

A Large Scale Neural Network Model for Crash Prediction in Urban Road Networks

Guangxing Wang¹, Jiwon Kim¹

¹The University of Queensland, Brisbane St Lucia, QLD 4072, Australia

Email for correspondence: jiwon.kim@uq.edu.au

Abstract

This study proposes a Neural Network (NN) classifier model for predicting crashes on freeways and arterials in urban road networks. NN is a biologically-inspired information processing paradigm which is composed of interconnected processing elements (neurons). This study considers the probability of crash occurrence as a class variable and applies an NN model to compute the crash probability to predict a crash occurrence within a given area in the network. Feature variables in this study are traffic condition variables (speed of links). This study develops a data-driven approach for building and evaluating NN models. The models are trained and tested using actual traffic and crash data collected in Brisbane, Australia from 2013 to 2016. The evaluation results show that the proposed models can successfully predict crash occurrence with a desired level of accuracy.

1. Introduction

In recent years, there is a growing trend to apply big data and machine learning techniques to real-time traffic management. Traffic crash prediction is one of the fields that can make use of such technological advance. If traffic managers can anticipate the time and location of a crash occurrence before it happens, the managers will be able to reduce the duration and impacts of unplanned traffic crashes more effectively and proactively manage the network to reduce or prevent the risk of crashes occurring.

A significant progress has been made in the field of traffic crash prediction since 2001. Various machine learning models are applied to this field, such as Bayesian model (Oh et al., 2001; Yu and Abdel-Aty, 2013a), log-linear model (Lee et al., 2002, 2003; Hourdos et al., 2006), logistic regression (Abdel-Aty et al., 2004, 2005; Lee et al., 2006; Abdel-Aty and Pemmanaboina, 2006; Abdel-Aty et al., 2007; Ahmed et al., 2012; Xu et al., 2014), probabilistic neural network (Abdel-Aty and Pande, 2005; Oh et al., 2005), Bayesian belief net (Hossain and Muromachi, 2012), Bayesian hierarchical models (Yu et al., 2013; Yu and Abdel-Aty, 2013a), support vector machine (Yu and Abdel-Aty, 2013b; Sun et al., 2014), and random forest (Xu et al., 2013).

There are, however, limitations in the existing studies, especially when it comes to the applicability of traffic crash prediction models in a real-world context for real-time traffic management. Some research gaps identified from the existing literature are summarised as follows:

- 1) Spatial coverage and network scale: the existing models were mainly tested for freeways and most study networks are limited to the corridor-level. To support traffic and

incident management covering the entire network, a network-level traffic prediction model should be developed and tested, considering various road types and modes.

2) Temporal coverage of crash prediction: the existing studies adopt a 5-minute interval as a prediction horizon, which means that the models can predict a crash occurrence 5 minutes before it may happen. While some studies attempt to predict crash occurrences 20 minutes into the future, most of the existing models focus on the 5-minute prediction horizon, which is considered to be too short for traffic managers to take meaningful actions to mitigate or prevent the potential crashes.

3) Handling imbalanced datasets: Since crashes are rare events, the ratio of crash data to non-crash data is very small (less than 5% of the total traffic observations). Such imbalanced datasets make the distribution of class variable highly skewed and tend to prevent the model from learning the class distribution properly. To address challenges faced with imbalanced datasets, the existing studies apply a sampling technique called *undersampling*, which is to filter a large portion of the non-crash data to create a balanced dataset. Although such a technique provides a temporary solution to handling challenges in model training, fundamental questions on practical solutions to this imbalanced dataset problem in crash prediction remain largely unanswered.

The objective of this study is to establish a traffic crash prediction model for real-time traffic management by addressing the above-mentioned limitations. Extending our previous study that applies a Naïve Bayes model for network-wide crash prediction (Wang and Kim, 2016), this study develops a large-scale Neural Network (NN) model for crash prediction that incorporates link speed information from the whole network and predicts a crash occurrence within a given area in the network up to 3 hours into the future.

2. Neural Network

A Neural Network (NN) is a biologically-inspired information processing paradigm which is composed of interconnected processing elements (neurons). An NN can be regarded as a non-linear function, whose estimation process is determining the value of the parameters of the neurons. In general, constructing an NN model requires the following three steps: (i) defining input (feature) and output (class) variables for the model, (ii) specifying the model structure, and (iii) specifying parameters of the neurons. Figure 1 shows an example of the NN model. The input layer is the input of the model, and the output layer is the output of the model. Each neuron in the input layer (A or B) represents a corresponding feature variable (i.e. predictor, attribute), which is the input of the model. Similarly, each neuron in the output layer (E) represents a corresponding class variable (i.e. target variable), which is the output of the model. The second step is to determine how many neurons are placed in the model and how they connect with each other. Neurons are connected through arrows, which represent the data flow. Each neuron has its own input and output. In Figure 1, the outputs of neuron A and B are the inputs of each of C and D; the outputs of C and D are the inputs of E. The parameters of a neuron are weights and bias, which are used for computing the linear transformation of the inputs. Let N be the number of inputs of the neuron; I_i be the i -th input; W_i be the i -th weight; B be the bias, L be the linear transformation outcome, formula (1) shows the linear transformation process.

$$L = \left(\sum_{i=1}^N W_i * I_i \right) + B \quad (1)$$

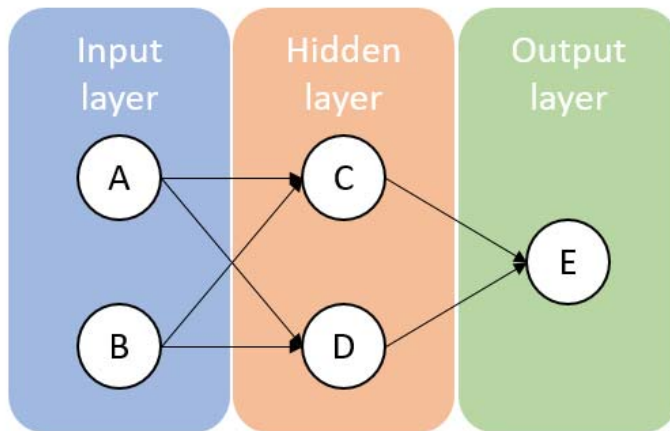
In order to introduce nonlinearity to the mapping from input to output of a neuron, an activation function is applied to compute the output of a neuron. One common selection of activation function is a sigmoid function. In such case, let A be the activation function (sigmoid), O be the output of the neuron, then:

$$O = A(L) = \text{sigmoid}(L) = \frac{1}{1 + e^{-L}} \quad (2)$$

To conclude, the computation of neuron output is sending inputs to a linear map, followed by a non-linear map. In the third step of NN model constructing, the weights and the bias for each neuron are estimated based on historical data.

After the model is constructed, the model output (class variable) can be computed based on model input (feature variable).

Figure 1: An example of the NN model



3. Data and Study Site

In this project, the following set of data was used for the *crash prediction modeling* work:

- **Link speed data** (2013-06-16 ~ 2016-04-05) from TMR through Public Traffic Data System (PTDS). The speed data were collected from a total of 744 links as shown in Figure 2.
- **Crash data** (2013-06-16 ~ 2016-04-05) from TMR through STREAMS Crash Management System (SIMS). The crash data were mapped to the cells shown in Figure 3 to define the class variable at the area-level and to estimate the probability of a crash occurring within each cell.

These two datasets are processed such that traffic speed and crash states are presented at every 3-minute interval, producing a total of 365,936 records.

Figure 2: Locations of 744 links from which traffic speed data were collected

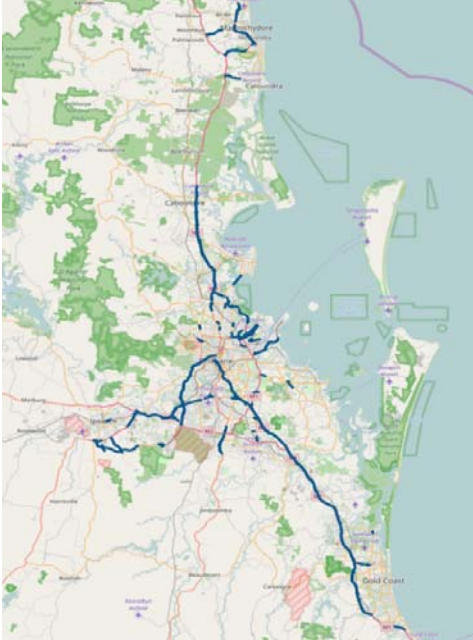
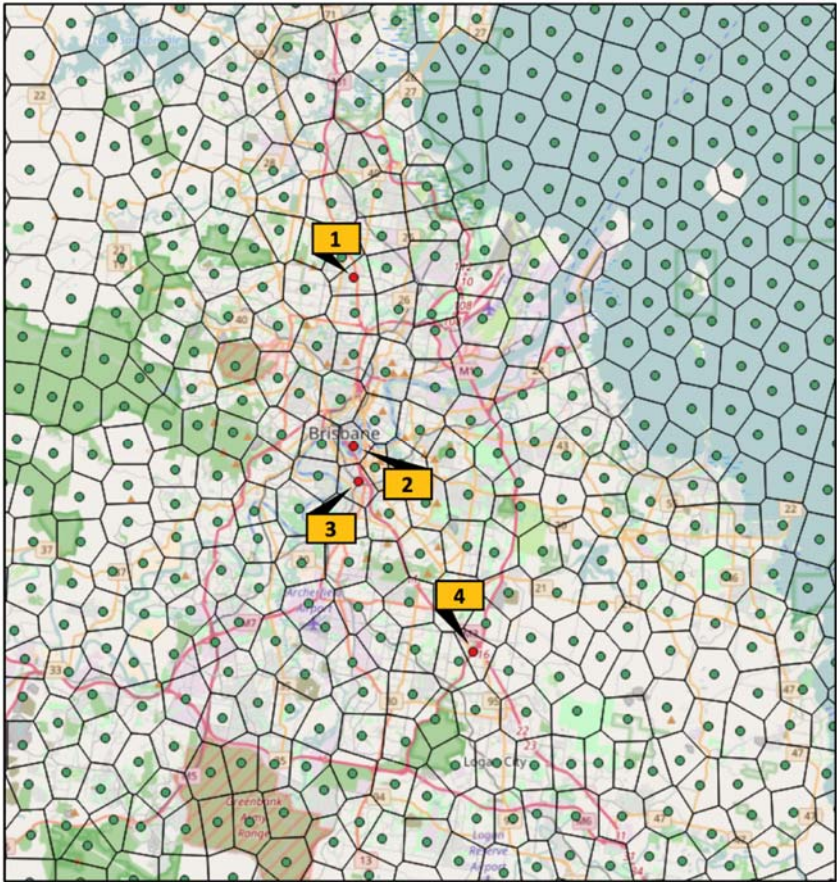


Figure 3: A view of spatial boundaries for which an area-wide crash prediction is performed.



4. Model Development

The value of a class variable in this study only represents the probability of crash occurrence without any location or occurrence time details. Therefore, multiple class variables are constructed for different combinations of Cells (polygon areas defined in Figure 3) and Time windows (time intervals in the future, defined in 5.1). Crash prediction models developed in this study are a set of NN models that have the same feature variables but different class variables. That is, we build a separate NN model for each class variable (i.e., the crash occurrence probability in the corresponding Cell and Time window) but use the same model structure and feature variables across all NN models.

4.1. Defining variables

In each crash prediction model, there are one class variable and a set of feature variables defined as follows:

- **Class variable:** $C_{i,j}$ represents the probability that at least one crash will occur in Cell i during Time window j . Each cell is a polygon area defined in Figure 3. A time window is a 10-minutes time interval in the future. Time window j represents the time interval from $(j - 1) * 10$ minutes later to $j * 10$ minutes later. For example, $C_{4,2}$ represents the probability of crash occurrence in Cell 4 during the period of 10 minutes later to 20 minutes later.

- **Feature variable:** $NS_{m,n}$ represents the normalized traffic speed value of Link m , n minutes ago. For example, $NS_{157,30}$ represents the normalized speed of Link 157, 30 minutes ago. Let $S_{m,n}$ be the corresponding speed (not normalized) of $NS_{m,n}$. Formula (3) shows the computation process of $NS_{m,n}$.

$$NS_{m,n} = \frac{S_{m,n} - \text{mean}(S_{m,n})}{\text{std}(S_{m,n})} \quad (3)$$

In this study, there are 744 links, so m can be one of the numbers from 1 to 744. The normalized speed of 60 minutes ago, 30 minutes ago, and present are selected as features, so n can be 0, 30, or 60. Therefore, each crash prediction model has $744 * 3 = 2232$ features.

As such, for each combination of cell and time window, the associated NN model includes a total of 2233 variables, which consist of 1 class variable and 2232 feature variables. Table 1 presents the list of variables and their definitions.

Table 2 provides an example of data used to build an NN classifier for Cell 3, Time window 5. The data are in the form of a **365936-by-2233 matrix**, where 2233 columns represent the 2233 model variables defined above and 365936 rows represent observations from the period of 2013-06-16 to 2016-04-05.

Table 1: Variables and their definitions for the proposed NN model

Variable Type	Name	Specifics	Variable Defined
Class variable	Crash occurrence probability in Cell i , Time window j ($C_{i,j}$)	- For historical data, the value can only be 0 or 1 because whether crashes have occurred is already known. - For computed outcome, the value range is from 0 to 1.	For each combination of cells and time windows
Feature variable	Normalized speed of Link m , n minutes ago ($S_{m,n}$)	- In each feature variable, the average value is 0 and the standard deviation is 1.	For each combination of links and the time when the speed recorded

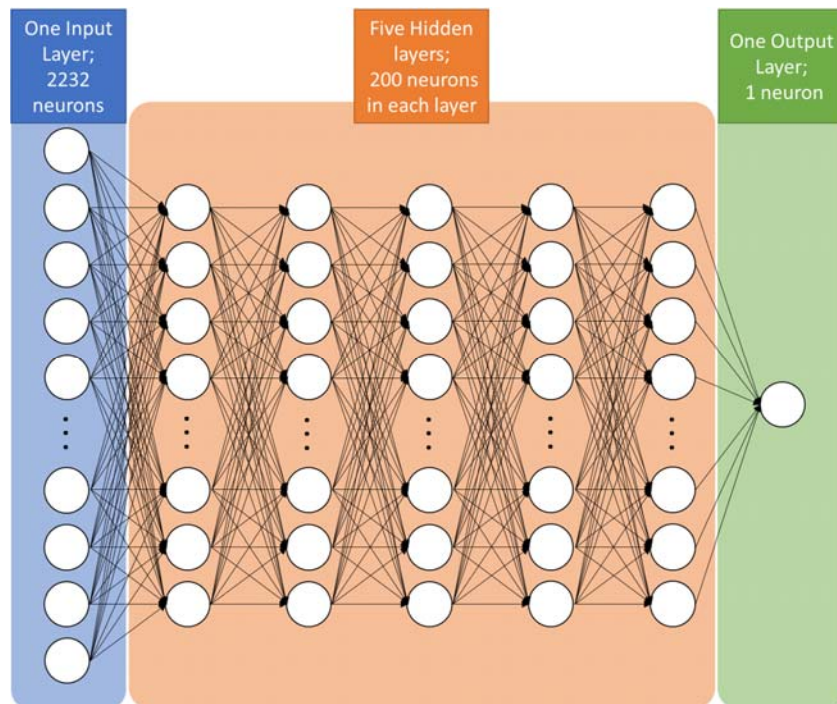
Table 2: An example of processed data for the crash prediction model associated with Cell 3, Time window 5; columns represent 2233 model variables, which consist of 1 class variable + 2232 feature variables, and rows represent observations.

$C_{i,j}$	$NS_{1,0}$	$NS_{2,0}$...	$NS_{744,0}$	$NS_{1,30}$...	$NS_{744,30}$	$NS_{1,60}$...	$NS_{744,60}$
0	0.392	2.022	...	0.014	-0.810	...	0.150	0.996	...	-0.931
0	-0.972	-1.230	...	-1.285	0.422	...	0.019	0.579	...	0.657
0	0.458	-0.096	...	0.181	0.516	...	0.599	0.861	...	-0.276
1	-0.242	-0.763	...	0.658	0.574	...	1.868	-0.760	...	-0.774
1	-1.608	0.248	...	1.284	-0.810	...	1.132	-0.555	...	-1.438
1	0.392	2.022	...	0.014	0.422	...	-0.773	0.937	...	0.072
...

4.2. Model overview

The crash prediction models in this study have the same feature variables and model structure. Figure 4 shows the template of the prediction models. In each model, there are one input layer, five hidden layers, and one output layer. In the input layer, there are 2232 neurons, each of which corresponds to one feature variable. In each hidden layer, there are 200 neurons. In the output layer, there is only one neuron that corresponds to a specific class variable.

Figure 4: The template of crash prediction models



5. Evaluation

Once models are built, it is important to assess their prediction performance based on historical data. For each NN model of this study, the full dataset (with 365936 records, as described in 4.1 and Table 2) are randomly divided into 3 subsets: a training set (305936 records), a validation set (30000 records), and a test set (30000 records). In the evaluation process, each NN model is first trained with training data, then different evaluation methods are applied to the model with test data.

5.1. Preparations and settings

5.1.1. Crash prediction models

In the evaluation process, 4 cells and 18 time windows are selected for assessing the prediction performance of the proposed NN models. Therefore, there are $4 * 18 = 72$ class variables, each of which has a corresponding NN model.

The four cells selected for this validation are shown in the region map in Figure 3. In Figure 3, the cells are constructed using the radius-based partitioning method with the desired cell radius of 1.5 km. Cells with red circles (labeled 1, 2, 3, and 4) are four study sites in Chermiside,

Brisbane City, Woolloongabba, and Eight Mile Plains selected for the model evaluation. The details of the study sites are provided in Table 3. These four sites were chosen because the road segments in these areas are known as crash hotspots, based on the information provided by TMR.

The eighteen time windows are 10-minutes time intervals covering the next 3 hours. In another word, any time point within the next 3 hours corresponds to a specific time window. The time windows are labeled 1 to 18.

For example, if we want to know the probability of crash occurrence 1 hour and 5 minutes later of Cell 3, the corresponding class variable is C_{3-7} . Then the corresponding NN model is called to compute the probability of crash occurrence C_{3-7} .

Table 3: Information on the four selected study sites used for model validation; site number represent the labels shown in Figure 3

Cell ID	Latitude	Longitude	Description
1	-27.3856	153.0309	Gympie Road at Cherside
2	-27.4784	153.0274	Riverside expressway at Brisbane City
3	-27.4894	153.0329	Pacific Motorway at Stanley St at Woolloongabba
4	-27.5834	153.1045	Pacific Motorway / Gateway in both directions at Eight Mile Plains

5.1.2. Threshold probability for classification

Once the crash probabilities are estimated, the next step is to assign a class label (either *Crash* or *Non-crash*) to each row based on a pre-defined threshold probability. The **threshold probability** is a threshold probability beyond which the test result is classified as positive when a specific condition is checked for. The specific condition considered in this study is “will there be a crash in Cell i in Time window j ?” and, if we set the threshold probability of 0.1, the test result for an instance will be classified as ‘positive’ or ‘yes’ (e.g., labelled *Crash*) if the estimated probability of the condition happening is greater than 0.1 and classified as ‘negative’ or ‘no’ (e.g., labelled *Non-crash*) otherwise. Since the choice of the threshold probability directly affects the classification outcome, care must be taken in choosing a threshold probability. For instance, increasing the threshold would result in fewer false positives (fewer cases that wrongly identifies a crash when it has not happened) but more false negatives (more cases that fail to identify a crash when it has happened). A common method for **selecting the optimal threshold probability value** for a given NN model is to find the value that maximizes F1 score. Details of F1 score is provided in the next section. For each model, its own optimal threshold probability value is identified and the prediction results produced under this threshold probability are used to evaluate the model performance.

5.1.3. Performance measures

The proportion of records that have a crash is less than 5%, indicating that the occurrences of the crash are quite rare. As such, evaluating model performance using a simple measure of classification accuracy, which is the percentage that the predicted class matches the actual class, is not meaningful. For example, when the probability of the crash is less than 5% as mentioned above, the classification accuracy will be always greater than 95% even if the classifier predicts everything as “Non-crash” and never assign any instance to the “Crash” class. To address this issue, there are more accurate **performance measures**, which are

Precision, Recall, F₁ score, ROC, AUC, and PRC. The definitions of these measures are provided below:

- **Precision**, also called **positive predictive value**, is the ratio of true positive prediction to all positive prediction, measured at a given threshold probability value. The higher the precision, the more accurate the prediction result. In this study, it equals to the correct crash predictions divided by all crash predictions. In other words, it represents the proportion of crash prediction that is right.

$$Precision = \frac{true\ positive}{all\ predicted\ positive} = \frac{true\ positive}{true\ positive + false\ positive}$$

- **Recall**, also called **sensitivity, true positive rate, or probability of detection**, is the ratio of true positive to all actual positive records, measured at a given threshold probability value. The higher the recall, the more accurate the prediction result. In this study, it is equal to correct crash prediction overall actual crash records. In other words, it represents the proportion of crash records that were correctly predicted by the model.

$$Recall = \frac{true\ positive}{all\ actual\ positive} = \frac{true\ positive}{true\ positive + false\ negative}$$

- **F1 score** takes both **precision** and **recall** into consideration as it is defined as the harmonic mean of precision and recall. As in precision and recall, the higher the F_1 score, the more accurate the result. F_1 score measures the average performance of the prediction at a specific threshold probability value. This thus can be used to select the optimal threshold probability value for a given model, which can be done by finding the threshold value that maximises F_1 score.

$$F_1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$$

- **ROC curve** (Receiver Operating Characteristic curve) is a graphical plot that shows the value of true positive rate (TPR or sensitivity) against the value of false positive rate (FPR or 1-specificity) at various threshold probability values. While Precision, Recall, and F_1 score are threshold-specific measures, that is, they are defined for individual threshold probabilities, ROC curve is a threshold-free measure as it shows pairs of TPR and FPR values calculated at all possible threshold probability values. As such, a ROC curve measures the overall performance of a model.
- **AUC** (Area Under Curve) is the area under the curve of ROC and provides a single score that captures the overall model performance measured by the ROC curve. The higher the AUC, the better the model performance. AUC is 0.5 for random guess and 1.0 for perfect classifiers. AUC scores are convenient to compare the performances of different models.
- **PRC** (Precision Recall Curve) is a graphical plot that shows the value of precision against the value of recall at various threshold probability values. Like ROC, PRC curve is also a threshold-free measure that indicates the overall performance of a model.

5.2. Results

5.2.1. Overview

For each of the 72 combinations of 4 cells and 18 time windows, a correspond NN network is tested. In Figure 5, the test results are averaged among 18 time windows for modeling in each cell. The results show that the precision is between 0.109 and 0.200 and the recall is between 0.109 and 0.137. The highest precision of 0.200 was obtained for Cell 1 at Gympie Road at Chermside, suggesting that 20% of the model predicted crashes were correct. The highest recall of 0.137 was obtained for Cell 4 at Eight Mile Plains, suggesting that 13.7% of actual crashes are correctly predicted by the model. While variations are observed in individual precision and recall values, the overall model performance captured by F_1 score appears to be consistent across sites.

Table 4: Performance measures derived from the crash prediction results for four cells

Cell ID	Precision	Recall	F1	AUC
1	0.200	0.109	0.117	0.851
2	0.109	0.125	0.102	0.827
3	0.168	0.115	0.109	0.838
4	0.165	0.137	0.123	0.866
Average	0.160	0.121	0.113	0.846

5.2.2. Comparison with Logistic Regression and SVM models

To understand if NN is a good model type for this study, we prepare Logistic Regression (LR) models and SVM models to compare with NN models. For each of the 72 NN model, an LR model and an SVM model are built and test with the exact same datasets of the NN model. To compare the prediction performance of the three models among 4 selected cells, Figure 5 and Figure 6 show the AUC and F1 score of the three model types among the 4 test cells. The NN models have the best prediction performance in all cells.

Figure 5: AUC among test cells

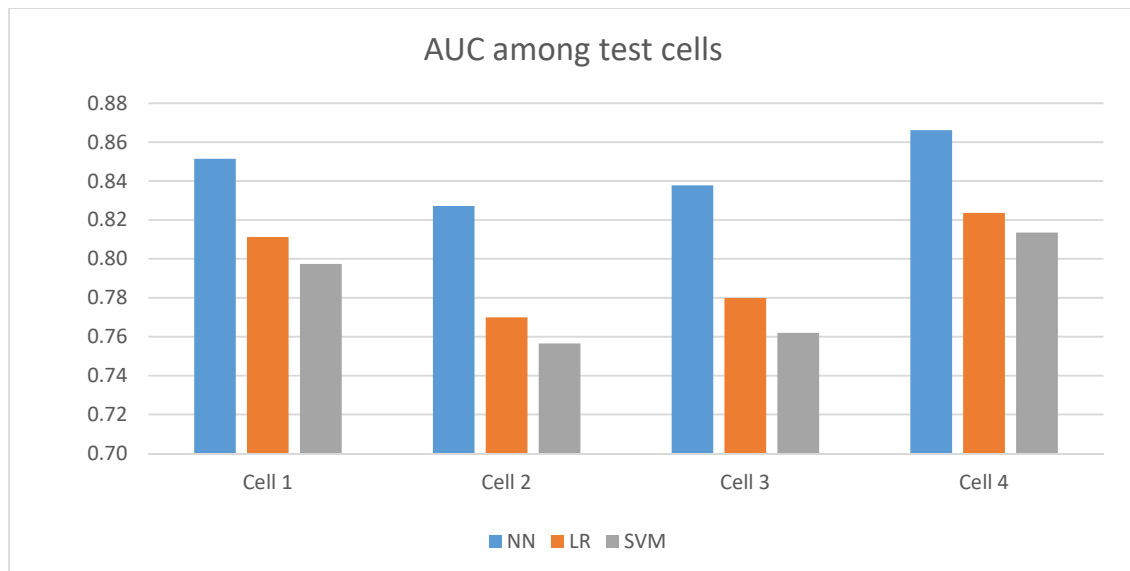
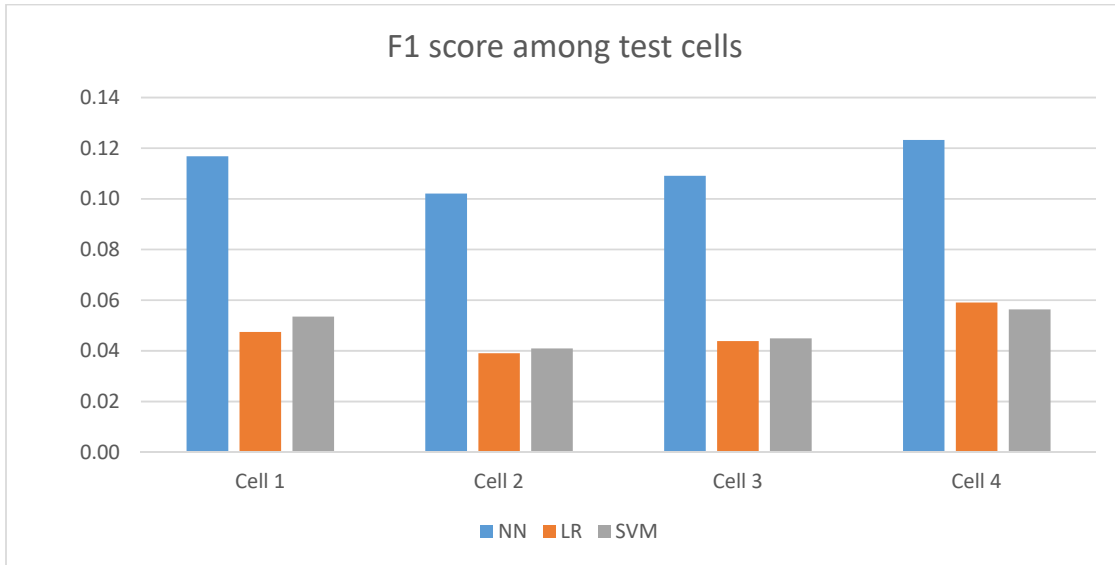


Figure 6: F1 score among test cells



To compare the prediction performance of the three models among 18 time windows, Figure 7 and Figure 8 show how the AUC and F₁ score of the three model types changes over the 18 time windows. The NN models have the best prediction performance in all time windows.

Figure 7: AUC among time windows

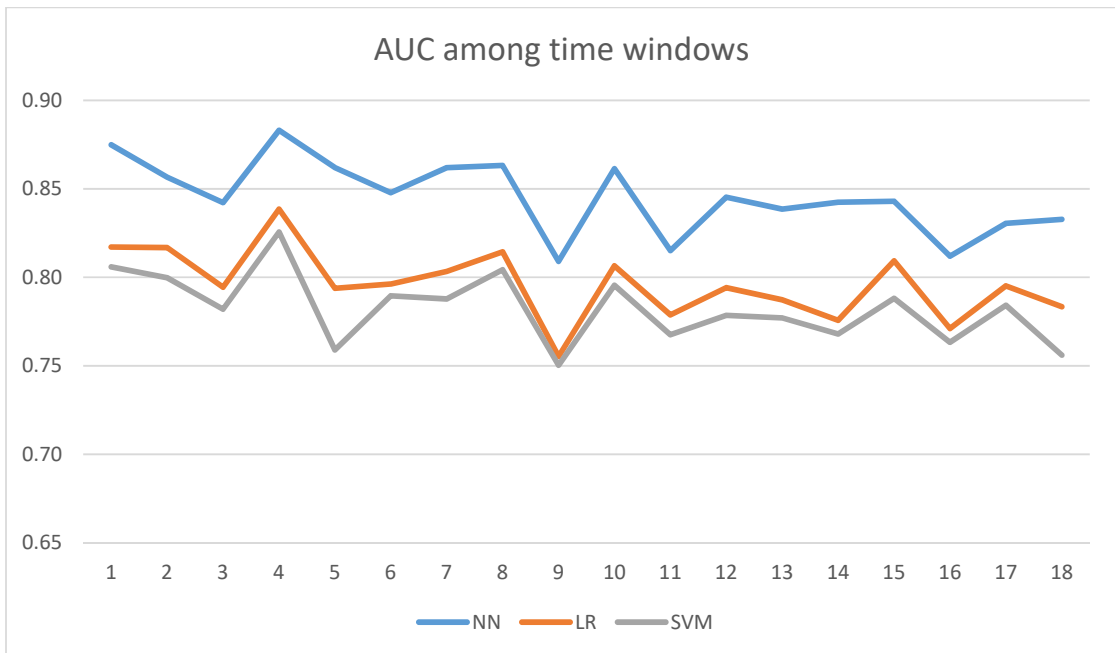


Figure 8: F1 among time windows

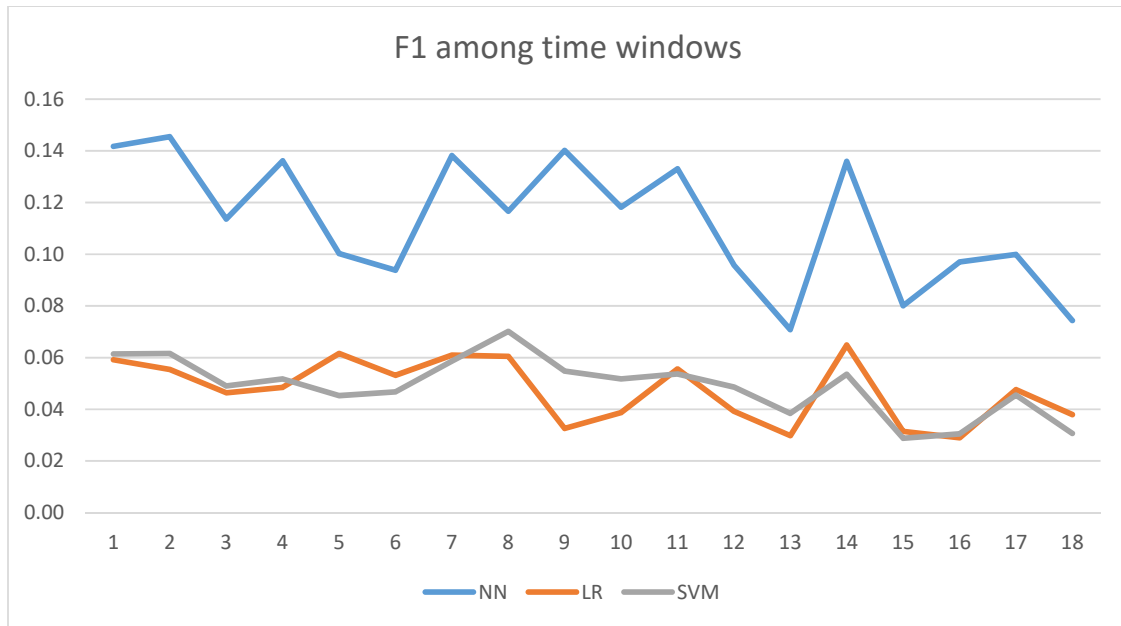


Figure 9 presents the ROC curves which show the overall performance of the three model types. Those are averaged ROC curves, each of which is generated by integrating the probability predictions of the test sets of the 72 models of the corresponding model type. In a ROC graph, the model with better performance is closer to (0, 1), because the lower false positive rate and the higher true positive rate represents better prediction performance. For more information on ROC, readers are referred to introductory articles such as (Davis and Goadrich, 2006; Fawcett, 2006). According to Figure 9, NN is the best model type in ROC test.

Figure 9: The ROC of three model types

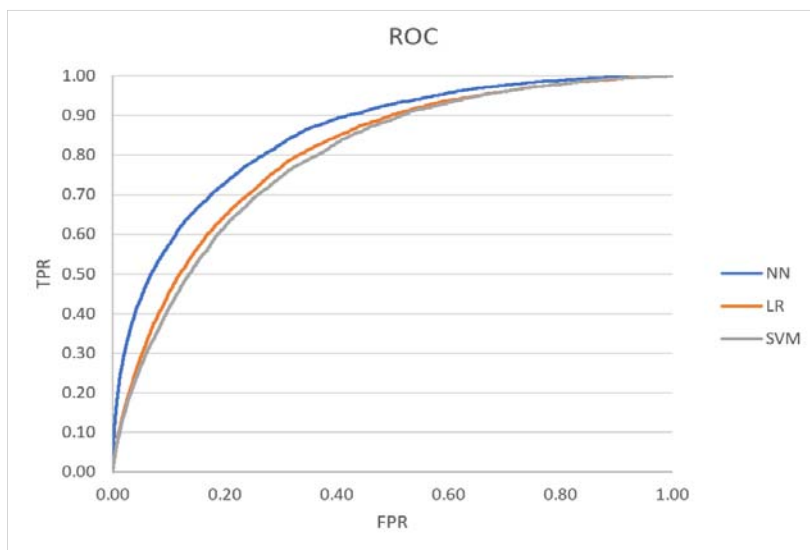
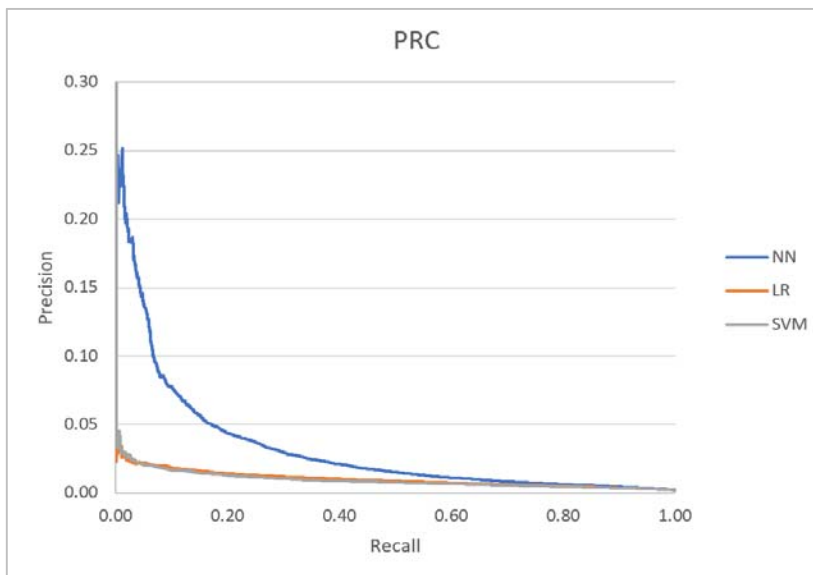


Figure 10 shows the PRC curves of the three model types. Like ROC, the curves are averaged PRC curves for the three model types. PRC shows the relation of the precision and recall relation as the threshold changes. Precision and recall are typically inversely related. To take an incident model in this study as an example, a lower threshold value results in more records that classified as “Crash”. Therefore, the precision of the prediction outcome decreases (less precise), while the recall increases (successfully predicts more crashes). Considering a larger precision and recall indicate better model performance, the curve closer to (1,1) (i.e. the precision and recall are both 1) corresponds to a better model. In other words, the upper right curve represents better performance because it has a higher precision when the recall is the same, and it has a higher recall when the precision is the same. According to Figure 10, NN is the best model type in PRC test.

Figure 10: The PRC of three model types



6. Conclusion

This study proposes a Neural Network (NN) classifier model for predicting crashes on freeways and urban road networks. Compared with other studies, the models in this study have following advantages: (i) The models in other studies can only be applied to a part of freeways, while models in this study can be applied to all freeways and urban road networks, (ii) The models in this study can predict crashes in the next 3 hours, which is significantly longer than the models in other studies, (iii) The models in this study is evaluated in the way that strictly simulates the real-time traffic management environment, (iv) The models in this study have high prediction performance.

The methodology for building and validating an NN classifier model described in this study is general and flexible, allowing the NN-based traffic prediction framework to be easily applicable to other regions and cities. The proposed model could be easily integrated into real-time traffic and crash management systems, where the model is built *offline* (with parameters learned from historical data) and the prediction crashes can be performed *online* by updating the values of feature variables based on real-time traffic condition information.

There are some limitations in the current study and future research directions are identified to address them. Firstly, this paper focuses on only four cells that are known as incident hot spots for model evaluation. In order to assess the model performance more accurately, more cells will be included in the further study. Secondly, only the data from Brisbane is used for model training and evaluation. When the proposed model is applied to other cities, the model performance may vary. Therefore, a further study will be needed to test the model using data from other cities.

Acknowledgments

This research was funded by Queensland Department of Transport and Main Roads (TMR) as part of TMR's effort to better understand the dynamic nature of the transport network and inform transport operational practices and services to better meet customer needs. We would like to thank TMR for access to the data on which the paper is based. The authors acknowledge the contributions to this research from Lachlan Faulkner and Edward Beak of TMR and Mark Hickman at the University of Queensland. The authors, of course, remain solely responsible for the content of this paper.

Reference

- Abdel-Aty, M., Pande, A., 2005. Identifying crash propensity using specific traffic speed conditions. *Journal of Safety Research* 36, 97–108. doi:10.1016/j.jsr.2004.11.002
- Abdel-Aty, M., Pande, A., Lee, C., Gayah, V., Santos, C.D., 2007. Crash Risk Assessment Using Intelligent Transportation Systems Data and Real-Time Intervention Strategies to Improve Safety on Freeways. *Journal of Intelligent Transportation Systems* 11, 107–120. doi:10.1080/15472450701410395
- Abdel-Aty, M., Uddin, N., Pande, A., 2005. Split Models for Predicting Multivehicle Crashes During High-Speed and Low-Speed Operating Conditions on Freeways. *Transportation Research Record: Journal of the Transportation Research Board* 1908, 51–58. doi:10.3141/1908-07
- Abdel-Aty, M., Uddin, N., Pande, A., Abdalla, F., Hsia, L., 2004. Predicting Freeway Crashes from Loop Detector Data by Matched Case-Control Logistic Regression. *Transportation Research Record: Journal of the Transportation Research Board* 1897, 88–95. doi:10.3141/1897-12
- Abdel-Aty, M.A., Pemmanaboina, R., 2006. Calibrating a real-time traffic crash-prediction model using archived weather and ITS traffic data. *IEEE Transactions on Intelligent Transportation Systems* 7, 167–174. doi:10.1109/TITS.2006.874710
- Ahmed, M., Abdel-Aty, M., Yu, R., 2012. Bayesian Updating Approach for Real-Time Safety Evaluation with Automatic Vehicle Identification Data. *Transportation Research Record: Journal of the Transportation Research Board* 2280, 60–67. doi:10.3141/2280-07
- Davis, J., Goadrich, M., 2006. The relationship between Precision-Recall and ROC curves.
- Fawcett, T., 2006. An introduction to ROC analysis. *Pattern Recognition Letters, ROC Analysis in Pattern Recognition* 27, 861–874. doi:10.1016/j.patrec.2005.10.010
- Hossain, M., Muromachi, Y., 2012. A Bayesian network based framework for real-time crash prediction on the basic freeway segments of urban expressways. *Accident Analysis & Prevention* 45, 373–381. doi:10.1016/j.aap.2011.08.004
- Hourdos, J., Garg, V., Michalopoulos, P., Davis, G., 2006. Real-Time Detection of Crash-Prone Conditions at Freeway High-Crash Locations. *Transportation Research Record: Journal of the Transportation Research Board* 1968, 83–91. doi:10.3141/1968-10

- Lee, C., Abdel-Aty, M., Hsia, L., 2006. Potential Real-Time Indicators of Sideswipe Crashes on Freeways. *Transportation Research Record: Journal of the Transportation Research Board* 1953, 41–49. doi:10.3141/1953-05
- Lee, C., Hellinga, B., Saccomanno, F., 2003. Real-Time Crash Prediction Model for Application to Crash Prevention in Freeway Traffic. *Transportation Research Record: Journal of the Transportation Research Board* 1840, 67–77. doi:10.3141/1840-08
- Lee, C., Saccomanno, F., Hellinga, B., 2002. Analysis of Crash Precursors on Instrumented Freeways. *Transportation Research Record: Journal of the Transportation Research Board* 1784, 1–8. doi:10.3141/1784-01
- Oh, C., Oh, J.-S., Ritchie, S., Chang, M., 2001. Real-time estimation of freeway accident likelihood, in: 80th Annual Meeting of the Transportation Research Board, Washington, DC.
- Oh, C., Oh, J.-S., Ritchie, S.G., 2005. Real-time hazardous traffic condition warning system: framework and evaluation. *IEEE Transactions on Intelligent Transportation Systems* 6, 265–272. doi:10.1109/TITS.2005.853693
- Sun, Jian, Sun, Jie, Chen, P., 2014. Use of Support Vector Machine Models for Real-Time Prediction of Crash Risk on Urban Expressways. *Transportation Research Record: Journal of the Transportation Research Board* 2432, 91–98. doi:10.3141/2432-11
- Wang, G., Kim, J., 2016. The Prediction of Traffic Congestion and Incident on Urban Road Networks Using Naive Bayes Classifier, in: Australasian Transport Research Forum 2016 Proceedings. Melbourne, Australia.
- Xu, C., Wang, W., Liu, P., 2013. A Genetic Programming Model for Real-Time Crash Prediction on Freeways. *IEEE Transactions on Intelligent Transportation Systems* 14, 574–586. doi:10.1109/TITS.2012.2226240
- Xu, C., Wang, W., Liu, P., Guo, R., Li, Z., 2014. Using the Bayesian updating approach to improve the spatial and temporal transferability of real-time crash risk prediction models. *Transportation Research Part C: Emerging Technologies* 38, 167–176. doi:10.1016/j.trc.2013.11.020
- Yu, R., Abdel-Aty, M., 2013a. Multi-level Bayesian analyses for single- and multi-vehicle freeway crashes. *Accident Analysis & Prevention* 58, 97–105. doi:10.1016/j.aap.2013.04.025
- Yu, R., Abdel-Aty, M., 2013b. Utilizing support vector machine in real-time crash risk evaluation. *Accident Analysis & Prevention* 51, 252–259. doi:10.1016/j.aap.2012.11.027
- Yu, R., Abdel-Aty, M., Ahmed, M., 2013. Bayesian random effect models incorporating real-time weather and traffic data to investigate mountainous freeway hazardous factors. *Accident Analysis & Prevention* 50, 371–376. doi:10.1016/j.aap.2012.05.011