Machine Learning IoT Application Design and Implementation

Carson Edmonds

Patricia Enrique

Michael Moll

Department of Engineering, University of San Diego

AAI-530: Data Analytics and Internet of Things

Prof. Marbut

February 26, 2024

GitHub Repository

Introduction

Understanding the traffic patterns in a major city allows for the department of transportation (DoT) to schedule maintenance and future developments. Analyzing the data gathered allows for transportation optimization and efficiency and to aid in the development of a smart city. This analysis aims to predict the traffic flow using the autoregressive integrated moving average (ARIMA) and the linear regression models. Both models are compared to determine which is a better predictor of the traffic volume. Additionally, a Recurrent Neural Network (RNNs) and a Long Short-Term Memory (LSTM) model are compared to predict the weather with a one-hour predictive horizon.

For this analysis the dataset selected is publicly available through the US Irvine Machine Learning Repository and contains environmental and traffic information from I-94 westbound in Minneapolis-St Paul, MN (Hogue, 2019). The data is collected by the MN DoT Automatic Traffic Recorder (ATR) station 301 located midway between Minneapolis and St Paul, MN using a collection of sensors. The sensor measurements are recorded hourly from 2012-2018 for a total of 48204 observations. Temperature and precipitation sensors are used to collect the environmental data while the traffic volume is measured using inductor loop detectors permanently installed in the pavement. Table 1 summarizes the variables observed in the dataset.

Table 1

Variables in the Metro Interstate Traffic Volume dataset

Variable	Data Type	Description
Holiday	Categorical	US National holidays plus regional holiday, Minnesota State Fair
Temp	Numerical	Average temp in kelvin
Rain_1h	Numerical	Amount in mm of rain that occurred in the hour
Snow_1h	Numerical	Amount in mm of snow that occurred in the hour
Clouds_all	Numerical	Percentage of cloud cover
Weather main	Categorical	Short textual description of the current weather
Weather description	Categorical	Longer textual description of the current weather
Date_time	Categorical	Hour of the data collected in local CST time
Traffic_volume	Numerical	Hourly I-94 ATR 301 reported westbound traffic volume

IoT System Design

The IoT system is comprised of temperature and precipitation sensors to collect the weather data and inductor loop detectors to detect cars (Turner, et al., 2010). Figure 1 illustrates the system design diagram for the IoT device and also includes the connectivity type and the messaging protocol. The IoT device is contained within a weatherproof container and placed along the road of interest as shown in Figure 2.

ATRs are permanent devices in the pavement surface that continuously and automatically collect data. The ATR station is able to collect the traffic flow from this device as well as distinguish between the types of vehicles and their speed. From the system design diagram, it can be seen that the inductor loop detector is the limiting sensor for operation as it has a minimal operational temperature of -20°C and the temperature and precipitation sensors are rated for operation until -40°C. When the temperatures in Minneapolis-St Paul, MN fall below this limit, the data collected by the sensors may be unreliable.

The data collected from the sensors is stored locally onsite and a basic analysis is completed. The preliminary analysis detects if there is a sensor failure or another anomaly which requires onsite maintenance or support. Next, the data is sent periodically to the centralized cloud computing service via 4G network for further analysis. The data messaging protocol used for this IoT device is Message Queue Telemetry Transport (MQTT) due to its robustness handling intermittent connections. Once the data has been received by the cloud service, a deeper analysis is completed using machine learning techniques. The data from one single ATR station can be combined with the data from other stations to deepen the understanding of trends in the city. Storing and handling the data on a cloud service allows for on-demand self-service, broad network access, resource pooling, rapid elasticity, and measured services to minimize the problems that arise from handling big data (Minteer, 2017).

Figure 1

System design diagram for the traffic IoT device

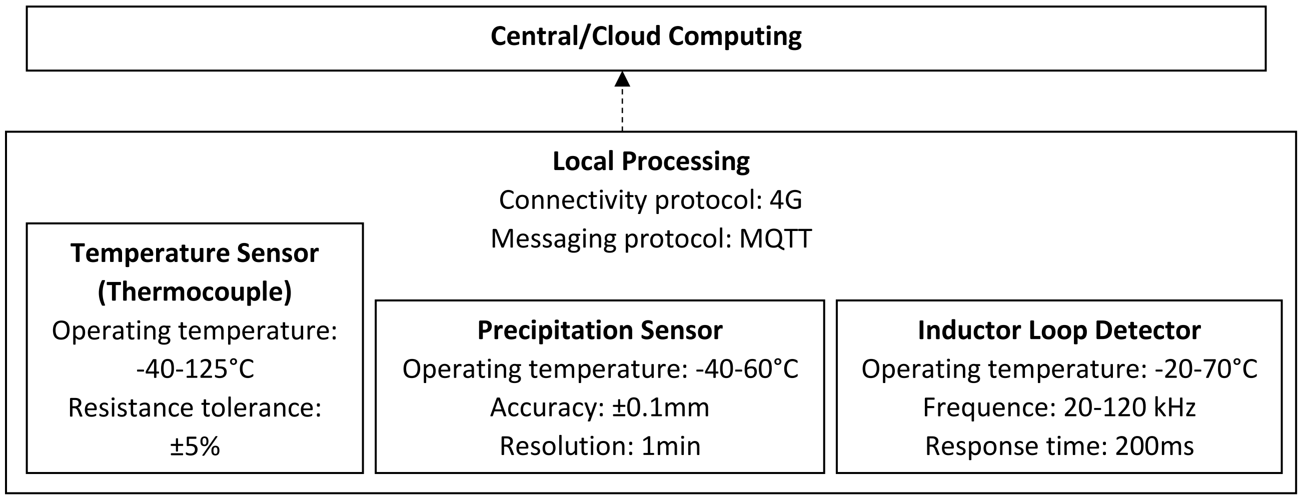

Central/Cloud Computing
Local Processing
Connectivity protocol: 4G
Messaging protocol: MQTT
Temperature Sensor (Thermocouple)
Operating temperature: -40-125°C
Resistance tolerance: ±5%
Precipitation Sensor
Operating temperature: -40-60°C
Accuracy: ±0.1mm
Resolution: 1min
Inductor Loop Detector
Operating temperature: -20-70°C
Frequence: 20-120 kHz
Response time: 200ms

Figure 2

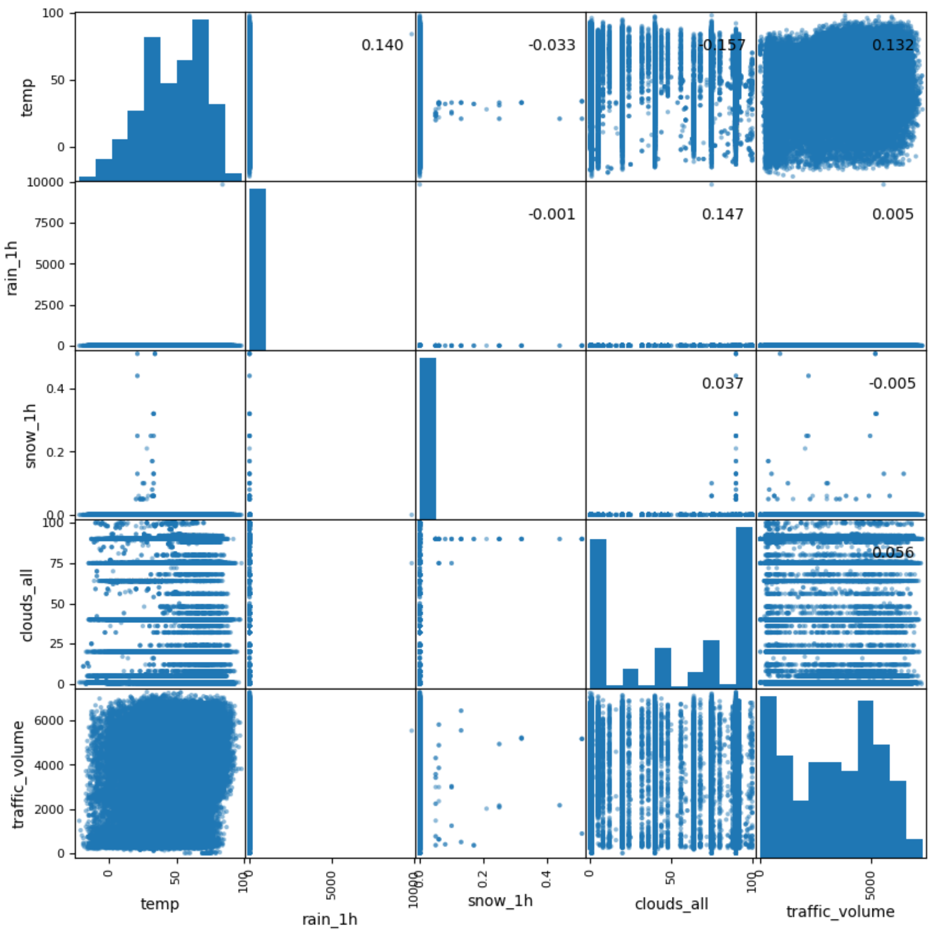
IoT device location (MN Department of Transportation. (n.d.))

Exploratory Data Analysis

To ensure that the data is able to be processed by the machine learning models, the data is first analyzed and cleaned. The date_time variable is converted to a datetime data type to allow for time series predictions and the temperature values are converted from kelvin to Fahrenheit for better user comprehension. Next, the data is checked for missing values which are not present before removing any duplicated inputs. To determine which variables may influence the traffic volume, the correlation matrix is generated as shown in Figure 3. From the matrix it can be seen that the correlation coefficients are all fairly neutral with the traffic volume having the highest correlation value with the temperature variable.

Figure 3

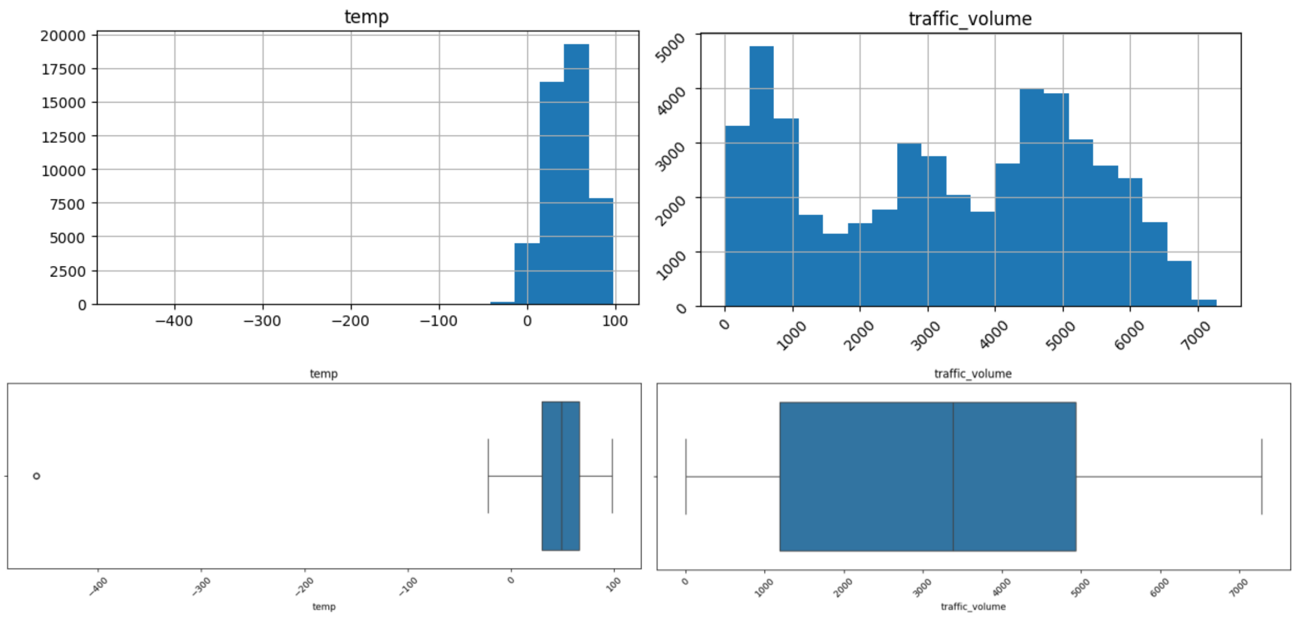
Correlation matrix for the dataset



The distributions for the numerical variables are represented using histograms and the graphs corresponding to the temperature and traffic volume variables can be seen in Figure 4 along with their corresponding box plots. From the boxplots generated, it can be seen that the temperature variable contains an outlier data point of -460°F which is deemed a sensor error and removed from the dataset.

Figure 4

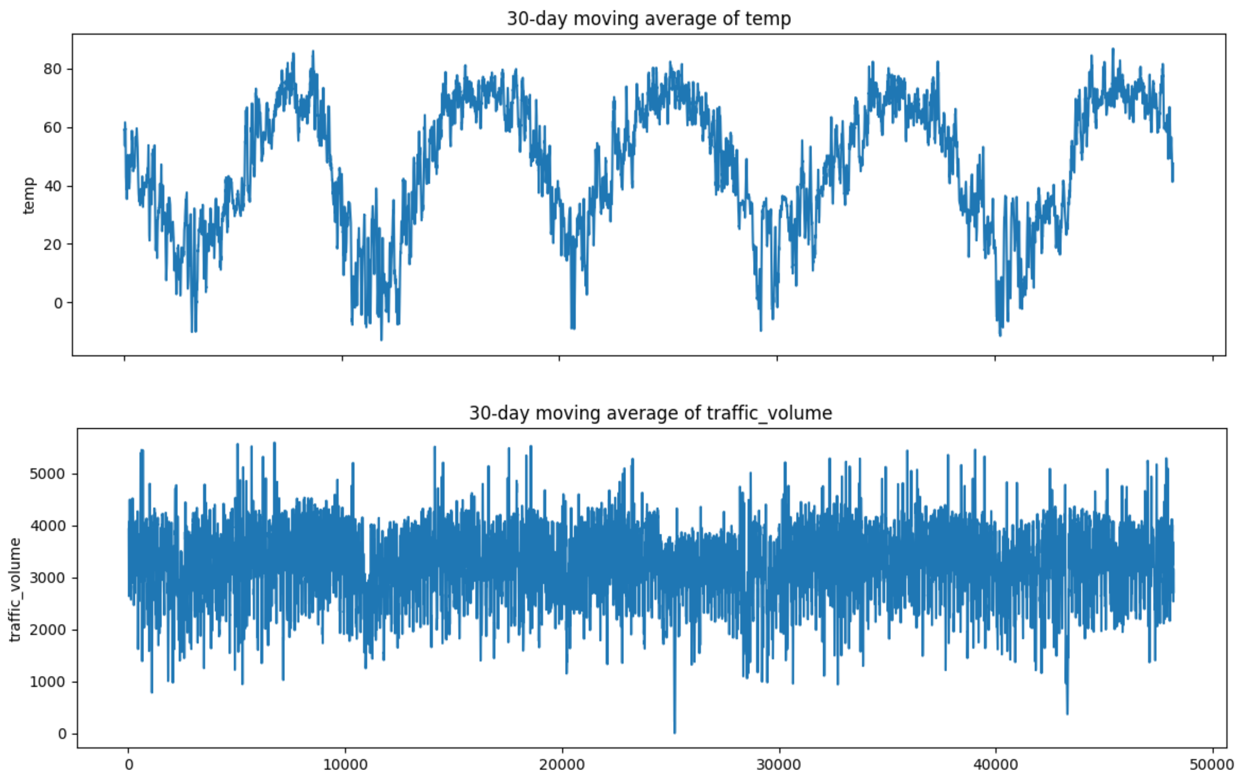
Numerical variable distribution



To identify trends in the variables, a 30-day moving average is applied to the temperature and the traffic volume as shown in Figure 5. From the graphs in Figure 5, it can be seen that the temperature variable follows a sinusoidal trend indicating that future values of the variable may be predictable, and it is a good choice for time series analysis. The traffic volume variable appears to have no trend and may be more difficult to predict using time series techniques. The variables do have similar peaks indicating that they may influence one another.

Figure 5

30-day moving trend for temperature and traffic volume



Time Series Prediction

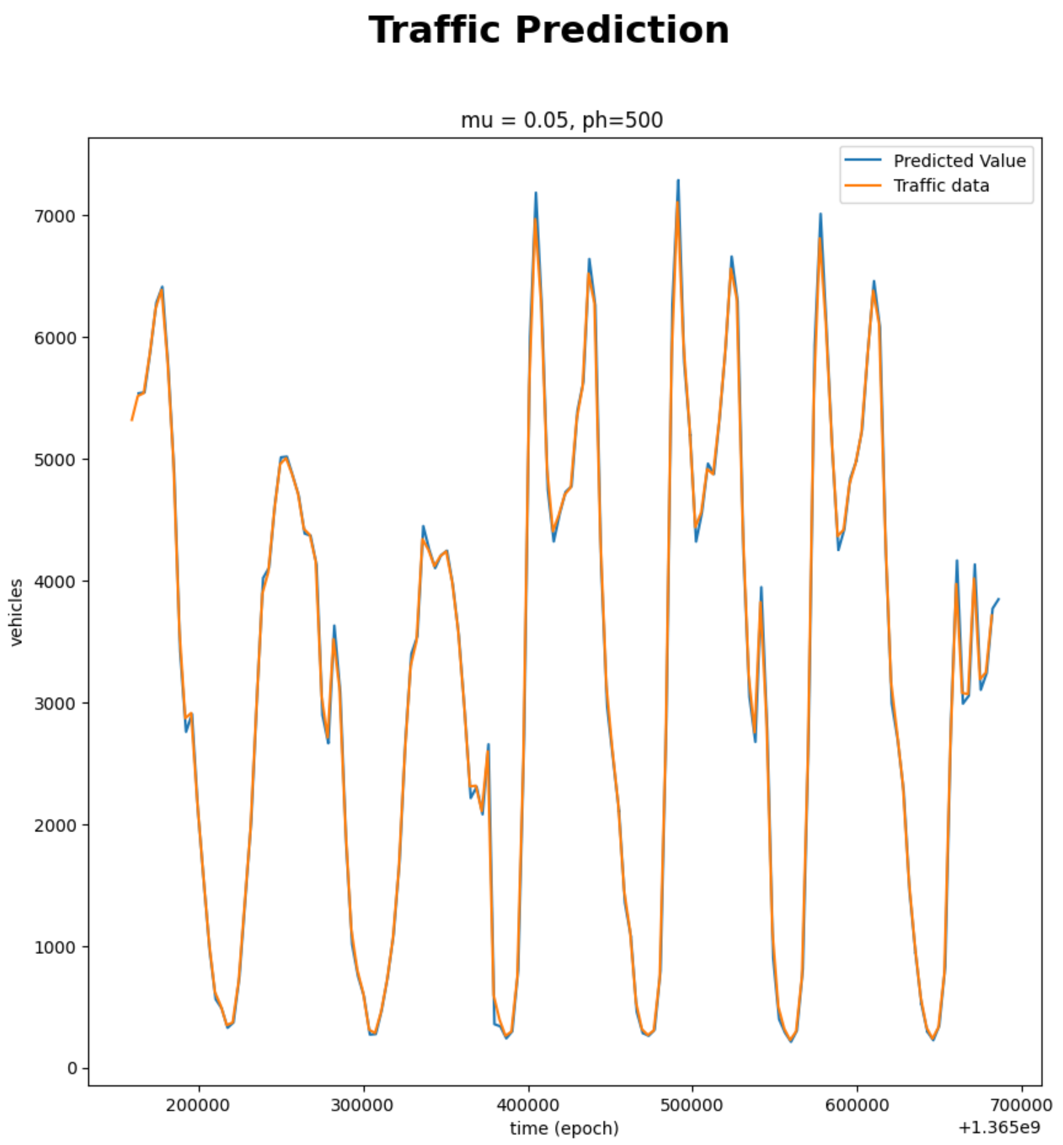
Linear Regression Model

Linear regression is a lightweight model that can be used to predict the traffic volume based on the historical data collected by determining a relationship between the dependent and independent variables. It is important to note that the dependent variable is not linear with respect to the independent variable but with the model parameter weights. This distinction allows for relationships such as exponential or sinusoidal to be modeled (Kapoor, 2019). For this analysis, the past traffic volume data is used to predict the traffic volume in the next hour. Three different models are compared with varying mu values determining the forgetting factor which exponentially decays the data from history.

The last 200 datapoints and their respective predictions for a mu value of 0.05 are shown in Figure 6. From the graph, it appears that the model is able to accurately predict the traffic volume with a 1 hour predictive horizon. To compare the three models, the mean squared error (MSE) is calculated and for mu values of 0.9, 0.05, and 0.01 the MSEs are 2099704, 5.660, 0.053 respectively. From these results it can be concluded that the linear regression model with a mu value of 0.01 has the best performance.

Figure 6

Last 200 datapoints and their predictions

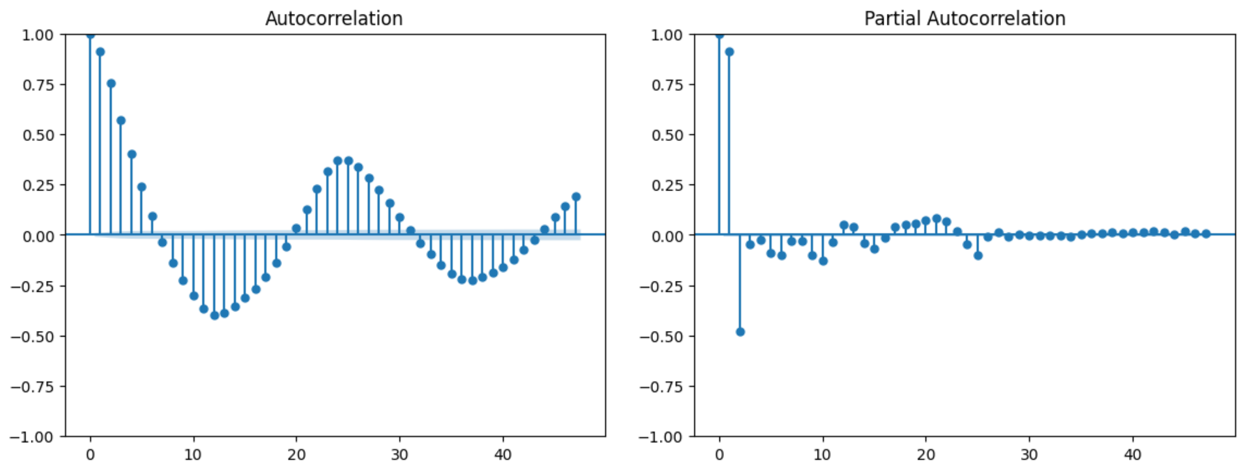


ARIMA Model

The ARIMA model is selected to predict the traffic volume due to its ability to include both trends and seasonality effects in the predictions. This model consists of a combination of moving average and autoregressive techniques that forecast future values by describing autocorrelations in the data (Minteer, 2017). Before designing the ARIMA model, an analysis of the data is done to determine if it is stationary indicating that the mean, variance, covariance, and standard deviation are not a function of time using the Dickey-Fuller test (Kumar, 2023). From this analysis it is determined that the data is stationary and the ARIMA model can be built using the predefined class available from StatsModels (Perktold, et al., 2013). Using the stationary test results, the order used for this model is (2,1,0). The autocorrelation and partial autocorrelation graphs generated are shown in Figure 6.

Figure 6

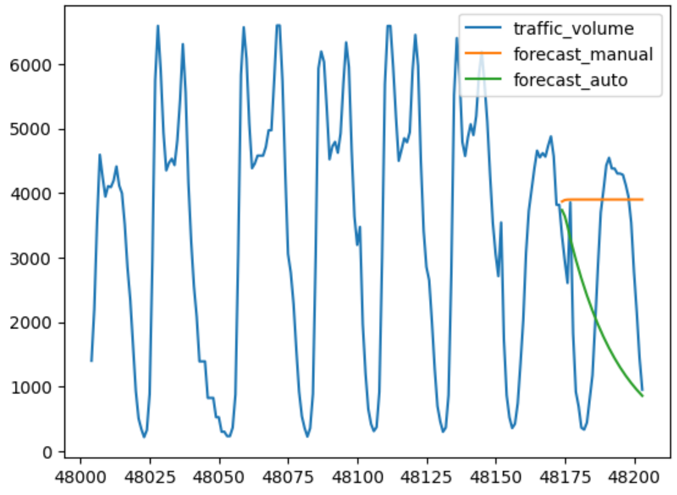
ARIMA model autocorrelation and partial autocorrelation



Using this model, the predicted traffic volume values are plotted along with the actual values in Figure 7. The performance of the model is determined using the mean average error (MAE) which is 1430 and 1815 for the manual and the auto predictions respectively. From the graph and the MAE values it can be seen that the ARIMA model is not the preferred model to predict the traffic volume when compared to the linear regression model.

Figure 7

ARIMA model predictions



Deep Learning Prediction

RNN Model

The RNN model is selected for the deep learning model in this analysis due to its performance with time sequence data. The feedback loops in this method allow for information preservation and for information to be passed from previous steps to the present (Kapoor, 2019). A simple RNN architecture is implemented using TensorFlow with one hidden layer with 100 units, a dropout layer, and a rectified linear unit activation function (Zhu & Chollet, 2023). The performance of the model is calculated using MSE and loss as seen in Figure 8. From the loss curve it can be seen that the model converges and is able to learn and improve over time. The test loss remains below the training loss indicating that no overfitting is occurring.

Figure 8

RNN training performance

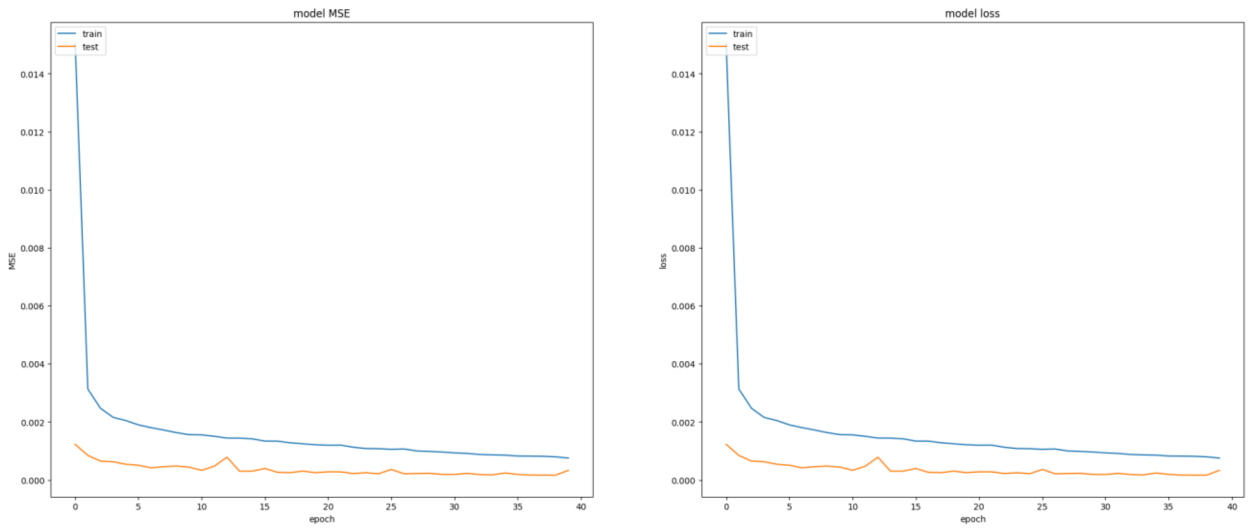
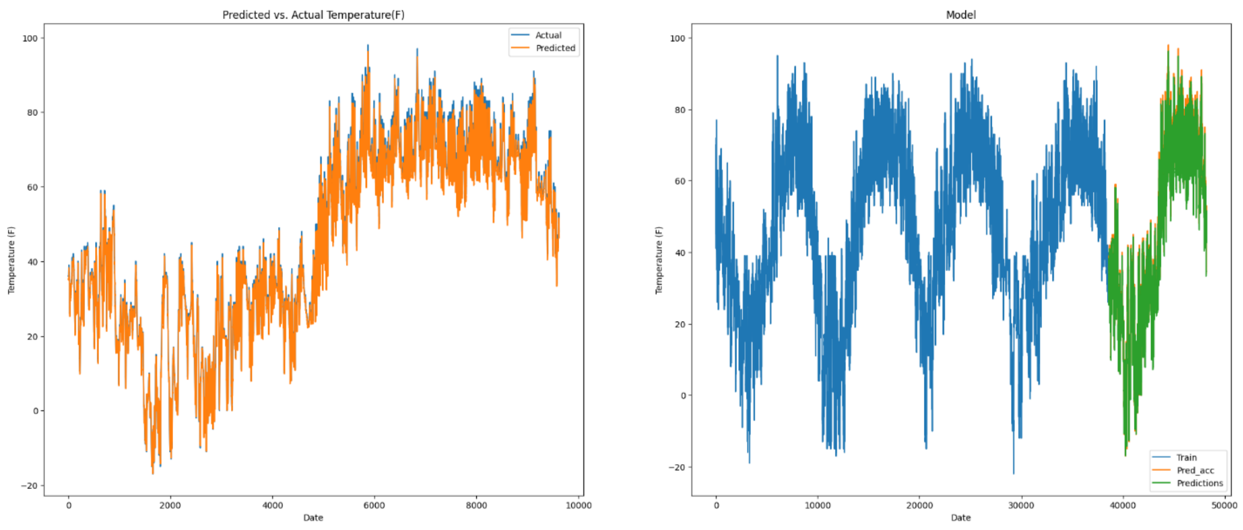


Figure 9 illustrates the model's predictions of future temperatures versus the true values using a testing dataset. The performance on the test dataset is measured using the MSE and loss, similar to the training dataset, as well as the root mean squared error (RMSE). From the results, it can be seen that this model is able to generalize to unseen data and produce fairly accurate temperature predictions.

Figure 9

RNN predictions using testing data



LSTM Model

An issue that arises with the RNN model is a vanishing gradient. To overcome this, the LSTM model can be used instead of the RNN in the hidden layer. To allow for comparison between the two models, the model architecture remains the same and only the hidden layer is updated. The performance of the LSTM model on the training data is shown in Figure 10. The loss and MSE curve for the LSTM model appears similar to the RNN model and indicates that the model is learning and improving as it converges with no overfitting.

Figure 10

LSTM training performance

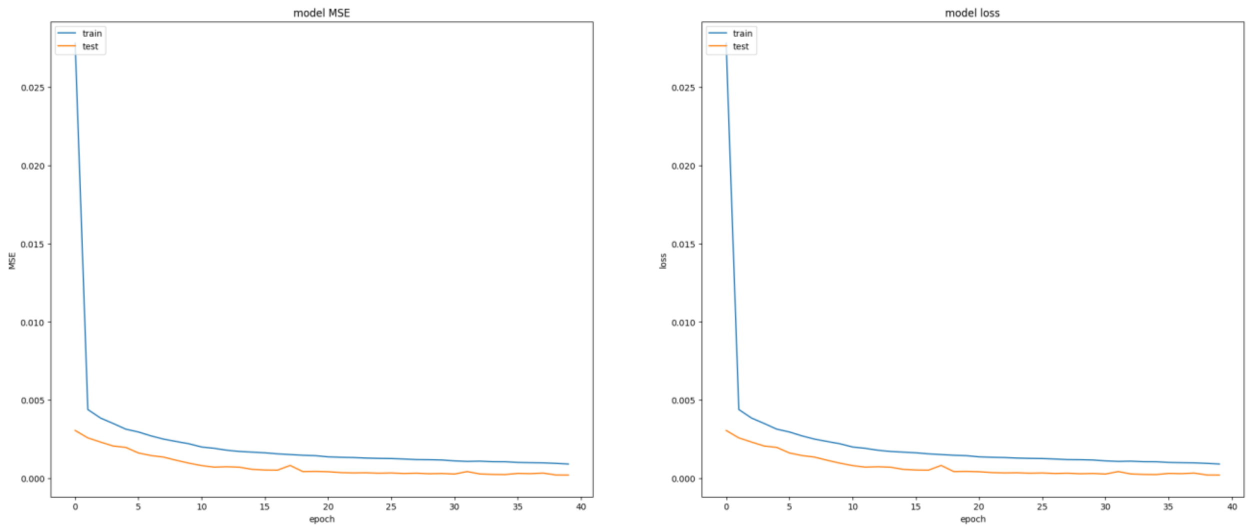
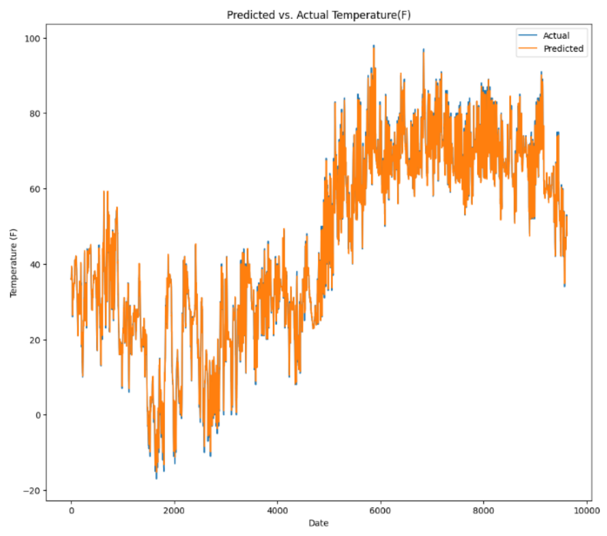
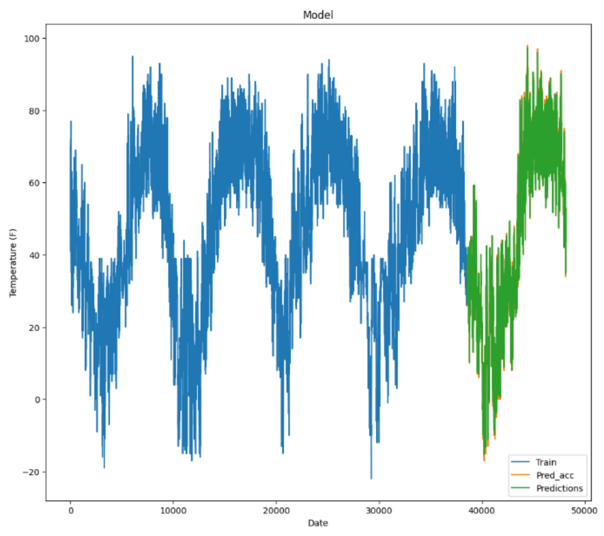


Figure 11 illustrates the model performance on the testing dataset which also appears to be similar to that produced by the RNN model, indicating that the LSTM model is also able to generalize to unseen data.

Figure 11

LSTM predictions using testing data





RNN and LSTM Model Comparison

To compare the performance of the two deep learning models, their MSE and losses are compared in Table 2. It can be seen that the models perform similarly throughout with the RNN model outperforming in training and the LSTM model outperforming in testing; however, both models have a similar testing RMSE. It can be concluded that the LSTM model may have a slight advantage when applied to unseen data.

Table 2

RNN and LSTM model performance

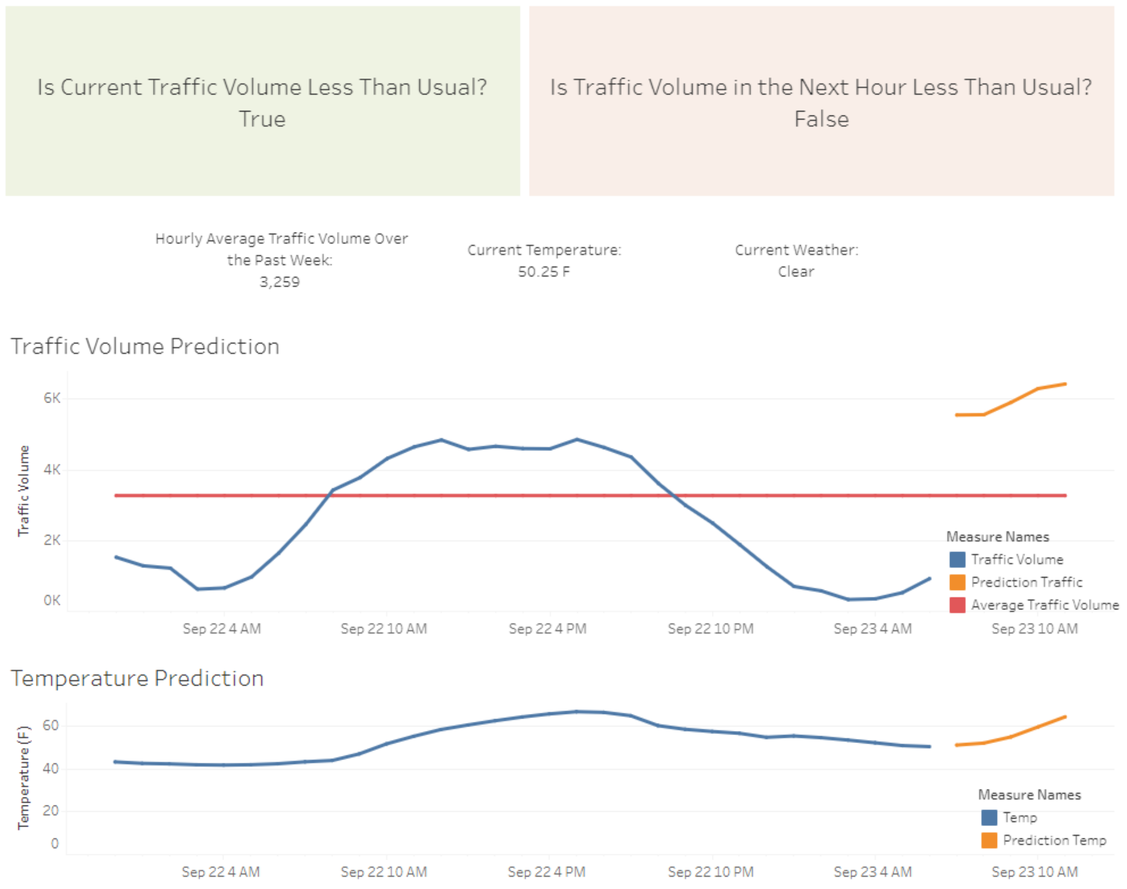
Deep Learning Model	Training MSE	Training Loss	Testing MSE	Testing Loss	Testing RMSE
RNN	1.724e-04	1.724e-04	2.663e-04	2.663e-04	0.297
LSTM	2.075e-04	2.075e-04	1.988e-04	1.988e-04	0.299

IoT Dashboard

For visualization of the data and the analysis, Tableau Public is used. An overview of the dashboard is provided in Figure 12. The most important information is placed in the top left corner followed by the tile on the top right. These provide information on the current volume of the traffic and the predicted status in the next hour and whether that volume is greater than or less than the average. The tiles are further shaded green to illustrate that the traffic is less than usual, or red to illustrate that the traffic volume is greater than the average. The tiles directly below these provide insight on the average traffic in the last week as well as the current temperature and weather. These tiles and text are smaller since the information provided is not as relevant or important to the user; however, it is information that is still nice to have available. Next, the visualizations for each of the machine learning insights are included. The data from the past day is shown along with the predictions for both the traffic volume and the temperature for the next 5 hours. Since the user visiting this dashboard is more interested in the traffic volume, the visualization with that information is larger and placed above the graphic with the temperature information.

Figure 12

IoT analysis visualization on Tableau Public



Is Current Traffic Volume Less Than Usual?
True

Is Traffic Volume in the Next Hour Less Than Usual?
False

Hourly Average Traffic Volume Over the Past Week:
3,259

Current Temperature:
50.25 F

Current Weather:
Clear

Conclusion

Applying machine learning methods to the data collected by the MN DoT ATR station 301 located midway between Minneapolis and St Paul, MN allows for a better understanding of the traffic patterns in the city. The analysis completed can be used to schedule maintenance and future developments as well as optimize transportation and to aid in the development of a smart city. A device diagram is created for the IoT device collecting the data to understand the limitations of the sensors within the system and the connectivity between the subsystems.

The two models compared for the prediction of traffic volume are the ARIMA and the linear regression models. From the analysis, it can be concluded that the ARIMA model is not preferred for this prediction; however, the linear regression model with a forgetting factor of 0.01 is able to accurately predict the traffic volume within the next hour with an MSE of 0.053. The deep learning models compared for the temperature prediction analysis are the RNN and LSTM models. Both models performed similarly with the LSTM model obtaining better results with the testing data, indicating it may have a slight advantage when applied to unseen data. Visualizations of the model results are displayed in an IoT dashboard to be accessible by the users.

References

Kapoor, A. (2019). *Hands-on artificial intelligence for IoT: Expert machine learning and deep learning techniques for developing smarter IoT systems*. Packt Publishing.

Kumar, V. G. (2023, December 21). *Statistical Tests to Check Stationarity in Time Series.* Analytics Vidhya. https://www.analyticsvidhya.com/blog/2021/06/statistical-tests-to-check-stationarity-in-time-series-part-1/

Hogue, J. (2019). *Metro Interstate Traffic Volume*. [Data set]. UC Irvine Machine Learning Repository. https://doi.org/10.24432/C5X60B.

Minteer, A. (2017). Analytics for the Internet of Things (IoT): Intelligent analytics for your intelligent devices. Packt Publishing.

MN Department of Transportation. (n.d.). *Traffic Forecasting & Analysis.* https://www.dot.state.mn.us/traffic/data/coll-methods.html

Perktold, J., Seabold, S., & Taylor, J. (2013). *statsmodels.tsa.arima_model.ARIMA.* StatsModels Statistics in Python. https://www.statsmodels.org/0.6.1/generated/statsmodels.tsa.arima_model.ARIMA.html

Turner, S., Carson, J., Wilkinson, L. J., Travis, K., & Zimmerman, C. (2010). *Traffic Monitoring A Guidebook.* U.S. Department of Transportation, Federal Highway Administration.

Zhu, S., & Chollet, F. (2023, November 16). *Working with RNNs*. TensorFlow. https://www.tensorflow.org/guide/keras/working_with_rnns#built-in_rnn_layers_a_simple_example