International Journal on Emerging Technologies 11(3): 1128-1133(2020)
ISSN No. (Print): 0975-8364
ISSN No. (Online): 2249-3255

A Travel Time Prediction by Applying Ensemble Machine Learning Techniques

Mrinmoyee Chattoraj[1] and V. Udaya Rani[2]
[1]Research Scholar, School of Computing & IT, REVA University, Kattigenahalli,Yelahanka, Bengaluru, India.
[2]Associate Professor,School of Computing & IT, REVA University, Kattigenahalli, Yelahanka, Bengaluru, India.

(Corresponding author: Mrinmoyee Chattoraj)
(Received 23 April 2020, Revised 29 May 2020, Accepted 30 May 2020)
(Published by Research Trend, Website: www.researchtrend.net)

ABSTRACT: Estimating and predicting the travel time is an important feature of Traveller Information System(TIS). In the past many approaches like k-nearest neighbours (k-NN), gradient boosting decision tree (GBDT) and support vector machines (SVMs) were used to predict the travel time. In this research an effort have been made to improve the accuracy of the predicted travel time using ensemble machine learning algorithms. The proposed model divides the prediction work into two parts by first forecasting the clusters of the two step clustering model and then forecast the tendency of forecast using the ARIMA and XGBoost model for the linear and the non-linear segments. Finally we find the summation of the weights of all the models and it has been observed that the predicted results out performs the other ensemble models.

Keywords: Auto Regressive Model, Bagging, Boosting, Clustering, Data driven method, Travel time prediction.

I. INTRODUCTION

Traffic congestion is one of the major problem in today's world which leads to a lot of adverse effects such as time delays, economic damage, extra consumption of fuel and environmental issues. Travel Time Prediction is a part of Intelligent Transportation System which plays a major role in planning different types of operations and helps to reduce the traffic congestion. Previously predicting the travel time was difficult due to the limited resources of obtaining travel time data .At present many vehicle detection devices are there such as inductive loop detectors, microwave radar detectors which are placed on the intersection of roads and the GPS gets the updated data of the moving cars with the help of these 'loop detectors' [1, 2]. These are fixed detectors and so cannot directly measure the travel time. Many algorithms have been proposed to detect the traffic data from these devices but it is unable to directly measure the travel time and there are some shortcomings of these algorithms [3-5]. Many advanced data collection techniques such as Floating Car Data (FCD) and automatic vehicle identification is used. Floating Car Data is a traffic monitoring system which has a low cost and covers a wide area. This technique depends on a probe vehicle which is a vehicle with sensors and moves along with traffic. Modelling the data from these devices in order to predict the travel time is a challenging job due to the following reasons:

– The latency during the peak hours is more as compared to normal hours.

– Travel time is influenced by the trend of the nearest historical travel time and it has been observed that free flow of traffic changes to a congested traffic at a particular time

– Many machine learning models are there which observes the non-linear pattern of travel time but due to overfitting it cannot give the accurate results.

In this paper we do Static travel time prediction where the travel time is predicted before the journey commences and the predicted value is constant throughout the journey. The type of prediction gives a general idea to the drivers about the travel time of the taxis despite the rare exception condition. We present an approach by which optimization is done using a mathematical model which is used in machine learning As we are training the model data we try to reduce the cost of errors between the data points and our model. Here we use the power of machine learning to forecast the travel time from a particular source to a specified destination.

The paper is systematized in this manner: Section II gives us information of the related works which have been obtained in travel time prediction. Section III analyses the definition of the problem which we need to solve during this research. Section IV describes the method applied and Section V defines the investigations performed and the results. Finally we conclude the results in Section VI.

II. RELATED WORKS

A lot of research work have been done to predict the traffic flow parameters such as the volume of the traffic and the speed of the traffic. Although limited effort have been spent on short term prediction of travel time but still we can find a lot of similarities between the traffic flow and the travel time. In order to predict the traffic flow there are basically three categories: the statistical models, the machine learning models and the hybrid models.

Statistical models like linear regression, Random Walk (RW) model, Autoregressive Integrated Moving Average (ARIMA) model, generalized autoregressive conditional heteroscedasticity (GARCH) model, Vector autoregressive (VAR) model and the locally weighted regression (LWR) model are widely used to predict the parameters of traffic flow. Van Der Voort *et al.*, developed a short term traffic forecasting model using the K ARIMA method in which a Kohonen map was tuned with the ARIMA model [6]. Williams *et al.,* explained the theoretical basis for modeling and forecasting univariate traffic condition data by using seasonal ARIMA process [7]. Till date various ARIMA models were applied [8, 9] but since the traffic flow involves a lot of nonlinear features which are complex in nature it was difficult for the statistical based model to handle it because there are lot of linear assumptions in these type of model due to which it was unable to capture the hidden nonlinear patterns of the time series related to traffic.

Due to the shortcomings of the statistical models many machine learning models came into existence. Machine learning models has powerful learning capabilities which are self-regulating. Bezuglov *et al*., studied three possible applications and the level of accuracy for three Grey System models which were used to predict the speed and travel time for short term traffic [10]. Artificial Neural Networks (ANN) [11], Support Vector Regression (SVR) was used to predict the traffic flow [12], Random Forest (RF) techniques such as Gradient boosting technique [13] was used by Zhang *et al*., to solve travel time prediction problem. Many other intelligent optimization techniques was also used to predict the flow of traffic and their empirical results show that they are superior to the statistical based model. Karlaftis, M. G., & Vlahogianni [14] did a comparative study of the statistical model and the ANN model and found that the ANN was a better predictor than the ARIMA model and the LWR model. Lippi *et al*., [15] conducted a comparison of machine learning models with time series models and found that Support Vector Regression outperforms the time series models. Deb *et al.,* [16] predicted the travel time keeping into consideration the impact of weather on traffic conditions. Huang *et al*., [17] predicted the travel time using tree based ensemble methods. Chen *et al*., [18] also did long term traffic prediction using gradient boosting. Although machine learning models helps us to predict the traffic parameter still they have their shortcomings. Firstly they fail to capture the changes in the normal trends. Secondly it is difficult to select the parameters of the machine learning model. In order to overcome these problems hybrid models are developed. In our model we will combine the linear properties of the statistical model and the nonlinear and memorizing capability of the machine learning model in order to overcome the over fitting problems. It will obtain better prediction than normal machine learning algorithms.

III. PROBLEM DEFINITION

In this part we predict the 'Static Travel Time' in context of the taxi service. We consider the taxi trip travel time as the total time a passenger stays inside the vehicle till the destination is reached .This is also called as in-vehicle travel time. Travel time and travel speed have a one-to–one correspondence so the modeller have the choice to model directly the travel time or can model the travel speed. Static travel time predicts the travel time at the beginning of the journey and does not collect data after the journey has started. Taxi trip T is defined as a tuple(s, d, t) where s and d are the geographical location of the source and destination, t is the time at which the trip begins. Source and destination are tuples which is represented as (latitude, longitude).Time of the day t is taken in seconds. The main aim will be to forecast the travel time from the source to destination as accurately as possible before the commencement of the journey.

IV. METHODOLOGY

Background: While predicting problems which involves unstructured data Artificial Neural Networks (ANN) was very helpful [19]. In case of small to medium structured data since ANN was not so successful so decision tree based algorithms were used. Decision tree based algorithms graphically represent a problem which makes decisions based on some conditions. Since a decision trees usually makes poor prediction model so Bagging was used later [20]. Bagging is an ensemble of Meta heuristic algorithm combining predictions from a number of trees. It is designed to improve the accuracy and stability of the machine learning algorithm. Later Random forest technique was used which is a bagging based algorithm where we consider only a subset of the selected features to build a collection of decision trees which is known as forest. The errors in these models were minimized and sequentially new models were built and so the performance of the previous models were increased or boosted. Thus this technique was known as boosting [21]. Later gradient descent algorithm was used to minimize the errors in sequential models. This technique was known as Gradient boosting [22]. For regression and classification problems we use a machine learning technique which is called gradient boosting. This technique produces a prediction model which is an ensemble of weak prediction models such as decision trees [23]. This technique builds the model in stages like any other boosting algorithm and then generalizes them by optimizing any arbitrary differentiable loss function. This technique of optimizing the loss function of the gradient boosting algorithm is also known as XGBoost (eXtreme Gradient Boosting) which is a scalable machine learning algorithm used for boosting decision trees [24]. XGBoost is a supervised machine learning problem which is used by training the models on multiple attributes(x1, x2, x3…) to predict a target variant .One important aspect of XGBoost models is that they scale well for multiple conditions with less requirement of resources than the existing prediction models. Parallel and distributed computing within XGBoost speeds up the model learning and enables quicker model exploration. XGBoost helps to regularize objective model to prevent overfitting and the algorithm is capable of handling all kinds of sparsity patterns.

Dataset: In this paper we use the taxi trip trajectory data set provided by New York City Taxi and Limousine Commission [25], which includes pickup time, geo-coordinates, number of passengers, and several other variables. This dataset contains millions of trip trajectories that took place in New York City. We are going to build a model that predicts the total ride duration of taxi trips in New York City. Since we are given each location coordinates, we calculate the Manhattan distances between each pair of points and count the longitude and latitude differences to get a sense of direction(East to West, North to South).

V. METHODOLOGIES APPLIED

Feature Selection: It is a technique to reduce the number of input variables. The most common feature selection techniques are Wrapper Selection Method and Filter Feature Selection technique. In our proposed technique we use wrapper selection technique where we create a model considering different combinations of features and finally we select those features whose performance metric is highest.

Two Step Clustering: It is a technique of making groups within the dataset which will not be apparent. The main steps in this clustering are as below:

Preclustering: Here outliers are handled by using the features of clusters.
Clustering: The optimal number of clusters are determined by merging sub clusters.
Clustering Membership assignment: The distance between sub-clusters are calculated.
Validating Results: The performance of clusters are calculated using Silhouette coefficient. If x is the mean distance between cluster and its sample and y is the distance between other clusters and the sample. Our main objective will be to maximize S given in Eqn. (1)

$$S = \frac{y-x}{\max(x,y)} \tag{1}$$

Parameter Determination using ARIMA Model: ARIMA is an auto regressive statistical model which is used to forecast and analyse time series.

VI. PROPOSED MODEL

Step 1: Initially we apply two-step clustering model to divide our data into various clusters according to their features and then the prediction results are calculated based on the boosting algorithm.
Step 2: We combine the ARIMA with a boosting algorithm to predict the tendency of the time series. It considers the strength of linear fitting from the ARIMA model and then using the residuals of the ARIMA model to do non-linear mapping using the boosting algorithm. The residual vector e=(r_1, r2, r3……r_n) is obtained by the ARIMA model which is the difference between actual and the predicted values.
Step 3: A combination model is developed by assigning weight w_1 to the model of Step 1 and w_2 to the model of Step 2.The objective is to minimize the MSE in the equation which is the mean of the error sum square.
Objective function:

Minimize $\mathrm{MSE}=\frac{1}{n}\sum_{k=1}^{n}[\widehat{Y_c}(K)-y(K)]^2$ (2)

In Eqn. (2) the forecasted value of the k^{th} sample of the combined model is denoted as $\widehat{Y_c}(K)$ and the actual values of the kth sample is denoted by y(K).The combined model of the k^{th} sample is calculated by Eqn. (3) where w_1 and w_2 are the respective weights assigned to the first and second model.$\widehat{Y_1}(K)$ and $\widehat{Y_2}(K)$ are the forecasted values of the first and second model.

$$\widehat{Y_c}(K) = w_1.\widehat{Y_1}(K) + w_2.\widehat{Y_2}(K) \tag{3}$$

In order to calculate the optimal values of w_1 and w_2 the equations are transformed into matrix operations. Matrix A in Eqn. (4) consists of the predicted values of the models in step 1 and 2 respectively.

$$A = \begin{bmatrix} \widehat{y_1}(1) & \widehat{y_2}(1) \\ \widehat{y_1}(2) & \widehat{y_2}(2) \\ \vdots & \vdots \\ \widehat{y_1}(n) & \widehat{y_2}(n) \end{bmatrix} \tag{4}$$

The matrix weight is given as $W = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}$ (5)

The matrix actual values are given by Y =[y(1), y(2),…….y(n)] (6)

Hence Eqn. (4) can be transformed into
A.W=Y (7)

Multiplying the transpose of matrix A on both the sides of Eqn. (7) we get

$$A^T.A.W = A^T.Y \tag{8}$$

Hence $W = (A^T.A)^{-1}.A^T.Y$ (9)

According to Eqn. (9) the optimal weights w_1 and w_2 calculated to forecast the best result of the combined model. The Flowchart of the proposed model is shown in Fig. 1 which shows the various stages of the model.

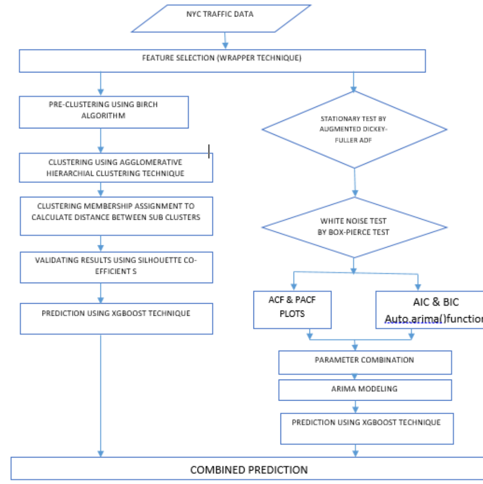

Fig. 1. Proposed Model of various stages.

VII. EXPERIMENTS AND RESULTS

Machine Learning and Statistical methods were used for comparison which includes the ARIMA model [26] and the clustering model [27]. The ARIMA model captures the autocorrelation and can generate the predictive intervals using three parameters p, q and r. In ARIMA (p, q, r), p is the number of auto regression terms, q is the difference order and r is the average window size. To specify these parameters initially we apply differencing lag-1 for a moving trend and then fit the ARIMA model to the different time series. In case of k-means clustering and traffic speed clustering the total time is divided into multiple time periods based on travel speed time series.
Evaluation Indexes for Model Performance: The performance of the combined new model was compared with the existing models such as ARIMA and k means clustering using the four model performance indicators as shown in Table 1 where the forecasted value of the k^{th} sample of the model is denoted as $\widehat{Y_c}(K)$ and the actual values of the kth sample is denoted by y (K).
In the clustering series there are 14 attributes as shown by the Feature importance graph in Fig 2, which tells us which feature has more impact in predicting the results. Generally Feature importance is a measure that specifies how important a particular feature is in the creation of the 'boosted decision trees' inside the model. If a particular attribute is used more in making decisions of a decision tree, its relative importance increases. For each attribute of a dataset we explicitly calculate its importance and then rank these attributes after doing a comparative study with each other. Each feature's importance is based on a single decision tree and how much the attribute splits from a particular point. The performance of the attribute improves on the amount it splits and the weight of the node depends on the number of observations. The measure of performance is called as Gini Index which is used to select the split points and it's also used to measure the error function. We then find the average of all decision trees inside the model to find the feature importance. In order to verify the performance of the proposed model the data set is partitioned into the training set, validation set, and the test set so as

to satisfy the requirements of different models. The data is used as follows: For the clustering series 80% of the data is used as training set 1 for the two step clustering model. The resulting samples from this two-step clustering is used to construct the boosting models. The test set with the remaining 20% of the data is selected to validate the clustering XGBoost model. Therefore we first partition the test set into corresponding clusters with the help of two step clustering model and then we check the validity of the clustering XGBoost model. For ARIMA XGBoost, 78% of the training set 2 samples are used to construct the ARIMA and with the help of validation set the ARIMA residuals are calculated. With the remaining 22% data we check the performance of the model by using it as the test data. Finally weights are assigned to both these models. For the test set the weights assigned are w_1=1.262 and w_2=0.273.

Experiments performed on the combined new model: The two step clustering algorithm is initially applied to the first training set. The clustering series are partitioned into 12 clusters which as homogenous in nature and the silhouette coefficient is considered 0.4. The ratio of the cluster sizes are kept small so that it is acceptable. Based on the feature importance scores the features are selected. For the cluster 3 XGBoost model the ME achieved was 0.389 and the value of MAE is 0.897.

Since XGBoost is a supervised learning algorithm so using appropriate input and output variables can optimize the algorithm. Parameter optimization does not have a major effect on these algorithms so only the main parameters like min child weight and the max depth are set in order to reach a balanced point. If the max depth of a model is more then it will lead to overfitting, hence we consider max depth to be in the range of 6-10 and the min child weight to be from 1-6. From Table 2 it has been observed that the ME and MAE are minimum when the depth is 9 and the child weight is 2. In this case the model is optimal. For the Arima XGBoost model we test the stationary and white noise of the training set 2. The p value of the ADF test and the Box pierce test are 0.01 and 3.3×10^{-16} which is less than 0.05 so we can use the Arima model for training set 2.With the help of the auto.arima() function the different combinations of parameters are determined using ACF and PACF plots. The plot of ACF has high trailing characteristics and PACF has a oscillating and decreasing tendency, hence we consider the first order derivative .The possible optimal values from the training set 2 gives the following values as shown in Table 3. As a result the optimal model is ARIMA (0,1,1) because its AIC has the best performance. To further determine the optimal model.

Table 1: Model Performance indexes.

Evaluation Index	Expression	Description
ME	ME= $\frac{1}{n}\sum_{k=1}^{n}(\hat{Y}_c\,(K) - y(K))$	Mean sum error
MSE	MSE= $\frac{1}{n}\sum_{k=1}^{n}[\hat{Y}_c\,(K){-}y(K)]^2$	Mean squared error
RMSE	RMSE= $\sqrt{\frac{1}{n}\sum_{k=1}^{n}[\hat{Y}_c\,(K){-}y(K)]^2}$	Root mean squared error
MAE	MAE= $\frac{1}{n}\sum_{k=1}^{n}[\hat{Y}_c\,(K) - y(K)]$	Mean absolute error

Table 2: Changes of ME and MAE based on depth and weight.

Depth and min child weight	Training set 1		Test Set	
	ME	MAE	ME	MAE
(8,3)	0.398	0.710	4.45	4.6
(9,2)	0.351	0.634	4.375	4.4
(10,2)	0.332	0.590	1.778	2.3

Table 3: AIC values of Arima using auto.arima () function.

ARIMA(p,d,q)	AIC		ARIMA(p,d,q)	AIC
Arima(2,1,2) with drift	2859.317		Arima(0,1,2) with drift	2852.301
Arima(0,1,0) with drift	2978.213		Arima(1,1,2) with drift	2852.183
Arima(1,1,0) with drift	2927.285		Arima(0,1,1)	2850.145
Arima(0,1,1) with drift	2851.392		Arima(0,1,1)	2851.545
Arima(0,1,0)	2984.210		Arima(0,1,1)	2851.328
Arima(1,1,1) with drift	2853.102		Arima(0,1,1)	2851.118

Table 4: Performance measure of A-XGBoost.

A-XGBoost	Validation Set	Test Set
Minimum Error	-0.002	-8.148
Maximum Error	0.002	23.476
Mean Error	0.00	1.213
Mean Absolute Error	0.001	4.487
Standard deviation	0.001	6.213
Linear Correlation	1	-0.147
Occurrences	70	32

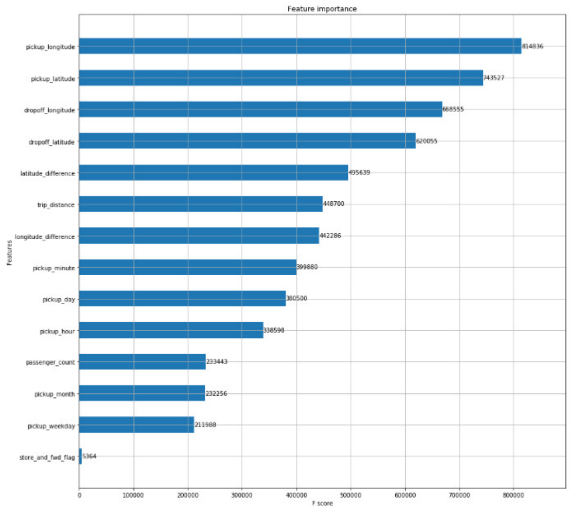

Fig. 2. Feature Importance.

Table 5: Performance evaluation of the various models.

Performance Evaluation metrics	ARIMA	XGBOOST	ARIMA XGBOOST	CLUSTERING XGBOOST	PROPOSED MODEL
ME	-20.213	-3.219	1.254	3.564	0.298
MSE	356.543	36.45	39.54	23.62	10.87
RMSE	21.941	6.56	6.873	4.784	3.54
MAE	21.245	5.98	4.234	3.821	2.57

We consider the RMSE values along with the AIC values. The performance evaluation of the Arima XGBoost model is shown in Table 4. The prediction between the Arima forecasts and the actual values are considered as the Arima residuals. In this way we calculate the predicted residuals using the trained set. The final prediction result is calculated by summing up the corresponding values of the Arima XGBoost residuals.By considering the minimal MSE we calculate the optimal model.

Models for Comparison: The models selected for comparative study are Arima model,XGBoost model, clustering XGBoost model, Arima XGBoost model and the new combined model.

According to Table 5 it has been observed that the proposed combined new model is better as compared to the other models because it has the minimum evaluation indexes.Although Arima XGBoost is inferior compared to the clustering model but it is superior compared to the XGBoost as well as the Arima model.

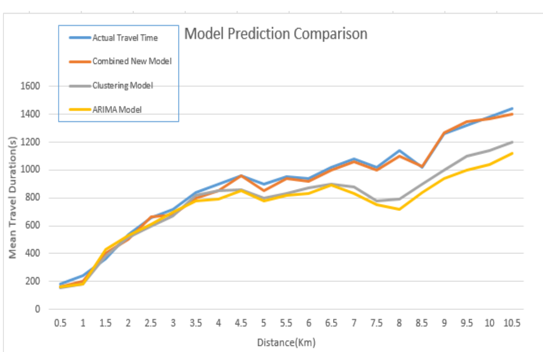

Fig. 3. Travel Time Prediction variation with distances shows that the Combined Model outperforms the other models for both long and short distances.

When we consider the travel time prediction it was observed that both Clustering and ARIMA model captures the variation for short distances but the combined new model is able to predict for both short and long distances as observed in Fig. 3 and 4 shows that the combined model is capable of reducing the travel time prediction error as compared to the other models.

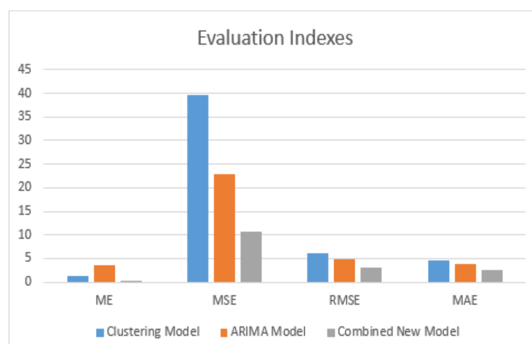

Fig. 4. Model Evaluation Indexes.

VIII. CONCLUSION AND FUTURE SCOPE

The proposed new model takes the advantages of both the Clustering and the ARIMA model and overcomes the disadvantages of both these models by using XGBoost along with it. As we have already seen that the Arima model can only handle the linear part so we applying XGBoost in order to deal with the nonlinear part of the data. We observe that the Root mean square error of the clustering model is 4.784 and for the ARIMA model it is 6.873 whereas it more accurate for the combined model and it is 3.54. The trained ARIMA model is used to predict the linear part of the data series and the nonlinear part is handled by XGBoost.

The combined new model is calculated by assigning proper weights to the forecasting results of the two models in order to get the optimized results. Common evaluation index like ME, MSE, RMSE and MAE is used to judge the performance of these three models and it has been demonstrated that the combined model outperforms the other models. In Future more changes can be made to the model in order to improve the performance metrics.

REFERENCES

[1]. Schäfer, R. P., Thiessenhusen, K. U., & Wagner, P. (2002). A traffic information system by means of real-time floating-car data. In ITS world congress.

[2]. Herrera, J. C., & Bayen, A. M. (2007). Traffic flow reconstruction using mobile sensors and loop detector data.

[3]. Mori, U., Mendiburu, A., Álvarez, M., & Lozano, J. A. (2015). A review of travel time estimation and forecasting for advanced traveller information systems. *Transportmetrica A: Transport Science, 11*(2), 119-157.

[4]. J. Zhang, S. He, W. Wang, & F. Zhan, (2015). Accuracy Analysis of Freeway Traffic Speed Estimation Based on the Integration of Cellular Probe System and Loop Detectors. *J. Intell. Transp. Syst. Technol. Planning, Oper. 19,* 411-426.

[5]. Lee, W. H., Tseng, S. S., & Tsai, S. H. (2009). A knowledge based real-time travel time prediction system for urban network. *Expert systems with Applications, 36*(3), 4239-4247.

[6]. Van Der Voort, M., Dougherty, M., & Watson, S. (1996). Combining Kohonen maps with ARIMA time series models to forecast traffic flow. *Transportation Research Part C: Emerging Technologies, 4*(5), 307-318.

[7]. Williams, B. M., & Hoel, L. A. (2003). Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: Theoretical basis and empirical results. *Journal of transportation engineering, 129*(6), 664-672.

[8]. Dong, C., Shao, C., Richards, S. H., & Han, L. D. (2014). Flow rate and time mean speed predictions for the urban freeway network using state space models. *Transportation Research Part C: Emerging Technologies, 43*, 20-32.

[9]. Vlahogianni, E. I., Karlaftis, M. G., & Golias, J. C. (2014). Short-term traffic forecasting: Where we are and where we're going. *Transportation Research Part C: Emerging Technologies*, *43*, 3-19.

[10]. Bezuglov, A., & Comert, G. (2016). Short-term freeway traffic parameter prediction: Application of grey system theory models. *Expert Systems with Applications, 62*, 284-292.

[11]. Schimbinschi, F., Moreira-Matias, L., Nguyen, V. X., & Bailey, J. (2017). Topology-regularized universal vector autoregression for traffic forecasting in large urban areas. *Expert Systems with Applications, 82,* 301-316.

[12]. Cheng, A., Jiang, X., Li, Y., Zhang, C., & Zhu, H. (2017). Multiple sources and multiple measures based traffic flow prediction using the chaos theory and support vector regression method. Physica A: Statistical Mechanics and its Applications, *466*, 422-434.

[13]. Zhang, Y., & Haghani, A. (2015). A gradient boosting method to improve travel time prediction. *Transportation Research Part C: Emerging Technologies, 58,* 308-324.

[14]. Karlaftis, M. G., & Vlahogianni, E. I. (2011). Statistical methods versus neural networks in transportation research: Differences, similarities and some insights. *Transportation Research Part C: Emerging Technologies, 19*(3), 387-399.

[15]. Lippi, M., Bertini, M., & Frasconi, P. (2013). Short-term traffic flow forecasting: An experimental comparison of time-series analysis and supervised learning. *IEEE Transactions on Intelligent Transportation Systems, 14*(2), 871-882.

[16]. Deb, B., Khan, S. R., Hasan, K. T., Khan, A. H., & Alam, M. A. (2019). Travel time prediction using machine learning and weather impact on traffic conditions. In *2019 IEEE 5th International Conference for Convergence in Technology (I2CT)* (pp. 1-8). *IEEE.*

[17]. Huang, H., Pouls, M., Meyer, A., & Pauly, M. (2020). Travel Time Prediction using Tree-Based Ensembles. arXiv preprint arXiv:2005.13818

[18]. Chen, C. M., Liang, C. C., & Chu, C. P. (2020). Long-term travel time prediction using gradient boosting. *Journal of Intelligent Transportation Systems, 24*(2), 109-124.

[19]. Braspenning, P. J., Thuijsman, F., & Weijters, A. J. M. M. (1995). Artificial neural networks: an introduction to ANN theory and practice *931,* Springer Science & Business Media.

[20]. Oza, N. C. (2005). Online bagging and boosting. In 2005 IEEE international conference on systems, man and cybernetics, *3*, 2340-2345.

[21]. Everingham, Y., Sexton, J., Skocaj, D., & Inman-Bamber, G. (2016). Accurate prediction of sugarcane yield using a random forest algorithm. *Agronomy for sustainable development, 36*(2), 1-9.

[22]. Zhang, Y., & Haghani, A. (2015). A gradient boosting method to improve travel time prediction. *Transportation Research Part C: Emerging Technologies, 58*, 308-324.

[23]. Liu, Y., An , A., & Huang, X. (2006). Boosting prediction accuracy on imbalanced datasets with SVM ensembles. In Pacific-Asia Conference on Knowledge Discovery and Data Mining, 107-118. Springer, Berlin, Heidelberg.

[24]. Chen, T., & Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In Proceedings of the *22nd ACM SIGKDD international conference on knowledge discovery and data mining,* 785-794.

[25]. [Online] URL:https://www.kaggle.com/c/nyc-taxi-trip-duration.

[26]. Alghamdi, T., Elgazzar, K., Bayoumi, M., Sharaf, T., & Shah, S. (2019). Forecasting traffic congestion using ARIMA modeling. In *2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC)* (pp. 1227-1232).

[27]. Feng, B., Xu, J., Lin, Y., & Li, P. (2020). A Period-Specific Combined Traffic Flow Prediction Based on Travel Speed Clustering. *IEEE Access*, *8*, 85880-85889.

How to cite this article: Chattoraj, M. and Rani, V. U. (2020). A Travel Time Prediction by Applying Ensemble Machine Learning Techniques. *International Journal on Emerging Technologies*, *11*(3): 1128–1133.