A Generic Approach for Extreme Condition Traffic Forecasting

Rose Yu, Yaguang Li, Cyrus Shahabi, Ugur Demiryurek, Yan Liu
Delivered by Watana Pongsapas

March 12, 2020

Contents

- Objectives of accurate traffic forecasting
- Methodology
- Experiments
- Results
- Discussion

Objectives of accurate traffic forecasting

Methodology

Basic concepts

- Autoencoder
- Recurrent Neural Network (RNN)
- Long Short Team Memory (LSTM)

Methodology

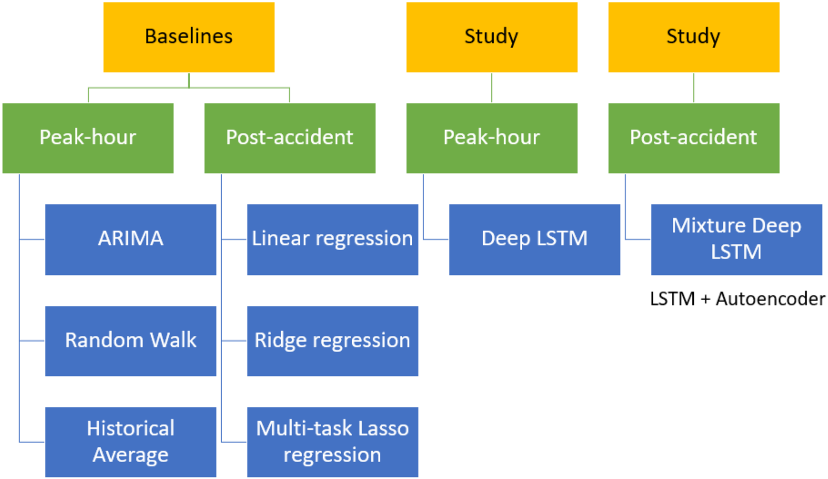
Limitations on the past models

Peak-hour Traffic Forecasting

- Majority of existing traffic forecasting models were designed for short-term forecasting and limited to single time stamp with fixed forecasting horizon.
- Autoregressive Moving Average (ARMA) or Autoregressive Integrated Moving Average (ARIMA) could only capture the linear temporal dependency of the sequences.

Post-Accident Traffic Forecasting

- Feeding in the sequences right before the accidents and generating the sequences after, Deep LSTM didn't yield satisfactory results.
- The researchers hypothesized this problem was due to the delayed of accident reports. The input sequences already contained the interruptions caused by the accidents.


Baselines
Study
Study
Peak-hour
Post-accident
Peak-hour
Post-accident
ARIMA
Linear regression
Deep LSTM
Mixture Deep LSTM
LSTM + Autoencoder
Random Walk
Ridge regression
Historical Average
Multi-task Lasso regression

Traffic data

- Normalized speed, time in a day, and day in a week
- From May 19, 2012 to June 30, 2012
- Source: Loop detectors

Accident data

- Rolling, major injuries, minor injuries, locations, and time
- 6,811 accidents
- Sources: Police reports, California Highway Patrol (CHP), LA Department of Transportation (LADOT), California Transportation Agencies (CalTrans)

Experiments

Data

Model

- 2 hidden LSTM layers with nodes of 64 and 32
- Dropout rate = 10%, Learning rate = 0.001, Epochs = 150
- Loss function: Mean Absolute Average Percentage Error (MAPE)
- Optimizer: RMSProp
- Activation function: Linear

Inputs and Forecast

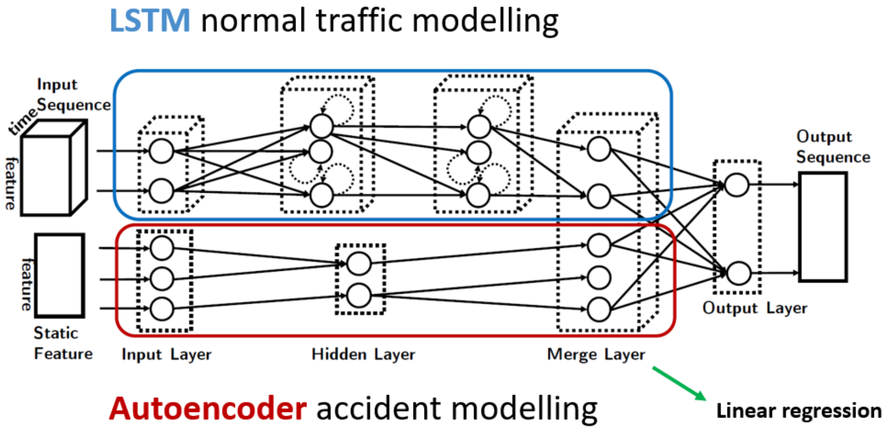
- Normalized speed, time in a day, and day in a week
- Forecast: Future time stamps

Model

- 2 hidden LSTM layers with nodes of 64 and 64
- Dropout rate = 20%, Learning rate = 0.0005, Epochs = 100
- Loss function: Mean Square Error (MSE)
- Activation function: Linear for the LSTM layers and the final prediction layer. Sigmoid for the Autoencoder.

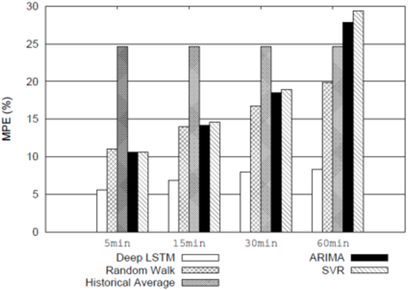
Inputs and Forecast

- Historical sub-sequence one week before the accidents into Deep LSTM
- 5 static accident features into Autoencoder
- Forecast: Predicted speed sequence 3 hours after the reported incident

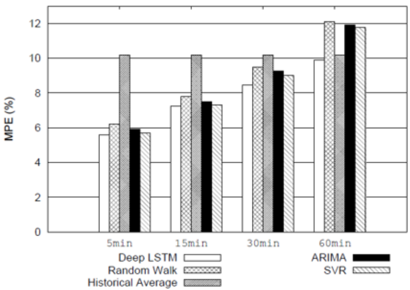

LSTM normal traffic modelling
Input Sequence
time
feature
feature
Static Feature
Input Layer
Hidden Layer
Merge Layer
Output Sequence
Output Layer
Autoencoder accident modelling
Linear regression

Results of Peak-hour Traffic Forecasting

Baselines vs Deep LSTM



(a) Peak hour

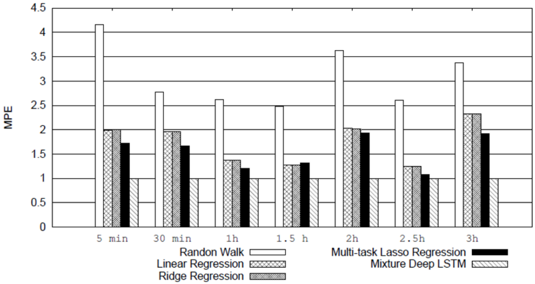


(b) Off-Peak hour

- During peak-hour traffic, Deep LSTM achieved as low as 5% MAPE and its performance was comparably consistent while baseline methods showed significant performance degradation.
- For off-peak hours, most of all methods had similar performances.

Results of Post-Accident Traffic Forecasting

Baselines vs Mixture Deep LSTM

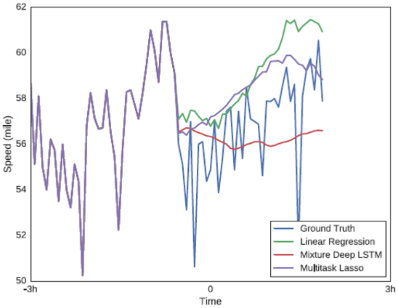


- Mixture Deep LSTM was roughly 30% better than baseline methods.
- Random walk couldn't response to the dynamics in the accident, thus suffers from poor performance.
- A regularization in the multi-task ridge regression and multi-task lasso regression avoided model overfitting, and was more adaptable to the post-accident situation.

Results of Post-Accident Traffic Forecasting

Baselines vs Mixture Deep LSTM (continue)

Method	MAPE
Mixture Deep LSTM	0.9700
Deep LSTM	1.003
Randon Walk	2.7664
Linear Regression	1.6311
Ridge Regression	1.6296
Multi-task Lasso Regression	1.4451
Feed-forward Network	3.6432



- Mixture Deep LSTM performed significantly better than other baselines and It could correctly predict the mean value of a highly dynamic long time sequence.

Discussion

- The traffic speed data, collected by sensors, had more than 20% of missing values.
- The researchers mentioned the speed data was captured by sensors on every five minute. However, they didn't mention in the paper about the speed of which driving lane was recorded.