Real time traffic flow prediction using CNN and LSTM

1. Introduction:

Traffic flow prediction (TFP) means predicting the volume and density of traffic flow, usually to control vehicle movement, reduce traffic jams, and create the optimal (least-time or energy-consuming) route. With the recent advancement in Artificial intelligence, Machine learning (ML), Deep learning (DL), and Big data, research in the field of predicting traffic flow has been expanded extensively.
TFP is the key component of Intelligent Transport Systems (ITS) and can assist ITS to forecast traffic flow. Large cities have exceedingly difficult traffic regulations. many countries have adopted ITS to reduce the costs associated with traffic congestion. This study reviews the application of artificial neural network (ANN), ML, DL and other techniques and models for TFP. Finally, we will propose our own predictive model using DL, train and test it, analyze the accuracy and compare the accuracy of our model with other models.
Comparing with conventional ML methods, DL models have the advantages such as simplifying data preprocessing procedures and outperforming other ML methods in terms of accuracy. Therefore, data-driven traffic flow prediction due to the availability of massive traffic data and DL schemes due to data preprocess procedures have received extensive attention recently in TFP.
Furthermore, In recent years there has been a vast increase in available data with the advancement of smart cities. This modernization can have a favorable impact on transportation networks in the area of ITS, reducing travel times, boosting productivity, and minimizing the environmental impact of vehicles. ML and DL technologies are fast-growing domains for predicting traffic flow. Traffic signals, accidents, weather conditions, and road repairs are the primary causes of traffic. Since real-time traffic data are largely produced exponentially, big data principles must be used to improve data transportation. This fact motivated us to to predict the volume of traffic flow between Minneapolis and St. Paul at a specific point in Minnesota. Our aim is to build a multi-step Recurrent Neural Network (RNN) with Long Short Term Memory (LSTM) model that makes a single prediction point of the traffic volume 2 hours into the future, given the previous 6-hour window.

2. Literature Review:

The survey of literature on TFP and ITS was conducted using several secondary sources of information such as articles, Books, and Research Reports published in various publications. The collected information is then reviewed to discover any potential TFP and ITS significant areas of concern.

The prediction of traffic congestion, in particular the short-term traffic forecast, is done by evaluating various traffic parameters. To predict traffic congestion, most research uses historical data. However, several reports offered real-time predictions of traffic congestion. In city transportation and area management, TFP has a wide range of applications. The TFP issue is a time series (TS) problem that involves estimating the urban road traffic flow at a future time using information gathered from one or more observation points during prior periods. Moreover, to predict traffic flow many AI techniques in particular various ML and DL models are applied. An ITS can effectively reduce traffic congestion by forecasting short-term traffic flow.

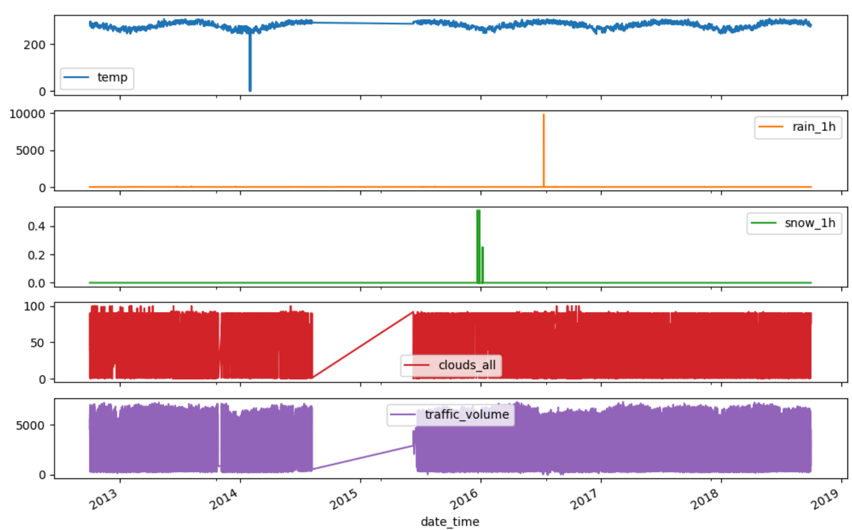
Graph neural networks (GNNs) are deep learning models that use graph data as input and are used for a variety of applications, including molecular property prediction and traffic forecasting. Google Maps employs machine learning to combine current traffic conditions with past traffic patterns for roads all around the world in order to reliably predict future traffic. The aim of traffic forecasting is to make traffic volume predictions using past volume and speed data. Authorities are paying more attention to observing traffic congestion due to the development of the transportation industry and the collection of traffic statistics. Traffic flow data is transformed into a 2D matrix for processing in traffic forecasting due to CNN's outstanding image processing capabilities.

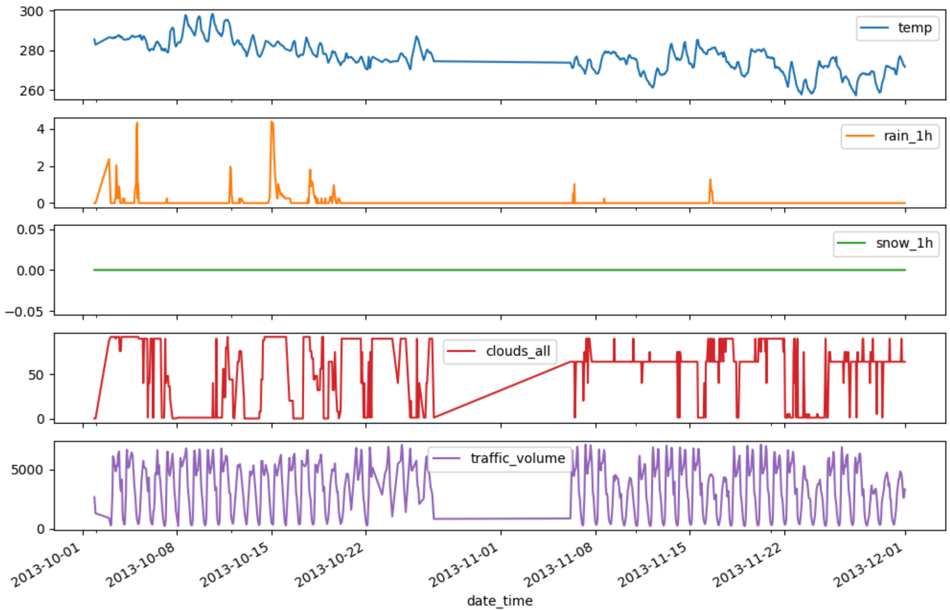
In recent years, new traffic flow prediction models and frameworks have been quickly developed to increase TFP performance in addition to applying artificial intelligence (AI) techniques like machine learning (ML). The majority of traffic data are small-scale, concentrated on highway traffic, or lacking in auxiliary data. Traffic conditions can be predicted using both online and offline data, notably searches on map apps.

B.Karthika et al. discussed the rich mobility data and deep learning about urban traffic predictions, Deep algorithms to forecast real-world traffic data and when the traffic data becomes big data, some techniques to improve the accuracy of traffic prediction are also discussed.
Apurv Chandel et al. used multiple linear regression models for the dataset and obtained best results in correspondence of gradient boosting regression after performing the tuning of hyperparameters with an accuracy of 83%.
Rohit Singh et al. in his research presents several regression machine learning techniques that combine the performance of various algorithms and compares it with existing Machine Learning Algorithms using the mean square error, root mean square error and other performance metrics.
Renhe Jiang et al. presented a comprehensive survey of all the deep learning models used for traffic prediction and implemented the dataset using pytorch and tensor flow for better prediction.
Kanokwan Khiewwan et al. described data mining techniques for the implementation and used DT, KNN and SVM for modeling and found out that DT performed better than other algorithms with an accuracy of 79.9%.

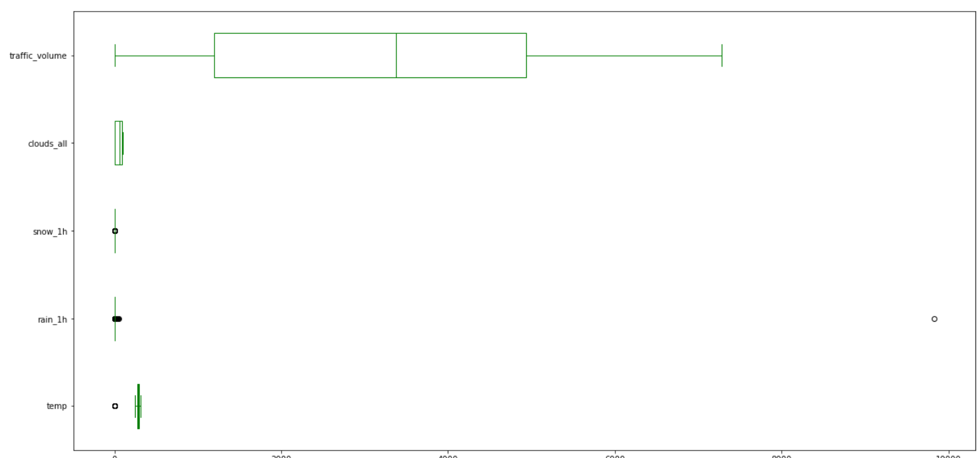
2. Output:

Before data cleaning:

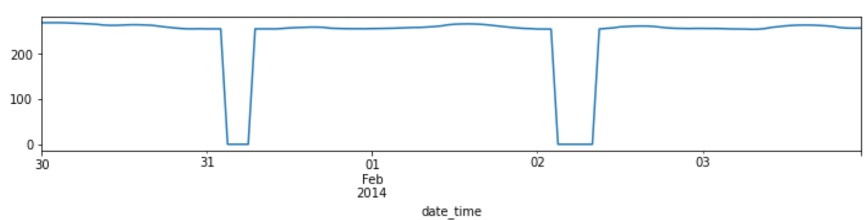




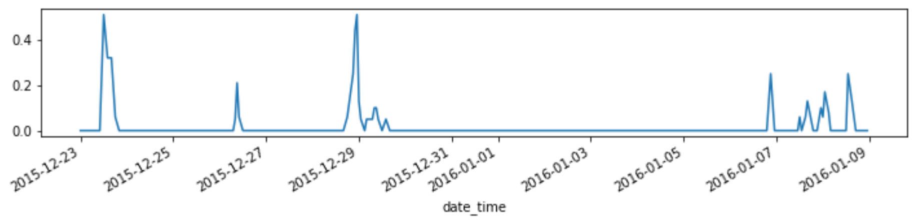
Outliers:



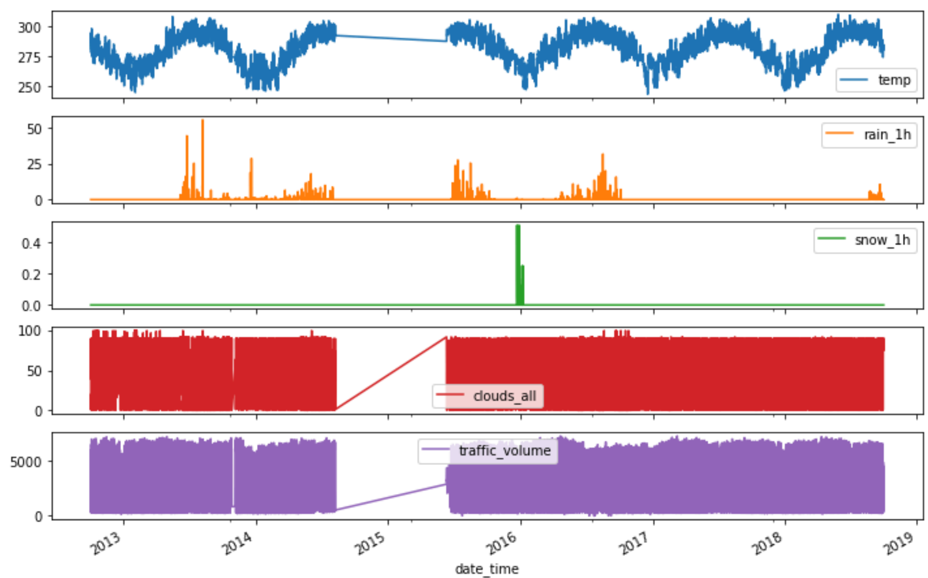
Traffic flow based on temperature:



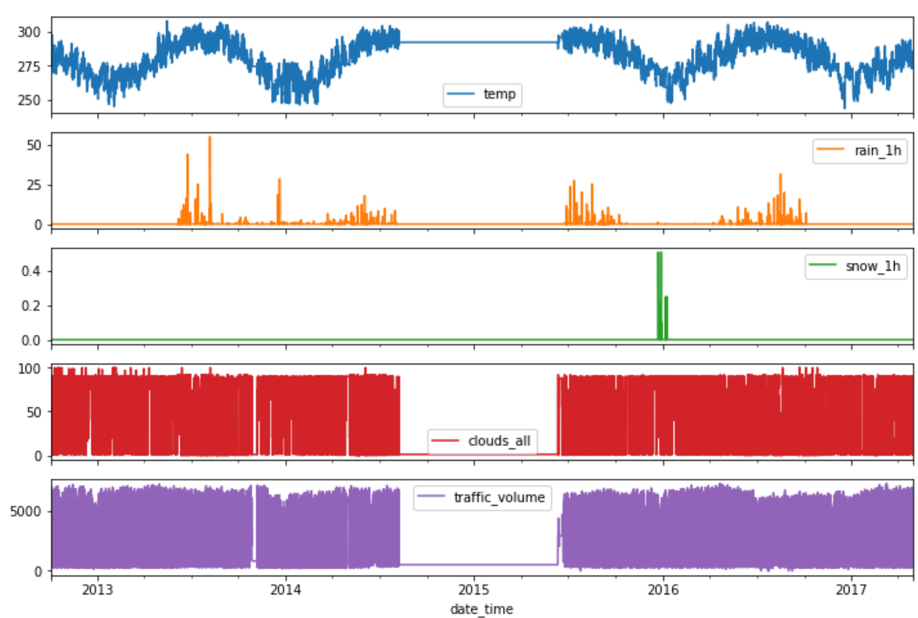
Traffic flow based on snow:



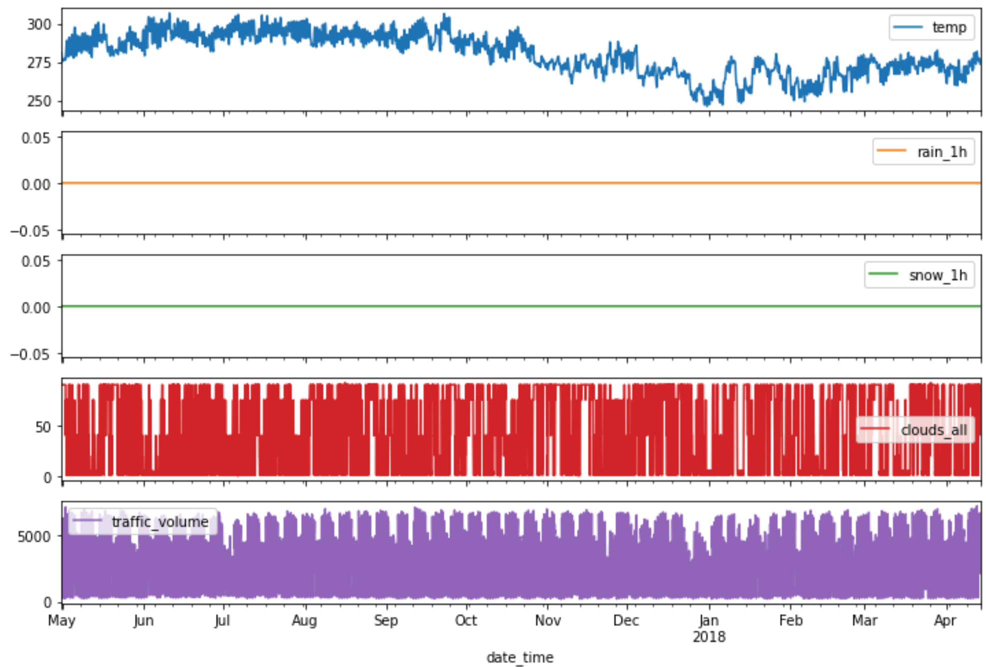
Cleaned data overview:



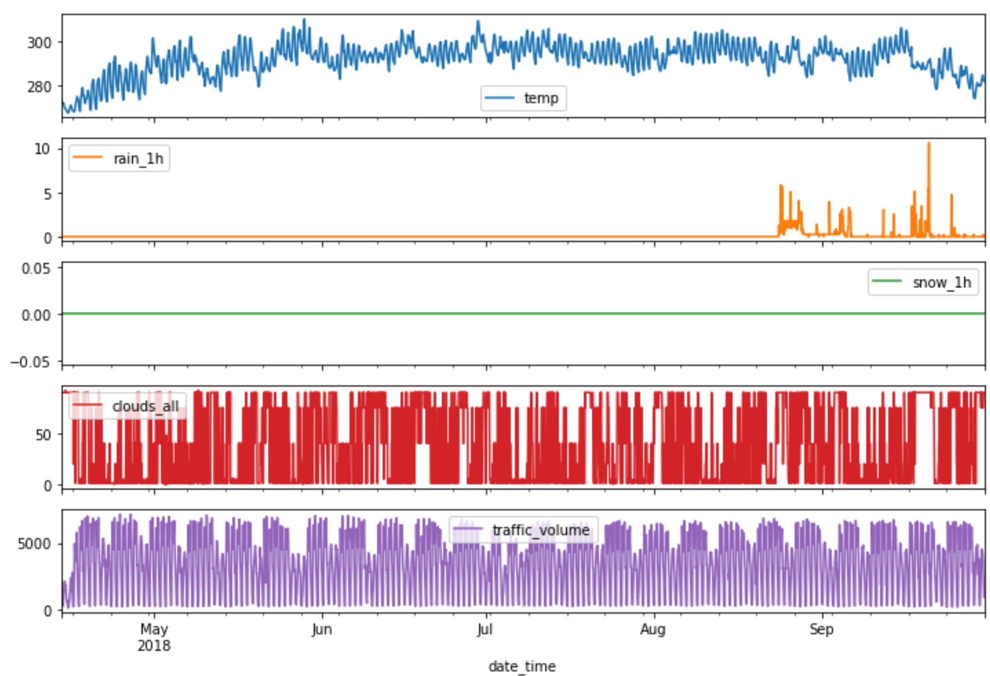
Trained data:



Validation data:

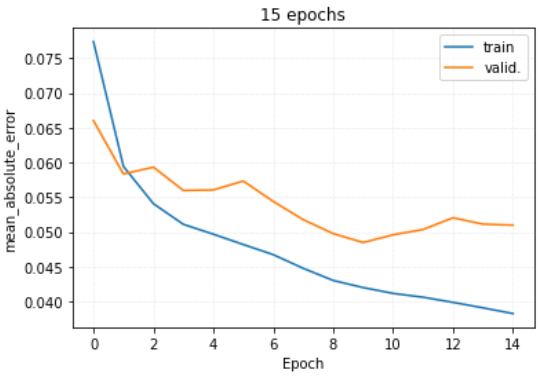


Test data:



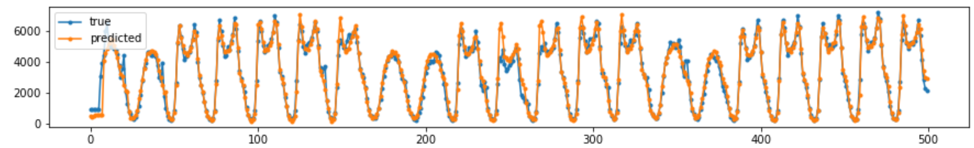
CNN:

Training v/s validation per epochs:

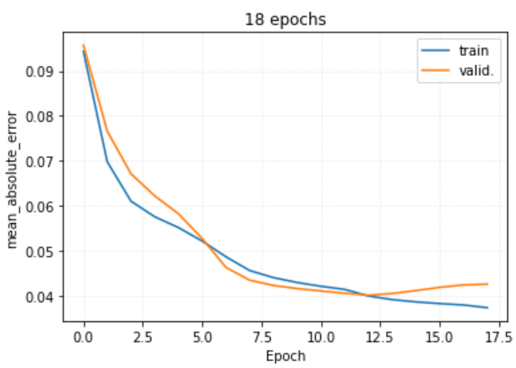


Prediction evaluation:

Predictions: 8352
MAE: 352.99 (0.0485)

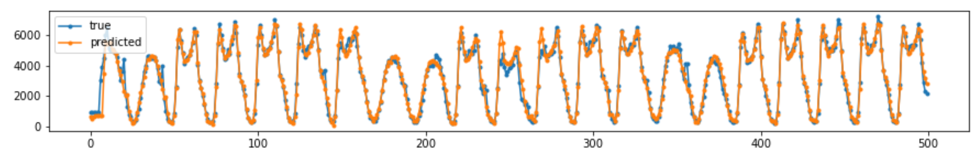


LSTM:
Training v/s validation per epochs:



Prediction evaluation:

Predictions: 8352
MAE: 291.92 (0.0401)



Comparative Analysis :

CNN

- Good at detecting patterns in a fixed window of time
- Can identify local patterns (like morning rush hour patterns)
- Faster to train than RNNs
- Works well for regular, repeating patterns

LSTM

- Better at remembering long-term patterns
- Can handle variable-length sequences
- Good at learning dependencies over time
- Better for irregular patterns or long-term trends

In comparing CNN and LSTM architectures for time series prediction, each demonstrates distinct advantages. CNNs excel at processing local patterns through their convolutional layers (using kernel_size=3 and 256 filters) and are computationally efficient, making them ideal for detecting regular, short-term traffic patterns. They benefit from parallel processing capabilities and faster training times. LSTMs, implemented with 32 memory units, are specifically designed to capture long-term dependencies and temporal relationships in sequential data, making them more effective at learning irregular patterns and long-term trends in traffic flow. While CNNs are more parameter-heavy due to their dense layers (512 units), they offer faster inference times. LSTMs, though slower in training and inference, require fewer parameters and show superior ability to remember important historical patterns. The choice between these architectures ultimately depends on the specific requirements of the traffic prediction task – CNNs for regular, short-term patterns and computational efficiency, LSTMs for complex, long-term temporal dependencies.

CNN and LSTM for time series prediction,

1. **Architecture & Processing**
    - CNN: Uses convolutional layers (kernel_size=3, 256 filters) for sliding-window pattern detection, followed by dense layers (512 units) for feature processing
    - LSTM: Employs recurrent cells (32 memory units) with internal memory gates for sequential data processing
2. **Pattern Recognition Capability**
    - CNN: Excels at detecting local, fixed-window patterns and regular trends in traffic data
    - LSTM: Specializes in capturing long-term dependencies and irregular patterns across varying time intervals
3. **Computational Efficiency**
    - CNN: Faster training and inference due to parallel processing capabilities, but requires more parameters
    - LSTM: Slower processing due to sequential nature, but more parameter-efficient with better memory utilization
4. **Memory Handling**
    - CNN: Limited by fixed kernel size, processes only recent time steps defined by convolutional window
    - LSTM: Maintains internal state memory, can remember and utilize information from much earlier in the sequence
5. **Performance Trade-offs**
    - CNN: Better for real-time predictions and regular patterns, easier to train, but limited temporal context
    - LSTM: Superior for complex temporal relationships, handles variable-length sequences, but requires more training time

3. Conclusion:

The LSTM model performed the best with the least variance.
We could not make LSTM networks gain better results by going deeper but having more LSTM units made a difference.
LSTM and GRU should help with the Vanishing Gradient Descent problem in deep networks.
We noticed that most of the models with fewer LSTM units were hardly overfitting and showed better validation scores than the training score.