

How to cite this paper:

Nadeem, M., K., & Fowdur, P. T. (2018). Performance analysis of a real-time adaptive prediction algorithm for traffic congestion. *Journal of Information and Communication Technology, 17* (3), 493-511.

PERFORMANCE ANALYSIS OF A REAL-TIME ADAPTIVE PREDICTION ALGORITHM FOR TRAFFIC CONGESTION

Khodabacchus Muhamad Nadeem & Tulsi Pawan Fowdur

*Department of Electrical and Electronic Engineering
University of Mauritius, Réduit, Mauritius*

muhamad.khodacchus1@umail.uom.ac.mu; p.fowdur@uom.ac.mu

ABSTRACT

Traffic congestion is a major factor to consider in the development of a sustainable urban road network. In the past, several mechanisms have been developed to predict congestion, but few have considered an adaptive real-time congestion prediction. This paper proposes two congestion prediction approaches are created. The approaches choose between five different prediction algorithms using the Root Mean Square Error model selection criterion. The implementation consisted of a Global Positioning System based transmitter connected to an Arduino board with a Global System for Mobile/General Packet Radio Service shield that relays the vehicle's position to a cloud server. A control station then accesses the vehicle's position in real-time, computes its speed. Based on the calculated speed, it estimates the congestion level and it applies the prediction algorithms to the congestion level to predict the congestion for future time intervals. The performance of the prediction algorithms was analysed, and it was observed that the proposed schemes provide the best prediction results with a lower Mean Square Error than all other prediction algorithms when compared with the actual traffic congestion states.

Keywords: Adaptive prediction, cloud server, Global Positioning System, real-time, traffic congestion.

INTRODUCTION

Road traffic congestion remains a major problem in today's era affecting both society and economic development. In the United States for example, over the last years, every city has experienced an augmentation in traffic congestion (TomTom Traffic Index, 2017). This increase in congestion is related to various problems like pollution, noise and consumption of time and energy in travel. Traditionally, several methods like improving road infrastructure and urban planning were employed to reduce congestion. However, they were both costly and time-consuming. Therefore in order to mitigate the problem, traffic congestion is predicted so that congested road can be avoided resulting in an improved performance and effectiveness of the public transport system. Previous studies have deployed model-based approaches as well as machine learning technique in the field of traffic congestion prediction. An overview of these previous works is given next.

Prakash (2015) proposed a system with K-Means clustering and Naïve Bayes algorithms to detect and predict the traffic congestion based on GPS data received from various GPS-enabled devices. Historical data, as well as the travelling speed, were used as input to the prediction model, and an accuracy of up to 89% was obtained from the system. Yang et al. (2015) had proposed a novel approach that uses the Traffic Flow Prediction (TFP) and Congestion State Fuzzy Division (CSFD) modules to predict the traffic congestion using the floating car trajectory data collected by taxi in Beijing. The Particle Swarm Optimization (PSO) algorithm in the TFP module optimised the parameter of the Support Vector Machine (SVM) in predicting the traffic volume. The study showed that the PSO algorithm outperformed all other optimisation algorithms in terms of prediction accuracy. Lwin & Naing (2015) made use of a Hidden Markov Model (HDM) for forecasting the traffic congestion using both the historical and real-time data. The system model was tested on different road segments during peak hours, and the HDM showed a promising prediction result with an average accuracy of 86%. Prathilothamai, Lakshmi and Viswanthan (2016) adopted the Apache Hadoop and Apache Spark framework for increasing the accuracy of prediction using an advanced data processing technique. The data was collected offline using an Ultrasonic and Passive Infrared sensor during peak time and off-peak time. As a result, the proposed prediction model had achieved a precise prediction of congestion levels during high traffic. A complex hybrid prediction model was proposed by Lopez-Garcia, Onieva, Osaba, Masegosa and Perillos (2016) whereby a combination of Genetic Algorithm and Cross-Entropy method (GACE) were used for forecasting short-term traffic congestion. The experiment was

performed using Matlab, and the results showed that the GACE achieved an excellent performance with the lowest prediction error. Moreover, Liu, Feng, Wang, Zhang and Wang (2014) proposed a Bayesian Network approach to predict urban traffic congestion including a directional dependence analysis algorithm to learn the Bayesian Network structure. Their research incorporated historical data to test the system and the resulting performance showed that the proposed system was capable of predicting the traffic congestion.

Although the above studies have implemented several prediction models, very few have focused on the use of an adaptive approach to improve the accuracy of the prediction. This paper proposes the use of an adaptive prediction model which could select between the most appropriate predictor for a given set of observations based on the Root-mean-Square-Error (RMSE) model selection criterion. The congestion estimation system consists of a Global Positioning System (GPS)/Global Systems for Mobile (GSM) tracking devices installed in a bus that relays the time and position of the bus to a cloud server in real-time. A control station will then access the cloud server and computes the congestion based on the vehicle speed which is calculated from the GPS data. Predictive analytics is then performed by the control station to select the best predictor among the five algorithms; Autoregressive Integrated Moving Average (ARIMA), K-Nearest Neighbors (KNN), Linear regression, polynomial regression and Moving Average, to provide an estimate of the congestion state for the next 0.3 kilometres.

The data was collected on two bus routes in Mauritius for ten weekdays during peak hours. It was observed that the adaptive algorithm significantly outperformed all the other traditional prediction algorithms by providing a MSE of only 0.1426 with respect to the actual congestion state.

PROPOSED CONGESTION PREDICTION SYSTEM

The proposed system consists of a tracking device, cloud server and control station. The tracking device consists of an Arduino board mounted with a Global Positioning System (GPS) and Global System for Mobile (GSM) module. The vehicle (bus) to be monitored is equipped with the tracking device which transmits the GPS information such as the coordinates and GPS time in real-time to a cloud server via the GSM module. The control station makes use of the Google API service to compute the distance travelled by the vehicle, from which the speeds of the vehicle and observed traffic congestions are calculated. The control station then applies predictive analytics to obtain

the congestion state for the next 0.3 kilometres to be covered by the vehicle. The prediction process is repeated using the GPS updates received from the tracking device. The next subsection describes the hardware and software configuration for the vehicle tracking device, cloud server and control station.

Figure 1 shows the overview of the proposed system.

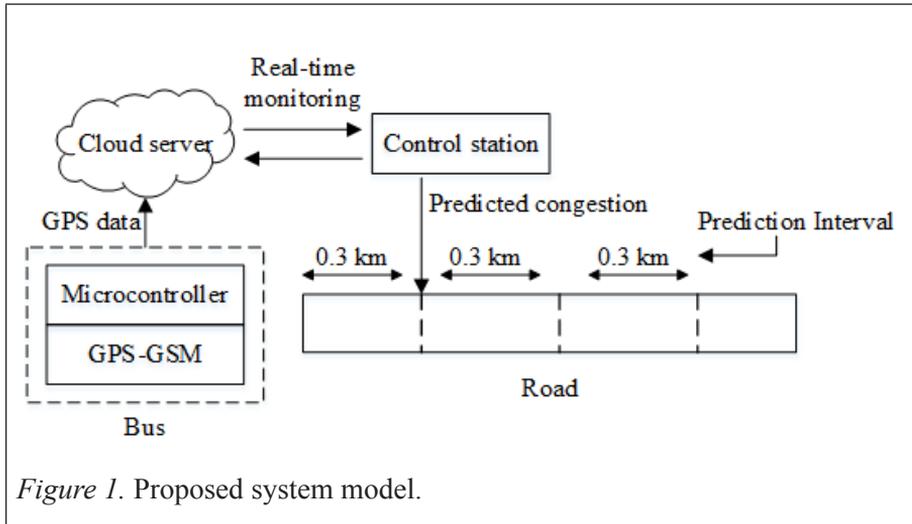


Figure 1. Proposed system model.

Hardware Configuration

The core elements incorporated to implement the vehicle tracking device are; the Arduino microcontroller, GPS module and GSM shield. Figure 2 shows the proposed circuit design and the interconnections among the hardware components.

The Arduino (Arduino Board Uno, 2017) is the brain of the system that holds the program inside its flash memory to control the modules mounted on the board. The GPS module (Google Maps Directions API, 2017) is used to acquire the vehicle location as well as GPS time from the navigation satellites. The GPS data is inserted in the query string of the cloud server URL address, and the GSM shield (SIM900 GPRS/GSM Shield, 2015) enables the tracking device to transmit the GPS data to the cloud server over the cellular network via the HTTP protocol. The GPS data is continuously transmitted to the cloud server with an interval of 10 seconds to avoid overlapping of GPRS data packets. The microcontroller and the modules mounted are powered by an external battery of minimum five volts.

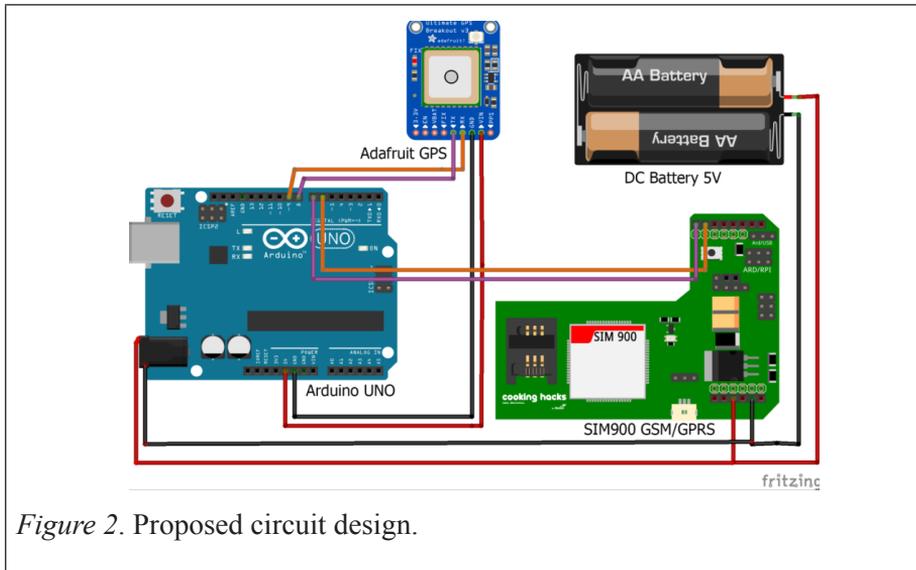


Figure 2. Proposed circuit design.

Cloud Server Setup

MySQL (MySQL, 2017) and PHP (PHP, 2017) are the main components of the cloud server which interface the microcontroller and the control station. The server stores the GPS data from the tracking device and provide access to the control station in order to monitor the vehicle in real-time.

MySQL is a database storage server that stores the GPS data in an organised form such as a table. The PHP language executes PHP scripts files upon the request of a web user. The tasks performed by the PHP in the proposed cloud system includes establishing connection with the MySQL server, inserting records in the database table, retrieving GPS data from query string of the URL and interacting with Google API (Google Maps Directions API, n.d.) service using an API authentication key to calculate the distance travelled.

Control Station Configuration and Predictive Analytics

The main application of the control station is developed on Java platform using the open source software Netbeans IDE. The primary function of the control station is to communicate with the cloud server, to monitor the vehicle in real-time and perform predictive analytics on the recorded traffic congestion states. The functions are described as follows.

With the help of a MySQL java file (“MySQL Connectors,” 2017), the control station constantly monitors the GPS data in the MySQL database server and computes the traffic congestion using the following equation.

$$\text{Congestion} = \frac{\text{Free flow travel speed}}{\text{Vehicle current speed}} \quad (1)$$

Where the speed of the vehicle is computed using Equation 2 and the free-flow travel speed refers to the ideal speed under zero congestion level. In this work, the free-flow travel speed is assumed to be 80kmh⁻¹.

$$\text{Speed of vehicle, kmh}^{-1} = \frac{\text{Distance travelled}}{\text{Time Taken}} \quad (2)$$

Where Distance travelled refers to the distance covered with reference to the last GPS record in the database.

The control station then applies prediction algorithms to forecast the traffic congestion for the next 0.3 kilometres. The range of 0.3 kilometres is chosen in this study since the average speed of a bus does not exceed 80km/h, and therefore this distance is long enough to improve the accuracy of the algorithms. The prediction algorithms developed in the control station are described as follow:

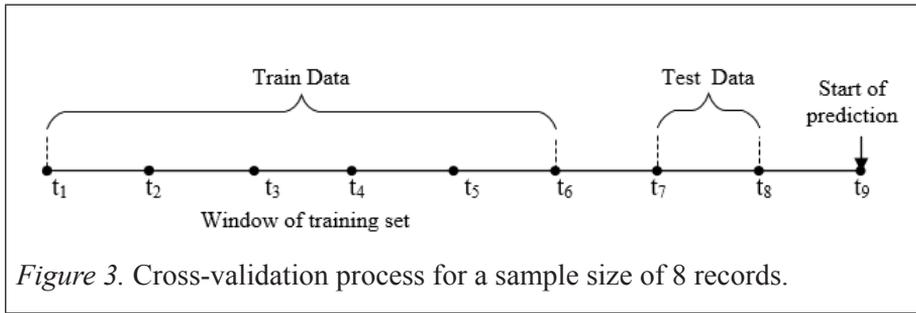
1. Moving Average – It is a time series prediction which is based on the average of previous observations. A window of the observations of a predefined size is selected for the prediction.
2. Autoregressive Integrated Moving Average (ARIMA) – It is a time series analysis that finds the best fit of a time series model to forecast future points in the series. ARIMA models are denoted by ARIMA (p, d, q) where p, d, q are numbers representing the order of autoregressive, degree of differencing and order of moving average.
3. Linear Regression – A regression technique that formulates a straight-line relationship between a dependent variable and independent variable (Zou, Tuncali, & Silverman, 2003). In this study, the dependent variable is the congestion level while the independent variable is the distance.
4. Polynomial Regression – A regression technique in which a dependent variable is regressed on the degree of an independent variable (Ostertagová, 2012). In this study, the second and third degree polynomial are used.
5. K-Nearest Neighbors – a simple machine learning model where the prediction is the average of k-nearest observations based on the Euclidean distance metric. The neighbourhood size, k is equal to the square root of the number of observations in the dataset (Duda, Stork, & Hart, 2000).

The above prediction algorithms are applied to the observed congestion states as described in the steps:

1. The vehicle information (speed, distance, congestion state) is stored in an array data structure.
2. The congestion state for the next 0.3 kilometres is predicted.
3. The array is updated with new vehicle data from the cloud server
4. The prediction process is repeated (Step 2-3) until no new updates are received from the cloud server.

Proposed Prediction Scheme

Prior to the prediction process, a cross-validation (Picard & Cook, 1984) is first performed on the recorded data set to generate a training and test dataset with a ratio of 80% to 20% respectively as shown in Figure 3. Each prediction algorithms uses the training set (t_1, t_2, \dots, t_6) to estimate a forecast for t_7 . The squared error deviation between the actual and forecast value is calculated using the formula given in Equation 3. The window of the training set is then shifted to $t_2 - t_6$ and the above process is repeated for t_8 . The error deviation is again computed between the actual and the forecast value of t_8 .



$$\text{Squared Error Deviation} = (p_i - p_0)^2 \quad (3)$$

Where p_i is the predicted value, and p_0 is the actual value. Once the error terms are computed, the RMSE is then used to select the predictor (lowest RMSE) for t_9 using the following equation.

$$\text{Root Mean Square Error (RMSE)} = \sqrt{\frac{1}{v} \sum_{i=1}^v (p_i - p_0)^2} \quad (4)$$

Where v is the number of data points in the test data. The prediction algorithm with the lowest RMSE is chosen as a predictor. There are two adaptive prediction schemes developed in the control station:

1. Adaptive prediction – uses the prediction algorithm with the lowest RMSE to predict the congestion.
2. Hybrid Neural Network (Hybrid NN) – combines the prediction algorithm with the lowest RMSE with a Neural Network model to predict the congestion. The proposed Neural Network architecture used in this work has the following model:

There are two neurons in the input layer (distance and congestion), seven neurons in the hidden layer which are found with a trial and error approach and one neuron in the output layer that provides the predicted congestion value. The activation function implemented is a sigmoid function which is used to determine the relationship between inputs and outputs of the network. The learning process is performed by a back-propagation algorithm which adjusts the weights on the neuron in the hidden layer (Amita, Singh, & Kumar, 2015). The proposed Hybrid NN is trained by passing a set of distance and measured traffic congestion at the input. The advantage of the Hybrid scheme is that the result of the predictor is correlated with the actual data measured and hence fine-tunes the prediction result which is then produced at the output layer of the NN model. The next section assesses the performance of the prediction algorithms developed.

SYSTEM TESTING AND PERFORMANCE ANALYSIS

The performance of the predictive algorithms and the adaptive schemes were assessed on two routes in Mauritius as shown in Figure 4. The parameters for the prediction algorithms used during the testing phase are given in Table 1.

Table 1

Parameter Set for the Prediction Algorithms during Testing Phase

Parameter	Value
Window size of Moving Average	30
KNN Neighborhood size	6
Neural Network Epoch	1000

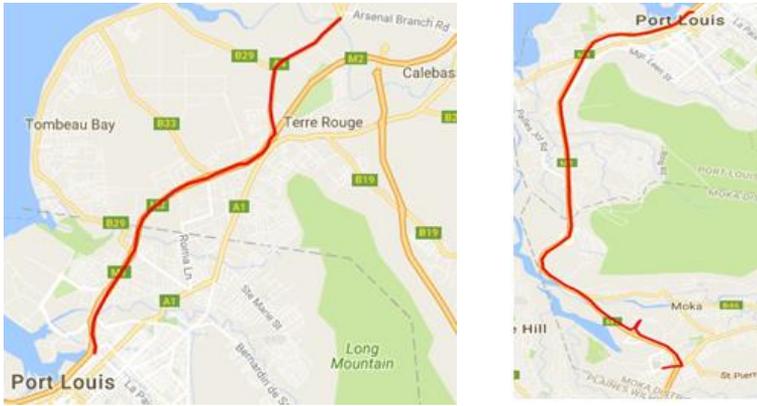


Figure 4. Google Map direction for Route 1 and Route 2 (Google Maps, 2017)

Table 2

Details of the Routes Selected for Testing Phase

	Route 1	Route 2
Source	Arsenal	Port Louis
Destination	Port Louis	Réduit
Distance	6.8 km	12 km
	Data Collection Interval	
Morning	7h00 – 7h30	7h30 – 8h15
Afternoon	16h00 – 16h30	15h00 – 15h30

The performance of the algorithms was assessed in terms of the predicted and actual congestion states for a range of distances. Mean Squared Error (MSE) was used as a metric to compare the performance of the algorithms. The tests were performed on ten weekdays. The results represent the average of the ten weeks recorded data sets.

Figure 5 and 6 show the graph of the predicted congestion states against the distance travelled for the eight algorithms as well as the actual congestion states. Figure 5 and 6 represent the morning and afternoon results for route 1. It is observed that the best performance is obtained with the adaptive algorithm (Adaptive RMSE) as it yields the closest match with the actual congestion state. In Figure 6, the Adaptive RMSE is closest to the actual congestion value at 3.3km.

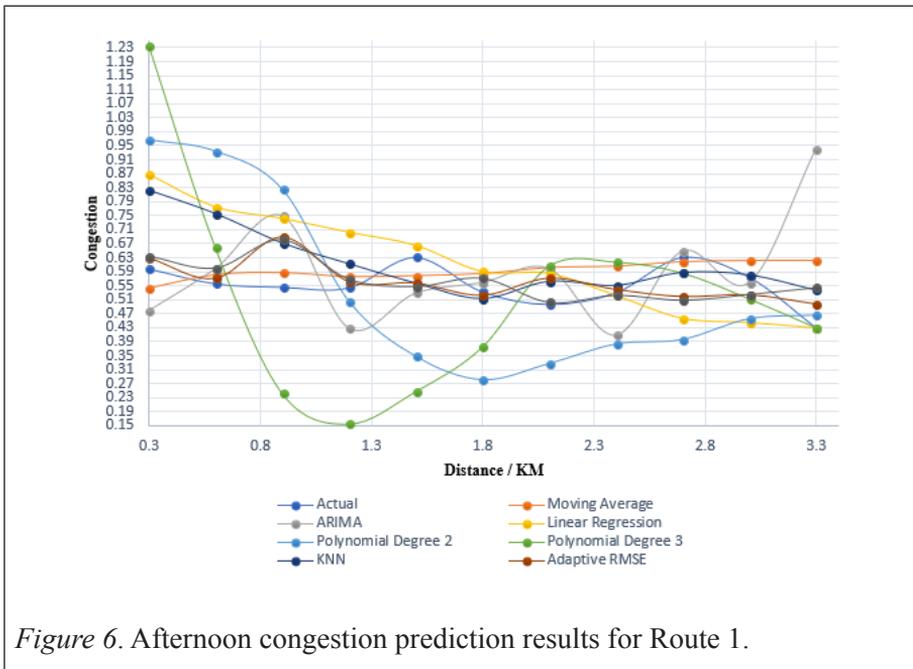
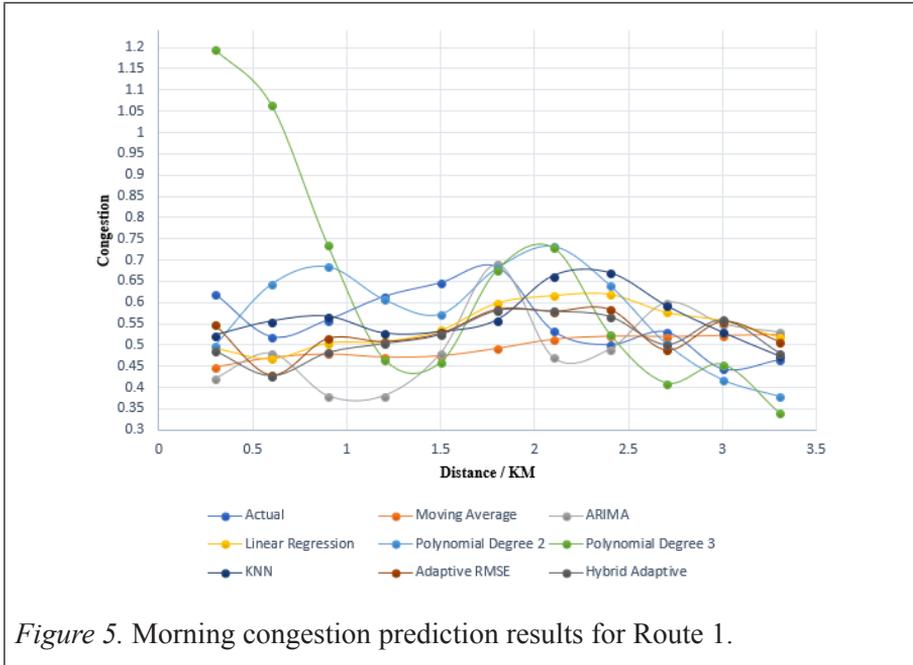
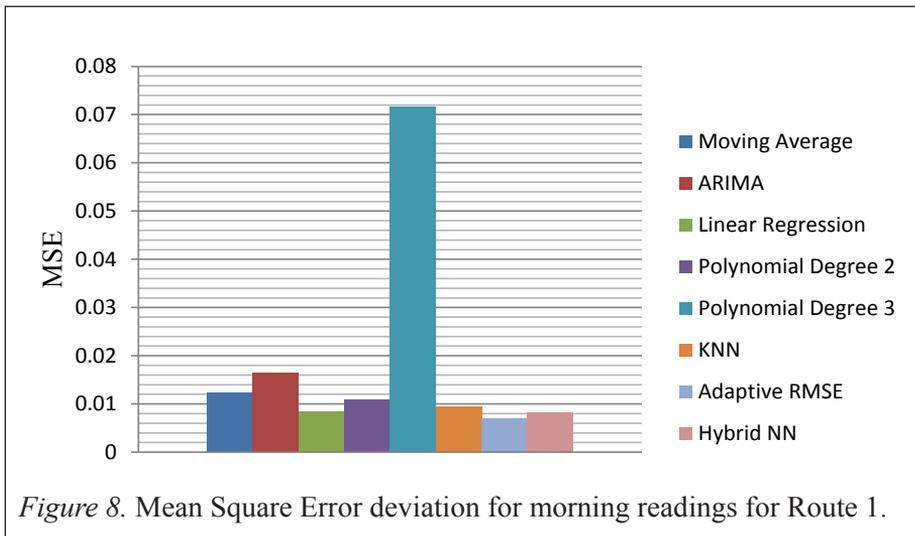
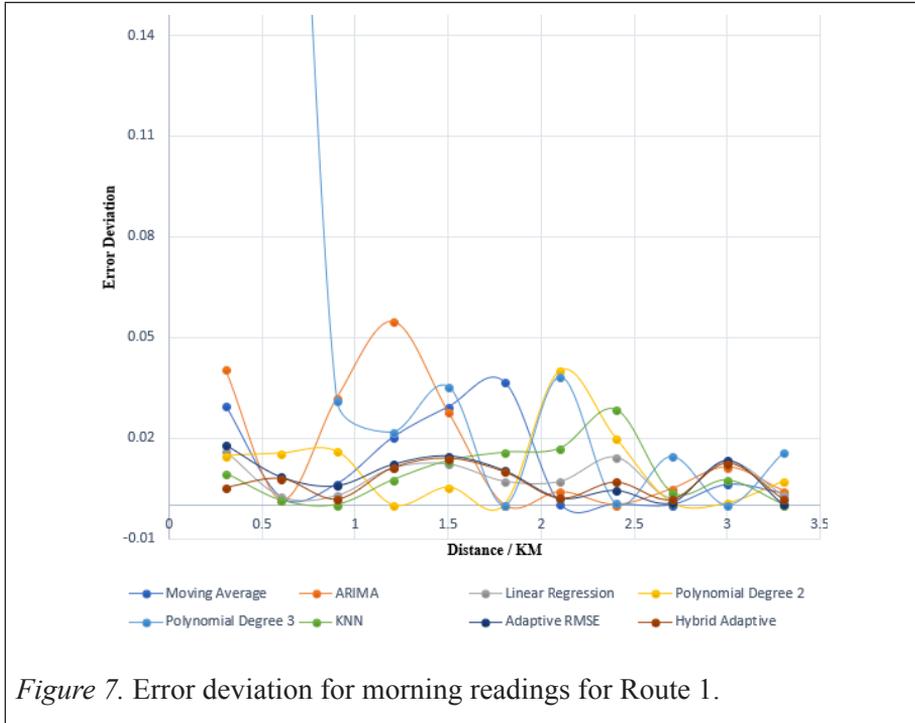


Figure 7 and 9 show the graph of error deviation against distance for the eight prediction algorithms for route 1. Figure 8 and 10 represents the MSE

in bar charts. It is observed that the adaptive schemes provide the lowest error deviation compared to the other prediction algorithms. The MSE results are given in Table 4.



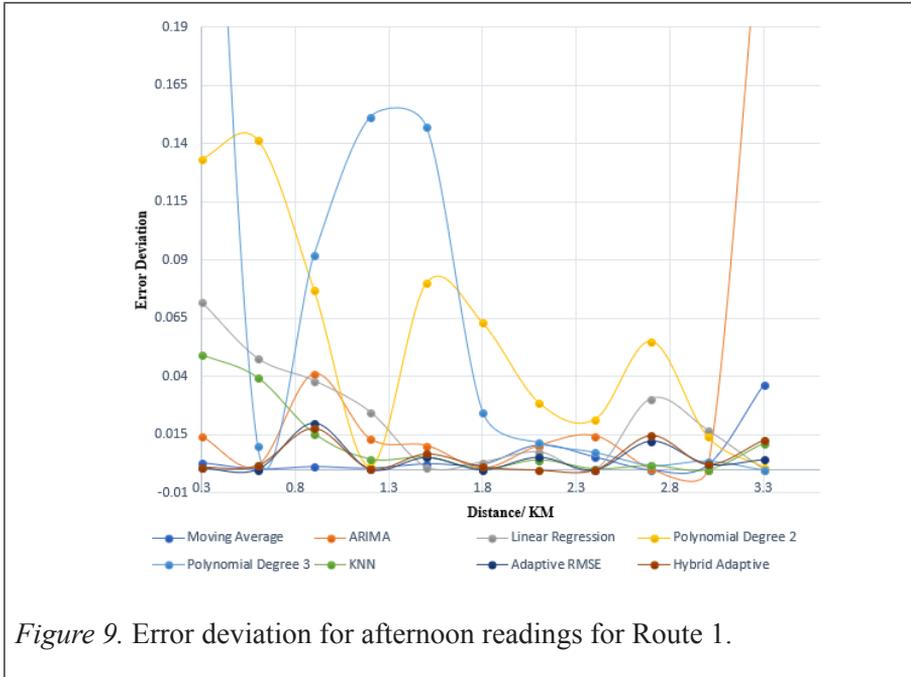


Figure 9. Error deviation for afternoon readings for Route 1.

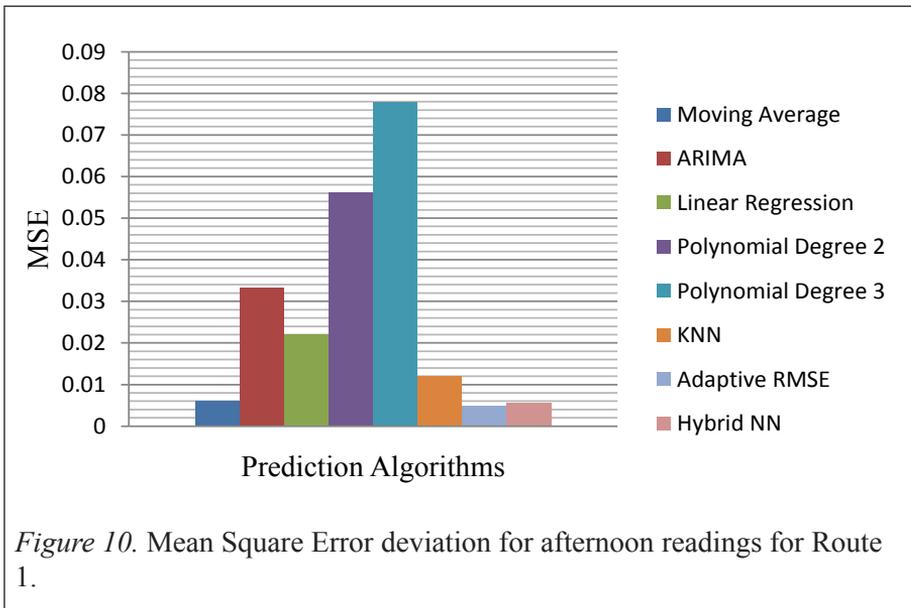


Figure 10. Mean Square Error deviation for afternoon readings for Route 1.

Figure 11 and 12 show the graph of the predicted and the actual congestion states against distance travelled for morning and afternoon readings of route 2. It is again observed that the adaptive schemes have the closest match to actual

value from 0.3km to 2.1km (Figure 11). In Figure 13 and 15, the adaptive schemes do not suffer large deviation compared to others prediction algorithms. Bar charts are given in Figure 14 and 16 to represent the MSE. Hence it can be concluded that the adaptive prediction algorithms provide the best performance.

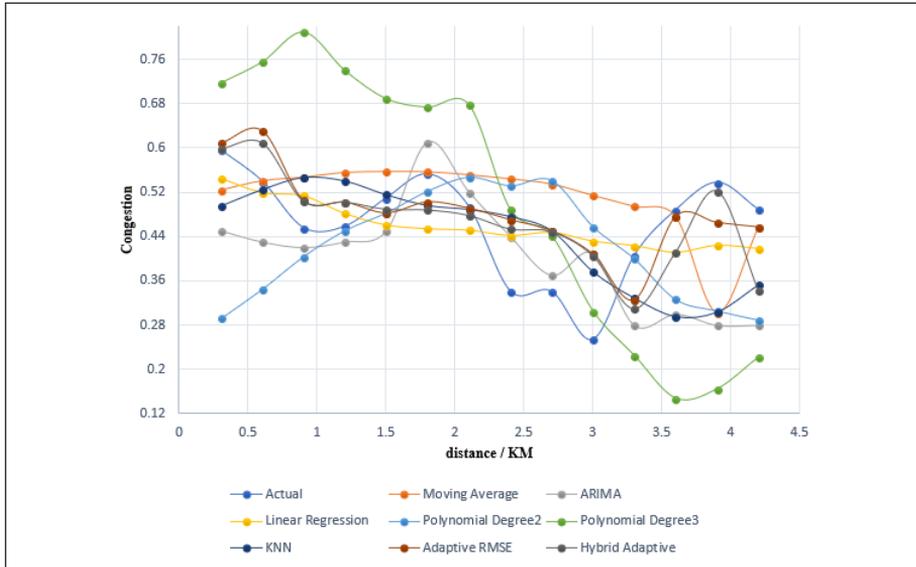


Figure 11. Morning congestion prediction results for Route 2.

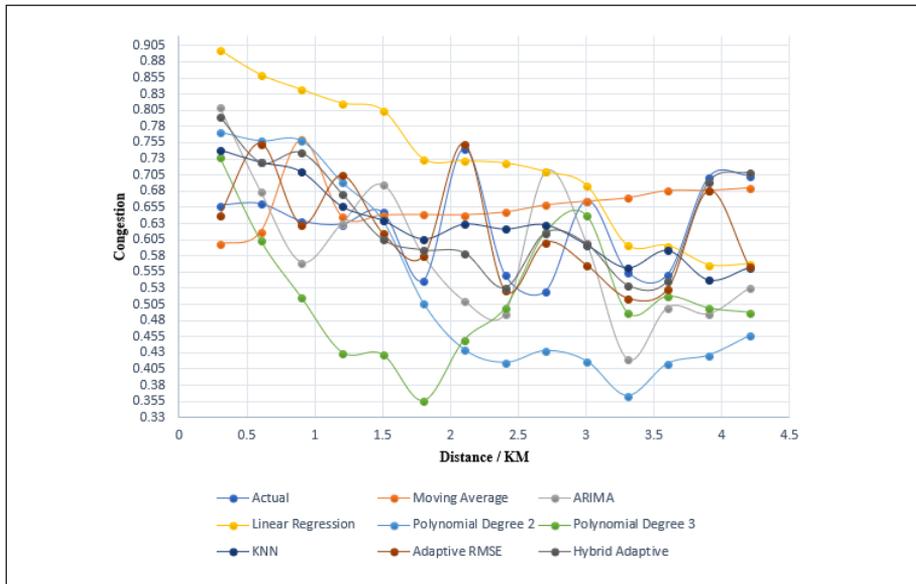


Figure 12. Afternoon congestion prediction result for Route 2.

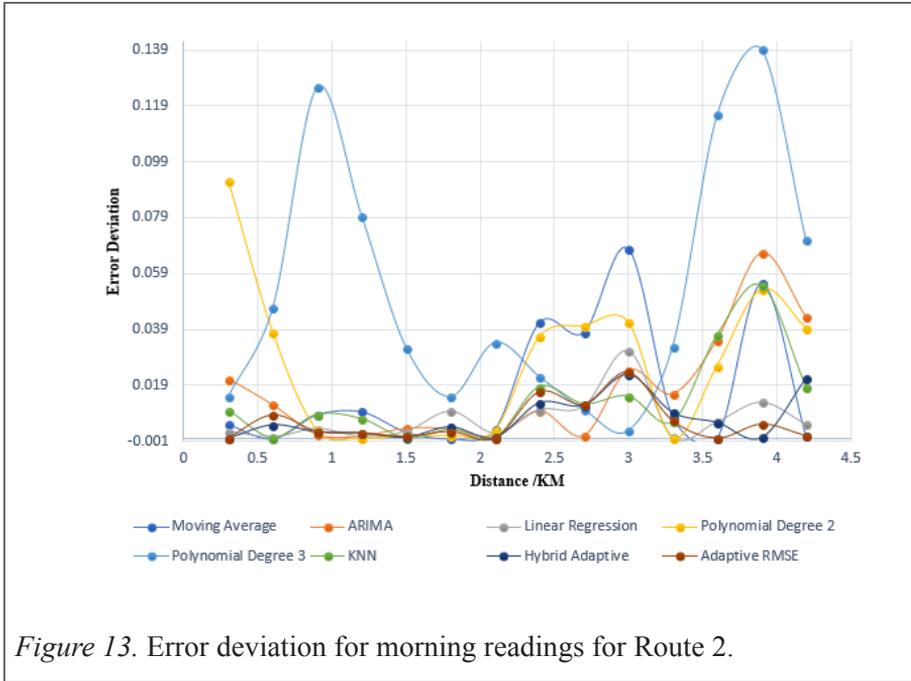


Figure 13. Error deviation for morning readings for Route 2.

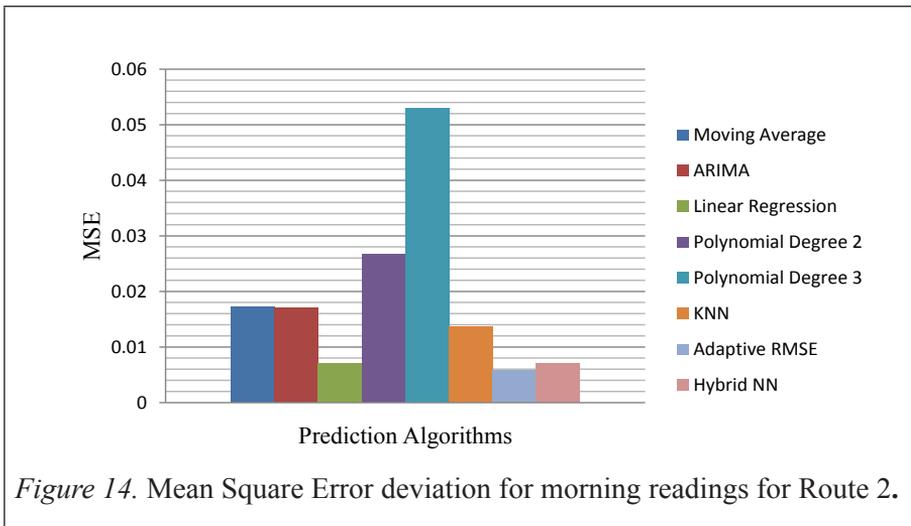


Figure 14. Mean Square Error deviation for morning readings for Route 2.

Figure 13 and 15 show the graphs of error deviation against distance for the route 2. The results show that the adaptive prediction (Adaptive RMSE) achieved the lowest error deviation compared with the other prediction schemes. It can also be observed that the Hybrid NN is the second best performing algorithm with an error deviation closest to the Adaptive RMSE.

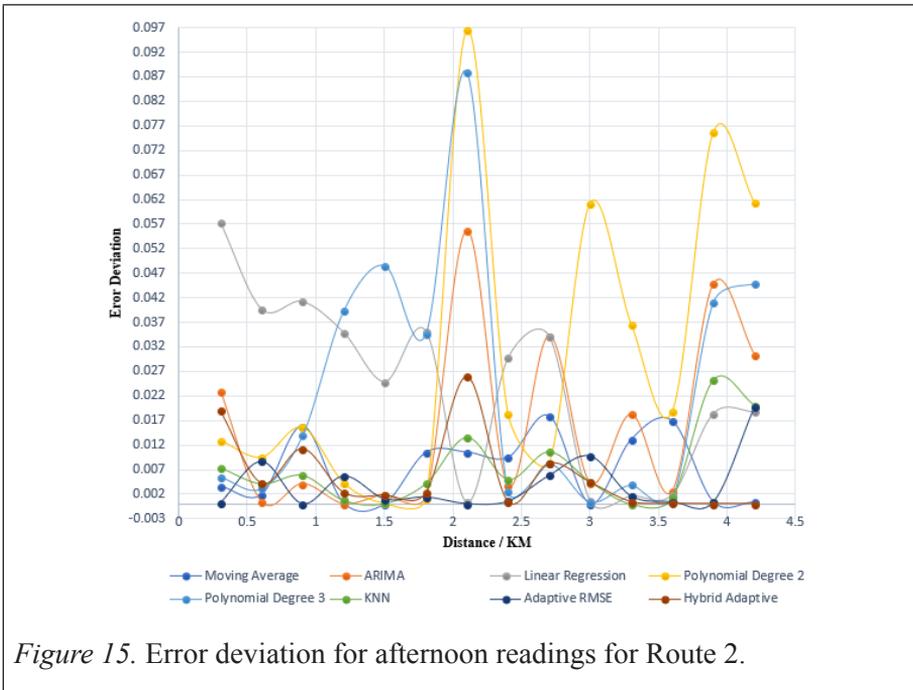


Figure 15. Error deviation for afternoon readings for Route 2.

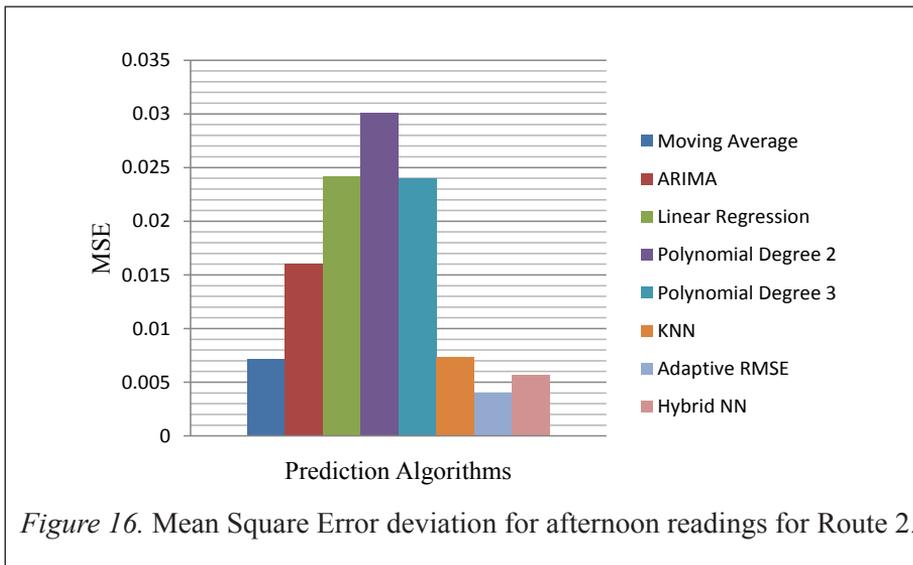


Figure 16. Mean Square Error deviation for afternoon readings for Route 2.

Table 3 provides the computation of RMSE for a sample data from morning readings for Route 1. Using Equation 3 and Equation 4, the RMSE is computed as follows:

The error deviation is calculated for each algorithm where p_i is the actual readings and P_0P_0 is the prediction result. The MSE is then calculated by summing all the error deviations and dividing by the total number of predictions which is 3($v=3$ in equation below). From the MSE, RMSE is obtained by applying the square root function. The results are given in Table 3.

Table 3

Computation of RMSE for Sample Morning Route 1 Data Set.

Actual	Moving Average	ARIMA	Linear Regression	Polynomial Degree 2	Polynomial Degree 3	KNN
0.62	0.44	0.42	0.49	0.49	1.19	0.52
0.51	0.47	0.48	0.46	0.64	1.06	0.55
0.55	0.48	0.38	0.50	0.68	0.73	0.56
RMSE	0.113	0.152	0.085	0.13	0.468	0.062

Table 4 gives the average MSE of the seven algorithms over the journey for route 2 and route 1. The overall performance indicates that the adaptive algorithm using RMSE only provides the lowest MSE and outperforms all other prediction techniques in terms of accuracy.

Table 4

Overall Performance Analysis for Route 1 and Route 2

Algorithm	Route 1		Route 2	
	Morning	Afternoon	Morning	Afternoon
Moving Average	0.91	0.70	0.53	0.27
ARIMA	1.18	0.66	0.46	0.53
K-Nearest Neighbors	0.28	0.41	0.49	0.17
Linear Regression	1.04	0.52	0.44	0.63
Polynomial Regression Degree 2	0.76	0.84	1.05	0.76
Polynomial Regression Degree 3	1.17	0.69	1.29	1.18
Adaptive RMSE	0.19	0.32	0.33	0.14
Hybrid NN	0.25	0.38	0.40	0.16

In Table 4, it is observed that on Route 1 the MSE of the adaptive algorithm (Adaptive RMSE) is significantly lower than the MSE of the regression

techniques by 33%. For time-series methods, it is noticed that the MSE is lowered to a small extent around 9%. Moreover, by comparing the results in Route 2, it is again observed that the MSE of the adaptive algorithm is lower than that of regression techniques and time-series algorithm by 30% and 3% respectively. The results show that the adaptive prediction scheme is a reliable approach to solve a complex problem with high variability of data like urban traffic flow.

CONCLUSION

This paper compared the performances of an adaptive prediction algorithm and a Hybrid NN prediction algorithm with five prediction techniques; Moving Average, ARIMA, Linear Regression, Polynomial Regression and KNN. A real-time cloud-based traffic congestion prediction system was proposed which consists of an in-vehicle tracking device and a control station. The tracking device was implemented using a microcontroller connected to a GPS and GSM/GPRS module which acquires and transmits the location of the bus to a cloud server in real-time. A control station interface has been implemented which accesses the location data of the bus, derives the traffic congestion based on vehicle's speeds and then performs a predictive analytics on the data. The RMSE criterion was used as a model selection criterion to select the best predictor to estimate the traffic congestion state. The performance of the proposed algorithm was evaluated and was found to achieve an average MSE of 0.2442 by the adaptive algorithm using RMSE. The study indicates that the adaptive prediction algorithm outperformed traditional prediction algorithms in terms of accuracy and is indeed a solution to improve the reliability of traffic information system. Further study may incorporate historical data to improve the prediction system as well as developing an onboard information system to avoid drivers taking congested areas.

ACKNOWLEDGMENT

The authors would like to thanks the University of Mauritius for providing the necessary facilities to conduct this research.

REFERENCES

Arduino Board Uno. (2017). Retrieved from <https://www.arduino.cc/en/Main/arduinoBoardUno>

- Amita, J., Sukhvir Singh, J., & Pradeep Kumar, G. (2015). Prediction of bus travel time using Artificial Neural Network. *International Journal for Traffic and Transport Engineering*, 410-424. doi: 10.7708/ijtte.2015.5(4).06
- Duda, R. O., Stork, D. G., & Hart, P. E. (2000). *Pattern classification* (2nd ed.). New York: John Wiley & Sons.
- Google Maps. (2017). Retrieved from <https://www.google.mu/maps/@-20.1793418,57.5370495,12z>
- Google Maps Directions API. (2017). Retrieved from <https://developers.google.com/maps/documentation/directions/>
- Liu, Y., Feng, X., Wang, Q., Zhang, H., & Wang, X. (2014). Prediction of urban road congestion using a Bayesian network approach. *Procedia - Social and Behavioral Sciences*, 671–678. doi:10.1016/j.sbspro.2014.07.259
- Lopez-Garcia, P., Onieva, E., Osaba, E., Masegosa, A. D., & Perallos, A. (2016). A hybrid method for short-term traffic congestion forecasting using genetic Algorithms and cross entropy. *IEEE Transactions on Intelligent Transportation Systems*, 557–569. doi:10.1109/tits.2015.2491365
- Lwin, H. T., & Naing, T. T. (2015). Estimation of road traffic congestion using GPS data. *International Journal of Advanced Research in Computer and Communication Engineering*, 1–5. doi:10.17148/ijarce.2015.41201
- MySQL Connectors. (2017). Retrieved from <https://www.mysql.com/products/connector/>
- MySQL. (2017). Retrieved from <https://dev.mysql.com/doc/>
- Ostertagová, E. (2012). Modelling using polynomial regression. *Procedia Engineering*, 500–506. doi:10.1016/j.proeng.2012.09.545
- PHP. (2017). Retrieved from <https://secure.php.net/>
- Picard, R. R., & Cook, R. D. (1984). Cross-validation of regression models. *Journal of the American Statistical Association*, 575–583. doi: 10.1080/01621459.1984.10478083

- Prakash Kaklij, S. (2015). Mining GPS data for traffic congestion detection and prediction. *International Journal of Science and Research*, 876–880.
- Prathilothamai, M., Sree Lakshmi, A. M., & Viswanthan, D. (2016). Cost effective road traffic prediction model using Apache spark. *Indian Journal of Science and Technology*. doi:10.17485/ijst/2016/v9i17/87334
- SIM900 GPRS/GSM Shield. (2015). Retrieved from http://linksprite.com/wiki/index.php5?title=SIM900_GPRS/GSM_Shield
- TomTom Traffic Index. (2017). Retrieved from http://www.tomtom.com/en_gb/trafficindex/list?citySize=LARGE&continent=ALL&country=ALL
- Yang, Q., Wang, J., Song, X., Kong, X., Xu, Z., & Zhang, B. (2015). Urban traffic congestion prediction using floating car trajectory data. *Algorithms and Architectures for Parallel Processing*, 18-30. doi: 10.1007/978-3-319-27122-4_2
- Zou, K., Tuncali, K., & Silverman, S. (2003). Correlation and simple linear regression. *Radiology*, 617-628. doi: 10.1148/radiol.2273011499