The Prediction of Delay Time at Intersection and Route Planning for Autonomous Vehicles

Genwang Gou[1], Yongxin Zhao*[1,2], Jiawei Jiang[1], Ling Shi[3]
[1]Shanghai Key Laboratory of Trustworthy Computing, East China Normal University, Shanghai, China
[2]Shanghai Trusted Industrial Control Platform Co., Ltd., China
[3]School of Information Systems, Singapore Management University

***Abstract*—Intelligent Intersections (roundabout and crossroads) management is considered as one of the challenges to significantly improve urban traffic efficiency. Recent researches in artificial intelligence suggest that autonomous vehicles have the possibility of forming intelligent intersection management, and likely to occupy the leading role in future urban traffic. If route planning method can be used for route decision of autonomous vehicle, the urban traffic efficiency can be further improved. In this paper, we propose an Intelligent Intersection Control Protocol (IICP) for controlling autonomous vehicles cross intersection, and recommend route for autonomous vehicles to reduce travel time and improve urban traffic efficiency. Firstly, we run IICP to obtain the original data, use SMOTE algorithm to synthesize balance data, and use RF, GBDT algorithms to predict delay time. Secondly, we use the iEigenAnt algorithm to find multiple short routes in traffic network. Finally, we recommend route for autonomous vehicles based on the minimum of driving time on the route and all delay time at each intersection to improve urban traffic efficiency.**

***Index Terms*—intersection management, autonomous vehicle, SMOTE algorithm, route planning.**

I. Introduction

Over the past half-century, road intersections are managed by traffic lights or traffic control signals, these control mechanisms used to be efficient. However, as a growing number of vehicles flooding into the urban traffic flow, the shortcomings of these management mechanisms begin to emerge. In many cases, vehicles are required to stop even if there is no vehicles inside the intersection, resulting in traffic congestion at intersection, the trip time is also increased. The study of American cities shows that the congestion problem caused urban Americans to travel an extra 8.8 billion hours and purchases an extra 3.3 billion gallons of fuel for a congestion cost of $166 billion [1].

In order to solve these problems and improve intersection condition, some works [5][12] focus on regulating traffic flow and optimizing signal control, the study [11] tries to explore new intersection management design. Some researchers focus on proposing intelligent transportation system and develop autonomous vehicles to improve intersection efficiency, and they have made many remarkable achievements. On one hand, various autonomous vehicles have been developed and tested at intersection. On the other hand, many intelligent transportation system have been demonstrated efficient, such as work [3] which manages the efficient passage of autonomous vehicles through intersections, increase the throughput of the intersections by up to 96.24% compared to common signalized intersections.

In this paper, we propose an IICP protocol to manage autonomous vehicles cross intersection. There are two kinds of objects in IICP, which are Intelligent Control Center (ICC) and autonomous vehicle, ICC makes two important calculations to manage autonomous vehicles cross intersection. First, it analyzes the transmitted message by autonomous vehicles and generates a priority sequence of all vehicles near the intersection by First Come, First Serve strategy. Second, it detects potential collision among vehicles in order of priority sequence, and uses two speed adjustment mechanisms to calculates the safety speed for each autonomous vehicle. The simulation result shows that IICP can greatly reduce delay time for each autonomous vehicle crossing intersection.

Afterwards, we apply machine learning algorithms on IICP to predict delay time at intersection under different traffic flow and parameter configurations. We run IICP to generate the original data set, whereas the original data set is imbalanced. In an imbalanced dataset, the training instance of minority data is obviously less than that of other data, as a result, these examples are more likely to be mispredicted. Hence many researchers have proposed numbers of algorithms to solve the consequences of imbalanced data. These algorithms can be categorized as three mainstream types, which are algorithm, sampling and oversampling, Among them, oversampling technology can avoid losing data information[8], and synthetic minority samples to form balance data, one of the most popular method of oversampling level is the synthetic minority oversampling technique (SMOTE) in [7]. Experiments on imbalanced UCI data reveal that using SMOTE algorithm can effectively improve the performance compared with sampling level [10], so we use SMOTE algorithm to preprocess data set to form balance data, and use RF and GBDT algorithms fit these balance data to predict delay time.

Finally, we use ant colony algorithm to solve route planning problem. In our previous work, we have proposed an iEigenAnt algorithm [9], which can find short route between the source node and destination node in reticular structure. We apply iEigenAnt algorithm to find multiple short routes in traffic network, and recommend route for autonomous vehicles according to the sum of driving time on the road and all delay

*Corresponding author: yxzhao@sei.ecnu.edu.cn

DOI reference number: 10.18293/SEKE2020-018

time at each intersection.

The main contributions of this paper includes:

- **Prediction of Delay Time**. We use SMOTE algorithm to preprocess the original data to form the balanced synthetic data, and use RF algorithm to fit the synthetic data to predict delay time under different traffic flow and parameters configuration, which get a higher prediction accuracy.
- **Route Planning**. We use iEigenAnt algorithm to find multiple short routes between the source intersection and the destination intersection, and recommend route for autonomous vehicle based on the minimum driving time, which improve the urban traffic efficiency.

The remainder of this paper is organized as follow. Section II presents the methods used in this paper. Section III describes IICP in detail. Section IV elaborates the process of the prediction of delay time. Section V includes the route planning for autonomous vehicles. Finally, this paper is concluded in section VI.

II. Preliminaries

In this section, we describe some basic knowledge about techniques and models. We use SMOTE algorithm to synthesize minority samples to generate balance data. Afterwards, we use RF and GBDT to train the balance training data to predict the value of delay time.

A. SMOTE Algorithm

The paper [6] suggested that the SMOTE algorithm can avoid the risk of overfitting by randomly duplicates minority class instance. The core idea of SMOTE algorithm is to analyze the minority class samples and synthetic new samples according to these minority class samples, and add new samples to the dataset. There are three steps for SMOTE algorithm generates a new sample. Fig.1 shows the principle of SMOTE algorithm.

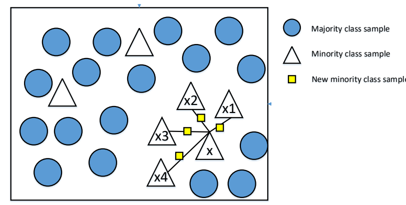


Fig. 1. Schematic diagram of synthetic data in SMOTE algorithm

- For each sample x in minority class samples S, calculates the distance from x and all samples in S according to the Euclidean distance, and then obtain its k-nearest neighbors.
- Determine the sampling ration n by the sample imbalance proportion. For each sample x, randomly choose several samples from its k-nearest neighbors, suppose as x_n.
- For each randomly selected neighbor x_n, use the equation (1) to synthetic a new sample x_{new}.

$$x_{new} = x + rand(0,1) * |x - x_n| \tag{1}$$

B. Random Forest (RF)

Random Forest is one of the most successful general-purpose algorithms in modern times. As an integrated training method, RF generates multiple prediction models and summarizes the results of the model to improve the accuracy of the prediction model. Figure 2 shows three main steps for RF to get the final prediction result or classification result.

- Get the training sets $S_1, S_2, ..., S_n$, (n represents the number of regression trees) according to the bootstrap [4] mechanism randomly with replacement.
- Training decision tree $T_1, T_2, ..., T_n$ based on training sets.
- The results of all regression trees are integrated to generate prediction value.

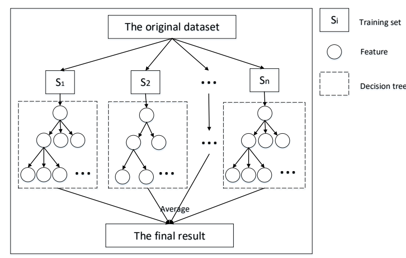


Fig. 2. Process of random forest algorithm

C. Gradient Boosting Decision Tree

Gradient Boosting Decision Tree is an iterative decision tree algorithm, which is composed of multiple decision trees, and the results of all trees are accumulated to make the final result. There are three steps for GBDT generates the final classification tree.

- Initial a weak tree with one root node, which can minimize the loss function.
- Calculate the negative gradient of loos function in current model, and take it as an estimate of the residuals. The core idea of Gradient Boost is to build a new model in the gradient direction of residual reduction to eliminate the previous residual, which is quite different from the traditional boost in weighting the correct and wrong samples.
- Estimate the regression leaf node area to fit the approximate residual value. Using linear search to estimate the value of leaf node region, minimizing the loss function, and update decision tree.

III. Intelligent Intersection Control Protocol

In this section, we present the intelligent protocol IICP, which aims at increasing the throughput of autonomous vehicles at intersection. Fig.3 shows the cross process of a new coming vehicle, it needs to go through four zones to pass intersection.

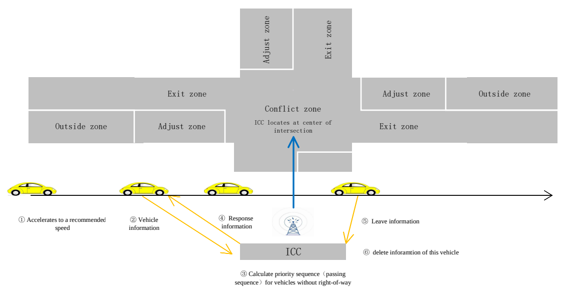


Fig. 3. The intersection layout and cross process of a new coming vehicle

In outside zone, all vehicles will accelerate to a recommend speed (as shown in Fig.4) to reduce travel time. In adjust zone, vehicles need to adjust their speed according to the instruction of ICC. In conflict zone, these vehicles will keep a constant speed until they cross the intersection. When autonomous vehicles arrival at the exit zone, which means vehicles have crossed the intersection, they can accelerate to the maximum speed to travel. Notice that all autonomous vehicles need to follow car-following strategy in these four zones, that is to say, the vehicles in the back must follow these in front.

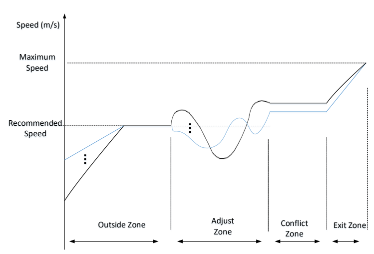


Fig. 4. Speed curve in IICP

A. Assumptions

In IICP, the human driving is replaced by autonomous driving, the whole control system should be rethought, so we introduce the following assumptions:

Intersection Assumptions: We model the intersection as a grid which consists of multiple cells, each cell has a unique identification number and can accommodate an autonomous vehicle. The intersection is controlled by ICC rather than the traffic light mechanism, and ICC is equipped with wireless communication device, powerful calculation device, etc...

Vehicle Assumptions: We assume all vehicles are autonomous vehicles, they have similar shape and physical, and they are equipped with sensor, positioning system, wireless communication device, so they can perceive nearby obstacles, obtain their position, interactive information with ICC.

B. Actions

ICC Actions:

1) ICC calculates the priority sequence and safe speed at fixed intervals by *priority determination* and *speed adjustment mechanism*, afterwards, it sends speed adjustment message to each vehicle accordingly.
2) ICC will ignore some vehicles when it is notified that these vehicles have crossed the intersection.

Vehicle Actions:

1) All vehicles cannot enter the intersection without receiving speed adjustment message from ICC.
2) Whatever vehicles are in conflict zone or adjust zone, they should periodically send vehicle driving information to ICC, including vehicle location, speed, etc...
3) If ICC has replied driving information to each vehicle, they must change their speed accordingly.
4) If some vehicles have crossed the intersection, they should immediately send message to notify ICC.

C. Priority Determination

IICP is a sequence-based protocol, ICC needs to generate a *priority sequence* for the adjust zone vehicles, lay the foundation for the subsequent speed adjustment mechanism.

Now, we define the notations that will be used later.

- (r, r): Intersection cell size.
- $CList$: Records the vehicles which inside conflict zone.
- $AList$: Records the vehicles which inside adjust zone.
- $D_{X,e}$: The distance between vehicle X and the entrance of intersection.
- CL: The cell list which vehicle needs to use to cross the intersection. CL_i means the ith cell of CL.
- $count_n$: Number of cells before vehicle X enters cell n.
- T_n: The occupy time of cell n.
- $T^X_{in_n}$: Time of vehicle X entering cell n.
- V_X: Speed of vehicle X.
- V^{new}_X: The desired speed of vehicle X to avoid collision.
- P: The priority sequence of all AList vehicle, P_X refers to the priority of vehicle X.
- HP_i: Record AList vehicles which have higher priority than ith highest priority vehicle.

With these notations, we now have: the time of vehicle X enters cell n consists of the time when vehicle X reaches the edge of intersection and the time when vehicle X passes through each cell before cell n (Equation (2)). Notice that the priority of AList vehicles is lower than that of CList vehicles, that is to say, the AList vehicles must avoid the CList vehicles.

$$T^X_{in_n} = \frac{D_{X,e}}{V_X} + \sum_{i=1}^{n-1} T_{CL_i} \tag{2}$$

D. Speed Adjustment Mechanism

When the priority sequence of AList vehicles is determined, ICC uses speed adjustment mechanism to adjust the speed of these vehicles. Assume there exist a vehicle X and a higher priority vehicle $X1$, if there's no collision between vehicles X and $X1$, ICC will accelerates vehicle X. The best situation is that vehicle $X1$ has just left a cell n and vehicle X enters it, so V^{new}_B can be calculated from Equations (3) and (4).

Algorithm 1: Speed Adjustment Mechanism

Assume there are m vehicles in CList and HP_i.
while *exists vehicles in adjust zone* **do**
  Calculate the time of each vehicle arrival at intersection, record vehicles in priority sequence P by FCFS strategy.
  Calculate $T^X_{in_n}$ for each CList vehicles and AList vehicles through Equation (2).
  for *i=0, i $<$ the length of P* **do**
    Choose the ith highest priority vehicle i from P, check whether there exist potential collision among vehicle i and m vehicles.
    if *There is no conflict* **then**
      for *j=0, j $<$ m* **do**
        Calculate the max allowed speed for vehicle i with vehicle j by Equations (3) and (4), suppose as V_i^j.
      Choose the $\min(V_i^1, V_i^2, ..., V_i^m)$ as the safety speed vehicle i can accelerates to.
    else
      for *j=0, j $<$ m* **do**
        Calculate the max allowed speed for vehicle i with vehicle j by Equations (5) and (6), suppose as V_i^j.
      Choose the $\min(V_i^1, V_i^2, ..., V_i^m)$ as the safety speed vehicle i can decelerates to.
    Add vehicle i to HP_i, $m = m + 1$.
  Reply specific driving information to AList vehicles.

$$\frac{(V_X + V_X^{new}) * T^{X1}_{out_n}}{2} = D_{X,e} + r * count_n \tag{3}$$

So we have:

$$V_X^{new} = \frac{2 * (D_{X,e} + r * count_n)}{T^{X1}_{out_n}} - V_X \tag{4}$$

In contrast, if there exists potential collision between vehicles X and $X1$, ICC will decelerates vehicle X. The way to calculate V_B^{new} is shown in Equations (5) and (6).

$$\frac{(V_X - V_X^{new}) * T^{X1}_{out_n}}{2} = D_{X,e} + r * count_n \tag{5}$$

So we have:

$$V_X^{new} = V_X - \frac{2 * (D_{X,e} + r * count_n)}{T^{X1}_{out_n}} \tag{6}$$

On the whole, the process of speed adjustment mechanism is to traverse each vehicle in order of priority sequence P and determine the safe driving speed. Assume that there are n vehicles in AList, so there will be n cycles in total. At the ith cycle, ICC gets the ith highest priority vehicle i from P, and check whether there will be potential collision among vehicle i, CList vehicles and HP_i vehicles. Note that potential collision means if these vehicles drive at current speed, there will be collision among them. There are two cases, which are *conflict* and *no conflict*.

Case 1. No conflict: In this case, ICC will accelerates vehicle i while ensuring safety. Assume the number of vehicles in CList and HP_i is m, ICC needs to calculate the maximum allowed speed with each vehicle in m vehicles according to equations (3) and (4), suppose as V_i^1, V_i^2, ..., V_i^m. Afterwards, ICC chooses the $\min(V_i^1, V_i^2, ..., V_i^m)$ as the safe driving speed that vehicle i can accelerate to.

Case 2. Conflict: In this case, ICC will decelerates vehicle i to avoid potential collision. Assume the number of vehicles in CList and HP_i is m, ICC needs to calculate the maximum allowed speed with each vehicle in m vehicles according to equations (5) and (6), suppose as V_i^1, V_i^2, ..., V_i^m. Afterwards, ICC chooses the $\min(V_i^1, V_i^2, ..., V_i^m)$ as the safe driving speed that vehicle i can decelerate to.

IV. Prediction of Delay Time

In this section, architecture of prediction of delay time is explained. Fig. 4 shows the process of the prediction of delay time. We run IICP to generate the original data set, and use SMOTE algorithm to balance the imbalanced data, then use RF and GBDT algorithms to predict the value of delay time under different traffic flow and algorithm parameters.

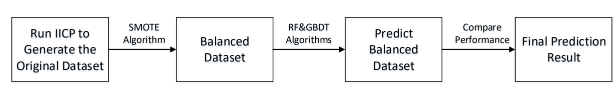


Fig. 5. Architecture of prediction of delay time

A. SMOTE Algorithm for Data Preprocess

We obtain the original data by running IICP on SUMO [2] firstly. There are five important parameters in this protocol, which are vNumber, acc, dec, minGap, maxSpeed, the meaning and the configuration of these parameters are shown in Table I. We set these features to the given value in Table I respectively. However, we find the original data is imbalanced

Algorithm 2: Prediction of delay time

Input: train dataset (x_i, y_i), test dataset (x_j, y_j), where $i = 1, 2, ..., m$, $j = m+1, m+2, ..., m+n$.
Output: the appropriate classification algorithm and fitted models.
Step 1: Divide training dataset into N binary subsets considering all classes.
Step 2: Use SMOTE algorithm synthesize balance data.
for $c = 1, c \leq N$ **do**
    Apply SMOTE algorithm on ith class data.
Combine all classes to form balance data.
Step 3: Apply RF and GBDT algorithms on the balanced data, select the better algorithm according to the evaluation method.
Return the appropriate prediction algorithm and fitted models.

(as shown in Fig.6), most data is in the range of 0 to 6, so we need to use SMOTE algorithm preprocess the original data.

TABLE I
KEY FEATURES IN IICP

Name	Meaning	Configuration
vNumber	the number of vehicles cross intersection in given time	$[25, 1000]$
acc	max acceleration of vehicle	$[2, 2.4]$, m/s^2
dec	max deceleration of vehicle	$[2, 2.4]$, m/s^2
minGap	min gap between continuous vehicles	$[10, 14]$, m
maxSpeed	max speed allowed on road	$[20, 29]$, m/s

Firstly, we segment the original data into 7 classes by the range of delay time. Secondly, we execute 7 SMOTE operations on the original dataset. On the ith SMOTE operation, we choose the ith class as the minority data and synthesize data according to the imbalance rate, and then we combine all classes to form the balance data. Finally, the synthetic balanced data will be the entry data of RF and GBDT algorithms, the appropriate prediction algorithm will be obtained by the evaluation methods. We have transform the data processing part as algorithm 2.

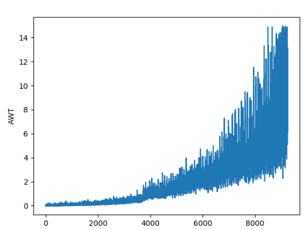


Fig. 6. Delay time under different traffic flow

B. Evaluation Methods

Performance evaluation metrics play an important role in assessing the prediction performance. We use Mean_Absolute_Error (MAE), Mean_Square_Error (MSE), R^2 Score (R^2) as the evaluation methods to evaluate the performance RF and GBDT algorithms in the original data and the synthetic data. y_i^t is the true value of y_i, y_i^p is the predicted value of y_i.

MAE: It illustrates the difference between the predicted value and the real value, the smaller, the better.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i^p - y_i^t| \tag{7}$$

MSE: It illustrates the square error between the predicted value and the actual value, the smaller, the better.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i^p - y_i^t)^2 \tag{8}$$

R^2: It illustrates the fitting degree of the prediction model and the real data, The best value is 1.

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i^t - y_i^p)^2}{\sum_{i=1}^{n}(y_i^t - \frac{1}{n}\sum_{j=1}^{n}y_j^t)^2} \tag{9}$$

C. Experiment Result

We apply RF and GBDT algorithms on the original data and synthetic data sets respectively, to evaluate the performance of using smote algorithm synthesizes data. The experiment result shows that using smote algorithm can significantly improve the prediction performance, and RF has a better prediction result than GBDT algorithm (as shown in Table II).

TABLE II
EXPERIMENT RESULT OF RF, GBDT ALGORITHMS ON THE ORIGINAL DATA AND THE SYNTHETIC DATA

Algorithm	Evaluation Methods	Original Data	Synthetic Data
RF	MAE	0.406	0.162
	MSE	0.681	0.146
	R^2	0.876	0.971
GBDT	MAE	0.384	0.268
	MSE	0.599	0.214
	R^2	0.891	0.956

V. Route Planning

In this section, we describe the way to provide route planning services for autonomous vehicles in detail. When human driving vehicles are replaced by autonomous vehicles, it's important to provide route planning services for autonomous vehicles.

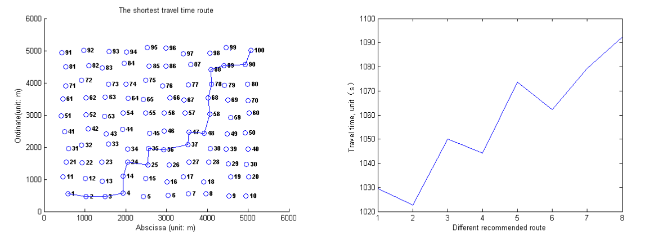
A. *iEigenAnt Algorithm*

In recent years, many researchers have put forward route planning algorithms, among them, intelligent bionics algorithm is rather useful when dealing with the problem of route planning under the condition of complex dynamic environment. In our previous work, we have proposed an intelligent bionic algorithm which called improved EigenAnt (iEigenAnt) algorithm, it can find short route between multiple points according to the way of both positive and negative feedback, we have successfully applied iEigenAnt algorithm on TSP problem. In this section, we use iEigenAnt algorithm to provide route planning service for autonomous vehicles.

B. *Experiment Design and Result*

In this section, we recommend route for autonomous vehicles according to the minimum of driving time on the route and all delay time at each intersection. For driving time on the route, we use iEigenAnt algorithm to find multiple short routes between source intersection and destination intersection. For delay time we randomly allocate traffic flow and parameter setting for each intersection to simulate the real scene, and use the model trained in section IV to predict the value of delay time. Finally, we choose the route which takes the least time as the recommend route.

We model a $10 * 10$ intersection network, where vehicles start from the source intersection (intersection 1) to the destination intersection (intersection 100), the horizontal and vertical distance between each two adjacent intersections is about 500 meters. Firstly, we use iEigenAnt algorithm find multiple short routes, and choose 8 routes as candidate recommended route, the length of each route is 8309.3, 8318.1, 8404.5, 8412.5, 8486.0, 8542.1, 8625.3, 8720.5 meters respectively. Secondly, we calculate the sum of delay time on each route, which is 32.43, 24.57, 41.53, 34.73, 55.47, 37.13, 44.26, 45.95 seconds respectively. Assume vehicles drive at 30km/h, the total travel time of 8 routes is shown in Fig.7. It's obviously, the route which takes the least travel time is route 2 instead of the shortest route 1.



(a) The recommended route (b) Total travel time of each route

Fig. 7. Applying iEigenAnt algorithm on route planning problem

VI. Conclusion

Autonomous driving can significantly decrease delay time for autonomous vehicles crossing intersection, and it will be the heart of urban transportation in the near future. So we propose an IICP to manage the intersection area to solve traffic congestion problem and seek the global benefit by dynamically allocating a safe time-space passage for each vehicle. Afterwards, we use iEigenAnt algorithm to recommend route for autonomous vehicles according to the minimum of driving time on route and all delay time at intersection, the experiment results show our idea is effective.

VII. Acknowledgement

This paper is partially supported by National Key Research and Development Program of China (No. 2019YFB2102600), Science and Technology Commission of Shanghai Municipality Project (No. 18ZR1411600) and the Open Project of Shanghai Key Laboratory of Trustworthy Computing (No. 08dz22304201804).

References

[1] 2019 urban mobility report and appendices. https://mobility.tamu.edu/umr/report/.

[2] simulation of urban mobility. http://sumo.sourceforge.net/.

[3] Reza Azimi, Gaurav Bhatia, Ragunathan Rajkumar, and Priyantha Mudalige. Ballroom intersection protocol: Synchronous autonomous driving at intersections. In *2015 IEEE 21st International Conference on Embedded and Real-Time Computing Systems and Applications*, pages 167–175. IEEE, 2015.

[4] Leo Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.

[5] Shujian Bu, Zhang Tong, and Du Li. Approximations and simulation of the optimal change interval for roundabout. In *International Workshop on Database Technology & Applications*, 2010.

[6] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16(1):321–357, 2011.

[7] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

[8] Douzas Georgios, Bacao Fernando, and Last Felix. Improving imbalanced learning through a heuristic oversampling method based on k-means and smote. *Information Sciences*, pages S0020025518304997–.

[9] Genwang Gou, Yongxin Zhao, Qin Li, and Qiwen Xu. A mathematical analysis of improved eigenant algorithm. *Journal of Experimental & Theoretical Artificial Intelligence*, 31(3):1–19, 2018.

[10] Li Ma and Suohai Fan. Cure-smote algorithm and hybrid algorithm for feature selection and parameter optimization based on random forests. *Bmc Bioinformatics*, 18(1):169.

[11] Tomaz Tollazzi, Marko Rencelj, and Saso Turnsek. Slovenian experiences with alternative types of roundabouts-" turbo" and" flower" roundabouts. In *Environmental Engineering. Proceedings of the International Conference on Environmental Engineering. ICEE*, volume 8, page 1220. Vilnius Gediminas Technical University, Department of Construction Economics ¡, 2011.

[12] Bai Yu and Kun Xue. Association of signal-controlled method at roundabout and stop rate. In *International Conference on Intelligent Computation Technology & Automation*, 2011.