



RELATED WORK

Traffic Flow prediction in a road network

EFSTRATIOS KARKANIS

P19064

Contents

Introduction	2
Paper 1: Traffic Flow Prediction Models: A review of deep learning techniques [1]	2
Paper 2: Predicting Short-Term Traffic Flow by LSTM RNN [2].....	2
Paper 3: Supervised deep learning based for traffic flow prediction [3]	3
Paper 4: Traffic Flow prediction with big data: A deep learning approach [4].....	3
Paper 5: Comparative Analysis for traffic flow forecasting in the city of Beijing [5]	3
Paper 6: DeepCrowd -A Deep Model for Large-Scale Citywide Crowd Density and Flow Prediction [6].....	3
Paper 7: Diffusion Convolutional Recurrent Neural Network (DCRNN) [7]	4
Paper 8: T-LSTM: a long short term memory network enhanced by temporal information for traffic flow prediction [8].....	5
Paper 9: Hybrid LSTM Neural Network for Short-Term Traffic Flow Prediction [9].....	5
Paper 10: Graph Hierarchical Convolutional Recurrent Neural Network (GHCRNN) for Vehicle Condition Prediction [10].....	6
Paper 11: Short-Term Traffic Flow Prediction Based on XGBoost [11]	7
References.....	8

Introduction

In this report, I collect many papers that solve the problem of traffic flow prediction in a road network. Most of these implementations use LSTM neural networks, MLPs and statistical models, such as ARIMA or SARIMA models. There are also few solutions with Graph Neural Networks, which are nowadays very popular to different kinds of machine learning problems.

The rest of this report is organized as follows: I mention each paper separately and explain the methodology that authors followed to solve the traffic flow forecast problem.

Paper 1: Traffic Flow Prediction Models: A review of deep learning techniques [1]

In this paper, authors conclude that the quality of models is determined by the quality of data. Moreover, traffic flow prediction depends on weather data, holidays and accidents that may happen in the road network. Due to the non-linearity of traffic flow patterns, deep learning techniques with many layers are used to extract more information. It is worth noting that after some research authors did, they found out that CNNs and RNNs require a big amount of data to be trained.

In this experiment, authors chose to implement an LSTM neural network with GRU techniques. The dataset they use is the PeMS Caltrans Dataset. Their code is in GitHub: [xiaochus/TrafficFlowPrediction](https://github.com/xiaochus/TrafficFlowPrediction). During their experiment, they compared different optimizers, such as ADAM, RMSprop and SGD. The best optimizer after comparison was RMSprop.

Paper 2: Predicting Short-Term Traffic Flow by LSTM RNN [2]

In this paper, authors provide the following valuable information about traffic flow forecasting: traffic flow prediction methodology can be divided into two groups, parametric methods and non-parametric ones. In parametric methodologies, model's structure is based in theoretical assumptions about the underlying data distribution or the functional form of the relationship between the input variables and the target variable. These methods cannot describe traffic flow patterns, due to non-linearity of traffic flow. Examples of these methods are ARIMA and SARIMA models and linear regression algorithm.

On the other hand, non-parametric methods in prediction methodology refer to approaches that do not make strong assumptions about the underlying data distribution or the functional form of the relationship between the input variables and the target variable. These methods are flexible and can be useful when the relationship is complex or not easily captured by a predefined model structure. Examples of these methods are Random Forests, Neural Networks, SVM, Encoders-Decoders etc.

In this paper, authors choose a non-parametric method, an LSTM-RNN model, which can capture the randomness and non-linearity of traffic flow. They use one hidden layer. Optimum units per layer are denoted using Grid Search Cross Validation method. The dataset they use is the Caltrans PeMS.

Paper 3: Supervised deep learning based for traffic flow prediction [3]

In this paper, authors use a fully connected Neural Network with dropout and batch normalization layers. As they mention, the difference between a fully connected neural network and their model is those two extra types of layers they use. The model is trained using back propagation algorithm.

Given past traffic flow value at previous timesteps, the model predicts the traffic flow value at the next timestep. The datasets they experiment in are the open data of Taipei City and CWB. Also, it should be stated that traffic flow prediction is forecasted only in one single road/edge of the road network.

Paper 4: Traffic Flow prediction with big data: A deep learning approach [4]

In this implementation, authors choose to use autoencoders, which are stacked one on the top of other. By this way, they form a deep network, which is trained using back propagation algorithm. Authors choose to predict the traffic flow at the future timestep, using traffic flow data at previous timesteps. The dataset they use is the PeMS dataset.

Paper 5: Comparative Analysis for traffic flow forecasting in the city of Beijing [5]

This paper contains useful information, about the problem of traffic flow forecasting. Authors compare three different techniques to solve this problem. The dataset they use contain speed data of every 5 minutes. The traffic flow predictions are only done within two bridges in Beijing city.

- The first technique is the use of a parametric model, ARIMA. This model estimates its parameters from past time steps. Then authors use the estimated model to forecast future timesteps of traffic flow. This type of model can only be applied to stationary time series.
- The second technique they use is neural networks (non-parametric method). This model can capture efficiently the non-linearity of the traffic flow. On the other hand, it takes time to find the optimal hyperparameters. Neural Networks require also a big amount of training data to perform good predictions.
- The third technique they use is non-parametric regression. In this case, authors use the K nearest neighbors' algorithm to group past values of time series. By this way, they classify time series with same patterns, making it easier to predict future traffic flow in each of them.

Paper 6: DeepCrowd-A Deep Model for Large-Scale Citywide Crowd Density and Flow Prediction [6]

The problem that this research tries to solve is traffic prediction. In other words, they predict not only how many objects will be in each location at the next timestamp (they call it crowd density prediction), but also how many objects will leave and visit each location at the next timestep (they call it crowd in out flow prediction).

To form the road network, they split a large urban area to a number of mesh-grids. Then, citywide crowd density in a continuous period can be represented with a four-dimensional tensor $R^{Timestep \times Height \times Width \times Channel}$ in an analogous manner to video data, where each Timestep can be seen as one video frame, Height and Width is two-dimensional index for mesh-grids, and each Channel stores an aggregated scalar value for each mesh-grid.

The model that they use is called DeepCrowd. This model consists of convolutional LSTMs in the form of pyramid. Convolutional LSTMs handles high dimensional data, while the pyramid architecture better utilizes low and high resolution feature maps.

The input to the model is multiple historical spatial domains of the road network. The output is the next traffic flow prediction result. Temporally, the next step of citywide crowd density and in-out flow can be correlated with historical data.

In this research, the dataset being used reflects the real-world citywide data. The data are of high quality (cover a large spatial area, many user samples, use of finer mesh-sizes). Also, this data is collected by GPS using a popular smartphone application. Using this data, researchers convert them into 4D tensors: $R^{Timestep, Height, Width, Channel}$. Moreover, data are from two big cities: Tokyo and Osaka. (to get this data, an application to the owner is required).

GitHub Link: <https://github.com/deepkashiwa20/DeepCrowd>

Paper 7: Diffusion Convolutional Recurrent Neural Network (DCRNN) [7]

In this research, the problem being solved is traffic forecasting on road networks. Authors predict the future traffic speeds of a sensor network, given historical data (previously observed traffic flow) from N correlated sensors. In this study, the data are collected using sensors, which are placed in many places in the road network, and thus creating a sensor network.

They represent the sensor network as a weighted directed graph $G = (V, E, W)$, where V is a set of nodes $|V| = N$, E is a set of edges and $W \in R^{N \times N}$ is a weighted adjacency matrix representing the nodes proximity (e.g., a function of their road network distance). Nodes are sensors and edge weights denote proximity between the sensor pairs measured by the road network distance.

The model being used is called Diffusion Convolutional Recurrent Neural Network (DCRNN). The structure of the model is based in the Seq2Seq architecture. Seq2Seq architecture consists of an encoder and a decoder. Both are Recurrent Neural Networks with DCGRUs (Diffusion Convolutional Gated Recurrent Units). Like Gated Recurrent Unit (GRU), DCGRU can be used to build recurrent neural network layers and be trained using backpropagation through time.

During training they feed the historical time series into the encoder and use its final states to initialize the decoder. The decoder generates predictions given previous ground truth observations.

What is the input and the output to the model? Researchers denote the traffic flow observed on G as a graph signal $X \in R^{N \times P}$, where P is the number of features of each node (e.g., velocity, volume). Let $X(t)$ represent the graph signal observed at time t , the traffic forecasting problem aims to learn a

function $h(\cdot)$ that maps T' historical graph signals to future T graph signals, given a graph $G: [X(t-T'+1), \dots, X(t); G] \rightarrow [X(t+1), \dots, X(t+T)]$.

The experiments are conducted on two big datasets: METR-LA and PEMS-BAY. The first dataset (METR-LA) contains traffic information collected from loop detectors in the highway of Los Angeles County. They select 207 sensors and collect 4 months of data ranging from Mar 1st, 2012, to Jun 30th, 2012, for the experiment. The second dataset (PEMS-BAY) is collected by California Transportation Agencies (CalTrans) Performance Measurement System (PeMS). Scientists select 325 sensors in the Bay Area and collect 6 months of data ranging from Jan 1st, 2017, to May 31st 2017 for the experiment. In both datasets, 70% of data is used for training, 20% are used for testing while the remaining 10% for validation.

GitHub Link: <https://github.com/liyaguang/DCRNN>

Paper 8: T-LSTM: a long short term memory network enhanced by temporal information for traffic flow prediction [8]

This paper uses the temporal information (time information) to predict traffic flow on a single road section. In other words, they use features associated with time to predict traffic flow. The prediction is short-term. Researchers believe that traffic flow is not the same in each time period of the day – traffic flow is high in rush hours and low in the early hours of each day.

The model being used is called Temporal information enhancing Long ShortTerm Memory neural network (T-LSTM). This model combines recurrent time labels with recurrent neural networks, which makes the best use of the temporal features to improve the accuracy of short-term traffic flow prediction. The input to the model is a vector $x = (x_1, x_2, x_3, \dots, x_n)$, where each x_i contains a time label and the traffic flow information. The input to the LSTM model is a 8×2 matrix (8 historical data are used to predict traffic flow of the next moment). The output of the model is a 2D vector $y = (\text{time label}, \text{traffic flow})$ of the next moment prediction.

The traffic detector data from Shibalidian Bridge to Hongyan Bridge of the East Fourth Ring Road in Beijing (March 1 to August 30, 2014) is selected to validate the T-LSTM. This dataset has the following columns: Speed, Flow, Date, Density. Also, there is not a link to this dataset. For purpose of research and analysis, the data are aggregated into the time intervals of 16 minutes. So, there are 90 ($720 \times 2 \div 16$) pieces of data and corresponding time labels for each day. The data of the first five months are used for training and the data of August are used for testing. Finally, the data are normalized to $[0, 1]$ by the Min-Max Scaler normalization method in the Scikit-learn library.

Paper 9: Hybrid LSTM Neural Network for Short-Term Traffic Flow Prediction [9]

This paper tries to predict the short-term traffic flow of cars in a real road network. Traffic flow here means the expected number of cars that will flow in the road network. The model is a Hybrid LSTM model, which is based on the LSTM model. Scientists did not use the simple LSTM model here as it does not provide good accuracy results on traffic flow prediction. The Hybrid LSTM model consists of dense,

dropout, LSTM, and activation layers. The Dropout layer means that some neural units are temporarily discarded from the neural network according to a certain probability during the training process, preventing over-fitting. The Activation layer is mainly used to improve the learning ability of the model.

What is the input and the output to the model? Training data (70%) of the format $X = [\text{Current Node}, \text{Source Node}, \text{Traffic Flow}]$. Output Y is the traffic flow prediction on each road network of the road.

The experimental dataset is the local road network traffic data from September 1st to September 8th in Yunyan District, Guiyang City, Guizhou Province. Inside the road network there are traffic lights. In this research, people tried to measure the flow between traffic lights every 30 seconds. The data collected are of the following format (have the following columns): Current Node (TrafficLightID, Source Node (FromID), Traffic Flow (traffic_flow) TrafficLightID denotes the traffic light identifier of the vehicle currently, while FromID denotes the traffic light identifier of the vehicle at the last moment.

Because the experimental data of each day is collected at a 30s interval from 6:00 a.m. to 8:00 p.m., the traffic_flow field is a matrix of dimension 1680, which is used to represent the traffic flows of a road section at a 30s interval. For example, traffic_flow[0] represents the traffic flow from road intersection FromID to road intersection TrafficLightID at t_0 .

In this paper, authors try to transform time series into supervised learning problem and evaluate the accuracy of model prediction. So, they divided the traffic flow data set into training set and test set, where the first 70% of the traffic flow data set is the training set and the last 30% of the traffic flow data set is the test set. After training a model based on the training set, the use of RMSE error metric will be used to evaluate the prediction accuracy.

Paper 10: Graph Hierarchical Convolutional Recurrent Neural Network (GHCRNN) for Vehicle Condition Prediction [10]

In this paper, the problem being solved is prediction of urban vehicle flow and speed. Authors use a Graph Hierarchical Convolutional Recurrent Neural Network as their final model. This model effectively extracts the hierarchical structure information of the road network, eliminates noise, improves efficiency and reduces memory usage rate.

The input to the model is a series of historical data of vehicle flow within the urban road network. The output is another series of forecasted traffic flow at future timesteps.

It is worth mentioning that the model proposed in this paper contains Pooling and Unpooling layers, which can effectively extract the hierarchical structure information of the road network and reduce the time and space complexity.

The speed data of Los Angeles is the average speed of each detection station in 5 minutes as one data. A total of 207 stations are selected and the period is 4 months, so there is total of 207×34272 data, the vehicle speed of each node in the graph is one data in a time period and input to the model.

Paper 11: Short-Term Traffic Flow Prediction Based on XGBoost [11]

In this paper, the authors aimed to solve the problem of short-term traffic flow prediction in a specified lane. They gathered traffic information from detectors, including the number of vehicles and cars, time occupancy, and average speed, as well as additional contextual features such as time, the number of lanes and sections, and traffic flow information from the previous section. The data was collected from 30 monitoring sections of the Fourth Ring Road in Beijing, with the training set consisting of data from June 24 to 29, 2008, and the testing set from June 30. They obtained a total of 862,859 training samples and 153,095 test samples.

To address the prediction task, the authors proposed a methodology that combined wavelet decomposition and reconstruction with the XGBoost model. Wavelet decomposition was applied to preprocess the training labels, removing high-frequency noise and retaining the long-term regularity of traffic flow. The reconstructed labels were then used as the target variable in the XGBoost model. They compared the performance of their approach with that of the Support Vector Machine (SVM) model and XGBoost without wavelet processing. The model parameters for XGBoost were determined through a grid search. Evaluation metrics such as Root Mean Square Deviation (RMSE) and Mean Absolute Percentage Error (MAPE) were used to analyze and compare the prediction results.

The results showed that the combined methodology of wavelet decomposition, reconstruction, and XGBoost outperformed the SVM model and XGBoost without wavelet processing. The wavelet techniques helped to reduce the influence of noise and improve the robustness of the prediction model. The XGBoost algorithm, with its gradient boosting framework, proved to be effective in capturing complex patterns and achieving high prediction accuracy. The comparison of observed traffic flow data and predicted values demonstrated the superiority of the proposed approach, with more accurate results for long-term traffic flow and smoother predictions for short-term periods.

The authors' technique involved the utilization of time-series data for short-term traffic flow prediction. They incorporated wavelet decomposition and reconstruction to preprocess the training labels and enhance the model's robustness. The XGBoost model was chosen as the predictive algorithm due to its powerful gradient boosting capabilities. By comparing their methodology with SVM and XGBoost without wavelet processing, the authors demonstrated the superiority of their approach in terms of prediction accuracy and efficiency. Overall, their combined methodology of wavelet decomposition, reconstruction, and XGBoost proved to be valuable for short-term traffic flow prediction, providing more accurate and smoother predictions for the specified lane.

References

- [1] Kashyap, A. A., Raviraj, S., Devarakonda, A., Nayak K, S. R., Santhosh, K. v., & Bhat, S. J. (2022). Traffic flow prediction models—A review of deep learning techniques. In *Cogent Engineering* (Vol. 9, Issue 1). Cogent OA. <https://doi.org/10.1080/23311916.2021.2010510>
- [2] Liu, X., IEEE Computer Society, Institute of Electrical and Electronics Engineers., IEEE International Conference on Social Computing and Networking (8th : 2015 : Chengdu, C., IEEE International Conference on Sustainable Computing and Communications (5th : 2015 : Chengdu, C., International Conference on Big Data Intelligence and Computing (2015 : Chengdu, C., & International Symposium on Cloud and Service Computing (5th : 2015 : Chengdu, C. (n.d.). *2015 IEEE International Conference on Smart City : SmartCity 2015 : proceedings : 19-21 December 2015, Chengdu, China*.
- [3] *Supervised_Deep_Learning_Based_for_Traffic_Flow_Prediction*. (n.d.).
- [4] Lv, Y., Duan, Y., Kang, W., Li, Z., & Wang, F. Y. (2015). Traffic Flow Prediction with Big Data: A Deep Learning Approach. *IEEE Transactions on Intelligent Transportation Systems*, 16(2), 865–873. <https://doi.org/10.1109/TITS.2014.2345663>
- [5] Rong, Y., Zhang, X., Feng, X., Ho, T. K., Wei, W., & Xu, D. (2015). Comparative analysis for traffic flow forecasting models with real-life data in Beijing. *Advances in Mechanical Engineering*, 7(12). <https://doi.org/10.1177/1687814015620324>
- [6] Jiang, R., Cai, Z., Wang, Z., Yang, C., Fan, Z., Chen, Q., Tsubouchi, K., Song, X., & Shibasaki, R. (2023). DeepCrowd: A Deep Model for Large-Scale Citywide Crowd Density and Flow Prediction. *IEEE Transactions on Knowledge and Data Engineering*, 35(1), 276–290. <https://doi.org/10.1109/TKDE.2021.3077056>
- [7] Li, Y., Yu, R., Shahabi, C., & Liu, Y. (2017). Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting. <http://arxiv.org/abs/1707.01926>
- [8] Mou, L., Zhao, P., Xie, H., & Chen, Y. (2019). T-LSTM: A long short-term memory neural network enhanced by temporal information for traffic flow prediction. *IEEE Access*, 7, 98053–98060. <https://doi.org/10.1109/ACCESS.2019.2929692>
- [9] Xiao, Y., & Yin, Y. (2019). Hybrid LSTM neural network for short-term traffic flow prediction. *Information (Switzerland)*, 10(3). <https://doi.org/10.3390/info10030105>

[10] Lu, M., Zhang, K., Liu, H., & Xiong³, N. (n.d.). Graph Hierarchical Convolutional Recurrent Neural Network (GHCRNN) for Vehicle Condition Prediction.

[11] Dong, X., Lei, T., Jin, S., & Hou, Z. (2018). Short-term traffic flow prediction based on XGBoost. Proceedings of 2018 IEEE 7th Data Driven Control and Learning Systems Conference, DDCLS 2018, 854–859. <https://doi.org/10.1109/DDCLS.2018.8516114>