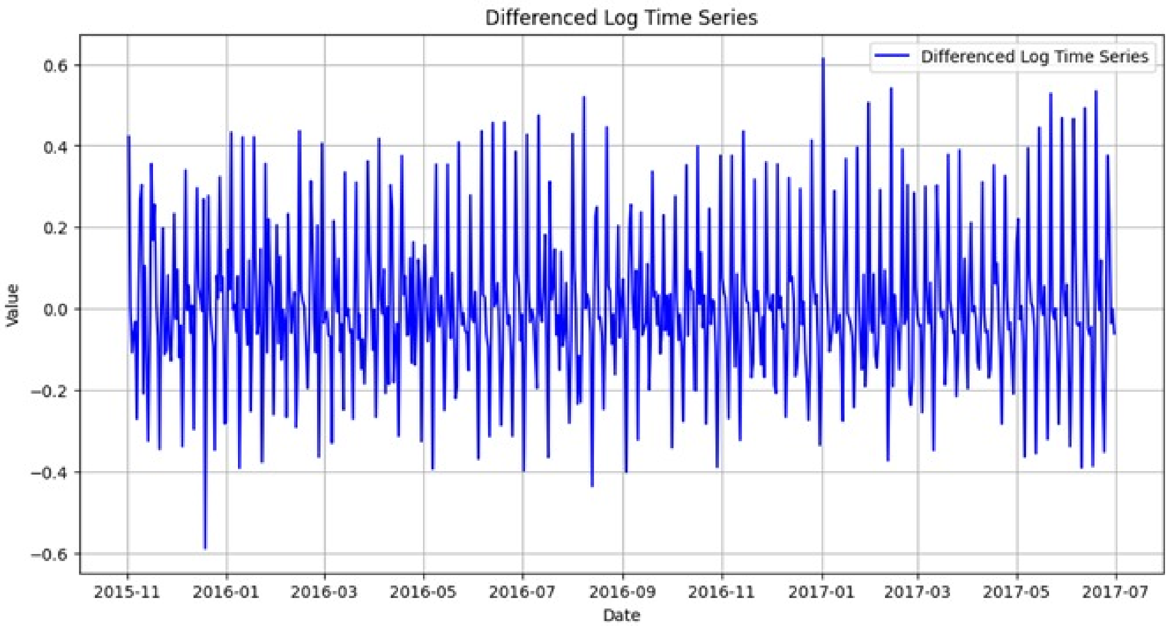

Figure 4: Differenced Log Time Series

ADF test is performed as follows:

- **p-value**: 6.643480614187697e-16. The time series is stationary ($p \geq 0.05$).

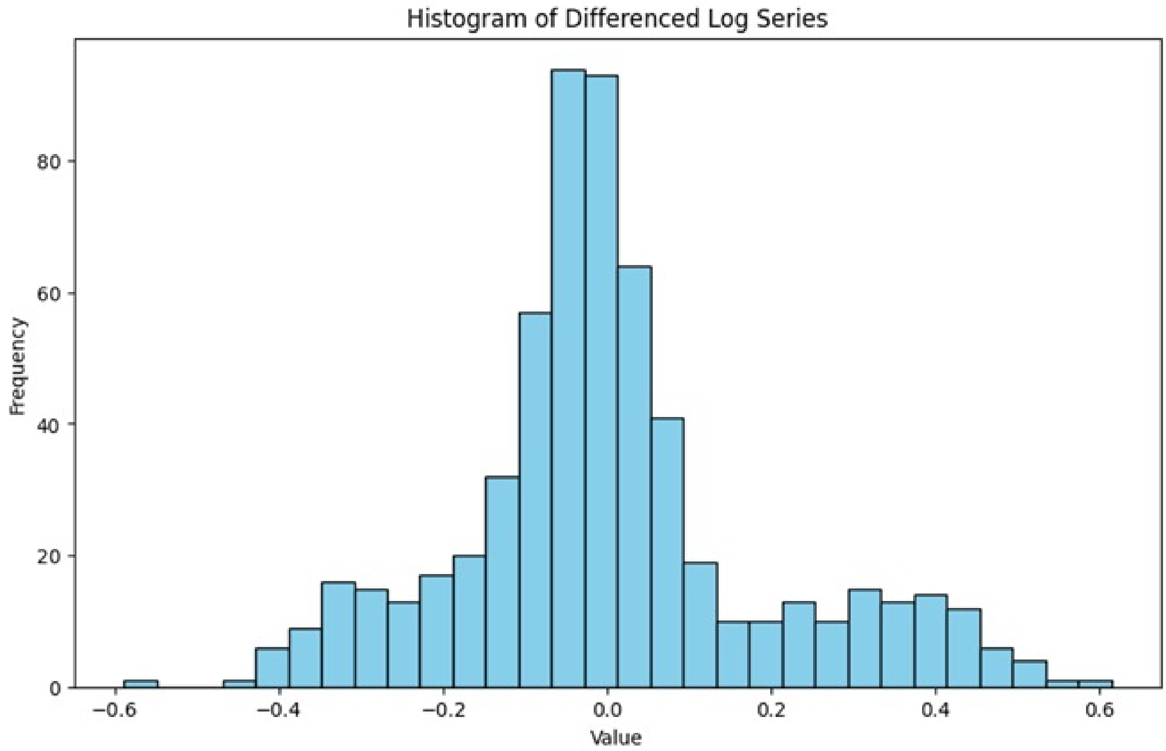

Figure 5: Histogram

Chunk	Mean	Variance
1	-0.00	0.04
2	0.02	0.04
3	-0.00	0.03
4	-0.01	0.04
5	0.01	0.04
6	0.00	0.03
7	-0.01	0.03
8	0.02	0.04
9	-0.00	0.03
10	0.01	0.05

Table 2: Mean and Variance for Each Chunk

When we look at the time series analysis results, there are now constant mean and constant variance values. This tells us that our time series has eliminated previous fluctuations and has become more consistent, and indicates that our time series is ready for the model. In order for our model to produce more accurate and consistent results during the prediction process, we divide our time series into two parts as train and test. The train part consists of 70In order to be able to do forcasting, we need to determine our SARIMA parameters. In these parameters, the PACF graph is looked at and the first delay point is taken into account to determine the p(AR) parameter, the AFC graph is looked at and the first significant delay is taken into account to determine the q(MA) parameter.

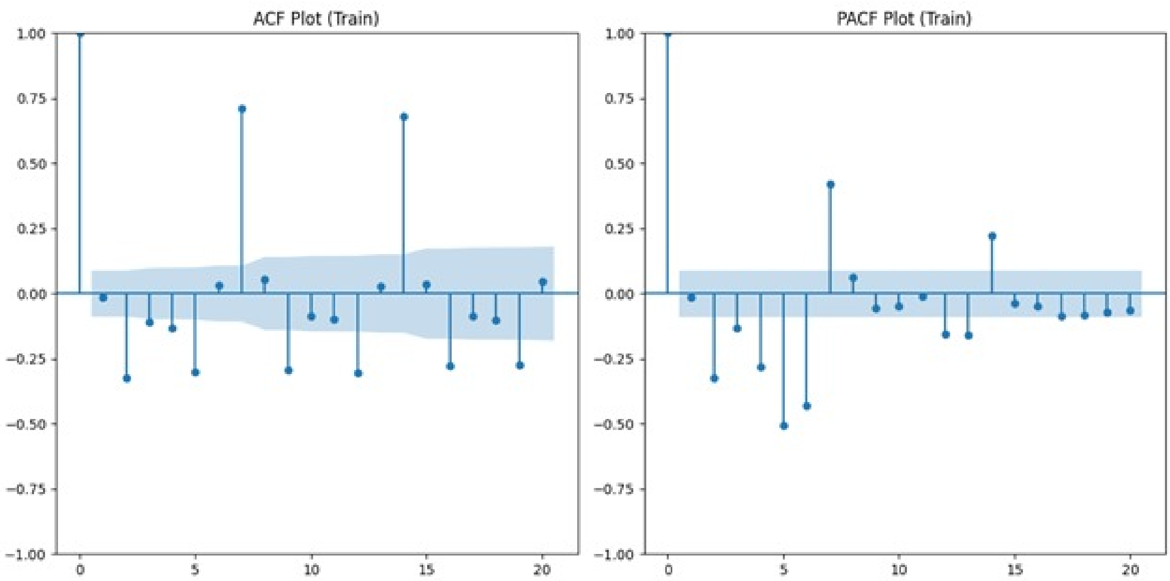

Figure 6: According to our graphs, our p=3 and q=1

Other parameters can be determined by using Grid Search, Random Search and ACF and PACF Analysis methods. Grid Search method was used to determine the parameters of this model. Grid Search performs a performance analysis by systematically trying the parameter settings and is a method that aims to find the parameter combination that gives the best results. In the SARIMA model, it determines the most appropriate values of the p,d,q,P,D,Q,s parameters.

5 Forecasting Results

The parameters used in the forcast were determined with Grid search and good parameters were estimated with SARIMA$(1, 0, 1, 1, 0, 1, 7)$. The most suitable parameters for our model are as follows: $p(1)$: This is the AR degree and indicates the number of delayed observations included in the model. $d(0)$: This is the difference degree and is used to make the model stationary. The reason for d=0 is that our model was made stationary in the pre-processing. $q(1)$: This is the (MA) degree and indicates the number of delayed estimation errors included in the model. $P(1)$: This is the seasonal autoregressive (SAR) degree and indicates the number of seasonally delayed observations included in the model. $D(0)$: This is the seasonal difference degree and indicates how many times seasonal difference is taken to make the data seasonally stationary. $Q(1)$: This is the seasonal moving average (SMA) degree and indicates the number of seasonally delayed estimation errors included in the model. $s(7)$:This is the seasonal period and expresses the length of the seasonal cycle. Our forcasting, which is the result of these parameters, is combined in the graph.

The SARIMA model's forecast performance is given in the graph and it is seen that it makes quite successful forecasts on the graph. A detailed evaluation of the model's forecast performance is given below:

Capturing the General Trend: The SARIMA model effectively learned the general trend of the time series and successfully reflected it in the forecasts. The increasing and decreasing directions of the data were modeled correctly. Following Seasonal Patterns: The model learned the seasonal patterns in the data strongly and preserved this structure in the forecasts. It is seen that seasonal fluctuations are clearly reflected in the forecasted values. Deviations and

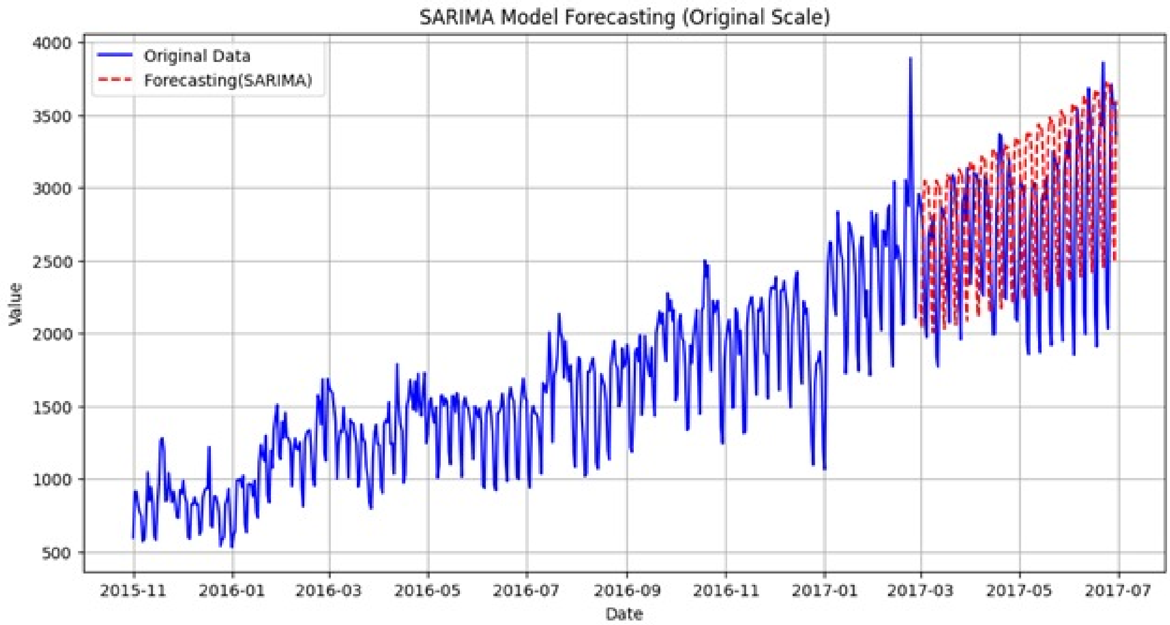

Figure 7: SARIMA MODEL FORECASTING

Differences: Some differences were determined between the actual values and the expected values, especially for sudden fluctuations. However, when these differences are not that serious, they do not negatively affect the forecast performance. Performance Measures The forecast performance of the SARIMA model was evaluated based on the following metrics: Root Mean Square Error (RMSE): The forecast error of the model was calculated as 853,400. This value expresses the average deviation between the estimated and actual values. Seasonal and Trend Performance: The model effectively learned both the general trend and seasonal characteristics and performed well in predictions.

Parameter	Coefficient	Std. Error	z	P>|z|	[0.025	0.975]
ar.L1	0.7222	0.034	21.075	0.0	0.655	0.789
ma.L1	-0.9707	0.015	-63.287	0.0	-1.001	-0.941
ar.S.L7	0.9998	0.001	1879.763	0.0	0.998	1.001
ma.S.L7	-0.9751	0.041	-23.887	0.0	-1.055	-0.895
sigma2	0.0091	0.001	17.847	0.0	0.008	0.01

Figure 8: The table summarizes the statistical parameters of the SARIMA model

- **ar.L1 (Autoregressive Value):**

  Coefficient: 0.7222. This coefficient shows that the current value is well affected by the previous value.

- **ma.L1 (Moving Average Value):**

  Coefficient: -0.9707. A negative coefficient indicates that forecast errors are corrective to the current forecast.

- **ar.S.L7 (Seasonal Autoregressive Value):**

  Coefficient: 0.9998. A coefficient very close to 1 indicates that the weekly seasonal pattern has a strong and consistent effect on the forecasts.

- **ma.S.L7 (Seasonal Moving Average Value):**

  Coefficient: -0.9751. This term reflects how seasonal forecast errors make a corrective contribution to the model.

- **sigma2 (Error Variance):**

  Coefficient: 0.0091. This small value indicates that the residual variance is very low, demonstrating the model's high accuracy in capturing the variability of the data.

- **Confidence Intervals (95%):**

  The confidence intervals ([0.025 - 0.975]) for all coefficients are narrow, indicating a high level of precision in the parameter estimates. For instance:

  - **ar.L1:** Confidence Interval [0.655, 0.789]
  - **ar.S.L7:** Confidence Interval [0.998, 1.001]
  - These narrow intervals confirm that the parameters are well-defined and reliable.

A recurrent neural network (RNN) is a type of artificial neural network that is best suited to recognizing patterns in sequences of data. An RNN is an extremely powerful algorithm that can classify, cluster, and make predictions about data, particularly time series and text. It has 3 layers which are input layer, a hidden layer and an output layer. The building block of RNNs is the recurrent unit. This unit maintains a hidden state, essentially a form of memory, which is updated at each time step based on the current input and the previous hidden state. This feedback loop allows the network to learn from past inputs, and incorporate that knowledge into its current processing. Early RNNs suffered from the vanishing gradient problem, limiting their ability to learn long-range dependencies. This was solved by the long short-term memory (LSTM) variant in 1997, thus making it the standard architecture for RNN.

Long Short-Term Memory (LSTM) is an artificial recurrent neural network (RNN) architecture used in the field of deep learning. Unlike standard feedforward neural networks, LSTMs have feedback connections. They can process not only single data points (such as images) but also entire sequences of data (such as speech or videos). LSTM recurrent unit tries to "remember" all the past knowledge that the network is seen so far and to "forget" irrelevant data. This is done by introducing different activation function layers called "gates" for different purposes. Each LSTM recurrent unit A standard LSTM unit consists of a cell, an input gate, an output gate, and a forget gate. The Internal Cell State conceptually describes the information that was chosen to be retained by the previous LSTM recurrent unit. The cell remembers values over arbitrary time intervals, and the three gates regulate the flow of information into and out of the cell. LSTM networks are highly suitable for classifying, processing, and making predictions based on time series data, as they can handle unknown durations of delays between important events in a time sequence. Input gate operates on the same signals as the forget gate, but here the objective is to decide which new information is going to enter the state of LSTM. At output gate, the input and previous state are gated as before to generate another scaling fraction that is combined with the output of tanh block that brings the current state. This output is then given out. The output and state are fed back into the LSTM block. At forget gate the input is combined with the previous output to generate a fraction between 0 and 1, that determines how much of the previous state need to be preserved (or in other words, how much of the state should be forgotten). This output is then multiplied with the previous state. There are many applications areas which are Natural Language Processing (NLP), speech recognition, video analysis, medical diagnosis, fraud detection, self driving cars, navigation systems and time series analysis.

Recurrent Neural Networks (RNNs) are designed to process sequential data by maintaining a hidden state that incorporates information from previous time steps. However, traditional RNNs struggled with learning long-term dependencies due to the vanishing gradient problem. This issue was resolved by the Long Short-Term Memory (LSTM) architecture, which introduced gates to regulate the flow of information. LSTM units consist of a cell state, input gate, forget gate, and output gate, enabling the network to ”remember” relevant information and ”forget” irrelevant details. These features make LSTMs highly effective for tasks such as time series forecasting, speech recognition, and text processing.

In our project, for the last part, we first applied preprocessing to our data. Then we separated our data into training for 80% and testing for 20%. For LSTM, we defined a function and created a sliding window, which is *look_back*.

Here, our aim is to create the LSTM input format. The *look_back* is used for determining the number of past time steps, and if this value is too high, then the model can learn too much. This means that an overfitting problem may occur. If this value is too low, then underfitting can occur. So, for these reasons, we chose the number of *look_back* as 14. After that, we defined the LSTM model. As an optimizer, we used Adam because Adam optimizer will provide us stability and faster results in learning. At the same time, Adam optimizer is used for deep networks due to its adaptability. Then we trained the model. In this part, the RNN has the hidden layer. In this layer, some activation functions are used. There are many activation functions, but we used ReLU (Rectified Linear Unit). In this function, $f(x) = \max(0, x)$. This means that if x is negative, the output will be 0, and if x is positive, the output will be the value of x. So, negative values translate into 0, and positive values remain constant. This ensures that the model learns nonlinear relationships. We chose the number of epochs as 150 because we tried to predict correctly. For every number of epochs, the model learns the relationship between the complex patterns in the data. We performed forecasting and translated it into the original scale. We looked at the RMSE values, and when comparing with SARIMA, the RMSE value decreased in the LSTM model. Lastly, we showed the results. The blue line represents the original data, the green line represents the training predictions, and the red line represents the test predictions.

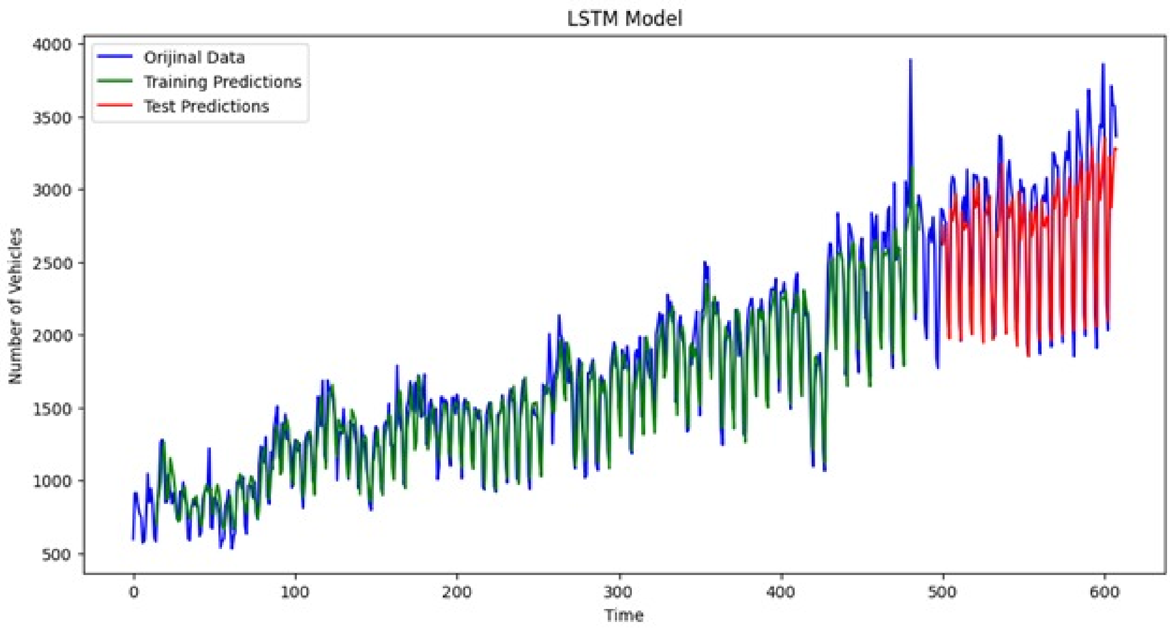

Figure 9: LSTM Model

6 Conclusion

In this study, our aim is to make road traffic prediction. Firslty, we used SARIMA model then for comparing, we used LSTM model that is a branch of RNN. In SARIMA, we modeled the seasonal and trend components of the traffic data. By using grid search, we found our parameters such as seasonal order (P, D, Q, m) and non-seasonal order (p, d, q). In LSTM, we used sliding window and then defined parameters like number of epoch and look back. As a result, in SARIMA we catched the seasonal pattern accurately, but in LSTM, the value of RMSE is less than SARIMA and this provide us more accurate model. When we comparing these two model LSTM is the best for performancing the model. SARIMA is reliable for seasonal time series data and struggled to adapt complex changes in the traffic data, LSTM is especially effective when working with large datasets with complex models. These models, it can help minimize the effects of traffic by making accurate forecasts. LSTM Model uses the approach of dividing the data into sliding windows to predict future values based on past observations. In addition to the sliding window, it has the ability to learn from complex and non-linear patterns in the data. LSTM is quite good at capturing long-term periods.LSTM is particularly effective in adapting to sudden changes or irregular patterns in traffic data that are often challenging.

Comparison and Results

SARIMA: It accurately captured seasonal patterns in the data. That is, it can be reliable for data sets with pronounced seasonality.However, it had difficulty adapting to sudden, complex changes in traffic flow in the data.Its performance was limited in scenarios with sudden spikes or irregular fluctuations in traffic data. **LSTM**: It achieved a lower RMSE compared to sliding window, which means that it performs better in this regard. This means a higher level of accuracy in their predictions. It effectively handled large datasets and captured both seasonal trends and nonlinear changes in traffic data, outperforming sarima in scenarios with complex patterns. Therefore, we can say that it is a better option for long-term and large-scale traffic forecasting tasks.

As a result, both models have advantages depending on the characteristics of the dataset. SARIMA remains a reliable option for datasets with well-defined seasonal components, but is less adaptable to sudden changes or complexities in the data; LSTM provides more accurate predictions in terms of overall accuracy and flexibility, especially in scenarios where the dataset exhibits nonlinear and disordered behavior.

When applied correctly, these models can significantly minimize the effects of traffic congestion by providing accurate forecasts. By providing better traffic management, they contribute to reducing travel delays, improving fuel efficiency and improving the overall quality of urban transport systems.

In this study, our aim is to make road traffic prediction. Firslty, we used SARIMA model then for comparing, we used LSTM model that is a branch of RNN. In SARIMA, we modeled the seasonal and trend components of the traffic data. By using grid search, we found our parameters such as seasonal order (P, D, Q, m) and non-seasonal order (p, d, q). In LSTM, we used sliding window and then defined parameters like number of epoch and look back. As a result, in SARIMA we catched the seasonal pattern accurately, but in LSTM, the value of RMSE is less than SARIMA and this provide us more accurate model. When we comparing these two model LSTM is the best for performancing the model. SARIMA is reliable for seasonal time series data and struggled to adapt complex changes in the traffic data, LSTM is especially effective when working with large datasets with complex models. These models, it can help minimize the effects of traffic by making accurate forecasts.

References

[1] You Wang, Ruxue Jia, Fang Dai, and Yunxia Ye, *Traffic Flow Prediction Method Based on Seasonal Characteristics and SARIMA-NAR Model*, 2022.

[2] S. V. Kumar and L. Vanajakshi, *Short-term traffic flow prediction using seasonal ARIMA model with limited input data*, European Transport Research Review, 2015.

[3] M. Asad, R. M. Yasir, N. Nower, and M. Shoyaib, *Traffic Congestion Prediction Using Machine Learning Techniques*, 2022.

[4] N. K. Manaswi, *Deep Learning with Applications Using Python*, Springer, 2020.

[5] "Recurrent Neural Network," *Wikipedia*, [Online]. Available: `https://en.wikipedia.org/wiki/Recurrent_neural_network`. [Accessed: 25-Jan-2025].

[6] "Vanishing Gradient Problem," *Wikipedia*, [Online]. Available: `https://en.wikipedia.org/wiki/Vanishing_gradient_problem`. [Accessed: 25-Jan-2025].