Available online at www.sciencedirect.com

ScienceDirect

Transportation Research Procedia 48 (2020) 1072–1083

Transportation Research Procedia

www.elsevier.com/locate/procedia

World Conference on Transport Research – WCTR 2019, Mumbai, 26-30 May 2019

Predicting Traffic Volume and Occupancy at Failed Detectors

Ishan Tarunesh[a], Edward Chung[b]

[a]*Indian Institute of Technology, Bombay, India*
[b]*Department of Electrical Engineering, The Hong Kong Polytechnic University, Hong Kong*

Abstract

Accurate Traffic flow prediction relies on correctness of the values received from detectors. It is often the case that detectors are not working correctly and provide with incorrect values. The aim of this work is to predict the traffic flow variables at the failed detectors using deep learning techniques such as neural network and autoencoders.
The major contributions are using neural network to model the complete network of detectors and use of autoencoders to reduce model size by exploiting spatial correlation between detectors. To the best of our knowledge deep learning has never been applied incase of detector failure.

Keywords: Traffic volume prediction; occupancy prediction; machine learning; autoencoder; neural network;

1. Introduction

Arterial Traffic Management Systems (ATMS) use a network of sensors and inductive loop detectors to measure volume, occupancy and speed of vehicles. These are then used to calculate the traffic signal timings at the intersections. For efficient working of ATMS it is essential that the data ATMS receive from detectors is correct. This is at times not the case due to detector failure.

Detector failures are difficult to identify in reality, most often uncovered by direct reporting of such case. Common practice involves performing handshakes which comprise of sending a signal to the detector and waiting to receive acknowledgement. There are multiple techniques [1] - [6] for real time failure detection but they are rarely used in practice due to complexity of scheme.

When a detector failure is identified, the ATMS has to “lock” the detector i.e. keeping it’s status as on all the time and running the phase involving that detector for the maximum time. This is highly inefficient because these timings are seldom retimed and the actual demand maybe less than the allocated maximum green time.

10.1016/j.trpro.2020.08.134

In the past, several machine learning and data mining approaches have been applied for traffic flow prediction [7]-[10]. However, in this paper we aim to tackle the problem of predicting traffic flow variables at failed detectors. The paper employs some of the widely known deep-learning techniques such as Artificial Neural Networks (ANN) and Autoencoders (AE) for traffic flow prediction. To the best knowledge of the authors, deep learning has not been applied for prediction of traffic volume and occupancy in case of detector failure.

The rest of the paper is structured as follows. Section 2 dwells more into the data, justifies the metric of performance and presents multiple deep-learning approaches to model the traffic flow. Section 3 gives direction for future work while section 4 provides concluding remarks.

2. Deep Learning for Predicting Traffic Volume and Occupancy

2.1 Site and Data Description

All analysis in this paper were performed on data collected from detectors installed at Maroochydore, Sunshine Coast, Queensland, Australia for a period of one year starting from 1st April, 2017 to 31st March 2018.

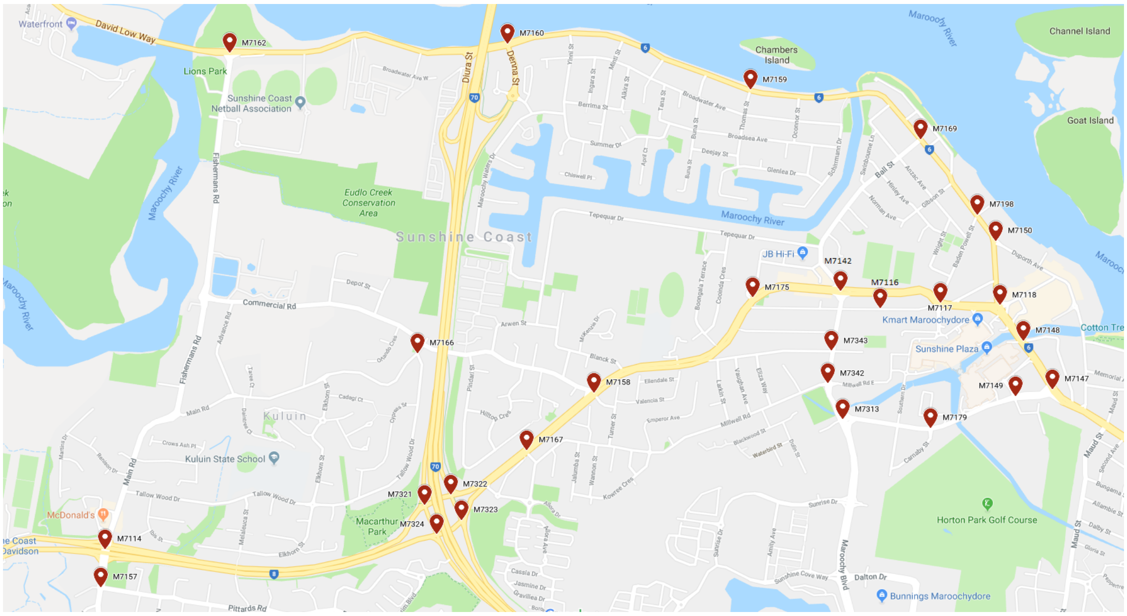

Fig. 1. Maroochydore, Sunshine Coast. The figure shows the 27 detectors pinned in red on which the analysis was conducted.

Maroochydore is an urban center of 55.5 sq. km area consisting of 27 intersections with a total of 232 loop detectors combined. The detectors measure the traffic volume and loop occupancy for different lanes which is used by ATMS for determining signal phase timings. The historical data from STREAMS [11] is aggregated at 15 mins interval throughout the day for a complete year.

The dataset does not contain label for failed detectors. We manually analyze the variation of traffic volume over different months and complete year to find anomalous detectors. The detectors which had either of these abnormality were classified as failed –

(i) Abrupt changes in volume in consecutive 15 min intervals over the day for multiple days

(ii) The graph of daily volume over the year shows sudden change. It is observed at many detectors for prolonged period of time. This could be due to partial failure in which case the detector only reports a fraction of the actual volume.

The detectors which were identified as failed in the above analysis were labelled and their data for the complete year was removed from the training dataset.

The above method for classifying detector failure is sensible but not exhaustive, later in the paper we also look at cross-correlation of detector readings which also serve as a sound method of identification of detector failure. We also observe that the detectors which our analysis fail to model accurately are precisely the ones which are uncorrelated with their neighboring detectors.

2.2 Metrics of Performance

The traffic signal timings are calculated based on the traffic volume and occupancy measured. Hence accurate prediction of volume and occupancy is important. To determine the accuracy of models prediction, the following metrics for volume and occupancy were used.

(A) Volume

Volume measurements at arterial intersections vary a lot depending on lane at which the detector is located. The high-volume detectors present on through lanes have volume ranges of up to 300 vehicles (per 15 min) compared to low-volume detectors present on turning lanes with range around 20 vehicles (per 15 min).

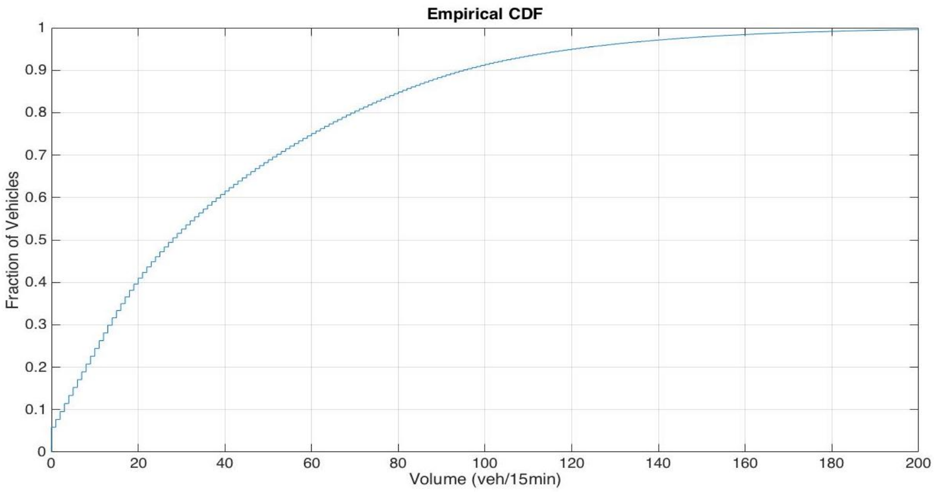

Fig. 2. Shows cumulative plot of fraction of car with given volume between period 6AM - 8PM for the complete year.

To calculate the effectiveness of the model we use a piece-wise loss function which uses mean absolute error (MAE) for low volume range and mean relative error (MRE) for high volume ranges. The point of division between low and high volume values was decided based on the cumulative graph of number of vehicles versus volume.

The time interval of 6AM to 8PM was chosen because it witnesses the maximum variation of volume at detectors.

(B) Occupancy

Occupancy is measured as a percentage hence absolute difference is used as evaluate the effectiveness of our model. We use two intervals of prediction ± 5% and ± 10% from measured occupancy as the prediction interval and report the accuracy based on both metrics.

We defined the accuracy of the proposed models as the number of predicted value that lie within the allowed tolerance of prediction around the measured (actual) value.

2.3 Input Variables

The three most important variables in determining the volume/occupancy at any detector is time, day and month. The traffic pattern on weekdays is different when compared to that on Saturday or Sunday as shown in Fig. 3.

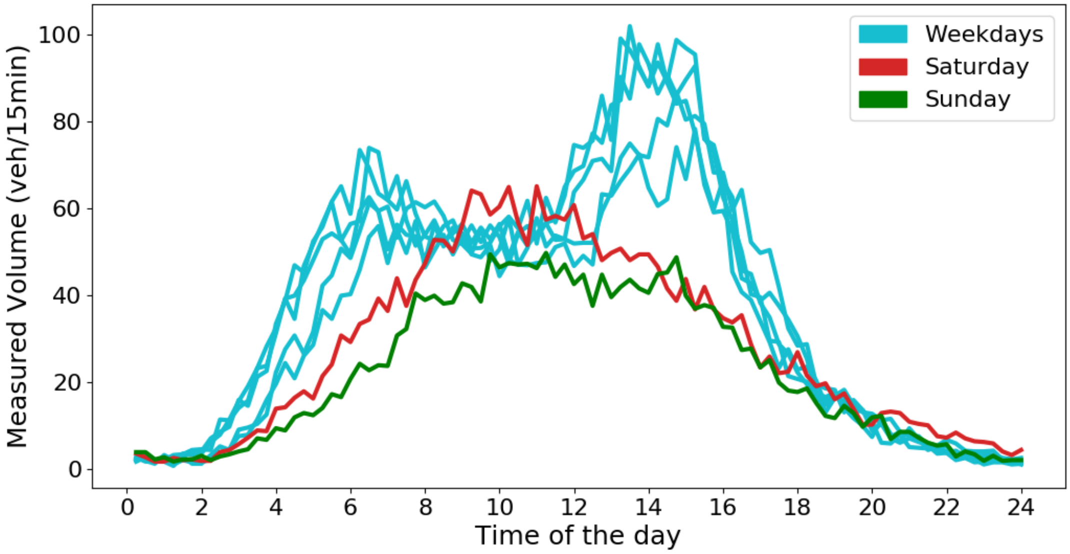

Fig. 3. Variation of volume at detector M7321/VD08 from 5th June (Mon) to 11th Jun (Sun)

There is a strong dependence of traffic flow on the season which can be easily captured using the month as another input parameter to our model [12] - [13].

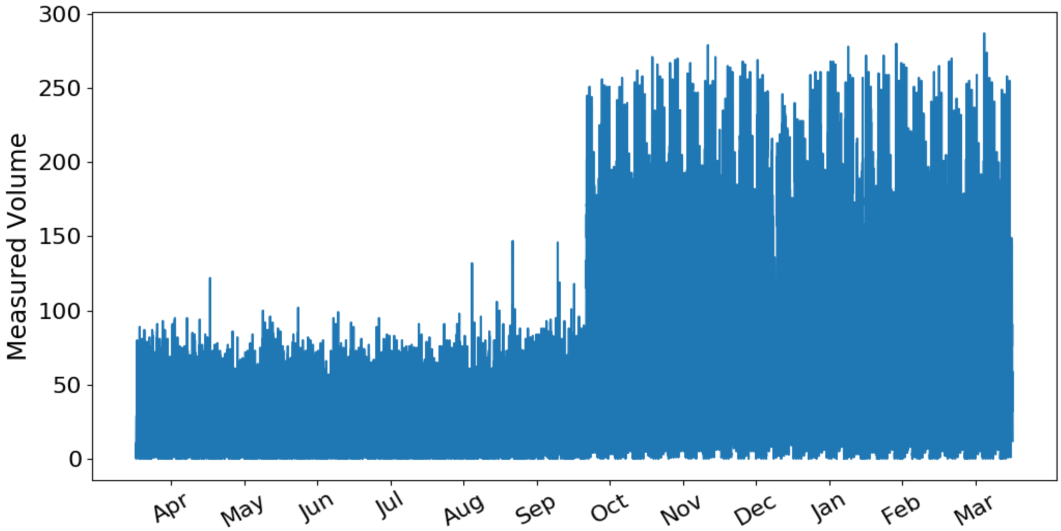

Fig. 4. Variation of volume at detector M7159/VD02 throughout the year.

Apart from these inputs, holidays witness different traffic flow patterns compared to non-holidays. We create a list of public and school holidays for the year April 2017 to March 2018 and added day type as another input (0 – normal day, 1 - school holiday, 2 - public holiday) [14]

2.4 Artificial Neural Network Models

This analysis was done in three steps of increasing complexity, i.e. single intersection, corridor of 5 intersections and complete network. Neural networks were used to predict the volume and occupancy at all detectors given the primary inputs (time, day, month and day type). We also analysed models that use past intervals traffic flow data along with primary inputs for prediction. These models used last 15-min (t-1), last 30-min (t-2) and last 60-mins (t-4) volume and occupancy.

We started with intersection M7142 on Maroochydore road which has 16 detectors. Regarding the structure of the model we needed to determine number of hidden layers and number of units per layer. We choose number of units from {32, 64, 128, 256, 512} and number of hidden layers from {1, 2, 3, 4, 5, 6, 7, 8}. After performing grid search we obtained the best architecture as mentioned in table 3. All models were trained at epoch = 500, learning rate = 0.01 using stochastic gradient descent as optimiser and mean squared error as the loss function.

The dataset for basic model consisted of a year of data at all detectors at M7142. After shuffling the dataset, 2000 rows were chosen for testing and the rest for training purpose (10%-90% split between testing and training).

Two testing strategies were used, first for partially failed detectors which were known beforehand we used the timespan when they were working and tested on them and second by artificially simulating detector failure. Detector failures were simulated by starting with 0 value at the failed detector at the start of the day (t_0=0) and consequently using the models output at the failed detectors as the past interval (t-1) detector reading for the next interval. Only values at the failed detectors were updated using the model's output. Since we did not have actual reading for failed detectors we had to heavily rely on this artificial simulation for testing and comparison. But later in our analysis we show that we could predict the flow at a detector reliably using the readings from nearby detectors due to heavy cross-correlation.

For models using past interval data (e.g. Basic & (t-1)) for prediction of next interval, we chose 30 random days and ran the model for throughout the day using the above mentioned procedure. The summary of the results is mentioned in table 1.

Table 1. Showing results of all models at a single intersection

Location	**Model**	**Volume Accuracy**			**Occupancy Accuracy**	
		≤ 20 veh/15 min	**> 20 veh/15 min**	**Total**	**± 5 %**	**± 10 %**
Single Intersection	Basic	88.7	79.2	85.1	71.5	88.2
	Basic & t-1	89.6	84.8	87.8	75.7	91.7
	Basic & t-2	90.4	85.9	88.6	76.6	92.5
	Basic & t-4	91.0	85.8	89.0	76.5	92.5

The second part of the analysis involved building similar models for a corridor of intersections. This aim was to understand the scalability of models to multiple intersection and observe whether model is able to learn the spatial correlation between detectors at nearby intersection. The corridor chosen consisted of roads Duporth Ave and Horton Parade with 5 intersections namely M7147, M7148, M7118, M7150 and M7198. The training and testing procedure used was the same as for single intersection. The summary of results is mentioned in table 2.

Observing the results of both single intersection as well as corridor of intersections, one could see that the model is able to pick up the spatial relations between detectors.
For the third part of the analysis we trained neural networks for predicting the traffic flow variables at all the 232 detectors. Both basic model and basic & t-1 models were tested (basic & t-2 model and basic & t-4 models were

unnecessarily large and hence not mentioned in the results)

Table 2. Showing results of all models at single intersection, corridor and network

Location	Model	Volume Accuracy			Occupancy Accuracy	
		≤ 20 veh/15 min	> 20 veh/15 min	Total	± 5 %	± 10 %
Single Intersection	Basic	88.7	79.2	85.1	71.5	88.2
	Basic & t-1	89.6	84.8	87.8	75.7	91.7
Corridor	Basic	86.9	79.0	84.1	70.7	87.4
	Basic & t-1	89.3	82.9	86.3	73.1	89.9
Network	Basic	82.9	78.1	81.6	68.1	84.5
	Basic & t-1	86.7	82.6	85.0	72.3	88.7

The dataset contained missing data values for some detectors and after removing those days we were left with around 160 days of clean data. Again 2000 random data rows were taken for testing basic model and 30 random days were chosen for models utilizing historical data. (Approx. 20%-80% split between testing and training).

Key observations from this table were that model accuracy does not decrease much when going from a single intersection to a complete network. This means that neural networks are able to learn spatial relations between detectors fairly accurately.

There is a significant increase of 3 percent on going from Basic model to Basic & t-1 model. This means that past interval data is very meaningful to predict the next interval. Adding more 15-min interval improves the prediction accuracy only marginally at the cost of larger model size and training time.

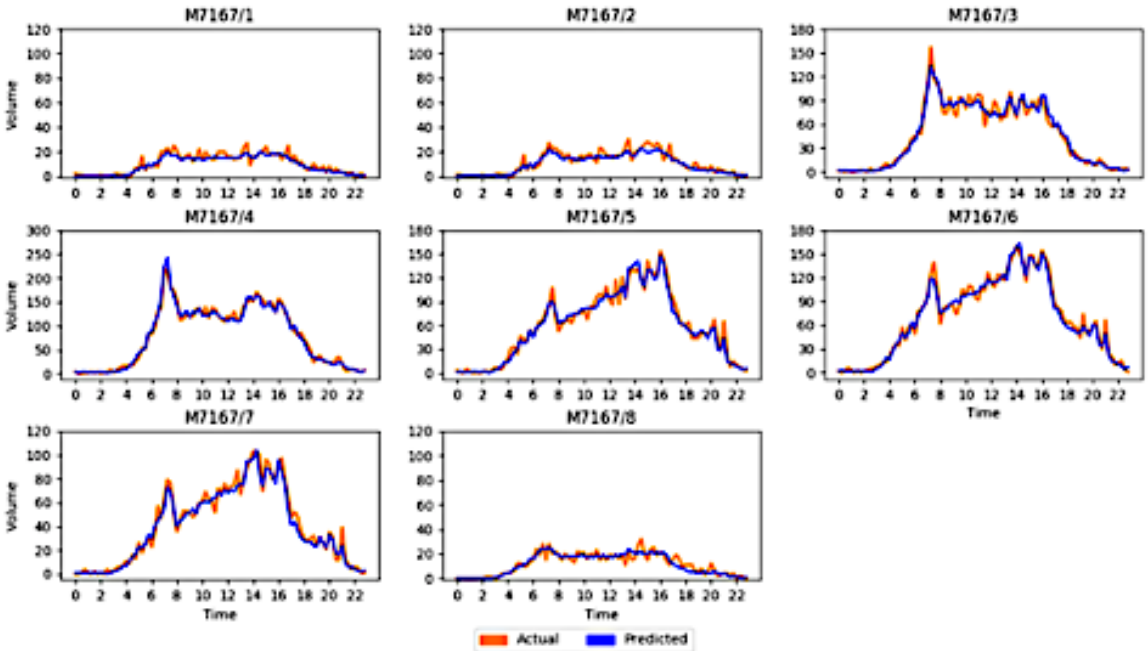

Fig. 5. Measured vs predicted for all detectors at intersection M7167 on 17th April, 2017 (Monday).

The model generalises well even for multiple detector failures. One of the key testing involved shutting down all the detectors at an intersection and using the model for prediction. Figure 5 shows the result of shutting down all 8 detectors at the M7167 intersection.

Another test for robustness was whether model is able to predict well on the weekends. Figure 6. shows the result of model predictions at 3 detectors of M7160 on a Sunday.

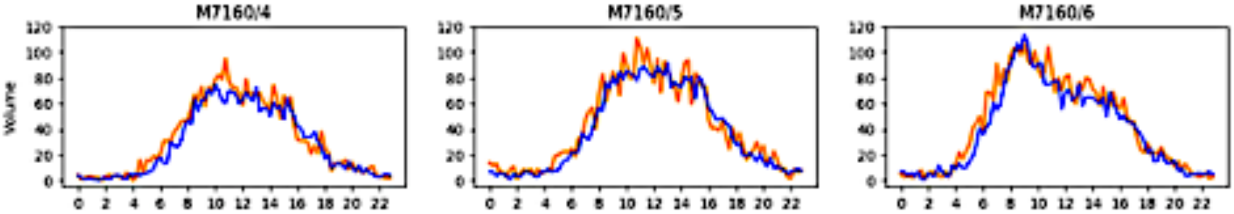

Fig. 6. Detector Failure at three detectors of M7160 on 11th November, 2017 (Sunday).

The result confirms that the model is able to accurately predict the traffic flow on weekends as well. There was still one issue with training models on complete network. The model size (and training time) is quite large. Table 3 shows the best architecture found in the grid search process for different models.

Table 3. Best Architecture found for each model through grid search

Location	Model	Volume		Occupancy	
		Hidden Layers	**Hidden Units**	**Hidden Layers**	**Hidden Units**
Single Intersection	Basic	4	128	5	128
	Basic & t-1	4	128	5	128
Corridor	Basic	5	128	5	128
	Basic & t-1	5	128	5	128
Network	Basic	7	256	6	256
	Basic & t-1	6	256	5	256

There is a 8-fold increase in number of parameters on going from 4 layers of 128 hidden units each (output size 16 and input size 20, Total parameters are 20*128 + 128*128*3 + 128*16 ~ 53.7K parameters) to 6 layers of 256 hidden units each (output size 232 and input size 236, Total 447.5K parameters). This reflected similarly in training time as mentioned in table 4

Table 4. Training time comparison for models at single intersection, corridor and network

Model	Architecture	Training Time
Single Intersection Basic & t-1	[128,128,128,128]	10-12 min
Corridor Basic & t-1	[128,128,128,128,128]	15-20 min
Network Basic & t-1	[256,256,256,256,256,256]	90-120 min

The training was done on 1.8GHz dual-core Intel Core i5 with 4GB of memory

As one can see that training neural networks on complete network requires 10-fold increase in training time. Note that time is for a single model and we train 40 models in grid search process (5 values of hidden units, 8 values of hidden layers). Hence the complete grid search process would have taken around 80 hours to complete.

The later part of our work involved visualizing correlations among detectors and using Autoencoders to reduce the model size without compensating a lot on the model accuracy.

2.5 Autoencoder Models

Autoencoders are neural networks trained to attempt to copy their input to their output [16]. The network consists of two parts, encoder whose purpose is to learn a latent and compressed representation of the input [15] and a decoder which uses this representation for reconstructing the input. Autoencoders are used for various tasks like denoising [17] - [18], dimensionality reduction [19] and information retrieval etc.

Using the last 15 min volume/occupancy to predict the next time horizon causes the input size to be large (number of detectors + basic inputs). If we observe the detectors closely we find out that many detectors show the same trend in volume/occupancy variation. Correlation coefficient is a numerical measure of correlation between two variables. It assumes value in range -1 (strong disagreement) to +1 (strong agreement). It is defined as

$$\rho_{xy} = \frac{cov(x, y)}{\sigma_x \cdot \sigma_y}$$

where $cov(x, y)$ is the covariance between x and y whereas σ_x and σ_y represents standard deviation of variable x and y. We calculated the correlation coefficient between all 232 detectors for the same 160 filtered days.

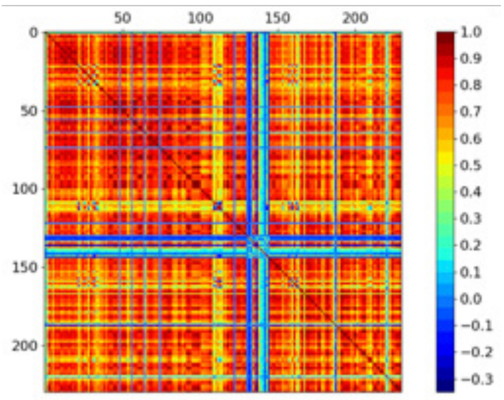

Fig. 7. Showing correlation coefficient matrix between 232 detectors (detectors at same intersection are adjacent)

In the above figure the blue lines represent detectors which are negatively correlated with all other detectors. On closer inspection it was found out that these are precisely the erroneous detectors. There were 18 detectors in that category and were removed from further analysis.

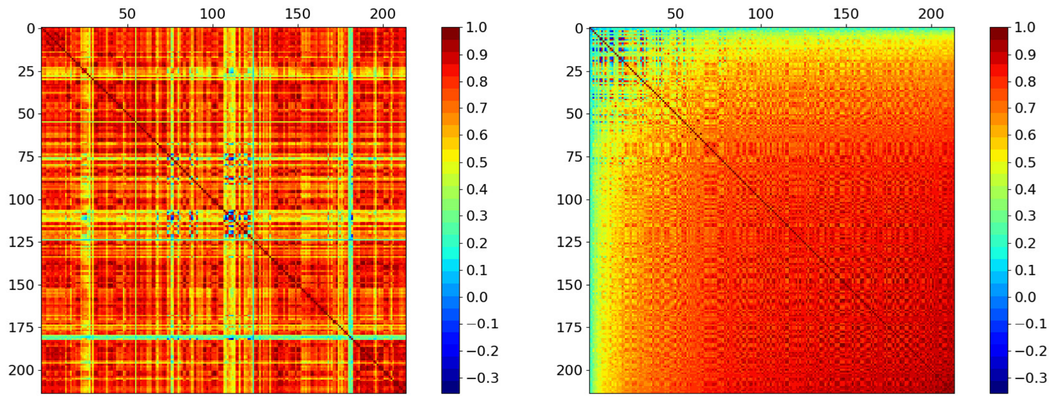

Fig. 8. Showing correlation coefficient matrix for remaining 214 detectors (left - unsorted, right - sorted)

The figure on right is obtained by sorting the rows and columns of matrix on the average correlation with other detectors. A large region of red on bottom right suggests that most of the detectors are very strongly correlated with each other. This observation supports our use of autoencoders to learn a compressed representation of the detector

values and use it along with time, day and month to predict the traffic flow at next interval.
In order to reduce model size, we split it into two parts – (i) encoder model and (ii) the prediction model. The encoder model uses Autoencoders to reduce the input size of 232 detectors to some smaller dimension by learning a compact representation. The prediction model is an ANN as before with lesser number of parameters since it uses the compressed representation of encoder model as input.

One advantage in training comes from the fact that the new model can be trained in separately rather than in an end-to-end fashion. The autoencoder is trained by minimizing input reconstruction loss (L2 loss between input and decoder output). The output of the encoder is the compressed representation of the detector values which we are interested in.

Both single layer autoencoder and deep autoencoder were used to learn the compressed representation of detector values. For single layered autoencoder we tried the following values for hidden layer sizes {5, 10, 15, 20, 30, 50, 100, 150}. For deep autoencoder we tried {3, 5} as the number of hidden layers with {5, 10, 15, 20, 30, 50, 100, 150} as the hidden layer sizes. We made sure than the configuration chosen had middle layer as the bottleneck and that the encoder and decoder part were symmetric in structure.
Table 5 shows the best architecture for both autoencoder as well as their reconstruction accuracy. One observation from the results is that deep autoencoder is able to able to compress the input into much smaller feature vector (size 30) compared to single layer autoencoder (size 100) without compromising too much on reconstruction accuracy.

Table 5. Reconstruction Accuracy of different autoencoders (AE)

Model		**Reconstruction Accuracy**		
Type	**Architecture**	**≤ 20 veh/15 min**	**> 20 veh/15 min**	**Total**
Single-Layered AE	[232,100,232]	91.2	90.5	91.0
Deep AE	[232,100,30,100,232]	90.7	89.3	90.2

The testing dataset used was 2000 random rows from 160 days (Approx. 20%-80% split between testing and training) and the same metric was used for accuracy.

The model prediction required a two step process now, (i) taking input of all detectors at time t and compressing it using autoencoder and (ii) using this compressed representation along with basic inputs to predict traffic flow at time t + 15 min.

Table 6. Accuracy of final models using both AE and ANN compared with only ANN model

Model	**Volume Accuracy**			**Occupancy Accuracy**	
	≤ 20 veh/15 min	**> 20 veh/15 min**	**Total Accuracy**	**± 5 %**	**± 10 %**
Only ANN	86.7	82.6	85.7	73.3	90.2
Single Layer AE + ANN	84.1	80.3	83.9	71.2	88.5
Deep AE + ANN	84.3	79.6	83.2	70.6	88.2

Using autoencoders for compressing the input causes the model accuracy to reduce by approx. 2%. On the other hand, if we look at the training time of AE + ANN models we see substantial improvement in training time (10-fold reduction). This was the tradeoff involved in the process of selecting smaller models.

Below are the comparative plots of all three models (Only ANN, Single Layer AE + ANN and Deep AE + ANN) on 3rd October 2017 (Tuesday) and 4th March 2018 (Sunday)

Table 7. Comparison of training time of AE + ANN models compared to only ANN model

Model	Architecture of NN	Training Time
Only ANN	[256,256,256,256,256,256]	90-120 min
Single Layer AE + ANN	[128,128,128]	15 min (AE) + 10 min (NN)
Deep AE + ANN	[128,128,128]	12 min (AE) + 10 min (NN)

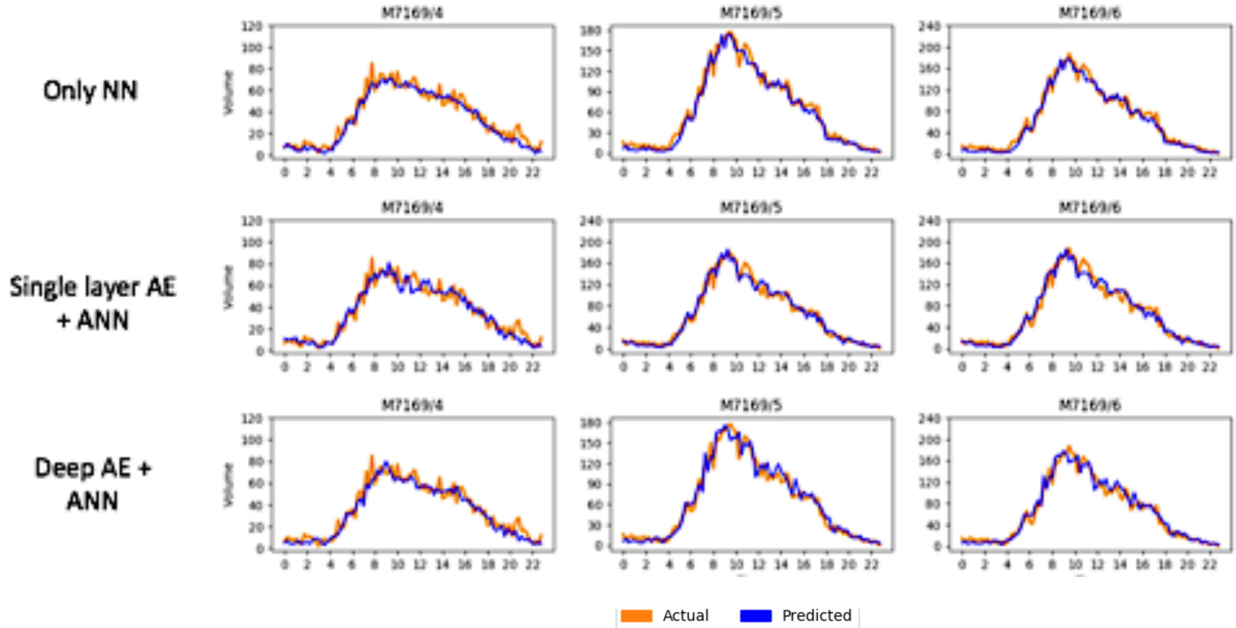

Fig. 9. Showing comparison between all three models on 3rd October Tuesday, 2017

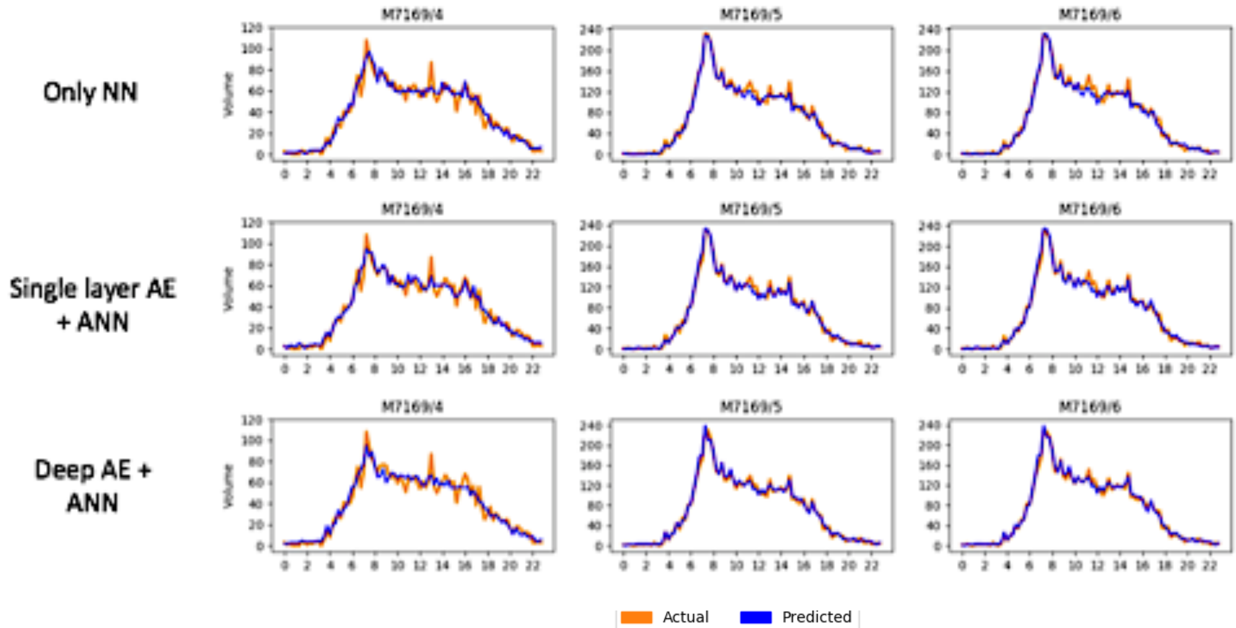

Fig. 10. Showing comparison between all three models on 4th March Sunday, 2018

The model performs very accurately in case of simulated detector failures. The model also predicts accurately on failed detectors (which were manually identified) within their working timespan. Since the dataset didn't have true readings at these failed detectors (for the failure timespan) we had to rely on simulated detector failures for testing. But we expect the model to predict well in the failure timespan as well since the model exploits strong detector cross-correlation.

3. Future Work

In this work, we could not analyse the time-dependent relation between detectors because the data available was aggregated in 15 min intervals. More sophisticated and accurate models could be built on datasets with finer resolution in time. We aim to look at one such real-time open dataset provided by Brisbane City Council. This data is collected from the SCATS (Sydney Coordinated Adaptive Traffic System) for each cycle.

It would be interesting to analyse the patterns in traffic for long weekends i.e. continuous holiday for 3-4 days due to public holidays on Thursday/Friday or Monday/Tuesday which according to knowledge of authors has not been taken into consideration in any models so far.

Another aspect of the work could be identification of anomalous days based on comparing the observed value to the model output.

4. Concluding Remarks

We observed that neural networks are very powerful models when it comes to predicting non-linear relations between detectors and basic inputs. Neural networks along with autoencoders reduces the model size substantially without compromising on accuracy. Though we aimed at modelling single detector failure, the model generalises well even for complete intersection failure.

There is an increase of 3-4% in accuracy if we use the last 15-min traffic flow as input to model. This serves as an evidence that the model is able to discover spatial correlation between detectors. The further increase in accuracy on adding more 15-min intervals is not substantial. This is because the temporal relations are very weak considering the interval size.

The final aim of predicting volume and occupancy incase of detector failure was to use those values to calculate signal timing at intersections. Now since we didn't have cycle by cycle volume/occupancy data at detectors but rather aggregated data at 15 min intervals we could treat the model's prediction as an average over 15 min horizon. Signal timings could then be calculated taking the model's prediction and using number of cycles in that 15 min interval.

Acknowledgement

The authors would like to thank Mr. David Apelt, Transmax and Mr. David Gyles, TMR (Department of Transport and Main Roads) for providing us with the traffic flow data from Maroochydore, Sunshine Coast for this study

References

[1] Leslie N. Jacobson, Nancy L. Nihan and Jeffery D. Bender, "Detecting Erroneous Loop Detector Data in a Freeway Traffic Management System", Transportation Research Record, vol. 1287

[2] Srinivas Peeta and Ioannis Anastassopoulos, "Automatic Real-Time Detection and Correction of Erroneous Detector Data with Fourier Transforms for Online Traffic Control Architectures", Transportation Research Record, vol. 1811

[3] Lelitha Vanajakshi and L. Rilett, "Loop Detector Data Diagnostics Based on Conservation-of-Vehicles Principle", Transportation Research Record, vol. 1870

[4] Chao Chen, Jaimyoung Kwon, John Rice, Alexander Skabardonis and Pravin Varaiya, "Detecting Errors and Imputing Missing Data for Single-Loop Surveillance Systems", Transportation Research Record, vol. 1855[] Investigating Effect of Holidays on Daily Traffic Counts: Time

Series Approach"
[5] Benjamin Coifman, "Using Dual Loop Speed Traps To Identify Detector Errors", Transportation Research Record, vol. 1683
[6] Koichiro Iwaoka and Yuzo Hirotsu, "Failure Detection Method using Aggregated Vehicle Detector Data", IEEJ Transactions on Industry Applications, vol. 137
[7] J. van Lint, S. Hoogendoorn, and H. van Zuylen, "Freeway travel time prediction with State-Space neural networks" Traffic Research Record, vol. 1811
[8] Nicholas G. Polson and Vadim O. Sokolov, "Deep learning for short-term traffic flow prediction", Transportation Research Part C: Emerging Technologies, vol. 79, pg. 1-17
[9] Hongsuk Yi, HeeJin Jung and Sanghoon Bae, "Deep Neural Networks for traffic flow prediction", IEEE International Conference on Big Data and Smart Computing (BigComp)
[10] Yisheng Lv, Yanjie Duan, Wenwen Kang, Zhengxi Li and Fei-Yue Wang, "Traffic Flow Prediction With Big Data: A Deep Learning Approach", IEEE Transactions on Intelligent Transportation Systems, vol. 16,
[11] Transmax, “STREAMS ITS Platform”, (https://www.transmax.com.au/what-we-do/streams/)
[12] J. C. Tanner, "Effect of Weather on Traffic Flow", Nature vol. 169, pg. 107
[13] Ju Sam Oh, Yong Un Shim and Yoon Ho Cho, "Effect of weather conditions to traffic flow on freeway", KSCE Journal of Civil Engineering 6(4), pg. 413-420
[14] Mario Cools, Elke Moons and Geert Wets,"Investigating Effect of Holidays on Daily Traffic Counts: Time Series Approach", Transportation Research Record, vol. 2019
[15] Yoshua Bengio†, Aaron Courville and Pascal Vincent, "Representation Learning: A Review and New Perspectives", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 35
[16] Pierre Baldi, "Autoencoders, Unsupervised Learning, and Deep Architectures", Proceedings of ICML Workshop on Unsupervised and Transfer Learning, PMLR vol. 27, pg. 37-49
[17] Pascal Vincent, Hugo Larochelle, Yoshua Bengio and Pierre-Antoine Manzagol, "Extracting and Composing Robust Features with Denoising Autoencoders", ICML 2008 Proceedings of the 25th international conference on Machine learning, pg. 1096-1103
[18] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio and Pierre-Antoine Manzagol, "Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion", Journal of Machine Learning Research 11 (2010), pg. 3371-3408
[19] Wei Wang, Yan Huang, Yizhou Wang and Liang Wang, "Generalized Autoencoder: A Neural Network Framework for Dimensionality Reduction", 2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops