Detect Travel Time Using ML Techniques

submitted By : Pradip Kumar Lenka,Asst. Prof. In Raajdhani Engineering College.
Satyabrata Das, Asst. Prof In Aryan Institute Of Engineering And Technology, Bhubaneswar
Srimanta Mohapatra, Asst. Prof In NM Institute of Engineering and Technology, Bhubaneswar
Anita Subudhi, Asst. Prof In Capital Engineering College, Bhubaneswar.

Abstract

Travel time assumes a significant part in the smart vehicle framework in metropolitan urban communities. Foreseeing precise Taxi trip venture out time assists suburbanites with arranging their excursion better and arrive at the objective on schedule. The greater part of the current tech-niques utilize managed learning models to appraise the movement time. Execution got from the regulated learning models is deficient. In this paper, we propose an original methodology that targets foreseeing travel time by utilizing both su-pervised and unaided strategies with an enormous notable dataset, and this clever strategy is contrasted and directed procedures. The grouping approach of un-administered learning alongside managed assists with improving the exhibition of a prescient model. Bunching helps in sectioning the close by area information into a comparable gathering which helps in tracking down the fundamental example inside the huge dataset. Then, at that point, a directed calculation is applied to this grouped information. AI (ML) strategies like Random Forest Regressor (RFR), XGBoost Regressor (XGBR), which are managed and RFR with k-implies, XGBR with k-implies which consolidates both directed and unsuper-vised methods are utilized to foresee the outing season of the taxi trips. The outcomes show that a mix of administered and unaided models perform better compared to just directed models. Additionally, the correlation shows that the RFR and RFR with k-implies perform better compared to XGBR and XGBR with k-implies separately. RFR with k-implies outper-structures different models with a precision of 84.6%. With better execution, RFR with k-implies likewise decreases the mistake pace of the model altogether.

1 Introduction

Accurate travel time estimation is crucial for the development of Intelligent Trans- portation Systems and has become the heart of functioning in location-based ser- vices. Advanced Travellers Information Systems (ATIS) and Intelligent Transport Systems (ITS) are few of the domains where travel time prediction is a key factor. Travel time prediction is critical in transportation planning as well as to en-route information in ATIS. In addition, estimation of travel time for any path is impor- tant from the perspective of route planning, ride sharing, traffic dispatching, and travel costs. Accurate travel time estimations help users to plan their trips better, avoid congested roads with the help of suggestions on optimal routes, and reach their destination in the shortest possible time. The two influential properties of the route guidance system are arriving on time and overall travel time [1]. This helps in reducing driver's frustration as he reaches well before the deadline and hence prevents accidents. It also reduces fuel consumption and air pollution due to reduced travel time.

Taxi Trip travel time prediction is very important in the intelligent transport system, developing mobility-on-demand systems and traveller information systems. Accurate travel time estimations of Public Transports like buses, taxis could also help users in choosing the desired departure time, and estimate the expected time of arrival based on transit delay and trip duration. This, in turn, helps alleviate traffic congestion and significantly slims down the growing traffic and fuel con- sumption. Travel time calculation is extremely sensitive to vehicle speed, weather, dynamic traffic situations as well as spatial and temporal properties. For example, disparities in peak hour traffic versus non-peak hours or winter versus summer traffic can be observed. It is difficult to estimate the traffic condition at a cer- tain segment at a later point of time / the future. However, the accuracy of the estimation plays an important role in user satisfaction.

Travel time is predicted indirectly from the available traffic data in the point measurement approach.To predict the travel time huge amount of historical/real time, travel/traffic data is being collected on a regular basis. Origin-Destina- tion Expected Time of Arrival(OD-ETA) is a MUlti-task Representation learning model for Arrival Time estimation (MURAT)[2]. The model learns better inter- pretations of the available input features(origin, destination and departure time) from the representation learning.

A variety of techniques can be opted to collect the travel data. Author Ravish R *et al .,*[3] implements sensor based solution. Placing the sensor on the vehi- cle, real time travel data can be collected. Also collected data is

uploaded in the cloud, which can be used as a repository. The innovations in mobile technology made availability of large amounts of trajectory data. Travel time can be measured accurately using these trajectories. To handle this massive amount of data author Lee *et al* [4] implements the rule-based classification on MapReduce technique to accurately measure travel time for a large scale spatial data.Tracking the object in motion and collecting the data helps for route guidance systems. Many existing services are query based, where the search result will be based on the traveller's experience. However this technique is not useful because of lack of data. To rec- ommend a route, data can be collected through social networks such as twitter or facebook. This technique improves precision and recall rate as it uses real-world data [5]. Location details are collected through user smartphones (GPS interface).

In this paper we have collected data from NewYorkCity(NYC), taxi trip records. After the collection of data, it is important to extract significant features from the dataset and one of the most used techniques is data clustering which helps to extract the hidden characteristics from the data.

Author Chen *et al* [6] implements adaptive based clustering which finds the similarities between the data and the cluster centres and groups the data into several clusters which helps to yield accurate time prediction. The data clustering technique is also used to reduce the size of the large dataset which helps in better execution time. Author Tomislav et al [7] used the k-means algorithm to cluster 6000000 computed speed profiles whose average storage space is 0.5GB into several clusters. In this paper, we have used the k-means algorithm to cluster the data to obtain better travel time prediction.

The current approach just uses limited modality data or single models with- out considering their one-sidedness. Author Zhiqiang *et al* [8] puts forward an optimized method based on ensemble method with multi-modality urban big data, namely Travel Time Estimation-Ensemble (TTE-Ensemble). The feature sub-vectors from the multi-modality data is given as model input and then the gra- dient boosting decision tree (GBDT) model is used to process the low dimensional simple simple features and the Deep Neural Network (DNN) model is adopted to handle high dimensional underlying features.

Many of the existing implementations use different ML algorithms (supervised learning) to predict the travel time(discussed in related work). In this paper the multiple techniques such as Random Forest Regressor (RFR), XGBoost Regressor (XGBR), RFR with k-means, and XGBR with k-means are applied to estimate the travel time. The performance of all four techniques are measured. Among the four techniques the later two are novel approaches which take the advantage of a combination of supervised and unsupervised ML techniques. The performance of the combined technique is improved compared to supervised technique. The further section of this paper is organized as follows, section 2 discusses the related work on travel time prediction. Section 3 illustrates implementation along with four different algorithms demonstrated. Results and Discussion is highlighted in section 4 and finally section 5 concludes the paper.

2 Related Work

Author Wu *et al.,* suggests Support Vector Regression (SVR) technique to pre- dict travel time [9] using traffic data provided by Intelligent Transportation Web Service Project (ITWS). Since SVR has the advantages of having a higher gener- alization ability and guarantees global minima on any given training data set.Here SVR is compared with other baseline models. Some common baseline methods such as *Current Travel-Time Prediction* Method and *Historical Mean Prediction* Method are used to compare SVR prediction methods. In Current travel time prediction method, the computation is done by taking the data available at the instant, whereas historical mean prediction method, the travel time is obtained from the average travel time of the historical data at the same time of the day and day of the week. Here, all the three predictors predict well for a long-distance of approximately 350km because as the travelling distance increases, the number of free sections increases due to which travel time of long-distance is dominated.

However, the results show that for short distance current travel time predictor is slow to adopt traffic dynamicity and historical mean predictor per- forms badly if traffic pattern does not exist in history. The SVR outperforms compared to both current travel time and historical mean prediction methods. Also, the SVR reduces Root Mean Square Error (RMSE) and Relative Mean Errors (RME) to less than half.

Author Chen *et al.* [10] suggests a Gradient Boosting technique combined with Fourier filtering process to predict travel time using Electronic Toll Collection (ETC) data of Taiwan Freeway No. 1. Gradient Boosting (GB) is an excellent tool for short-term travel time prediction problems but this paper shows that by modifying the gradient boosting process it can also be used for long-term travel time prediction. The GB method is used to generate base models from the training data to strategically find the optimal combination of trees. The GB method has two steps, first generate weak models and the second step is to assemble the weak models into a strong model. The prediction of GB is the result of the loss function of the data points with N numbers and the Sum of Squared Errors (SSE) is used as a measurement for loss function. The N dimension loss function forms the gradient function which leads the prediction into the right direction. The high frequency noise present in the data affects the accuracy of prediction and is reduced by the Fourier transform as shown in equation 1.

$$\Sigma \infty \quad a_0 + \quad (a_n \cos nx + b_n \sin nx) \quad (1)$$

where

$$a_0 = \frac{1}{2\Pi} \int_{-\pi}^{\pi} f(x)\,dx \quad (2)$$

$$a_n = \frac{1}{\Pi} \int_{-\pi}^{\pi} f(x)\cos nx\,dx \quad n = 1, 2, 3, \ldots k \quad (3)$$

$$b_n = \frac{1}{\Pi} \int_{-\pi}^{\pi} f(x)\sin nx\,dx \quad n = 1, 2, 3, \ldots k \quad (4)$$

The Fourier series is used with GB to eliminate the high frequency noise which results in better accuracy. The overall performance is measured using MAPE, MAE and RMSE, however RMSE explains the error better than MAPE and MAE. The values of RMSE, MAPE and MAE of prediction period are smaller which inferences that by incorporating the Fourier series with GB weakens the noise effect and improves the prediction accuracy.

Travel time prediction for urban traffic is tough and depends on the time due to various travel demands, interruptions caused by traffic control devices, and weather conditions. However to predict the travel time for urban traffic we need to analyze both real-time and historic data. Author Chein [11] developed a model that estimates travel time efficiently using both real-time and historic data. The data from the Transportation Operations Coordinating Committee is used that consists of data collected from the roadside terminals (RST) installed on the New York State Thruway. The path-based travel time and link- based travel time are the two types of input data considered by the author for the proposed prediction model. The path-based and link-based travel time are the two types of input data considered by the author in the proposed prediction model. The path-based travel time depends on the difference between the entry and exit paths of the vehicles recorded. Whereas, link-based travel time is the summation of the vehicle's travel time in the consecutive individual links that constitute the entire path.The Kalman filtering algorithm is used in the prediction of travel time considering both live and historic data. The proposed model is given by the equation 5:

$$y(t) = \varphi(t-1)\, y(t-1) + w(t-1) \quad (5)$$

Where y(t) denote the travel time at time interval t that is to be predicted, (t) denote the transition parameter at time interval t which is externally determined and w(t) denote a noise term. As there is no traffic parameter other than travel time is involved, the observation equation associated with the state variable y(t) is given by equation 6:

$$u(t) = y(t) + s(t) \quad (6)$$

Where u(t) denotes the observation of travel time on time interval t and s(t) denotes the measurement error at time interval t. The Kalman filtering algorithm updates the state variable as new observation during prediction, hence it is used for travel time prediction. The models need a historical seed to adapt to the traffic conditions that are going to be present at the time of prediction and the historic data collected for the path which is being predicted by the model is contemplated as a historic seed. The performance of the proposed model is measured using mean absolute relative error (MARE) and root relative square error (RRSE) and and their error values are very less indicating it is a robust dynamic travel time prediction model.

Precise travel time prediction is very important for riders using applications like cab aggregator. In one of the paper [12], the author implements static travel time prediction of taxi trip using XGBoost algorithm. The dataset used is the taxi trip trajectories by New York City Taxi and Limousine Commission under Freedom of Information Law (FOIL). The dataset contains millions of record having passenger pick-up geo-coordinate, passenger drop off geo-coordinate, timestamps for pick-up and drop-off, medallion id, driver id, trip time(in seconds), trip distance(in miles), a fair amount, tax amount and many other attributes related with taxi services. As 70% of the dataset considered as in-liers trips, after data preprocessing and filtering of extreme-conditioned trips, XGBoost algorithm is used to predict the travel time. On-time series XGBoost regression predictor model outperforms even with outliers.The technique is compared with the Neural network, Support Vector Machine, online map service and simple baseline model. XGBoost outperforms the other prediction models with reasonable accuracy handling minimum varia- tions. However, the XGBoost model is applied only to predict static travel time prediction, not for dynamic travel time prediction and the factors influencing the travel time of the taxi were not considered. In the paper, author Chen *et al .,*[13] proposes a novel approach to search the route based on location using split and combine method. The proposed algorithm divide the route into sub route and join them to suggest new routes. In order to handle massive amount of data, the task is divided into sub-tasks which are handled independently and parallel.

A traffic flow prediction model built on field data collected by loop detectors at signalized inter- sections is integrated with time-based model,which are used to signal optimization, route choice, traffic monitoring, etc is presented by author

Bing Feng et al [14]. K-means clustering is performed on travel speed to search for each clustering center. Based on the decomposed three periods - peak, flat- peak and low-peak period, the period-specific Combined Predictive Model (CPM) for 24-hour traffic flow is developed. A combined prediction model based on time partition is proposed for 24-hour traffic flow forecasting, which adopts the grey theory model for flat-peak and low-peak periods and back-propagation artificial neural network for peak hours, respectively.

The road traffic flow varies depending on the location, travel day and time. This majorly affects the estimation of ac- curate travel time. Author Nath *et al.,* [15] implements travel time prediction using the Modified K-means Clustering ap- proach (MKC) which takes care of these variations. The author also compares the results with the Successive Moving Average (SMA), Chain Average (CA) and Na¨ıve Bayesian Classification (NBC) method. The proposed MKC method pro- vides the better option with high accuracy and takes ad- vantage over SBC, CA, SMA. The method considers uncertain situations and it works on historical data. Unlike k-means clusters, positioning of the centroid for each cluster is veri- fied and chosen in such a way that inter cluster centroid difference is maintained very high. The user gives start time,

source and destination as an input. Then the route is divided into different segments.Unlike k-means clusters, positioning of the centroid for each cluster is verified and chosen in such a way that inter clus- ter centroid difference is maintained very high. The user gives start time, source and destination as an input. Then the route is divided into different segments. Applying successive repetition approximate on these segments, travel time can be measured from source to destination. The algorithm first finds the frequency of each travel time from different records. Then defines prediction relation from the three attributes such as frequency(f), travel time(y), velocity(z). Finding the greatest value of frequency fmax (xp), a tuple 'p' having p(xp,yp,zp) is chosen as a centroid of cluster1. Compare each tuple Ta(xa,ya,za) with the P(xp,yp,zp) using the formula. Then finds the cost using the equation 7,

$$Cost(P, Ta) = |xp - xa| - |yp - ya| - |zp - za| \quad (7)$$

Tuple having maximum cost is chosen as a centroid of cluster2. Once the two clusters with the centroid are built, define cluster membership based on the nearest centroid. Re-estimate the cluster centre until no change in clusters. Obtain the travel time from each cluster using the below equation 8,

$$\tau_i = \sum_{i=1}^{N} (f_i * t_i) / \sum_{i=1}^{N} f_i \quad (8)$$

Where τ_i is the travel time obtained from an i-th cluster, N is the total number of a tuple in an associated cluster, fi is the frequency of the i-th tuple, and ti is the Travel time of the i-th tuple. Then find the average of the two, to predict the travel time.However, MKC method requires a large amount of historical data to improve the accuracy of prediction. The method did not take certain event/seasonal pat- terns, data associated with uncertainty and distance versus prediction accuracy.

When the user needs to switch between multiple buses while travelling from one location to another, the proposed method can be used to calculate the bus

travel time [16]. Bus travel time involves waiting time at multiple places and jour- ney time. Journey time of each line/segment of multiple lines is predicted using Long Short Term Memory Model (LSTM). The Partitioning and Combination Framework (PCF) takes care of non-uniform waiting time and bus travelling with different travel speed and frequency. The study also includes influencing factors such as traffic signals, traffic condition, travel distance, number of stop points etc. The method adopts the data-driven approach. To predict journey travel time, bus trajectory (collection of bus stops and arrival time of the bus at the stops) are ex- tracted from historical data. LSTM technique is applied to predict the bus travel time using features extracted from these trajectories and other related data in- cluding road characteristics. Hence the journey time is calculated using the LSTM approach and halting time at bus stops are calculated using Interval Based Histor- ical Approach (IHA). To obtain final travel time PCF-sum can be used to calculate the direct sum of all the components or use Linear Regression (PCF-LR) to com- bine all components to form the final result. The proposed implementation uses both PCF-sum and PCF-LR to evaluate the performance. However, the proposed method requires large data as the model works on historical data. Also prediction model needs to be retrained for the new set of data frequently. In real-time scenario uncertainties such as weather condition, events, festivals etc. are not considered which majorly influences travel time.

Most of the travel time predictions are based on dividing the road network into the segment and summing up individual segment travel time[16]. However, this method gives inaccurate result because of accumulated error. The author Wang *et al.,* [17] proposes a travel time estimation taking end to end part directly. The proposed method uses convolution operation combining geographic information into it, extracting spatial correlation. Author names this end to end (collective) Travel Time Estimation (TTE) framework as DeepTTE. The technique extracts spatial and temporal characteristics and their dependencies from raw GPS data. The technique works on historical data which contains latitude, longitude, times- tamp, start time, day and climatic condition. DeepTTE contains three elements such as *attribute*, *spatial-temporal learning* and *Multitask learning*. Attributes of historical data are categorical values which cannot be fed to the neural network. Using the embedding technique, these categorical attributes are converted into vector attribute. Later embedded vector is combined with travel distance and fed to the second stage, this is named as *attr*. The second element of *spatial-temporal* components is divided into two parts. In the first part, geo convolution network finds the correlation between consecutive GPS location from raw GPS data. In the second part, the recurrent neural network finds the temporal correlation from the first part. Finally, *multi-task learning* components can be used to predict travel time. This is done by both segments of the road and between end to end paths. During the training stage, multi-task learning includes both segment and end to end path. However during the testing stage only end to end path of estimating travel time is considered. The proposed model performs well with high accuracy. In the ITS domain, estimated time of arrival (ETA) is one of the essential services. The author Wang *et al.,* [18] considers that the estimation of travel time is a pure spatial-temporal regression problem, which uses machine learning tech- nique (using floating car data) to solve the problem. The proposed method uses Wide-Deep-Recurrent (WDR) learning model for prediction along the travel path given the start time. The model takes the advantage of linear models, deep neural

network and recurrent neural network and trains jointly. The dataset contains dif- ferent trip paths, departure time and arrival time. The user gives the input which includes origin, destination and departure time as a query to find the actual travel time. The features extracted for estimating travel time are Road segment (trip path) and intersections, time of travel, traffic information, vehicle and driver pro- files, and weather information. Global statistical information is extracted through wide deep recurring learning, in which the wide model converts features into a high dimensional feature, whereas deep model projects sparse input into dense features. However, in order to extract, local information LSTM technique is used. The proposed solution is deployed on DiDi platform(services customer requests and provides solution). Also, it is observed that the technique is more powerful than existing deep learning models and minimises the mean absolute error per- centage. However, the model needs to be adopted for location-based problems.

Combinatorial Optimization problems related to transportation and mobility are trending and challenging for sustainable development of a city. Environmental friendly solutions are the requirement of this era. In this paper [1] the author aims to optimally solve the vehicle routing problem taking two factors into consider- ation: arriving on time and total travel time. A semi-decentralized multi-agent based approach for vehicle route guidance is been formulated. This approach con- sists of vehicle and infrastructure agents. Vehicle agents are drivers whose sole responsibility is to follow the route guidance specified by infrastructure agents. The infrastructure agents collect intentions (i.e., deadlines and destinations) from vehicle agents and render guidance to them by solving route assignment problem. They are associated with all traffic lights at road intersections. This approach integrates both the factors by representing them as two objective functions. For arriving on time, the best-suited one is the probability tail model which maximizes the probability of arriving at destination before the deadline. Total travel time is formulated as Mixed-integer Quadratic Programming (MIQP) problem that has a weighted quadratic term which will minimize expected travel time using route assignment.

Route guidance system suggests the best route to reach the destination given start time working on both historical and/or real-time data. The author Asghari *et al.,* [19] states that best route is not always a reliable one. Reliability is based on how the prediction model works during an emergency. Such as travelling with a deadline (arrival on time to important work). This requires analysis of probability distribution of the journey time over each link along the path between source and destination. Many techniques have been proposed to compute Probability density function(Pdf) to know the route travel time. However, all these techniques as- sume that probabilistic link travel times(pltt) is known in advance. The author suggests techniques by exploring different algorithms to calculate pltt, in a road network. Different algorithms are addressed using characteristics such as repre- sentation, time dependency and correlation. Pltt computation is important as it directly reflects the accuracy of pdf of the entire road network. Pltt can be dis- crete or continuous and works on both historical and real-time data (current data). Three different techniques are used to estimate probabilistic travel times and the correlation between them. Suppose ts is the start time of travel and tq is the time
of the query, to obtain pdf between link i to j, if ts≥tq, simple and first approach would use available historical data. However if the model works on historical data
irrespective of tq, prediction model gives the same output. Hence better choice and

the second approach is to use current traffic condition along with historical data when ts=tq. This is performed through linear interpolation between current traffic condition and summarized historical link travel time. In the third approach, based on predicted time (predicted on current traffic), further analysis is done taking similar historical value . However it requires further investigation carrying similar study in different parts of the globe. The reason is, a different set of parameter and their values vary according to location.

3 Implementation

The Four techniques implemented are presented in this section.

3.1 Background

An ensemble learning model is a part of machine learning that constructs a set of classifiers and then classifies the data points by voting or averaging for prediction [20]. For solving travel time prediction problems, the tree-based ensemble methods act as good candidates since they provide results that can be easily interpreted, handle a variety of predictor variables, and fit complex non-linear relationships [21].Boosting is a subset of ensemble learning, is an iterative method that com- bines models of the same type to improve the performance. Gradient Boosting is one the popular algorithm in machine learning that greatly explores the complex relationship between the variables [22]. It also produces accurate results and gives a chance to interpret the influence of different variables and nonlinear relationships between variables and predicted results. eXtreme Gradient Boosting(XGBoost) al- gorithm is an improvement of Gradient Boosting [12]. It is a supervised learning that uses a decision tree to improve the performance of the model and it takes multiple attributes to train the model and predict the target value.

Bagging is also a type of ensemble learning similar to boosting that combines similar models to improve the performance [23]. However, in boosting the models are built iteratively and the performances of previously built models influence the new model whereas in bagging the models are built separately and independent of each other. Bagging as- signs equal weights to the entire instance and uses vote or average for predicting the output. Random forest is an ensemble learning method that combines both bagging and random subspace [24]. It is a supervised algorithm that constructs independent multiple decision trees randomly using the same dataset and the output is obtained by applying the bagging method. k-means, an unsupervised learning algorithm that uses an iterative approach to partition the dataset into predefined k clusters where each observation can belong to only one of the k clusters. It assigns each data observation to a cluster if the sum of the squared distance between the data observation and the cluster’s centroid is minimal.

3.2 Dataset

The data set for the travel time prediction is taken from NYC Limousine Open data which includes trip records of all the trips completed by green taxis in NYC in 2015 for the month of January [25]. It consists of 10,48,575 records. For a given trip, the data set contains details such as passenger pick-up and drop-off latitudes and longitudes, pick-up and drop-off time-stamps, trip distance (in miles), fare amount, and passenger count. Table. 1 shows the sample records with the related attributes.

Pickup_ datetime	Dropoff_ datetime	Pickup_lon gitude	Pickup_latit ude	Dropoff_lon gitude	Dropoff_la titude	Trip_ distan ce	Passe nger_ count
2015-1-1 00:34:00	2015-1-1 00:38:00	-73.9225921	40.75452805	-73.91363525	40.765522	0.88	1
2015-1-1 00:34:00	2015-1-1 00:47:00	-73.9527511	40.67771149	-73.98152924	40.6589775	3.08	1
2015-1-1 00:34:00	2015-1-1 00:38:00	-73.8430099	40.71905518	-73.84658051	40.7115669	0.9	1

Table 1: NYC green trip sample dataset

3.3 Data Preprocessing

The data preprocessing involves selecting attributes necessary for the implemen- tation and removing outliers. The important attributes selected are:
Trip Distance(T_D): The distance(in miles) between the pick-up and drop-off points plays an important factor in travel time prediction since,

$$Distance = Speed \times Time$$

Distance is given in the dataset.
Trip Duration(T_d):Td (in seconds) is derived indirectly from the dataset. This is obtained by subtracting drop-off and pick-up timestamps.
Speed(S): Speed(in miles per hour- mph) is obtained by using the formula

$$Speed = \frac{Distance}{Time}$$

Day of the week:It corresponds to the travel day which is indirectly obtained from the pick-up and drop off timestamps. Here, 0 corresponds to Monday, 1 indicates Tuesday and subsequent number indicating the remaining days of a week respectively as shown in Table 2. It helps in analyzing the speed and duration of

the travel. The weekdays experience slow speed and long duration due to the working of business, educational institutes, Government offices etc.,

Time of the day:Time of the day corresponds to pick-up hour and drop-off hour which is obtained using pick-up and drop-off timestamps. It is observed that the peak hours experience high traffic congestion affecting travel time.

Removing Outliers:

In the implementation, each attribute of the dataset is visualized using a boxplot graph to identify the outliers. As a result,the records having speed between 6 to 140 mph, pick-up and drop off longitude values greater than 0 and pick-up latitude values between 38 and 45 coordinates are retained in the dataset.

Table. 2 shows the sample dataset with attributes T_D , T_d, S, day of the week, pick up and drop-off hours obtained after data preprocessing.

Pickup_datetime	Dropoff_datetime	Trip_distance	Day_week	Pickup_hrs	dropoff_hrs	Duration	Speed
2015-1-1 00:34:00	2015-1-1 00:38:00	0.88	3	0	0	240.0	13.0
2015-1-1 00:34:00	2015-1-1 00:47:00	3.08	3	0	0	780.0	14.0
2015-1-1 00:34:00	2015-1-1 00:38:00	0.9	3	0	0	240.0	13.0

Table 2: Pre-processed sample data set

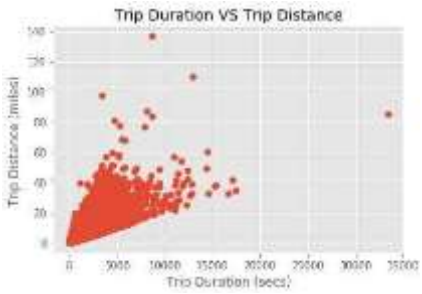

Fig. 1: Trip distance versus trip duration

The graph in Fig.1 illustrates the trip distance versus trip duration. The trip duration and trip distance are dependent on each other and when trip distance increases the duration of the trip also increases. However the graph is not linear with respect to trip distance, this is because in real time factors such as traffic, weather condition, time of the day affects the travel time. Hence it is important to

consider those details which affects travel time to accurately estimate the travel time.

3.4 Algorithms

To predict the travel time, Random Forest Regressor(RFR), XGBoost Regres- sor(XGBR), RFR with k-means and XGBR with k-means are the four different methods used. The block diagram in Fig. 2 illustrates the working of the first two methods and Fig. 3 shows the working of later two implementation methods. The historical taxi trip dataset is pre-processed and split into training and testing set. This is given as an input to the models. The training set is used to train the model and the test set is used to predict the travel time.

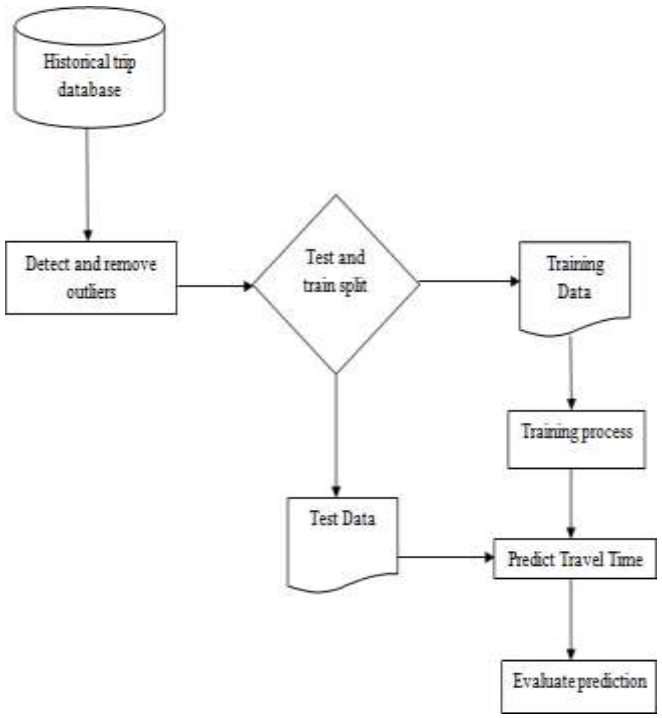

Fig. 2: Block diagram of RFR and XBGR model

In RFR with k-means and XGBR with k-means, pick-up and drop-off locations are clustered and then given as input to the models RFR and XGBR as shown in Fig. 3. All four models' performance is evaluated using standard evaluation met- rics.

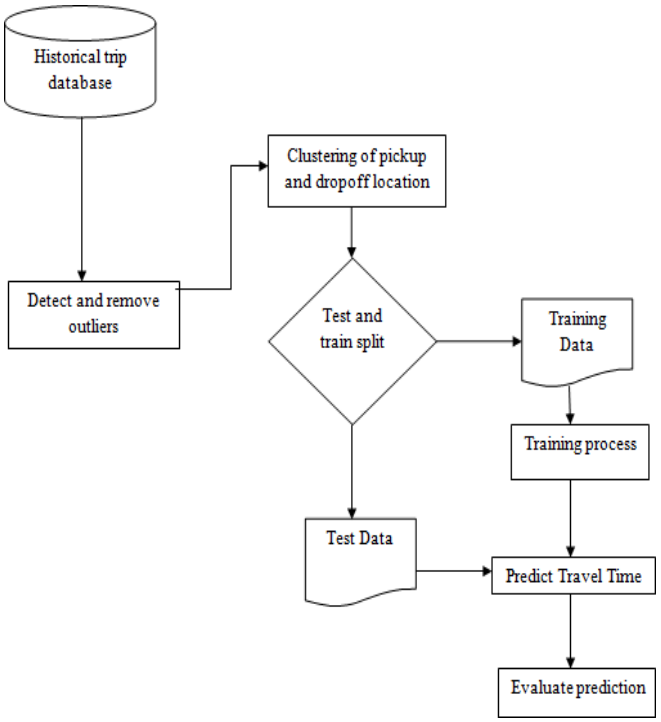

Fig. 3: Block diagram of RFR with k-means and XBGR with k-means model

3.4.1 Random Forest Regressor

Random Forest Regressor (RFR) is an ensemble model that predicts the outcome by combining the decisions from several base models where each base model is a simple decision tree. Formally, it can be written as shown in the equation 9:

$$h(x) = g_0(x) + g_1(x) + g_2(x) + ...g_n(x) \quad (9)$$ where the final model 'h(x)' is the result of the sum of base models g0,g1,g2 gn,

'n' is the number of decision trees and gn is the decision from the nth tree. In our implementation, n takes the value of 50. To predict the travel time, the dataset consisting of attributes pick-up and drop off latitude and longitude, trip distance, pick-up hours, day of the week, and duration is randomly split into 80% of the training set(X) and 20% of the test set(Y). The model is trained on X and tested on Y to evaluate the model performance

Algorithm 1 : Random Forest Regressor Input: Preprocessed dataset D
Output: Predicted travel time(T_d)
Begin
for all features do:

- Split D into X and Y
- Assign the number of decision trees (n)
- Apply RFR on X
- Fit the model for Y
- Evaluate the model output T_d using standard metrics

End For End

3.4.2 XGBoost Regressor

XGBoost Regressor(XGBR) is an ensemble learning model that uses the objective function to train the model which is a sum of training loss and a regularization term. The objective function is given by equation 10:

$$obj(\vartheta) = L(\vartheta) + \Omega(\vartheta) \quad (10)$$

where $L(\vartheta)$ is the loss function which tells how well the model can predict on the training data and $\Omega(\vartheta)$ is the regularization term which is used to control the overfitting.The preprocessed dataset consisting of attributes pick-up and drop off latitude and longitude, trip distance, pick-up hours, day of the week, and dura- tion is split randomly into 80% training set (X) and 20% test set (Y). To predict travel time, XGBR model is trained on X and tested on Y and its performance is evaluated.

Algorithm 2 : XGBoost Regressor Input: Preprocessed Dataset D

Output: Predicted time travel (T_d)
XGBoost Regressor(D):
For all the features in D do:

- Split D into X and Y.
- Apply the value of the decision tree, n.
- Apply XGBR on X.
- Fit the model for Y.
- Evaluate the model's output T_d using standard metrics.

End for End

3.4.3 Random Forest Regressor with K-Means

In this model, pick-up(P) and drop-off(D) clusters are formed separately using the k-means algorithm. Within each of P and D form k clusters, where k takes the value of 15. These 15 clusters within P are formed by using pick-up latitude- longitude and assigning each observation to the cluster whose centroid is closest defined using Euclidean distance resulting in pick-up locations nearest to each other forming a cluster. Similarly, 15 clusters within D are formed using drop-off

latitude-longitude which results in the drop off locations that are close to each other. Fig. 4 and Fig. 5 shows the P and D clusters respectively. As shown in the figure, pick-up and drop-off longitude is plotted against X-axis and pick-up and drop-off latitude is plotted against Y-axis. The P and D help to analyze the route being traveled from one P cluster to the D cluster. The P and D clusters are added into the dataset as a set of new attributes. The Euclidean distance is calculated
using the equation 11:

$$d = \sqrt{\sum_{i=1}^{n} (u_i - v_i)^2} \qquad (11)$$

where u and v are a pair of samples. The dataset consisting of attributes pick- up and drop off latitude and longitude, trip distance, pick-up hours, day of the week,duration and routes obtained from the clusters P and D are randomly split into X and Y. To predict the travel time, the model is trained on X using RFR and tested on Y to evaluate the model performance.

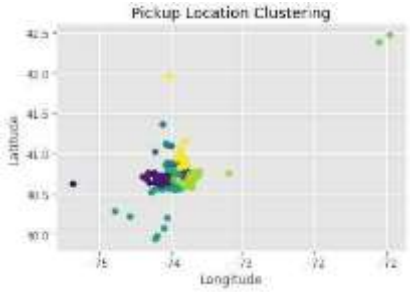

Fig. 4: pick-up location clusters

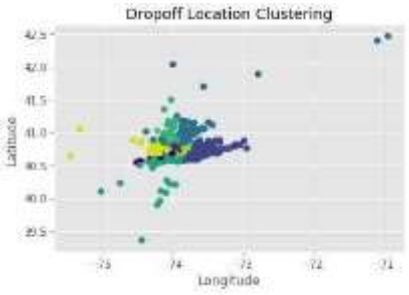

Fig. 5: drop-off location clusters

Table 3 shows two new attributes k-means pick-up and k-means drop-off which represent the pick-up and drop-off clusters formed using k-means. For instance, the 1st observation values 5 and 14 of k-means pick-up and k-means drop-off re- spectively indicates that taxi is traveling from 5^{th} P to 14^{th} D cluster.

Pickup_datetime	Dropoff_datetime	Trip_distance	Day_week	Pickup_hrs	dropoff_hrs	Duration	Speed	Kmeans_pickup	Kmeans_dropoff
2015-1-1 00:34:00	2015-1-1 00:38:00	0.88	3	0	0	240.0	13.0	5	14
2015-1-1 00:34:00	2015-1-1 00:47:00	3.08	3	0	0	780.0	14.0	10	0
2015-1-1 00:34:00	2015-1-1 00:38:00	0.9	3	0	0	240.0	13.0	2	12

Table 3: Sample Dataset containing pick-up and drop-off clusters

Algorithm 3 :RFR with k-means

In this technique RFR is applied on the result of k-means. The implementation is split into two parts. In the first part(Part 3A) we obtain the clusters within each pick-up and drop-off location. Part two(Part 3B) includes applying RFR on this clusters.

Part 3A: Clustering pick-up and drop off locations using K-means

Input: Dataset D with attributes pick-up latitude & longitude, Drop-off latitude & longitude

Output: Dataset D containing attributes P and D Begin

For all features do:

pick-up cluster (P):

- Assign the number of clusters, k
- For each of the k clusters compute the cluster centroid
- Assign each observation to the cluster whose centroid is closest to the input feature defined using Euclidean distance
- P is formed drop-offcluster(D):

- Assign the number of clusters, k
- For each of the k clusters compute the cluster centroid
- Assign each observation to the cluster whose centroid is closest to the input feature defined using Euclidean distance
- D is formed

End for End

Part 3B: Prediction of travel time(T_d) using RFR

Input:Dataset D Output: The predicted value of travel time(T_d) Begin for all features do: for all features do:

- Split D into X and Y

- Assign the number of decision trees (n)
- Apply RFR on X
- Fit the model for Y
- Evaluate the model output T_d using standard metrics

End For End

3.4.4 XGBoost Regressor with K-Means

In this model, Part 3A (forming a clusters P and D) remain same as demonstrated in Random forest regressor with k-means. To predict travel time, the XGBoost model is trained on X and tested on Y and its performance is evaluated using a standard evaluation metrics.

Algorithm 4: XGBR with k-means clustering

Part 4A: Clustering of pick-up and drop-off location using k-means clustering It is similar to Part 3A.
Part 4B: Prediction of travel time using XGBR Input: Dataset D
Output: Predicted time travel (T_d) XGBoost Regressor(D):
For all the features in D do:

- Split D into X and Y.
- Apply the value of the decision tree, n.
- Apply XGBR on X.
- Fit the model for Y.
- Evaluate the model's output T_d using standard metrics.

End for End

4 Results and Discussions

Standard performance metrics such RSE, MSE, RMSE, MAE and MAPE are con- sidered to evaluate the model performance. The results in table 4 illustrates all the above metrics of the four models. RSE score is a statistical measure and is defined as a ratio of the variance of the dependent variable explained by the independent variable.

Fig. 6 shows that the RSE score of RFR with k-means is highest with a value of 84.6%. RSE is calculated using the equation 12:

$$RSE = \frac{Explained\ variation}{Total\ variation} \quad (12)$$

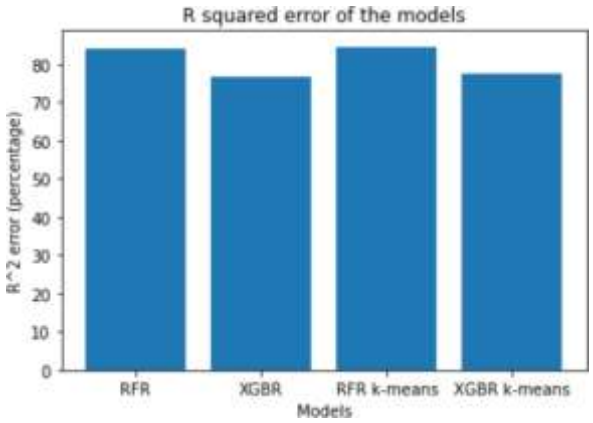

Fig. 6: RSE score of the models

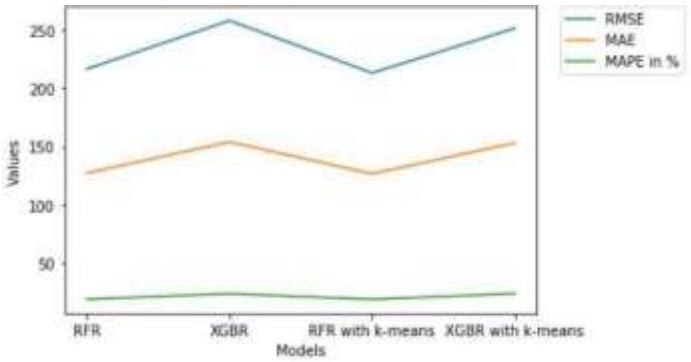

Fig. 7: Graph representation of RMSE,MAE,MAPE of four models

Fig 7 illustrates the RMSE,MAE, MAPE(in %) for all the four models. MSE which measures the average squared difference between the estimated values and the estimator. The smaller the MSE, the closer it is to find the line of best fit. RMSE is the root of MSE. MSE and RMSE values in Table 4 indicate that RFR with K-means has a smaller value and it is closer to the best fit line. MAE is used to measure the average error in the prediction. Lower the MAE value, better the model. MAPE is a measure of error that tells how accurate a forecast system is. Low MAPE represents low error. The RFR with k-means model reduces the MSE, RMSE, MAE, and MAPE compared to the other three models. The RFR with K-means not only improves the performance of the model but also reduces the error prediction of the model significantly. Hence, it is a better model compared to the other three models. Also, RFR and RFR with k-means performs better than XGBR and XGBR with k-means respectively.

Table 4: Performance comparison of four models

Metrics/Models	RFR	XGBR	RFR with k-means	XGBR with k-means
RSE(in %)	84.2	77.6	84.6	78.3
MSE	46777.46	66482.20	45356.93	63193.86
RMSE	216.28	257.84	212.97	251.38
MAE	127.13	153.92	126.62	153.05
MAPE(in %)	19.16	23.85	19.05	24.00

5 Conclusion

XGBoost and Random Forest algorithms can perform both the general classifica- tion and regression task, its capability in time-series analysis task is explored in this paper. The static travel time of taxi trips is predicted using RFR, XGBR, RFR with k-means, and XGBR with k-means, and the models are compared us- ing RSE, MSE, RMSE, MAE, MAPE evaluation metrics. Among the four models, RFR with k-means performs better than the other three models with an RSE value of 84.6% and less MAE, MSE, MAPE and RMSE. The XGBR model performs the least with an RSE value of 77.6% and has more MAE, MSE, MAPE and RMSE value. We can conclude that by integrating K-means clustering with RFR and XGBR which uses both supervised and unsupervised learning perform better than their respective supervised models i.e RFR and XGBR. Also when Random for- est regressor and XGBoost Regressor are compared, RFR and RFR with k-means perform better than XGBR and XGBR with k-means respectively. However, the performance of all four models can be improved by considering real-time scenarios like weather, uncertain situations/events, traffic, etc. In our future work, we plan to incorporate all such situations (real-time data) along with historical data to predict the travel time.

6 Compliance with Ethical Standards

Conflict of Interest: The authors declare that they have no conflict of interest.

Funding: This study was not funded.

References

1. Cao, Zhiguang and Guo, Hongliang and Zhang, Jie,"A multiagent-based approach for ve- hicle routing by considering both arriving on time and total travel time",ACM Transactions on Intelligent Systems and Technology (TIST), vol-9,no-3,pp 1-21,ACM New York, NY, USA,(2017)
2. Li, Yaguang and Fu, Kun and Wang, Zheng and Shahabi, Cyrus and Ye, Jieping and Liu, Yan,"Multi-task representation learning for travel time estimation",Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining,pp 1695-1704, (2018)
3. Ravish, R., Nadagouda, P., Hombal, K., Ramkumar, L., Nayak, P., Shah, P., Jayakumar, R., Suresh, P. and Rangaswamy, S.,September. "IoT Based Road Travel Time Detection". In 2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI) (pp. 2231-2236). IEEE,(2018)

4. Lee, HyunJo, Seungtae Hong, Hyung Jin Kim, and Jae-Woo Chang. ”A travel time pre- diction algorithm using rule-based classification on MapReduce.” In Database and Expert Systems Applications, pp. 440-452. Springer, Cham, 2015
5. Comito, Carmela. ”Travel routes recommendations via online social networks.” In Pro- ceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, pp. 1168-1173. 2019.
6. Chen, Chi-Hua, Feng-Jang Hwang, and Hsu-Yang Kung. ”Travel time prediction system based on data clustering for waste collection vehicles.” IEICE TRANSACTIONS on Infor- mation and Systems 102, no. 7 : 1374-1383, (2019).
7. Erdeli´c, Tomislav, Martina Ravli´c, and Ton˘ci Cari´c. ”Travel time prediction using speed profiles for road network of Croatia.” In International Symposium ELMAR, pp. 97-100. IEEE, 2016.
8. Zou, Zhiqiang, Haoyu Yang, and A-Xing Zhu. ”Estimation of Travel Time Based on Ensem- ble Method With Multi-Modality Perspective Urban Big Data.” 24819-24828, IEEE Access 8 (2020).
9. Wu, Chun-Hsin and Ho, Jan-Ming and Lee, Der-Tsai,”Travel-time prediction with support vector regression”,IEEE transactions on intelligent transportation systems,vol-5,no-4,pp 276- 281,IEEE,(2004)
10. Chen, Che-Ming, Chia-Ching Liang, and Chih-Peng Chu. ”Long-term travel time pre- diction using gradient boosting.” Journal of Intelligent Transportation Systems 24, no. 2 : 109-124,(2020).
11. Chien, Steven I-Jy, and Chandra Mouly Kuchipudi. ”Dynamic travel time prediction with real-time and historic data.” Journal of transportation engineering 129, no. 6 : 608-616, (2003).
12. Kankanamge, Kusal D and Witharanage, Yasiru R and Withanage, Chanaka S and Hansini, Malsha and Lakmal, Damindu and Thayasivam, Uthayasanker,”Taxi Trip Travel Time Prediction with Isolated XGBoost Regression”,Moratuwa Engineering Research Con- ference (MERCon), pp 54-59, IEEE,(2019)
13. Chen, Lisi, Shuo Shang, Christian S. Jensen, Bin Yao, Zhiwei Zhang, and Ling Shao. ”Effective and efficient reuse of past travel behavior for route recommendation”, In Proceed- ings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 488-498. (2019)
14. Feng, Bin, Jianmin Xu, Yongjie Lin, and Penghao Li. ”A Period-Specific Combined Traffic Flow Prediction Based on Travel Speed Clustering.” , 85880-85889: IEEE Access 8 (2020).
15. Nath, Rudra Pratap Deb and Lee, Hyun-Jo and Chowdhury, Nihad Karim and Chang, Jae-Woo,”Modified K-means clustering for travel time prediction based on historical traffic data”,International Conference on Knowledge-Based and Intelligent Information and Engi- neering Systems,pp 511-521, Springer, (2010)
16. He,Peilan and Jiang, Guiyuan and Lam, Siew-Kei and Tang, Dehua,”Travel-time predic- tion of bus journey with multiple bus trips”,IEEE Transactions on Intelligent Transportation Systems, vol-20,no-11,pp 4192-4205,IEEE,(2018)
17. Wang, Dong and Zhang, Junbo and Cao, Wei and Li, Jian and Zheng, Yu,”When will you arrive? estimating travel time based on deep neural networks”,book:Thirty-Second AAAI Conference on Artificial Intelligence,(2018)
18. Wang, Zheng and Fu, Kun and Ye, Jieping, ”Learning to estimate the travel time”,Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Dis- covery and Data Mining, pp 858-866,(2018)
19. Asghari, Mohammad and Emrich, Tobias and Demiryurek, Ugur and Shahabi, Cyrus,”Probabilistic estimation of link travel times in dynamic road networks”,Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems,pp 1-10,(2015)
20. Dietterich, Thomas G. ”Ensemble methods in machine learning.” In International work- shop on multiple classifier systems, pp. 1-15. Springer, Berlin, Heidelberg, 2000
21. Zhang, Yanru, and Ali Haghani. ”A gradient boosting method to improve travel time prediction.” Transportation Research Part C: Emerging Technologies 58 : 308-324,(2015).
22. Cheng, Juan, Gen Li, and Xianhua Chen. ”Research on travel time prediction model of freeway based on gradient boosting decision tree.” IEEE Access 7 (2018): 7466-7480.
23. Ian H. Witten, Eibe Frank, Mark A. Hall Book: Data Mining Practical Machine Learning Tools and Techniques, Third Edition,Morgan Kaufmann Publishers is an imprint of Elsevier
24. Song, Wanchao, and Yinghua Zhou. ”Road Travel Time Prediction Method Based on Random Forest Model”, In Smart Trends in Computing and Communications, pp. 155-163. Springer, Singapore, 2020