



BM20A6100 Advanced Data Analytics and Machine Learning

**Forecasting Traffic Based on the Weather**  
Project Work, Level A - Final Submission

18.12.2025  
Ina Laurila  
Samuel Ojala  
Hanna Vetikko

## 1. Introduction

The goal of this project is to forecast hourly traffic volume using past data on traffic and weather conditions. We use real traffic data from Interstate 94 in the Minneapolis–St. Paul area.

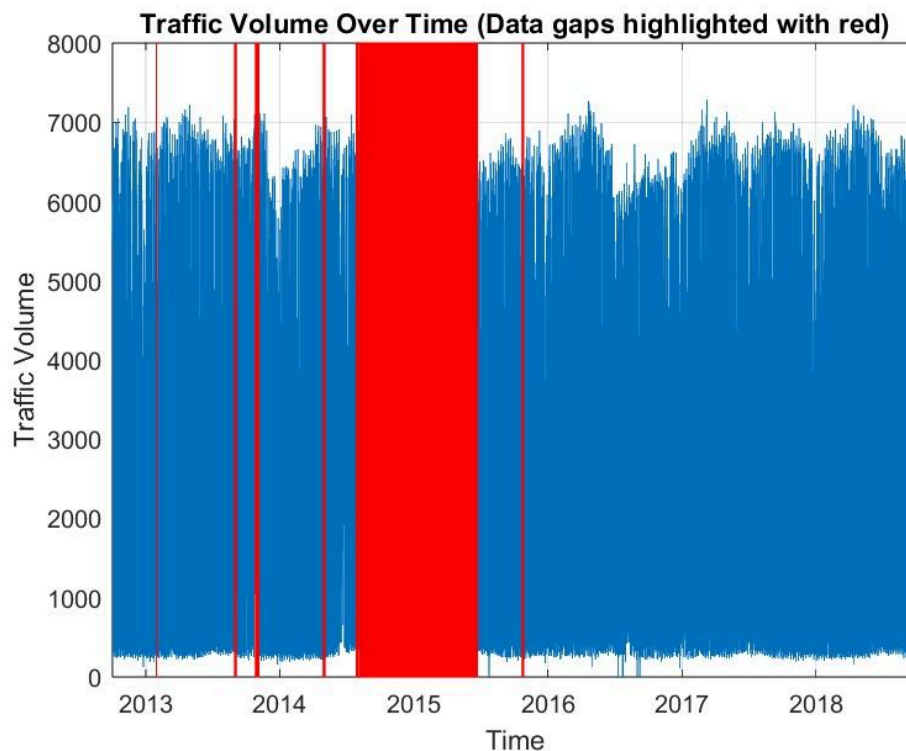
## 2. Data analysis

The dataset includes hourly measurements of westbound traffic on Interstate 94 in Minnesota, collected by the Department of Transportation at station 301 between Minneapolis and St. Paul. It also contains hourly weather information, such as temperature, rain, snow, and visibility, as well as data on holidays that may affect traffic levels. There are 9 features in total, and 5 of them are numeric.

### 2.1 Explanatory data analysis and visualization

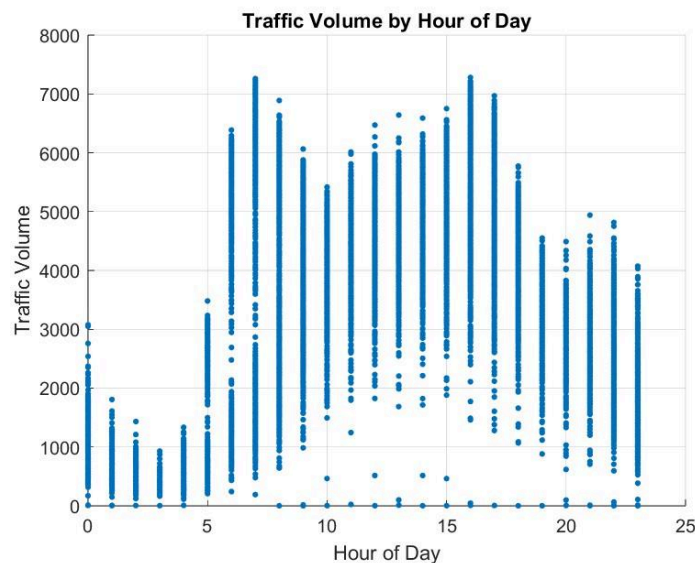
Traffic volume seems quite similar over time. Lowest values seem to remain at a constant level, whereas the highest values vary more but do not show any clear behavior or exceptions.

There are some timeframes with no data, most notably from the 9th of August 2014 until the 11th of June 2015. Missing data periods are highlighted in red in Figure 1. There are also some duplicate samples for the same datetimes. These will require attention when resampling or imputing data.



**Figure 1.** Data visualized as traffic volume over time

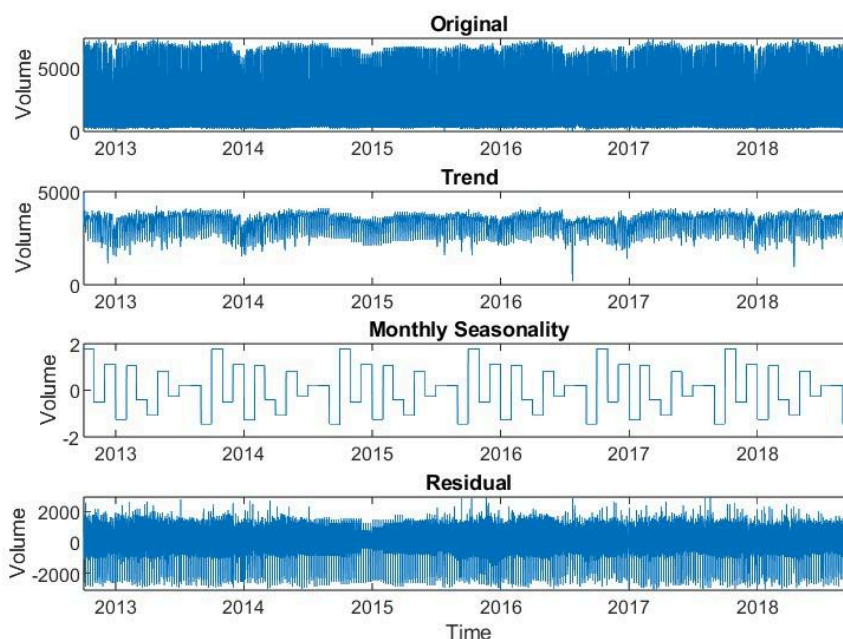
Hourly patterns reveal typical commuting peaks around 7–9 AM and 4–6 PM, and very low traffic volumes during nighttime hours. This confirms a strong daily seasonality in the dataset.



**Figure 2:** Data visualized as traffic volume by hour of day

## 2.2 Time-series decomposition analysis

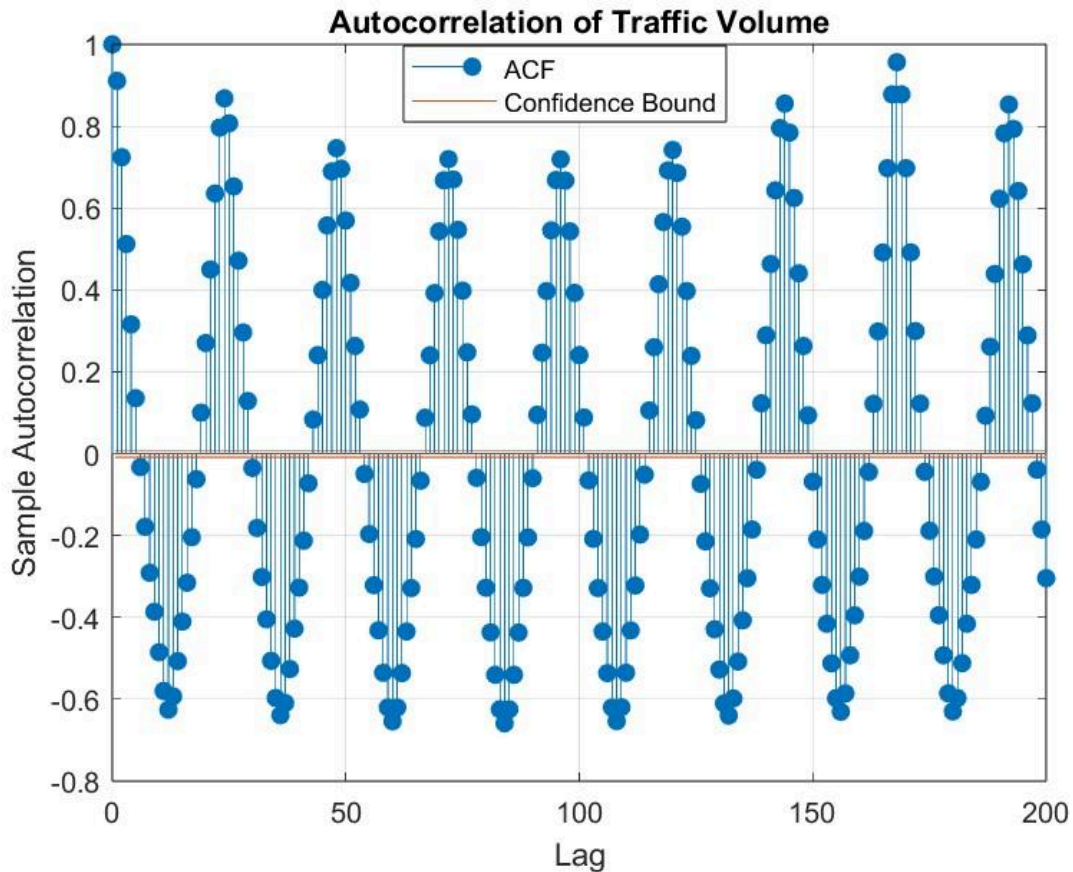
The decomposition separates the overall traffic pattern into three parts, which are visualised in Figure 3. The trend remains mostly stable over time. The seasonal component captures repeating monthly patterns in the data, but there are also daily and weekly seasonal components. The residual shows short-term variation and noise after removing trend and seasonality. Missing data periods are still visible as gaps in all components. The daily seasonal component varies roughly between -100 and +100, indicating that traffic volume typically fluctuates by about 200 vehicles.



**Figure 3:** Time-series decomposition of long-term trend, seasonality, and residuals

## 2.3 Autocorrelation analysis of the dataset

As can be seen in Figure 4, the traffic volume has high autocorrelation at lag 1, meaning the previous hour is a good indicator for the next hour's traffic volume. Peaks can be seen every 12 lags, which indicates that the night and day cycle is a good indicator of traffic volume. The biggest peak is present at 168 lag ( $7 \times 24$ ), indicating the weekly cycle is also a good indicator of traffic volume.



**Figure 4:** Autocorrelation of traffic volume.

## 2.4 Time-series data partition

The time-series data will be split chronologically into train and test sets (70% / 30%). The training data will be used to build and tune the forecasting model, and the testing data will be used for final evaluation on unseen observations. No separate validation set was used; instead, training progress was monitored via training loss, and final performance was evaluated on the held-out test set.

### **3. Methodology**

#### **3.1 Data Pretreatment**

The measurements are by hour, but there are no data points for each hour continuously, so interpolation is needed to fill them in. While simple linear interpolation could be used, we used seasonal data to interpolate values by averaging the values recorded on the same hour of the day, week, month, or year ago. This is especially important for the large gaps in the data where simple linear interpolation has nothing to draw on.

As all the variables in the data share the same timestamps, and because all the datapoints have values in their variables, there are no missing values in the data. The data fulfills the definition of synchronous data.

The time-series STL decomposition can be used for outlier detection. The residuals plot had detrended and deseasonalized the data so that data points that deviate greatly from the median can be considered outliers. For this project, only the most extreme outliers, such as those near zero, are removed as they are likely false. Other values that deviate strongly from the median are kept as they likely represent real traffic behaviour, such as holidays, which may still distort short-term predictions. The dataset also had duplicate samples for the same timestamps, in which case the duplicates were removed.

##### **3.2.1. Sub-sequencing and seasonality**

Long time-series can be divided into shorter sequences using fixed-length sliding windows or by segmenting the data based on local stationarity or cyclical patterns (e.g., daily, weekly, yearly). (Shengsheng et al., 2022; Silva et al., 2021)

Seasonality strongly affects subsequence design: window lengths should cover at least one full seasonal cycle to capture recurring patterns. STL decomposition can also be used to create meaningful sub-sequences by separating the series into trend, seasonal, and residual components, each modeled independently with LSTM (Chen et al., 2020).

Multiple studies show that removing or modeling trend and seasonality before LSTM improves forecasting accuracy (Chen et al., 2020; Rehman et al., 2023). The STL-LSTM hybrid approach achieved significantly lower errors than a raw-data LSTM by predicting trend, seasonal, and residual components separately (Chen et al., 2020).

### **3.2.2. Standardisation methods**

Standardisation is an important preprocessing step for neural network models, as it ensures that input features are on comparable scales. Common standardization methods include Z-score and min-max scaling. The standardization method should be chosen based on the properties of the data like distribution and existence of outliers. It is also important to only use training data for standardization to prevent data leakage. Min-max scaling is easily affected by the data as it may take defining values from an outlier.

For example, Mahesha et al. (2024) propose a combined normalisation strategy for LSTM-based time-series forecasting using min-max scaling, z-score standardisation, and max-normalisation. Similarly, Rehman et al. (2023) standardise the input data using both min-max normalisation and z-score standardisation to ensure comparable feature scales.

In this project, all numerical input features were standardised using z-score standardisation based on the training set.

## 4. Results

Four different models are implemented for the forecasting task. Autoregressive Model (AR), Recurrent Neural Network Model (RNN), Long Short-Term Memory Model (LSTM), and Multivariate Forecasting Transformer Model (MVFT).

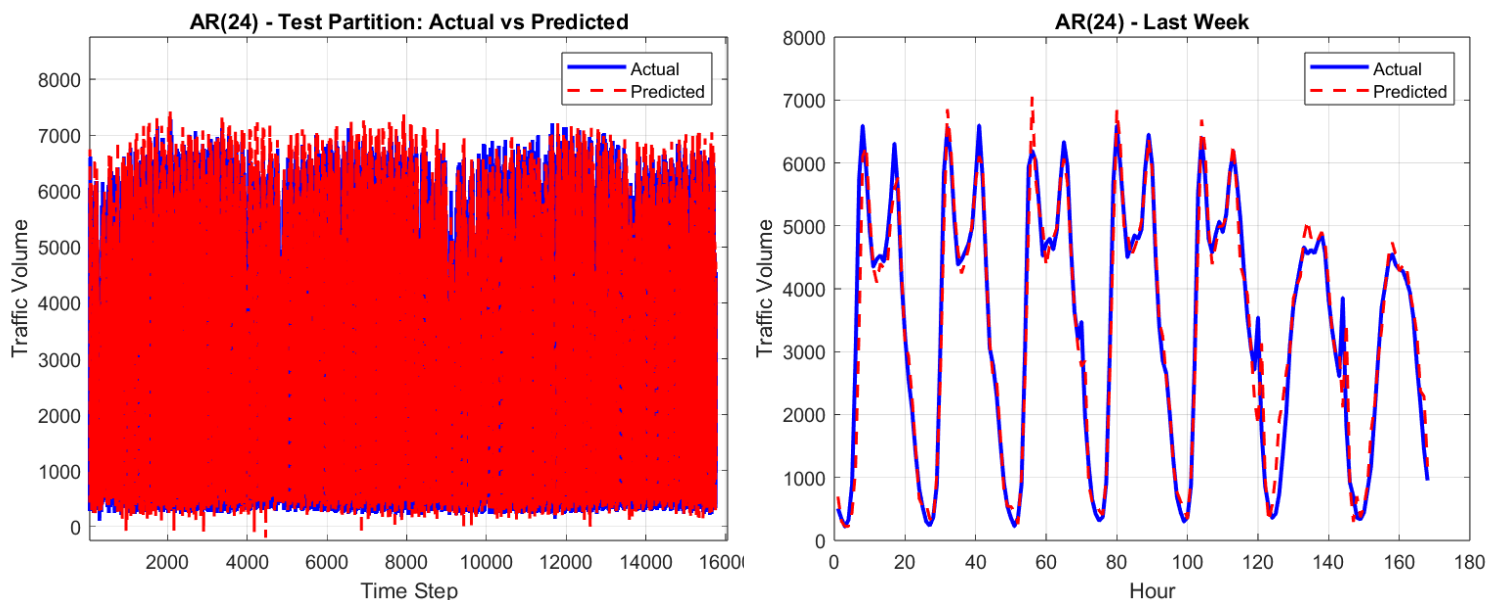
**Table 1:** Model Performance Metrics Comparison

Metric	AR	RNN	LSTM	MVFT
MSE	202504.01	64637.61	90736.69	69443.35
RMSE	450.00	254.24	301.23	263.52
MAE	301.18	165.26	181.71	172.77
R-squared	0.948	0.984	0.977	0.982

The performance of each model is analysed more deeply in chapters 4.1-4.4.

### 4.1 Baseline Model

To compare the other models, an autoregressive model is created as the baseline model, which considers the last day or rather the last 24 hours and is used to forecast traffic for the next hour for the entire test set as seen in figure 5. The baseline model is already able to capture the strong weekly and daily patterns.



**Figure 5:** Autoregressive Basemodel Predictions on Test Set

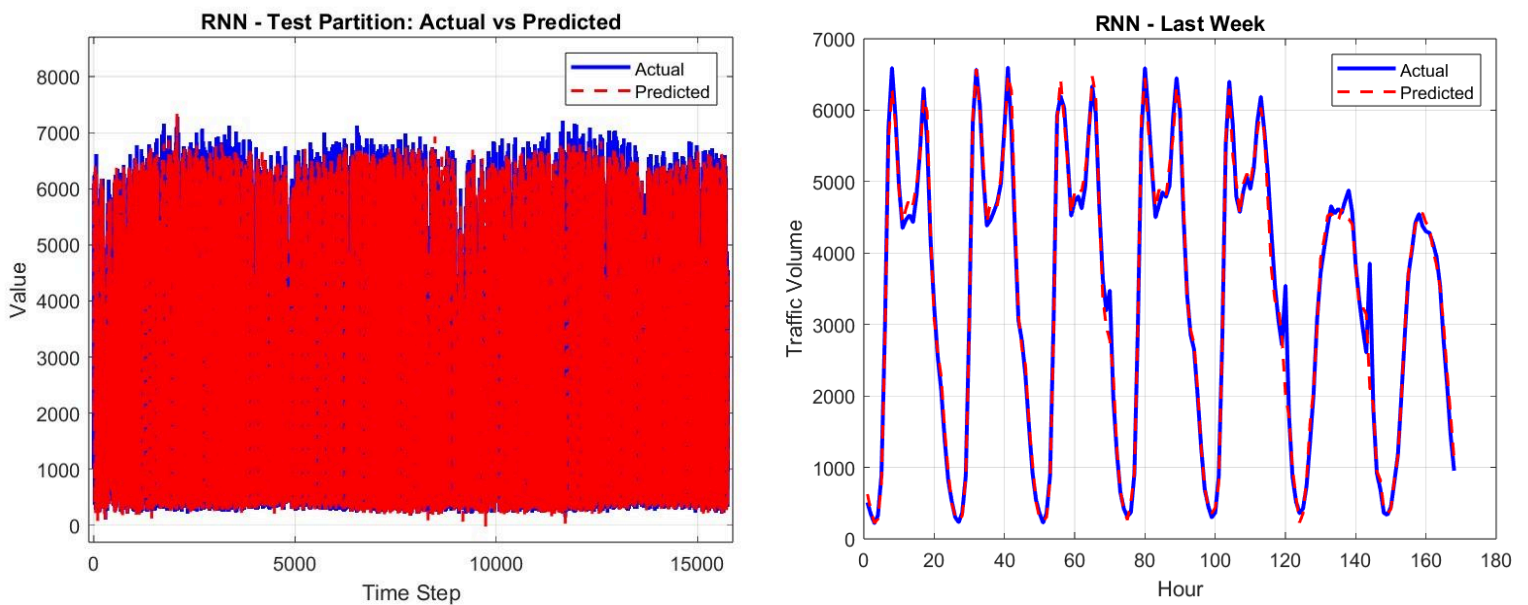
### 4.2.1 RNN Model Implementation

A Recurrent Neural Network (RNN) model predicts the next hour's traffic using the knowledge of the previous hour's traffic and this process is repeated iteratively across the full test set. The resulting predictions from using a model with 30 epochs of training are shown in Figure 6.

### 4.2.2 Model Results

The RNN predictions fit the actual data very well according to the performance metrics in Table 1. The predicted results are plotted against the actual data in Figure 6, showing that the clearest error is in predicting the high values. The predicted values follow a similar pattern as the actual values but are significantly lower. There are also minimal values that are predicted lower than the actual.

The resulting performance is uncharacteristically high for a simple RNN model, this may result from the utilization of the time of hour and weekday as additional features for the model which may help it recognize daily and weekly patterns without being able to observe more than one recorded data point.



**Figure 6:** RNN Model Model Predictions on Test Set

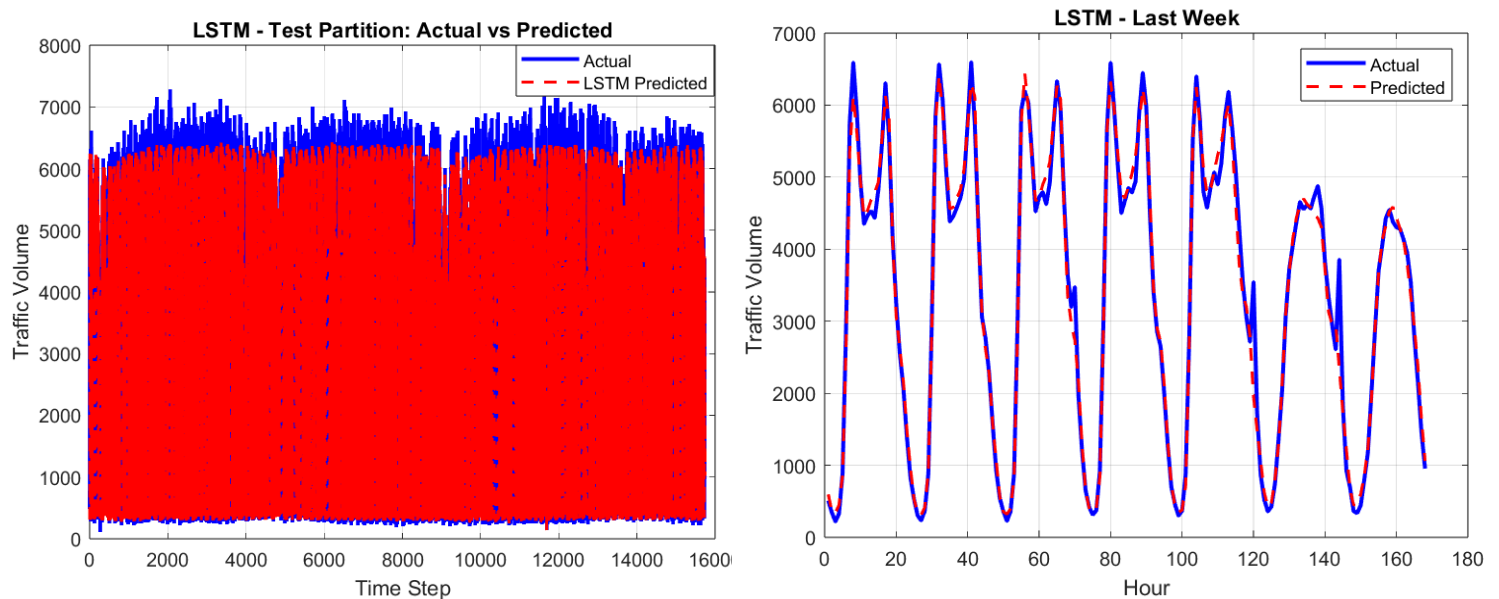
### 4.3.1 LSTM Model Implementation

The Long Short-Term Memory (LSTM) model was constructed for comparison to the regular RNN model. LSTMs are explicitly designed to retain information over longer time spans, making them well-suited for capturing seasonal patterns present in time series data. The lookback window was limited to 24 hours due to computational complexity. The resulting predictions from using a model with 30 epochs of training are shown in Figure 7.

### 4.3.2 Model Results

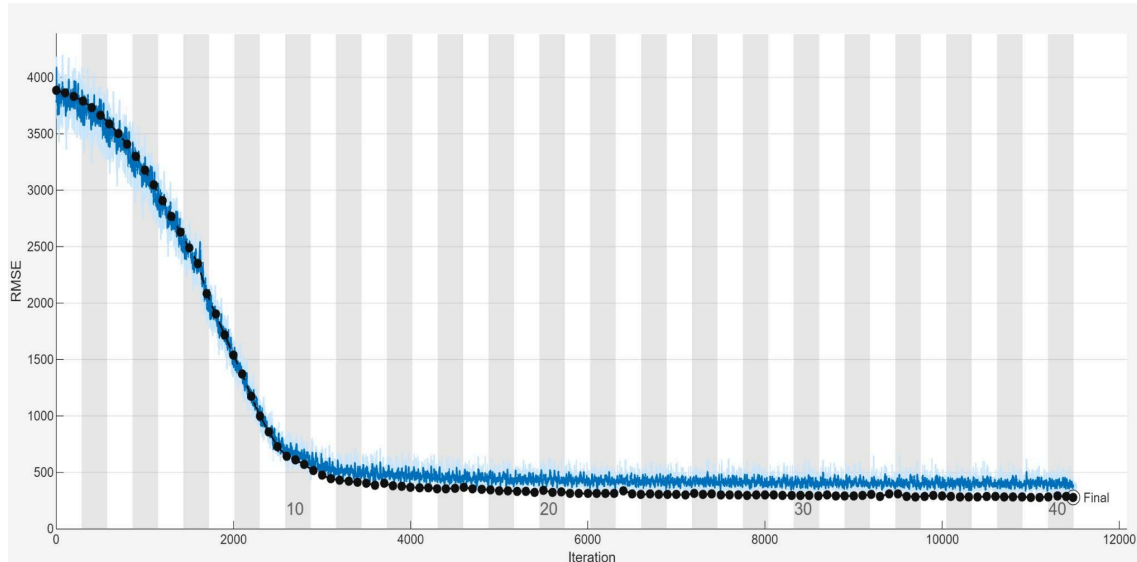
LSTM performed worse than RNN for all metrics, but the difference is not large. It is important to notice that the LSTM results are affected by computational limitations to keep the model simple enough to train in a timely manner.

In Figure 7, the model's issue with predicting peaks is clearly visible, caused by the lack of temporal evolution in monthly patterns, which are not covered by the sequence length of 1 day. However, predicted lower values are more stable than with the RNN and are mostly estimated to be larger than the actual values.



**Figure 7: LSTM Model Predictions on Test Set**

Figure 8 shows that the training loss decreases sharply during the initial epochs, indicating rapid learning and effective convergence. While reaching 10 epochs, the loss begins to stabilize, indicating stable training and diminishing performance gains with further epochs. Overall, the training plot suggests healthy convergence and a typical training process for LSTM.



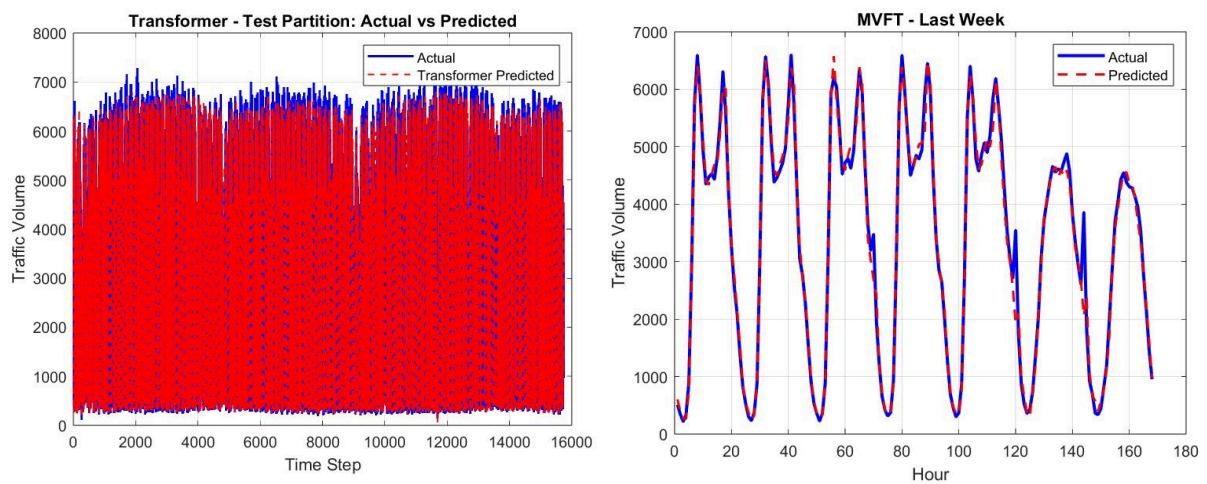
**Figure 8:** Training Plot for LSTM Model Trained on 40 epochs

#### 4.4.1 Multivariate Forecasting Transformer Model

The multivariate forecasting transformer is a model designed for very long sequential data, from which the model compares the relationships of the variables throughout the entire dataset to learn global patterns. The complexity of the approach makes the model very computationally demanding.

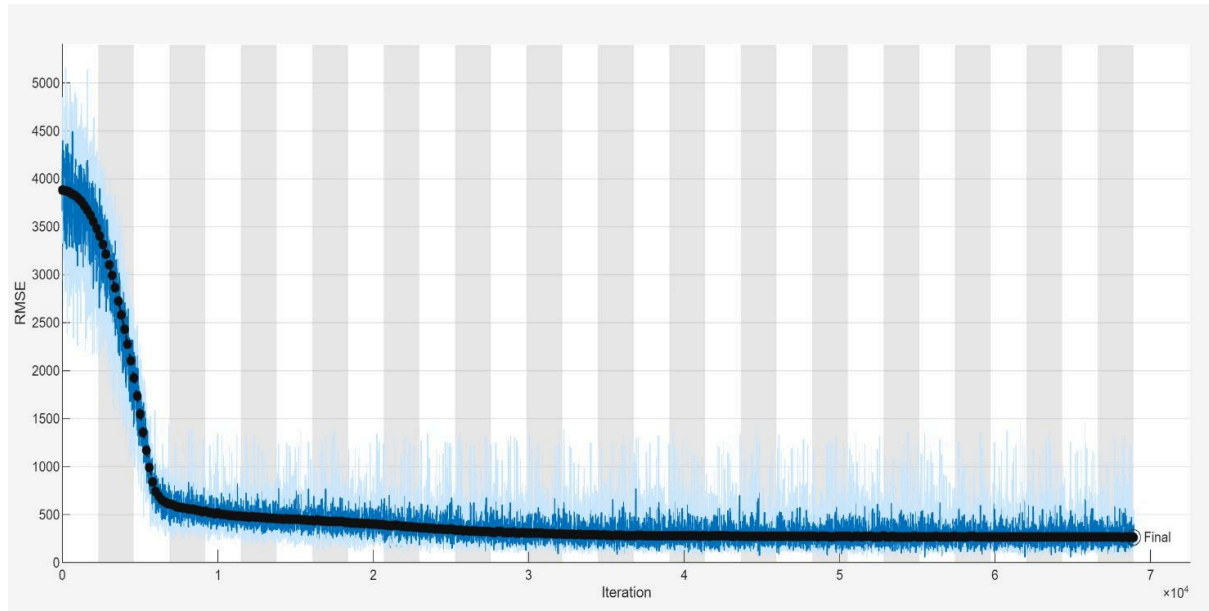
#### 4.4.2 Model Results

The predictions from the transformer model are quite similar to the predictions obtained from the LSTM model. The performance metrics in Table 1 indicate that the transformation model performs slightly worse than the RNN. By visually assessing Figure 9, this model shows a similar issue to the other models, underestimating the peak values.



**Figure 9:** Multivariate Forecasting Transformer Model Predictions on Test Set

Figure 10 shows that the training loss decreases sharply during the initial epochs, indicating rapid learning. However, after reaching 3 epochs, loss settles into a flat but noisy pattern. This may indicate the model is too simple for the task. In the end even with possible issues present, MVFT achieves the 2nd best performance, indicating that there may be room for improvement with a more complex model using this approach.



**Figure 10:** Training Plot for MVFT Model Trained on 30 epochs

## **5. Discussion**

### **5.1 Improving the RNN Model**

One simple and often effective way to improve the model is to adapt the delay. If the model predictions were less accurate it would most likely result from the loss of temporal information during training. Because the model only sees one hour at a time, it could be unable to capture longer seasonal patterns. By providing the model with longer sequences from the past, it would be able to learn recurring traffic patterns more effectively and reduce short-term prediction errors. In our case, there are pre-known cycles in traffic behaviour that make it easier to find possible options to try as a delay sequence.

### **5.2 Improving the LSTM Model**

LSTM has more adaptable parameters than RNN, so there are more ways to improve the model. Similarly, the length of the lookback window used for predicting the next values can be adjusted to better fit the data. Extending the lookback window beyond 24 hours would allow the model to better capture weekly traffic patterns in addition to daily cycles. The model's capacity could also be increased by tuning the number of LSTM units or adding additional layers.

The biggest issue with the model is predicting peaks, as the LSTM tends to smooth sharp changes and underestimate high traffic volumes. This is typical when peaks are rare in the training data. The model could handle them better by adding more examples of peak periods, or including features like hour of day or weekday that help the model recognise when peaks usually occur.

### **5.3 Improving the Multivariate Model**

The improvement ideas for the previous models also apply to the MVFT model but there are also several different options to try for optimizing it. Starting from the ways to do the normalization, to forming hybrid models. Most importantly the implemented model may not be complex enough to capture the data patterns as shown in figure 10 and could benefit from more layers or increased size of layers.

The biggest issue with the model is the peak value underestimation. One addition to the architecture for this issue could be `globalMaxPooling1dLayer` which can help maintain the peaks in the data. Additionally, decomposition can be performed inside the transformer or in the pre-processing stage, where trend and seasonal components can be fed as separate channels. Another thing to test is different loss functions. Some might encourage the model to smooth the prediction to follow the mean. Loss function can also be adapted to penalize the underestimations. One option could be combining the transformer with a classical model, and/or trying a different transformer model that is meant for strongly seasonal data.

## 6. Conclusions

The aim of this project was to forecast hourly traffic volume on Interstate 94 using weather and time-related features. Exploratory data analysis showed strong daily and weekly seasonality and high short-term autocorrelation.

All neural network models substantially improved the forecasting accuracy compared to the autoregressive baseline model. This shows that deep learning approaches can better capture the nonlinear relationships and complex seasonal patterns in traffic data. A common challenge across all models was the underestimation of peak traffic volumes, suggesting that improvement would require improving the features to allow for these patterns to be learned by the models.

Among the tested approaches, the simple RNN achieved the strongest overall performance in one-step-ahead forecasting. However, more complex models (LSTM, transformer) may become more useful when forecasting multiple time steps ahead, where longer-term dependencies play a larger role. Future work should therefore focus on multi-step forecasting and improved handling of peak traffic volumes.

## References

Chen, D., Zhang, J., Jiang, S. 2020. Forecasting the Short-Term Metro Ridership With Seasonal and Trend Decomposition Using Loess and LSTM Neural Networks. *IEEE Access*, 8, 91181-91187. Available: 10.1109/ACCESS.2020.2995044

Mahesh, N., Babu, J. J., Nithya, K., Arunmozhi, S.A. 2024. Water quality prediction using LSTM with combined normalizer for efficient water management. *Desalination and Water Treatment* 317. Available: <https://doi.org/10.1016/j.dwt.2024.100183>

Rehman, K. A., Shahrizalad, A. R. M, Noorasiah, M. 2023. Improving long-term wave forecasting through seasonal adjustment based on STL and CNN-GRU network. *Journal of Sustainability Science and Management*, 18 (4), 120-138. Available: <http://doi.org/10.46754/jssm.2023.04.009>

Silva, R.P.; Zarpelão, B.B.; Cano, A.; Junior, S.B. 2021. Time Series Segmentation Based on Stationarity Analysis to Improve New Samples Prediction. *Sensors*, 21, 7333. Available: <https://doi.org/10.3390/s21217333>

Shengsheng, L., Weiwei, L., Wentai, W., Feiyu, Z., Ruichao, M., Haotong, Z. 2021. SegRNN: Segment Recurrent Neural Network for Long-Term Time Series Forecasting. Available: <https://doi.org/10.48550/arXiv.2308.11200>