

Analysis and Prediction of Delay at Signalized Junctions in Bangalore

Naveen R¹, Nikhil T R², Akshay Hanagodimath³

¹M. Tech student, Civil Engineering Department, MSRUEAS, Bengaluru, India

^{2,3}Assistant Professor, Civil Engineering Department, MSRUEAS, Bengaluru, India

Abstract - The day by day Increase in Urbanization is increasing the traffic congestion which is affecting, Level of Service (LOS) and entire traffic system at signalized intersection, which constitute a major concern in every country. Queue Length and Delay are important parameter influencing the Level of Service (LOS) at the signalized intersection. Research's are going-on to improve the Level of Service (LOS) and Artificial intelligence/Machine learning/Deep learning is one of the superior method to improve Level of Service (LOS) at the signalized intersection. In this investigation, Highway Capacity Manual (HCM-2000) is used for determination of Delay and Level of Service (LOS) at the signalized intersections. Linear Regression Model is designed to predict the accurate delay. Fuzzy Logic Traffic controlled Signal is designed to get the required amount of green extension time based on Queue length and Waiting time. For determination of Delay at signalized intersection, Highway Capacity Manual-2000 and Regression Analysis is used. Python is more robust and open source which is very easy to install and user friendly where lot of solutions are found online. Delay value from the Highway Capacity Manual and Linear Regression analysis are almost Similar to each other. To determine Delay from Linear Regression, only 4 input parameters are need to be collected at signalized intersection. To determine Delay from Highway Capacity manual 18 input parameters are needed to be collected at signalized intersection. For determination of delay at new signalized intersection, artificial neural network is superior because only 4 parameters need to collected. Fuzzy Logic signal control system is designed to get the exact extension green time based on the Vehicle queue length and waiting time. Strength.in strength.

Key Words: LOS- Level of Service, HCM-Highway Capacity Model

1. INTRODUCTION

Day-by-day Increase in Urbanization is increasing the traffic congestion, which is affecting the entire traffic system and constitute a major concern in every country. Increase in Traffic congestion is taking a valuable time from the people, which is indirectly affecting the behaviour of the driver. Change in behaviour of a driver is also becoming a common factor for increase in Accidents especially at the signalized intersection. According to the ministry of Road transport and Highway 1317 crashes and 413 deaths every day. The number of road crashes have increased by 31% from 2007 to 2017. Level of Service (LOS) is used to analyze the intersection by categorizing traffic flow and assigning Level of service for the signalized intersection. Different signalized

intersection will be serving at different Level of Service (LOS) depending upon geometry of the road, Vehicle volume, traffic signal timing.

1.1 Delay and LOS

We have all been there. When driving along suddenly, traffic comes to a standstill. Then, after inching along for what seems like an eternity, just as suddenly as it stopped, traffic starts to move again, the road opening up as if by magic. It can be dangerous to become a part of a sudden traffic jam, and what causes traffic jams and car crashes is based largely on the number of distractions in cars today that can affect the way you react to certain events. Highway capacity manual (HCM) developed by the transportation research board of USA provides some procedure to determine level of service. It divides the quality of traffic into six levels ranging from level A to level F. Level A represents the best quality of traffic where the driver has the freedom to drive with free flow speed and level F represents the worst quality of traffic.

1.2 Regression Analysis

In statistics, linear regression is a linear approach to modeling the relationship between a scalar response (or dependent variable) and one or more explanatory variables (or independent variables). The case of one explanatory variable is called simple linear regression. For more than one explanatory variable, the process is called multiple linear regressions. This term is distinct from multivariate linear regression, where multiple correlated dependent variables are predicted, rather than a single scalar variable.

1.3 Fuzzy Logic

Fuzzy logic is a form of many-valued logic in which the truth values of variables may be any real number between 0 and 1 both inclusive. It is employed to handle the concept of partial truth, where the truth value may range between completely true and completely false. By contrast, in Boolean logic, the truth values of variables may only be the integer values 0 or 1.

2. LITERATURE REVIEW

Javed alam and Dr.Manoj Kuamr Pandey(2005).et.al: In this paper the two stage traffic light system is designed using fuzzy logic and its performance is compared with the pre-timed signal controller. Extension time decision module(ETDM) will calculates the green light time i.e.

extension time of the respected phase which is having higher urgency can be calculated according to the higher number of vehicles. Two inputs are used for the design of fuzzy logic such as Queue-lane-1, queue-lane-2 and the extension green time is used as the output from the fuzzy logic signal controller. queue-lane-1 is used to determine the number of vehicles in lane 1 and queue-lane-2 is used to determine the number of vehicles in the lane-2.

C. M. Mwangi, S. M. Kang(2014).et.al: This paper proposes a fuzzy logic system to control traffic signals on a signalized intersection. The fuzzy logic controller (FLC) dynamically controls the traffic light timings and phase sequence to ensure smooth flow of traffic, decrease traffic delays and thus increase the intersection capacity. A fuzzy logic traffic control simulation model is developed and tested using MATLAB/ SIMULINK software. Queue length and waiting time are used as the input for the fuzzy logic traffic control. Phase selection and green phase extension model is taken as output. The fuzzy logic system will use information from the inductive loop sensors to determine the number of number of cars between the two set of the sensors.

Shen Rong and Zhang Bao-wen(2018) The paper herein introduces the algorithm and model of the field of machine learning. Linear regression model is used to analyze the sale of iced products of company and the effect of temperature variation on the sale. Firstly, we cleanse the data collected one year ago and analyze data at the same time. Then we choose forecast temperature as the independent variable and the sale of iced products as the dependent variable to establish simple linear regression model for analysis. We use the object-oriented programming language Python3.6 and introduce the linear regression function. Programming language can also make data analysis in the field of data mining easier. The final result correctly leads the company to adjust the production and sale of iced products flexibly according to the variation of temperature, which definitely provides great commercial value and offers crucial theoretical foundation for the sale of other companies who produce iced products.

3. METHODS AND METHODOLOGY

3.1 Methodology:

The traffic movement survey is carried out in the Signalized junction of Bangalore. The delay is calculated for each signal using Highway Capacity Manual 2000. The results obtained are trained in the Linear Regression Model. The output from the Highway Capacity Manual-2000 and the Linear Regression model is compared for similarity. Suitable solution for the reduction in delay at the junction. From the research papers it is seen that out of 18 parameters which are required to calculate delay in Highway Capacity Manual-2000 can be reduced to only 4 parameters(PCU, green/cycle length time, cycle time and width of road) to predict the delay for the signalized junction. Addition to 4 parameters for more specific results, an additional parameter is added as

input which is “time” showing delay for each hour irrespective of the peak time. The obtained delays are trained in the regression model. The regression model is selected as the data provided are in sequenced manner. The python having the inbuilt functions builds the model with call functions and fits the model and predicts/provides the results depending on the data input. The predicted data are compared with the actual results obtained from the manual. Suitable solution is developed depending on the delay at each signal.

3.2 Traffic Survey

Traffic movement surveys were carried out in the following intersections of Bangalore for determination of delay, as shown in below Tables.

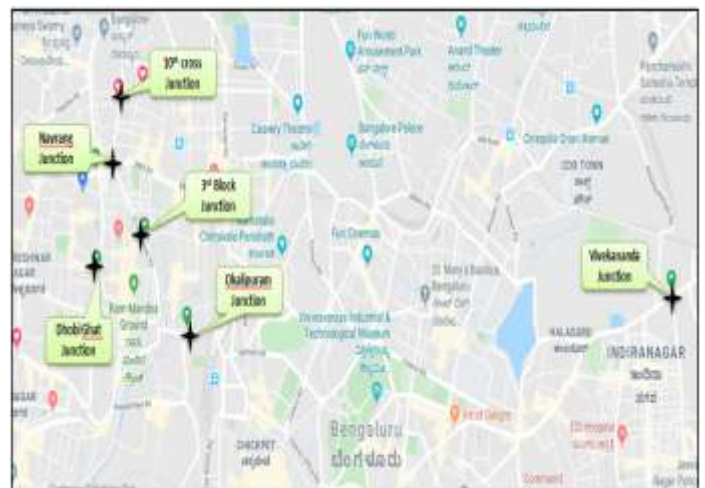


Figure 1: Locations of Traffic Survey in Bangalore

3.2 Highway Capacity Manual

The delay for the signal is calculated using HCM-2000 for every hour for all the legs. The delay is obtained using 18 parameters which are mentioned in the manual. The obtained parameters define the delay at the junction. The delays calculated are then trained in the linear regression model so that the model predicts the required delay at any particular time.

3.3 Linear Regression

Linear regression is chosen for the prediction of delay. Out of 18 parameters only 4 parameters are taken as input for the model. The obtained delay is bifurcated as training data and testing data. The models is developed using Python3.6 which is robust and having many inbuilt functions. The model is thus trained using the 4 inputs and the obtained delay from HCM-2000. The model in turn trains itself and predicts the delay at any point of time.

```
import pandas as pd
import math
import numpy as np
```

```
import warnings
warnings.filterwarnings('ignore')
from sklearn.model_selection import KFold
from sklearn.linear_model import
LinearRegression,Lasso,Ridge
from sklearn.model_selection import train_test_split

ld=pd.read_excel(r'import the file from the
location',encoding='latin')
ld.head()
ld["Timings"].nunique()
ld.columns
l2_dummies=pd.get_dummies(ld["Timings"])
ld=pd.concat((ld,l2_dummies),1)
ld=ld.drop("Timings",1)
l2_dummies
for col in ['Movement']:
    ld[col]=ld[col].astype('str')
lp_dummies=pd.get_dummies(ld["Movement"])
ld=pd.concat((ld,lp_dummies),1)
ld.dtypes
ld=ld.drop("Name of signal intersection",1)
ld.dtypes
ld=ld.drop(["Movement"],axis=1)
ld.dropna(axis=0,inplace=True)
ld_train,ld_test=train_test_split(ld,test_size=0.2,rand
om_state=2)
lm=LinearRegression()
x_train=ld_train.drop(["Delay (Sec/veh)"],1)
y_train=ld_train["Delay (Sec/veh)"]
y_train.head()
x_test=ld_test.drop(["Delay (Sec/veh)"],1)
y_test=ld_test["Delay (Sec/veh)"]
lm.fit(x_train,y_train)
x_test.head()
residual=p_test-y_test
l_rmse=np.sqrt(np.dot(residual,residual)/len(p_test
))
l_rmse
coefs=lm.coef_
features=x_train.columns
list(zip(features,coefs))
# Finding best value of penalty weight with cross
validation for ridge regression
alphas=np.linspace(.0001,10,100)
# We need to reset index for cross validation to work
without hitch
x_train.reset_index(drop=True,inplace=True)
y_train.reset_index(drop=True,inplace=True)
rmse_list=[]
for a in alphas:
    ridge = Ridge(fit_intercept=True, alpha=a)
    # computing average RMSE across 10-fold cross
validation
    kf = KFold(n_splits=10)
    xval_err = 0
    for train, test in kf.split(x_train):
        ridge.fit(x_train.loc[train], y_train[train])
        p = ridge.predict(x_train.loc[test])
```

```
err = p - y_train[test]
xval_err += np.dot(err,err)
rmse_10cv = np.sqrt(xval_err/len(x_train))
# uncomment below to print rmse values for
individual alphas*
# print('{:.3f}\t {:.6f}\t '.format(a,rmse_10cv))
rmse_list.extend([rmse_10cv])
best_alpha=alphas[rmse_list==min(rmse_list)]
print('Alpha with min 10cv error is : ',best_alpha )
ridge=Ridge(fit_intercept=True,alpha=best_alpha)
ridge.fit(x_train,y_train)
p_test=ridge.predict(x_test)
residual=p_test-y_test
rmse_ridge=np.sqrt(np.dot(residual,residual)/len(p
_test))
rmse_ridge
alphas=np.linspace(0.0001,1,100)
rmse_list=[]
for a in alphas:
    lasso = Lasso(fit_intercept=True,
alpha=a,max_iter=10000)
# computing RMSE using 10-fold cross validation
kf = KFold(n_splits=10)
xval_err = 0
for train, test in kf.split(x_train):
    lasso.fit(x_train.loc[train], y_train[train])
    p =lasso.predict(x_train.loc[test])
    err = p - y_train[test]
    xval_err += np.dot(err,err)
    rmse_10cv = np.sqrt(xval_err/len(x_train))
    rmse_list.extend([rmse_10cv])
# Uncomment below to print rmse values of
individual alphas
    print('{:.3f}\t {:.4f}\t '.format(a,rmse_10cv))
    best_alpha=alphas[rmse_list==min(rmse_list)]
    print('Alpha with min 10cv error is : ',best_alpha )
    lasso=Lasso(fit_intercept=True,alpha=best_alpha)
    lasso.fit(x_train,y_train)
    p_test=lasso.predict(x_test)
    residual=p_test-y_test
    rmse_lasso=np.sqrt(np.dot(residual,residual)/len(p
_test))
    rmse_lasso
    list(zip(x_train.columns,lasso.coef_))
    x_test['y_pred']=p_test
    x_test['y_actual']=y_test
```

x_test.head()

3.4. Results from Linear Regression Model

Sl No	Delay from HCM-2000 (sec/vehicle)	Delay from the regression model		
		Linear	Lasso	Ridge
1	53.63	66.45	68.26	67.62
2	123.47	136.58	128.66	132.27
3	2760.36	3428.22	3437.73	3373.89

4	323.94	302.34	315.17	332.08
5	2751.64	2764.06	2766.48	2764.22

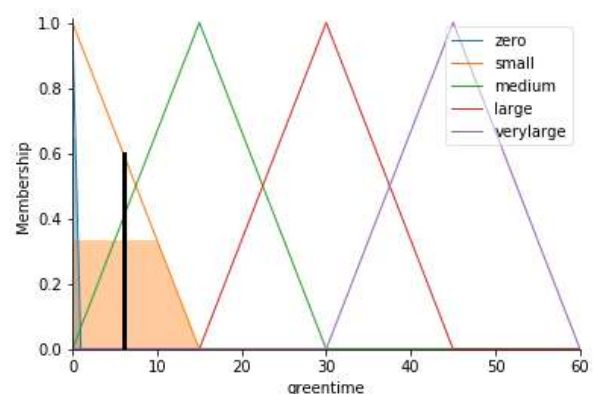
3.5. Design of Fuzzy Logic Using Python

```
import numpy as np
import warnings
warnings.filterwarnings("ignore")
import skfuzzy as fuzz
from skfuzzy import control as ctrl
waitingtime = ctrl.Antecedent(np.arange(0, 61, 1),
'waitingtime')
queuelength = ctrl.Antecedent(np.arange(0, 61, 1),
'queuelength')
greentime = ctrl.Consequent(np.arange(0, 61, 1),
'greentime')
waitingtime.automf(5)
queuelength.automf(5)
greentime['zero'] = fuzz.trimf (greentime.universe,
[0, 0, 0])
greentime['small'] = fuzz.trimf (greentime.universe,
[0, 0, 15])
greentime['medium'] = fuzz.trimf
(greentime.universe, [0, 15, 30])
greentime['large'] = fuzz.trimf (greentime.universe,
[15, 30, 45])
greentime['verylarge'] = fuzz.trimf
(greentime.universe, [30,45,60])
waitingtime.view()
rule1 = ctrl.Rule(waitingtime['poor'] |
queuelength['poor'], greentime['zero'])
rule2 = ctrl.Rule(waitingtime['poor'] |
queuelength['mediocre'], greentime['zero'])
rule3 = ctrl.Rule(waitingtime['poor'] |
queuelength['average'], greentime['small'])
rule4 = ctrl.Rule(waitingtime['poor'] |
queuelength['decent'], greentime['small'])
rule5 = ctrl.Rule(waitingtime['poor'] |
queuelength['good'], greentime['medium'])
rule6 = ctrl.Rule(waitingtime['mediocre'] |
queuelength['poor'], greentime['zero'])
rule7 = ctrl.Rule(waitingtime['mediocre'] |
queuelength['mediocre'], greentime['zero'])
rule8 = ctrl.Rule(waitingtime['mediocre'] |
queuelength['average'], greentime['small'])
rule9 = ctrl.Rule(waitingtime['mediocre'] |
queuelength['decent'], greentime['medium'])
rule10 = ctrl.Rule(waitingtime['mediocre'] |
queuelength['good'], greentime['large'])
rule11 = ctrl.Rule(waitingtime['average'] |
queuelength['poor'], greentime['small'])
rule12 = ctrl.Rule(waitingtime['average'] |
queuelength['mediocre'], greentime['small'])
rule13 = ctrl.Rule(waitingtime['average'] |
queuelength['average'], greentime['medium'])
rule14 = ctrl.Rule(waitingtime['average'] |
queuelength['decent'], greentime['medium'])
```

```
rule15 = ctrl.Rule(waitingtime['average'] |
queuelength['good'], greentime['medium'])
rule16 = ctrl.Rule(waitingtime['decent'] |
queuelength['poor'], greentime['small'])
rule17 = ctrl.Rule(waitingtime['decent'] |
queuelength['mediocre'], greentime['small'])
rule18 = ctrl.Rule(waitingtime['decent'] |
queuelength['average'], greentime['medium'])
rule19 = ctrl.Rule(waitingtime['decent'] |
queuelength['decent'], greentime['large'])
rule20 = ctrl.Rule(waitingtime['decent'] |
queuelength['poor'], greentime['large'])
rule21 = ctrl.Rule(waitingtime['good'] |
queuelength['poor'], greentime['medium'])
rule22 = ctrl.Rule(waitingtime['good'] |
queuelength['mediocre'], greentime['medium'])
rule23 = ctrl.Rule(waitingtime['good'] |
queuelength['average'], greentime['medium'])
rule23 = ctrl.Rule(waitingtime['good'] |
queuelength['decent'], greentime['medium'])
rule24 = ctrl.Rule(waitingtime['good'] |
queuelength['good'], greentime['large'])
```

```
#rule2.view()
tipping_ctrl = ctrl.ControlSystem([rule1, rule2,
rule3])
tipping = ctrl.ControlSystemSimulation(tipping_ctrl)
def calculateGreentime(waitingTime,queueLength):
    tipping.input['waitingtime'] = waitingTime
    tipping.input['queuelength'] = queueLength
    tipping.compute()
    print(tipping.output['greentime'])
    greentime.view(sim=tipping)
calculateGreentime(10.0,5.30)
```

3.6. Results from Fuzzy Logic Model



4. CONCLUSION

- For determination of Delay at signalized intersection, Highway Capacity Manual-2000 and Regression Analysis is used.

- Python is more robust and open source which is very easy to install and user friendly where lot of solutions are found online.
- Delay value from the Highway Capacity Manual and Linear Regression analysis are almost similar to each other.
- To determine Delay from Linear Regression, only 4 input parameters are need to be collected at signalized intersection.
- To determine Delay from Highway Capacity manual 18 input parameters are needed to be collected at signalized intersection.
- For determination of delay at new signalized intersection, artificial neural network is superior because only 4 parameters need to collected.
- Fuzzy Logic signal control system is designed to get the exact extension green time based on the Vehicle queue length and waiting time.

REFERENCES

- [1] J. Mohamed Rizwan , 2016. Multi Layer Perception type artificial Neural Network based traffic control. International Journal of Engineering and Innovative Technology.
- [2] Javed alam., 2015. Design and analysis of a two stage traffic light system using fuzzy logic. Journal of information technology.
- [3] mk, J. a. a. P., 2015. Design and Analysis of a two stage traffic light system using Fuzzy Logic. Journal of information Technology and software Engineering.
- [4] murat, O. B. a. Y. S., july 2006. Modelling Vehicle delays at signalized intersections: Artificial Neural Network. Journal of Scientific and Industrial Research.
- [5] Ozgur Baskan,, 2006. Modelling vehicle delays at signalized intersections: Artificial neural networks approach. Journal of scientific and industrial research.
- [6] Sahar araghi, ., 2014. Optomal Design of traffic signal controller using Neural Networks and Fuzzy Logic System. International joint conference on Neural Networks(IJCNN) .