Efficient Journey Planning and Congestion Prediction Through Deep Learning

Muhammad Shalihin Bin Othman
School of Computing
National University of Singapore
e0205029@u.nus.edu

Sye Loong Keoh
School of Computing Science
University of Glasgow
SyeLoong.Keoh@glasgow.ac.uk

Gary Tan
School of Computing
National University of Singapore
gtan@comp.nus.edu.sg

***Abstract*—The advancements of technology continuously rising over the years has seen many applications that are useful in providing users with sufficient information to make better journey plans on their own. However, commuters still find themselves going through congested routes every day to get to their destinations. This paper attempts to delineate the possibilities of improving urban mobility through big data processing and deep-learning models. Essentially, through a predictive model to predict congestion and its duration, this paper aims to develop and validate a functional journey planning mobile application that can predict traffic conditions, allowing road users to make better informed decisions to their travel plans. This paper proposes a Multi-Layered Perceptron (MLP) deep learning model for congestion prediction and supplements a Linear Regression (LR) model to predict its duration. The proposed MLP-LR model performed reasonably well with an accuracy of 63% in predicting an occurrence of congestion. Some critical discussions on further research opportunities stemming from this study is also presented.**

***Keywords*—congestion prediction; journey planning; deep learning**

I. INTRODUCTION

The rising population in many big cities and the increased demands in transport have exacerbated the traffic congestion problems over the years. Commuters are experiencing longer traveling time and having difficulty planning their journey efficiently. In order to reduce traffic in the cities, Singapore government have introduced Electronic Road Pricing (ERP), while London city has enforced Congestion Charges (CC) in an effort to reduce traffic congestions. There are also vast investments in the upgrading of public transports through the integration of technology such as mobile applications that provides ease of tracking bus schedules, arrival times, etc.

Although these regulation and solutions are available to tackle congestion issues, it appears that congestion still happens regularly and the general public still has to take the congested route to get to their desired location though they may be aware of congestion. In particular, Singapore is one of such countries facing an increase of traffic congestion problems over the years [1].

This paper approaches the traffic congestion issues by exploring the use of existing machine-learning tools such as WEKA [2] and Google's Tensorflow [3] Deep Learning library, to better analyze how deep learning can contribute to resolve them. Using Singapore as a case study, the availability of various data sources from the Singapore Land Transport Authority, Google and Open Maps weather APIs, has allowed the exploration of machine learning algorithms to analyze, learn, predict and determine congestions. Specifically, we investigated the use of deep-learning models to study the roads and efficiently aid the end-user application in providing the best route for road users, thus reducing their journey time. After extensive research and testing, a Multi-Layer Perceptron (MLP) coupled with a Linear Regression (LR) model was finally chosen as the proposed model to efficiently predict congestion and its duration, thus offering a new perspective in end-user's journey planning process.

Section II presents related work followed by an overview of the proposed solution in Section III. Section IV details the prototype implementation, describing the MLP-LR learning model, and the journey planning mobile application. In Section V, we explain how we arrive at the decision to use the MLP among other classifiers and present an evaluation of the MLP-LR model as well as the results of its performance. Finally, we conclude with future work in Section VI.

II. RELATED WORK

The use of machine learning algorithms to address issues in urban mobility is a common research area today. Ghosh *et. al.* [4] proposed Bayesian Support Vector Regression (BSVR) to provide some measure of confidence associated with the forecast duration of the incidents. The results concluded that the detection efficiency, for variations in prediction error, of BSVR could achieve reasonable accuracy through sensitivity and specificity analysis.

The Commonwealth Scientific and Industrial Research Organization based in Australia, has also worked on some projects aimed to use machine learning techniques to make transport networks safer and smarter [5]. One of the projects include predicting near future traffic jams and hot spots of congestion through a combination of association rule mining and dynamic Bayesian network to construct causality trees from congestion, from which, the propagation probabilities based on temporal and spatial information was estimated. Frequent sub-structures of these causality trees reveal recurring interactions among spatial-temporal congestion as well as potential bottlenecks in the designs of existing traffic networks.

Son [6] introduced Bayesian Optimization (BO) to deal with challenges in simulation-based optimization. BO constructs surrogates of the true function and evaluates the most promising points based on the surrogates. Then, the number of function evaluations can be effectively reduced without compromising performance. High dimensional BOs were then experimented on analytic functions and traffic simulation. Gupta *et. al.* presented a framework that integrates the optimization of control strategies with the generation of predictive travel time guidance within DynaMIT2.0 [7], so that actual outcomes are consistent with the expectation of users when responding to the predictive information and network control.

These papers attempts to detect congestion and predict its duration through different means. Through these related works, the data and strategies used in this research work was shaped and guided to achieve the most desirable result in predicting congestion and its duration most efficiently.

III. Proposed Solution

This section outlines the design requirements and the proposed congestion and duration prediction model using MLP-LR. A Multi-Layered Perceptron is a feed-forward neural network that has three or more hidden layers. Data passes through it in one-way (forward) from input to output. Multi-Layered Perceptron can solve problems that are not linearly separable. Its mathematical equation can be written as:

$$y = \varphi(\sum_{i=1}^{n} w_i x_i + b) = \varphi(\mathbf{w}^{\mathbf{T}}\mathbf{x} + b)$$

The process from input to output generally involves transforming the data to become linearly separable before going through the hidden layers. Multiple real-valued inputs are computed by forming a linear combination according to its input weights, adapted using some gradient-based optimization algorithm, before computing the output with some nonlinear activation function.

With this, the classification from the MLP can then be fed into a Linear Regression model to predict the congestion duration based on the time and day on a linear scale.

A. Design Requirements

Using Singapore as a case study, traffic data and weather information were studied and analyzed to extract critical attributes that would form the most effective feature in identifying traffic congestion. After several research, discussions and careful study, we observed that in order to effectively identify traffic congestion in Singapore;

- Congestion occurring on weekdays has no relation to the possibility of congestion occurring on weekends and vice-versa.
- Congestion durations may vary across different regions in Singapore as the traffic flow varies for different parts of the city; The city is split into five regions, namely *Central, East, West, North,* and *South*.
- The same congestion type in different region may have varying congestion duration as well.

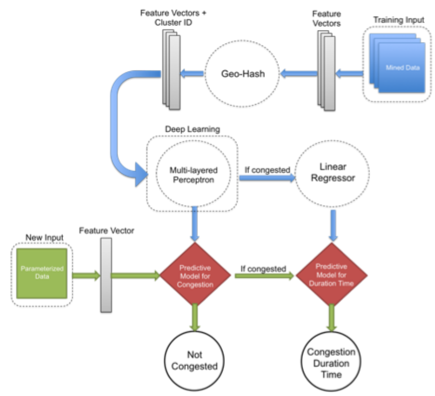

Fig. 1. Overview of Congestion and Duration Prediction Model

- Non-recurring incidents occurring on special events or holidays (such as New Year or Christmas), may last longer than usual; Such events do not reflect the "*normal*" traffic conditions in Singapore.

The proposed solution adopts a classification model to predict congestion. As for the prediction of congestion duration, a regression model is used. The combination of these two models are discussed next.

B. Architecture Design

Figure 1 describes the training and prediction flows of the proposed deep-learning system. With the availability of real time transportation data in Singapore provided by myTransport.sg, Land Transport Authority (LTA) Data Mall and the weather data from One Maps Weather API, it is possible to train the proposed MLP-LR model with high accuracy.

1) Features for Training: A critical finding revealed that although a certain algorithm may produce the best accuracy on a particular set of data, another might produce even better accuracy on another set of data. After careful study and numerous discussions, a few important attributes, namely the latitude, longitude, day, time and weather, were found to be of high importance in identifying traffic congestion in Singapore. The actual duration of congestion will then supplement the feature as an additional attribute to predict the duration of the congestion type that was predicted. Therefore, the following defines the feature for training the proposed congestion prediction model:

1) Day of week ($w_i \in \{1, 2, 3, 4, 5, 6, 7\}$, where 1 represents Monday. . . 7 represents Sunday)
2) Time of day ($t_i \in \{0 \ldots 2359\}$, in 24-hour time format)
3) Weather (Integers as defined by the Open Maps Weather API),
4) Congestion Type ($c_i \in \{0, 1, 2, 3, 4, 5\}$, where 0-5 are defined in Table I)

As for the duration prediction model for the predicted congestion, the model relies on the following feature for training:

1) x - Day of week concatenated with time of day (e.g., Monday 12:30pm would be represented as 11230)

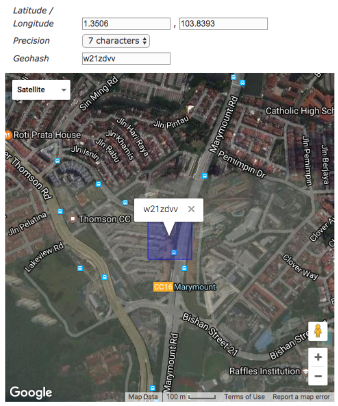

Fig. 2. GeoHash Example

2) y - Duration of congestion in minutes (where $d_i > 0$)

It is observed that different causes of congestion may carry different durations in different regions. Therefore, a classifier that is only able to classify a situation as *congested* or *not congested* is not adequate. Since the LTA data mall has the ability to distinguish different traffic condition, the proposed model is trained to determine different types of congestion as shown in Table I.

TABLE I
CONGESTION TYPES

ID	Congestion Type
0	Not Congested
1	Heavy Traffic
2	Accident
3	Vehicle Breakdown
4	Other Obstacles
5	Road Block

2) Geo-Hashing: In order to efficiently separate regions in Singapore where congestion types may lead to different duration, we need to cluster a group of latitudes and longitudes to form a region.

A geo-hashing algorithm basically identifies a rectangular cell, as marked with a blue bounding box in Figure 2. All latitude and longitude points that fall within the same rectangular cell would translate into the same geohash. Each extra character in a geohash identifies one of the 32 sub-cells. The cell width reduces moving away from the equator (to 0 at the poles). Therefore, the number of characters defines the precision of the rectangular cell. After extensive studies and tests, precision 7 was concluded to best suit the needs of clustering stretches of roads in Singapore, with a cell size of $\leq$ 153m $\times$ 153m.

3) Congestion and Duration Prediction: For each cluster, each congestion type will have its own linear regression model. The MLP model will then be able to predict the congestion type for a cluster.

Subsequently, the linear regression model for the congestion type, for that cluster, can then be retrieved to make the congestion duration prediction. With the day and time concatenated as the x value, the duration of the congestion y can be trained and predicted.

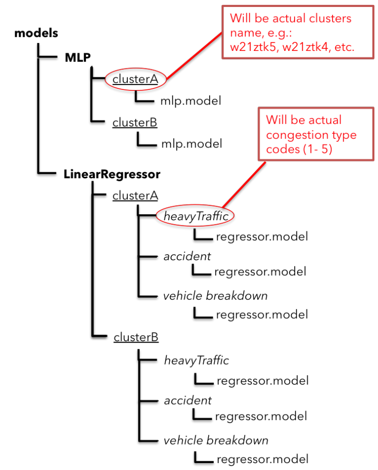

Fig. 3. Model Directory Structure

Hence, if the model for congestion prediction is accurate enough, the classified congestion type for a defined cluster would already be specific enough to enable a better than average prediction of duration using a linear regression model. Figure 3 illustrates the directory structure for storing the models.

IV. Prototype Implementation

We have developed a prototype including the deep learning models to accurately predict different congestion types and its congestion duration; an android application and a web service that enable end users to effectively plan their journey in Singapore based on the developed MLP-LR model via the Internet.

A. Deep Learning Models

Using Google's Tensorflow Deep Learning library, the Multi-Layered Perceptron (MLP) model and the Linear Regressor (LR) was built.

Through the MLP classifier, the congestion type for a cluster, given the day, time and weather, can be predicted through the feed-forward algorithm. The model can then perform its training updates through the back-propagation algorithm that is already implemented in the DNNClassifier by Tensorflow. Following this, the congestion duration prediction can be achieved by retrieving the appropriate linear regression model for the cluster. The linear regression model for machine learning uses the following formula:

$$y = \beta_0 + \beta_1 \times x$$

- y is the congestion duration to be predicted.
- x is the day and time concatenated and passed as an input variable.
- β_0 and β_1 are coefficients that needs to be estimated, moving the line around.

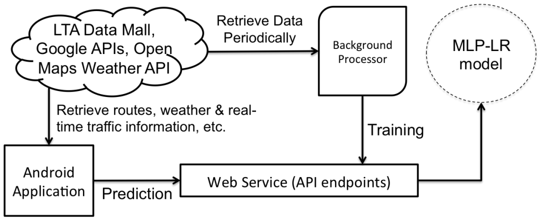

Fig. 4. Model-Service-Mobile Architecture Overview

With these models, a web service was then implemented to train and make predictions by converting features into a stream of command-line arguments, then calling the Python executor through the built-in Java *execute()* method, passing in command-line arguments for the Python scripts, containing the main method, to process and return the prediction as output, or to report the training status.

B. Web Service Application Programming Interface (API)

The web service is a maven dynamic web project developed as an API service to execute training and predictions on the predictive models. The web service project was compiled into a war file and deployed onto an instance of the AWS EC2 server. Figure 4 shows the overview of the model-service-mobile architecture. There are two main API endpoints for prediction as follows:

- *Multiple prediction on the full MLP-LR model* – it is used to predict the duration for multiple clusters along the route to a desired destination. For example, from Point A to Point B, the route is made up of multiple clusters. Through this API, the congestion duration for each cluster is predicted and returned.
- *Single prediction on the LR model alone (for predicting congestion duration only)* – it predicts the duration for a single cluster where the congestion type is already known. For example, after the multiple prediction was called and the predicted duration for each cluster has been determined, cluster A was predicted with a congestion duration of 0, however, the real-time updates from LTA Data Mall indicates that there is actually a congestion of type x currently happening. The duration can then be predicted through this API by passing congestion type x. This helps to improve the overall prediction of congestion duration.

Through these two API endpoints, a mobile app can be built to take in user's input for start and end locations, as well as date and time, and outputs the predicted congestion spots along the route and its time duration to clear up. The next section will explain how these APIs are used by the mobile app.

C. Android Application

An Android Application was developed to utilize this model. It serves as a simple journey-planning tool that allows a user to enter their start and end points, specifying the time and date to depart by or arrive at. It will then call the web service API upon submission, to predict possible congestion

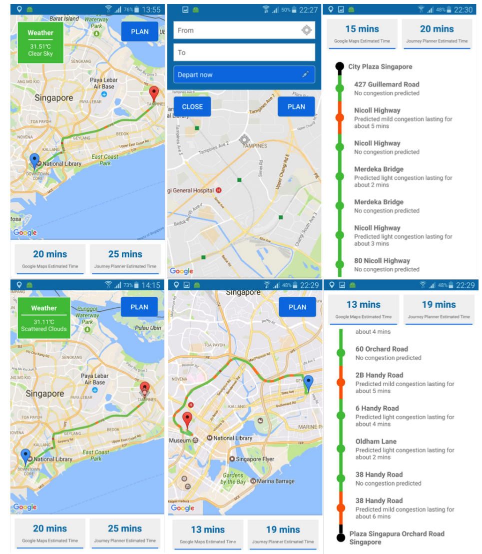

Fig. 5. Mobile Application Screen shots

along the route. The system relies on Google Directions API to retrieve all possible routes.

For each route, the polyline of latitudes and longitudes are grouped into clusters through a geo-hashing algorithm. Subsequently, the parameters are formed to make the prediction through the web service API. The congestion duration for each cluster can then be determined, and collated to display the traffic conditions along the route leading up to the destination. Figure 5 shows the screen shots of the mobile application.

The mobile app has a feature to process the request with real-time data if the request made was to depart immediately. Using LTAs real-time API, it checks if there is any real-time congestion along the route that was not predicted by the model. The application will then make another call to the web service API to predict the duration based on the congestion details. This involves a prediction on the linear regression model only. The total congestion time is then updated accordingly to compute the overall estimated time.

This mobile application is not meant to show users the way to get to their destination, as Google Maps already does that. Instead, it aims to delineate the condition of the road for each possible route, with a fusion of real-time and predictive data.

V. Evaluation and Results

Multiple classifiers were tested and evaluated before the decision of using the MLP was made. Following which, further evaluations on the different types of optimizers and configurations were done as well. As for the accuracy validation tests, they were carried out in two phases that aims to evaluate firstly, the accuracy of predicting congestions against real-time traffic data provided by the LTA, followed by the accuracy of predicting the duration time of congestion.

A. Evaluating Classifiers

Through utilizing the helpful Weka graphical tool, six classifiers were trained and evaluated. The Mean Absolute Error

(MAE) and Root Mean Squared Error (RMSE) are used as the performance indicator to compare the accuracy of predicting a congestion. In essence, a total of 1614 mock features were trained and evaluated on these models. Figure 6 shows the results from WEKA, indicating that MLP outperforms Logistic Regression, K-Nearest Neighbour, Stochastic Gradient Descent, Naive Bayes, and Support Vector Machine (SVM) with the lowest MAE and RMSE.

=== Summary ===	Logistic Regression	K-Nearest Neighbour Classifier	Stochastic Gradient Descent	Multi-Layered Perceptron	Naïve Bayes	Support Vector Machine
Correlation coefficient	0.6475	0.5157	0.3867	0.9305	0.5959	0.7169
Mean absolute error	0.2275	0.2032	0.2469	0.0228	0.1695	0.1359
Root mean squared error	0.3694	0.3633	0.3922	0.1553	0.3406	0.2998
Relative absolute error	80.9849 %	72.3435 %	87.9076 %	8.1061 %	60.3409 %	48.371 %
Root relative squared error	87.1271 %	85.679 %	92.5018 %	36.6356 %	80.3199 %	70.6973 %
Total Number of Instances	1614	1614	1614	1614	1614	1614

Fig. 6. Weka Test Results

The MLP model was evaluated and fine-tuned by evaluating different optimizers, activation functions, learning rate, etc. Table II shows the comparisons of accuracy scores for different variations of the MLP model, trained with over 1000 features.

TABLE II
ACCURACY OF MLP OPTIMIZER

	Version 1	Version 2	Version 3	Version 4
Optimizer	SGD	AdaGrad	AdaDelta	Adam
Accuracy	50.1%	36.9%	55.6%	30.9%

After thorough analysis, we concluded that the MLP model with an Adaptive Delta Optimizer at a 0.01 learning rate and 0.95 decay rate, along with 3 hidden layers of 10, 20 and 10 neurons respectively provides the best prediction results.

B. Congestion Prediction Accuracy

Validation was done by retrieving real-time congestion from LTA's Data Mall and calling the API service to predict the occurrence of a congestion. Markers were plotted on a map, identifying successful predictions of congestion against failed prediction of actual congestion that happened. Figure 7 shows the results for Monday to Sunday and its accuracy, collected over a 2-week period, for 2 separately trained but identical models with 15,000 and 30,000 features trained respectively.

The red markers indicate that a real-time congestion occurred at that cluster but the predictive model failed to predict it (true-negative), while the green marker indicates that the predictive model successfully predicted congestion at the cluster where there is congestion in real-time (true-positive). The results for 15,000 trained features shows a satisfactory accuracy with an average of 57%. However, the prediction for Sunday falls way below average. This is likely due to insufficient data since it is clear from the figure that very little congestion occurs on Sundays.

The results for 30,000 trained features prove the concept of improved accuracy through incremental learning. With a

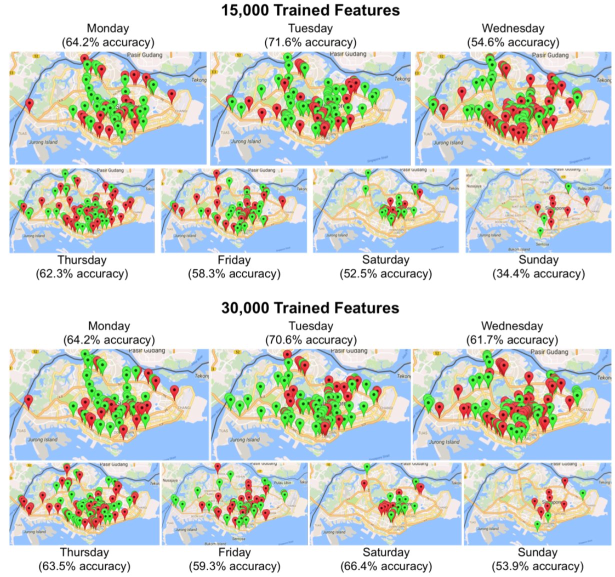

Fig. 7. Results of Congestion Prediction in Singapore

good average accuracy of 63%, the model also scores well above 50% for each day of the week. Thus it is conclusive that the model is capable of achieving high accuracies with carefully cleaned data and incrementally trained over time. The lower accuracies found for some days, such as Fridays, may be due to the higher probabilities of non-recurring congestion where more vehicles on the roads may be traveling through non-routine routes. Such cases would be harder to have a high accuracy in prediction and hence, the statistics resulting from the predictive MLP model is justified to produce a good measure of predicting congestion in traffic.

C. Duration Prediction Accuracy

To evaluate and validate the accuracy of predicting the congestion duration, a prediction was made for every real-time congestion before the congestion actually ended. When the congestion was cleared up, the duration was recorded to compare the predicted duration against the actual duration. Each entry was organized into 4 regions; Central, East, West and North. For each region, the predicted durations were plotted against the actual durations for each entry to visualize the precision of the predicted durations. Figure 8 shows the results for the two separately trained but identical models with 15,000 and 30,000 features trained respectively.

The central and east regions are the most common regions of congestion. Hence, they were chosen as examples shown in Figure 8. The blue line represents the predicted durations while the red line represents the actual durations. From the graph, it can be seen that for all days in each region, the predicted duration does not stray more than 10-20 minutes from the actual duration (i.e. the gap between the 2 lines at all points). This amount of difference is not satisfactory although the difference may or may not affect a users speed too

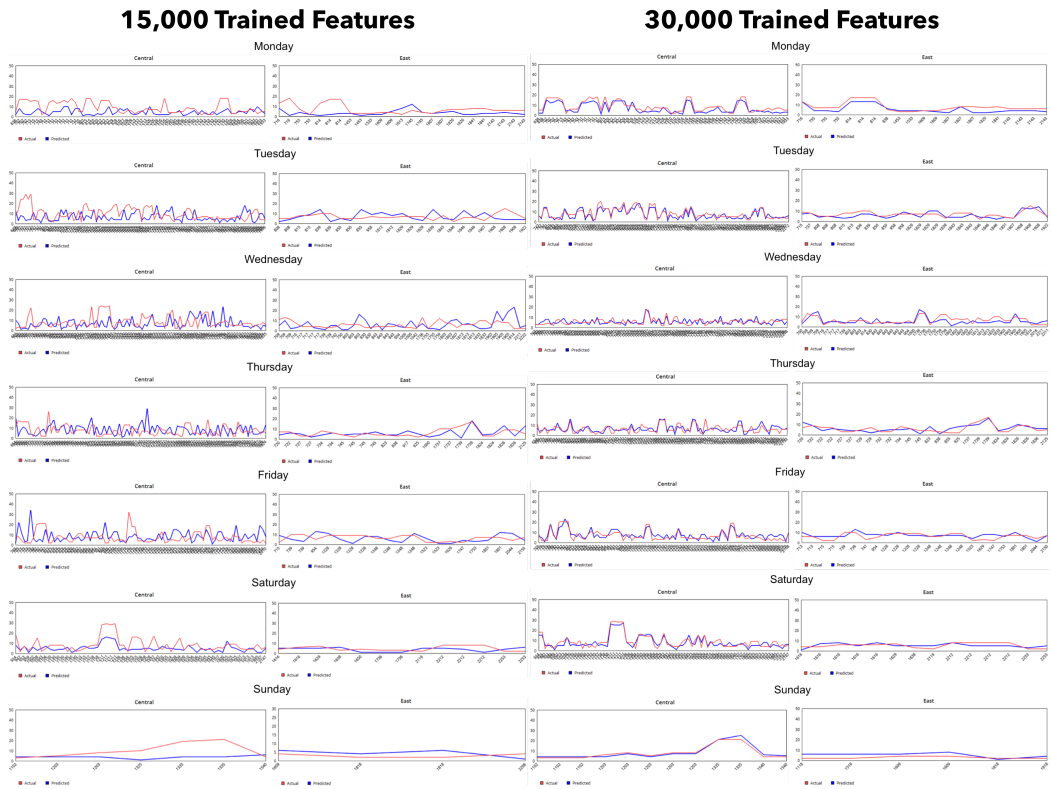

Fig. 8. Congestion Duration Comparison Charts

significantly. However, such high variance may still produce unreasonably inaccurate journey time estimation.

Finally, the model with 30,000 features trained shows very promising results with only about 5-10 minutes difference for each prediction. With such a small variance, it will barely make much difference to the estimated journey time even if compared to a prediction with 0 minute difference for all. This proves that the LR model, supplemented with the result from the MLP, is able to provide a good prediction for the duration of congestion, given enough training data.

VI. Conclusions

We have shown a novel integration of deep learning models and mobile applications to mitigate congestion issues in Singapore. Through careful studies and tests, the outcome of this rigor process aligned with the intentions of this paper. Hence, it is conclusive that through big data processing and the selection of the right deep-learning model, it is very much possible to effectively predict traffic congestion.

Further improvement possibilities may include possible supplements to the model that can serve as an aid to predict more critical information that may contribute to the advancement of future urban mobility. There were also some issues revealed through evaluation tests such as the undesirable loading time for predicting congestion along longer routes. This is due to the prediction of much more clusters on a single model hosted on a server with very limited computing power. Hence, some solutions suggested would be to optimize the speed by upgrading the instance with a higher computing power and perhaps host the predictive models on multiple Hadoop [8] clusters for faster processing.

In conclusion, this paper has certainly opened up several avenues for research and investigations with a real-time application capable of proving the possibility of the concept, which is to mitigate congestion issues through deep learning.

References

[1] L. L. Yuan, "A case study on urban transportation development and management in singapore," in *Second International Expert Panel Meeting on Urban Infrastructure Development, Bangkok, Thailand.* Citeseer, 1997, pp. 8–9.

[2] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software: an update," *ACM SIGKDD explorations newsletter*, vol. 11, no. 1, pp. 10–18, 2009.

[3] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, "Tensorflow: A system for large-scale machine learning," in *Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI). Savannah, Georgia, USA*, 2016.

[4] B. Ghosh, M. T. Asif, and J. Dauwels, "Bayesian prediction of the duration of non-recurring road incidents," in *Region 10 Conference (TENCON), 2016 IEEE.* IEEE, 2016, pp. 87–90.

[5] DATA61. (2015, feb) Advanced data analytics in transport. [Online]. Available: https://research.csiro.au/data61/advanced-data-analytics-in-transport/

[6] J. Son, "High dimensional bayesian optimization with application to traffic simulation," M.Sc Thesis, National University of Singapore, 2016.

[7] S. Gupta, "Real-time optimisation of network control strategies in dynamit2.0," in *TRB 95th Annual Meeting, Washington, USA*, jan 2016.

[8] Apache Software Foundation. (2017) Apache hadoop. [Online]. Available: http://hadoop.apache.org/docs/current/