Computer Simulation Study of Vehicle Type Classification Using Machine Learning Techniques with Mobile Phone Location Data

Chaninart Eosanurak, Nattida Wongtrakoon, Ei Ei Mon, Chaodit Aswakul

Wireless Network and Future Internet Research Unit, Department of Electrical Engineering, Faculty of Engineering, Chulalongkorn University, Thailand
chaninart.e@student.chula.ac.th, nattida.w@student.chula.ac.th, 6071456921@student.chula.ac.th, chaodit.a@chula.ac.th

Abstract

Road traffic data are conventionally obtained by fixed-location detectors e.g., induction loop coil sensors and closed-circuit television systems. However, these conventional on-road sensors for traffic data collection cost highly both in installation and maintenance, and they are only available in a limited part of the road network. Thanks to the current ubiquity of cellular communication infrastructure, the mobile phone location data, obtained from the cell towers, are an alternative to collect real-time on-road traffic data with low costs for traffic management operations. In this paper, the classification of vehicle types is considered from the mobile phone location data that can be made available at the cell towers whose locations are known. Considering data that can be mixed from mobile phones carried by passengers in the different modes of urban transportation, this work deals with the motorcycle, car, and Skytrain via the Simulation of Urban Mobility (SUMO) with the Sublane model. The random forest, k-nearest neighbor, and support vector machine algorithms are applied for the vehicle type classification. From the random forest and k-nearest neighbor algorithms, the classification accuracies are greater than 85% and 80%, respectively, with the cell tower inter-spacing less than 500 meters and greater than 80% for the average classification accuracy for both random forest and k-nearest neighbor algorithms. On the other hand, the support vector machine algorithm's classification accuracy is greater than 85% only in the case of 100 meters of cell tower inter-spacing. Since the existing cell tower inter-spacing is typically in the order of 500 meters for the current 4G mobile phone technology, and as lows as 100 meters for the upcoming 5G technology, the SUMO-based finding in this paper suggests a clear potential of using this ubiquitously available mobile phone data for the vehicle type classification and other subsequent travel information extraction analysis in the future.

1 Introduction

Increasing traffic congestion in Bangkok is a severe problem that affects transportation quality and vehicle fuel consumption as well as human health because of pollution from vehicle emission. Traffic congestion is caused by many factors ranging from the lacking of efficient urban plans, the increasing of vehicle population inadequately to available road capacity, and the insufficiently informed traffic management. To alleviate the traffic problem, this paper is concerned with the last factor, i.e., to find ways of providing new potential sources of traffic data that can be used in real-time for traffic management operations.

Conventionally, traffic congestion and control management rely on sensor data from induction loop coil sensors and closed-circuit television (CCTV) systems [1]. Despite past research showing the sensor and CCTV usage efficiency, the cost of installation and operating is still expensive, and their operational deterioration is a difficult burden. That is especially true for areas with vast coverage like Bangkok or other metropolitan cities. Such high expenses thus prevent the full deployment of those conventional sensors in countries with restrictive economic constraints.

However, recent advances show another potential source of data by using the mobile phone infrastructure that has already been made ubiquitous in urban areas. In this regard, based on the cellular communication architecture, mobile user equipment must try to connect with the neighbor base stations since the location of those base stations is known to the mobile phone service providers, and the rough location of that mobile user equipment can also be extracted. Mobile user equipment carried by passengers in moving vehicles can be used to infer about the vehicle mobility and hence the overall traffic congestion status.

In the past, there exist researches that try to utilise data from mobile phone systems. In [2], traffic data are obtainable by using mobile phone signals to reduce the service costs and to increase the richness of the obtainable data. Particularly, the mobile phone location data have been used to classify the road conditions into three traffic congestion levels, i.e., low, medium or high congestion. In [3], the paper aims to approximate traffic volumes and the number of commuters within each vehicle. Then, the travel mode is classified into drive-alone, carpooling or bus usages. In [4], the information from both mobile devices and Bluetooth signals have been used to detect moving vehicles and classify them into car, truck, and semi-truck categories. In [5], by exploiting mobile phone location data, the congestion level can be estimated and classified into fluency, light congestion and heavy congestion categories. In [6], the paper handles the cellular phone location data to extract vehicle traffic flow rates. In [7], the paper uses mobile phone location data to estimate the accuracy of travel time. In [8], human mobility patterns are extracted from mobile phone location data for the development of transportation planning purpose. In [9], the paper exploits the mobile device location for traffic monitoring and compares the accuracy obtainable from a range of traffic data sources.

The existing researches have shown a promising potential to use mobile phone location data for vehicle movement data extraction. However, the past researches have considered not all plausible types of vehicles that may be present in many urban road networks. In particular, to the best of our knowledge, there is not yet any consideration on the existence of two-wheel vehicles like motorcycle, or rail transportation modes. These two transportation modes are important both because of their usage popularity in many cities and their interesting movement patterns that can affect the accuracy of the traffic data to be extracted. Research in the scenario with various combinations of vehicle types, i.e., cars, motorcycles, and rails, is believed worthwhile. To carry out this research, an efficient simulation platform that is ready with the ability to mimic the movement of various vehicle types becomes an essence.

Fortunately, both the vehicle movements and the corresponding traffic data assumed available at mobile phone base stations can be simulated and generated via SUMO (Simulation of Urban Mobility) (e.g. [10], [11]). As the first step, in this paper, we are interested to analyze the simulated results to classify vehicle types. Vehicle type recognition is important in traffic prediction because vehicle type classification can help filter out some data that are not relevant to road traffic congestion. To estimate the congestion levels on surface road networks, the data noises can be injected by the mobile phone location data from moving passengers inside the so-called *Skytrain*, which is rail-mode transportation that runs on the up-lifted infrastructure built separately high above the underlying road surface. Moreover, due to the smaller size of motorcycle, the mobile phone location data from moving passengers driving motorcycles should be considered with lower importance than cars. Without the classification, the travel time obtained from the mixed passengers of Skytrains, cars and motorcycles would result definitely in data extraction inaccuracy.

This paper aims to investigate the factors that affect the accuracy of vehicle-type classification algorithms. The vehicle types in this paper are car, motorcycle, and Skytrain. As the first step, a single straight road is considered with regularly spacing traffic light signals sharing the locations with the stopping station of Skytrains. From similar work with only subsets of vehicle types considered therein, three interesting classification algorithms have been investigated (e.g. [4]). In this research, we have selected to compare three standard algorithms, i.e., the random forest, k-nearest neighbor, and support vector machine algorithms.

Random forest algorithm [12] is an ensemble method, developed by combining decision trees. The decision tree algorithm is the supervised learning algorithm with a tree-like structure to decide with each branch of the tree represents each condition. Instead of splitting all input features for every tree, the random forest algorithm randomly selects the input features and samples by using bootstrap to create the trees. The majority vote of all trees in the forest is used for evaluation.

K-nearest neighbor algorithm [13] is the supervisor algorithm using for classification an unknown class data. By using labeled training data, the k-nearest neighbor algorithm classifies the class of unknown class data by considering k numbers of nearest known class data and take the majority vote class to be class of the considered data.

Support vector machine algorithm [14] with a linear kernel, considered in this paper, applies hyperplane, a linear combination equation, for classification. Thereby, the optimal hyperplane, causing the highest accuracy, is chosen by the mathematics method to calculate the hyperplane with the largest distance between the hyperplane and each class.

The influencing factors interested in this paper are the cell tower inter-spacing that affects the level of data resolution obtainable; the traffic light signal inter-spacing, the number of vehicles per hour and the length of green and red time intervals, all of which affect the overall road congestion level.

In the remainder, this paper is structured as follows. Section 2 presents the problem formulation. Section 3 then gives the experimental results and discussion. Finally, Section 4 provides the conclusion.

2 Problem Formulation

In this paper, the road configuration is set up by using SUMO [10] version 1.4.0. One-dimension road with 3 lanes 3-kilometre-long for car and motorcycle and another railway with the same length for Skytrain are implemented as shown in Figure 1. This simplified network topology allows then the vehicle's behavior to be analyzed easily while still capable of simulating the motorcycle penetrating behaviour on the lateral area between moving or stopping cars. Traffic light signal inter-spacing and Skytrain station inter-spacing are varied from 500, 1000, and 1500 meters which are chosen by the approximated length of the maximum and minimum Skytrain station inter-spacing in Bangkok. The traffic light signal and the Skytrain station are always in the same position to simulate the worst-case scenario when the data of mobile phone from Skytrain, moving on the railway, interrupt the data of mobile phone from the vehicles on surface road underneath with the same stopping points.

The whole 3-kilometer-length of the road is equally divisible by the traffic light signal inter-spacing. So, the traffic light signal could be in-space periodically located. Additionally, cell tower inter-spacing is varied to evaluate the effect of the classification accuracy at 100, 250, 500, 1000, and 1500 meters. Moreover, the road with 3-kilometer-long is divisible by all cell tower inter-spacing, resulting in the simple model of the cell tower locations.

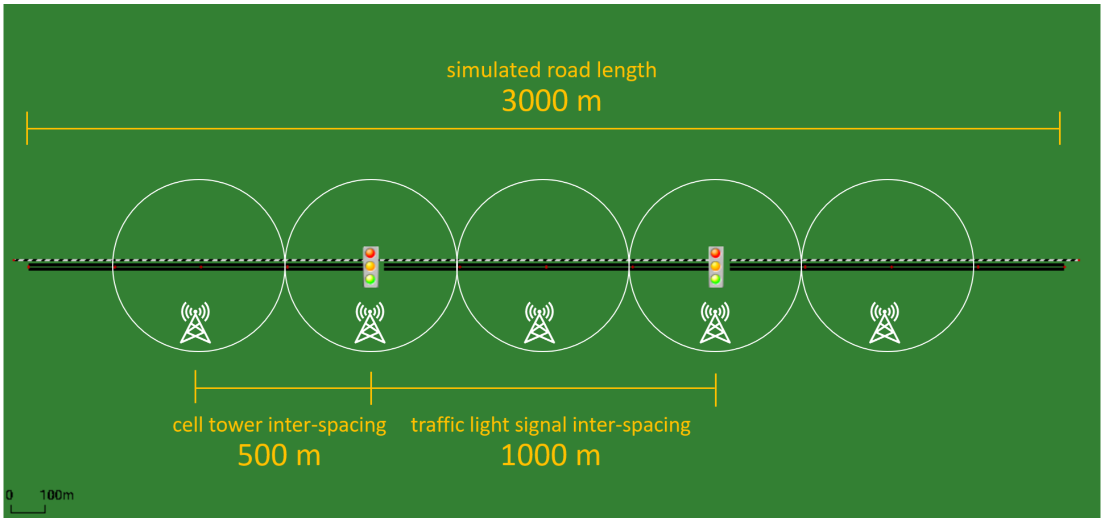


Figure 1: Model with traffic light signal inter-spacing of 1000 meters and cell tower inter-spacing of 500 meters

In practice, the cell tower radius of the 5G network is recommended to be less than or equal to 50 meters [15]. So, the highest resolution in this paper is set for practical design as a cell tower inter-spacing 2×50 or 100 meters. And the cell tower inter-spacing is varied to find out the effect of the resolution.

The length of the green and red time intervals of the traffic light signals is varied according to practical operational measurements in [16], which aims to observe the start-up lost time in Bangkok. Specifically, the cycle length is set to a constant value at 230 seconds while the green time is varied from 50, 100, and 150 seconds according to the minimum and maximum green-time range in [16]. Although the standard yellow light is practically set to 3 seconds, this state does not affect the traffic in Bangkok significantly so the yellow light can be ignored from this computer simulation study. The teleported vehicle is neglected because the teleportation in the simulation causes the missing of data which is not considered in this paper.

Table 1: The parameters of motorcycle and car for SUMO simulation

Vehicle Type	**maxSpeed**	**accel**	**decel**	**width**	**length**	**latAlignment**	**minGap**
Motorcycle	12.5	4.0	4.0	0.5	1.9	compact	0.4
Car	12.5	2.0	2.5	1.78	4.62	center	2.37

Vehicle Type	**minGapLat**	**lcSublane**	**lcPushy**	**lcAssertive**	**lcLaneDiscipline**
Motorcycle	0.05	10000	10	10	-
Car	-	100	0	1	1

Three types of vehicle considered in this paper are the car, motorcycle, and Skytrain, which are represented by the vehicle type of passenger car, motorcycle, and rail urban in SUMO simulator. The motorcycle model has been adjusted using the Sublane model [17] with resolution 0.7 meter, the sub-lane width of 0.7 meters with the parameters as shown in Table 1. And the ratio of cars and motorcycles is 0.56:0.44 which this ratio is based on the actual population of those vehicle types in Bangkok [18]. These parameters are set for the motorcycle configuration so the motorcycles could penetrate through the stopping cars alike what happens in reality. The stop offset is set to 5 meters at all traffic light signals for the motorcycles to stop in front of the car stopping line, simulating the stopping line layout in Bangkok, as shown in Figure 2.

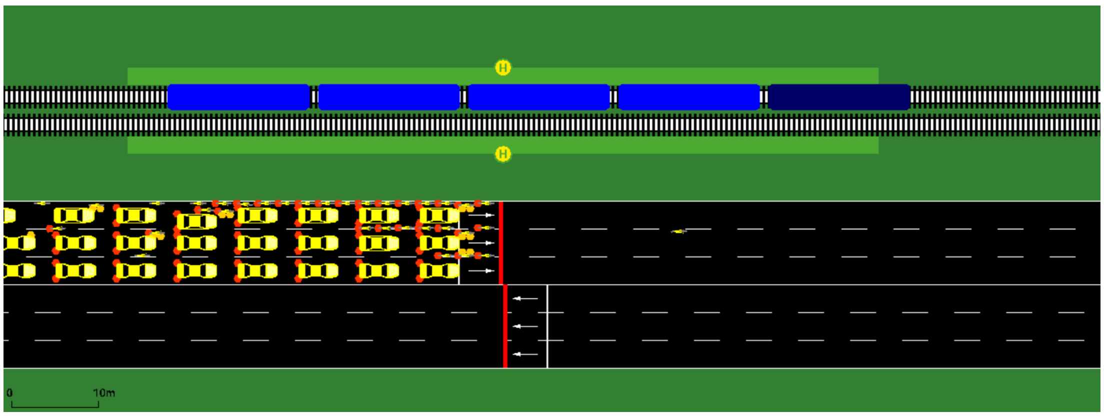


Figure 2: Stopping behaviors of motorcycles, cars and Skytrains

The formulation assumptions of this paper are as follows: (1) The cell towers are assumed to be installed alongside the road and be equidistant. So, the cell towers could connect to the mobile devices located within the vehicles, moving along the street and the effect of cell tower inter-spacing could be discovered. (2) The location data of mobile phone are only from the mobile devices that are located within vehicles of 3 types, which are car, motorcycle, and Skytrain. (3) A mapping ratio between the vehicles and the mobile phone is 1:1 to consider the worst-case scenario for classification. For instance, if multiple mobile phones are originated from a single Skytrain, then the location data of those mobile phones will become easily clustered and distinguishable from the location data of mobile phones from cars and motorcycles. (4) The mobile phones are connected to the closest cell towers to gain the highest intensity of mobile phone signal. This is a custom design of cellular networking principle. Consequently, the vehicle location could be implied by the mobile phone signal's connected cell towers.

The data are collected from the SUMO simulator by using TraCI (Traffic Control Interface) [19] in the form of the positions, time values, vehicle IDs, and vehicle types with 1 Hertz sampling frequency for 2 hours in each simulation. From the criteria of this paper, the mobile phone locations and the vehicle locations are considered as a 1:1 mapping; thus, the number of mobile phones and the number of vehicles are the same. And the mobile phone locations are supposed to be the same as those of the vehicles.

Based on TraCI data, the movement data of mobile phones are converted by a simple programming logic to the cell tower IDs with time values where mobile phones are connected. The conversion is follow the rule that each mobile phone could connect to one cell tower at a time which is the closest cell tower. All these events are time-stamped in SUMO, reported in TraCI, so finally the cell dwelled time at each cell tower can be calculated. Herein, to generate the data point for each vehicle traversing all the cell towers in its movement direction, the cell dwelled time C_i at a given i-th cell tower is computable from the difference between the first timestamp that the cell first sees a mobile phone on its arrival and the last timestamp that the cell last sees that mobile phone on its departure as shown in Figure 3.

Vehicle ID	Vehicle Type	Cell ID	First Time	Last Time	Cell Dwelled Time = Last Time - First Time+1
veh00	Car	cell00	21760	21769	10
veh00	Car	cell01	21770	21777	8
veh00	Car	cell02	21778	21784	7
veh00	Car	cell03	21785	21793	9
veh04	Skytrain	cell07	21927	21934	8
veh04	Skytrain	cell06	21935	21966	32
veh04	Skytrain	cell05	21967	21978	12
veh04	Skytrain	cell04	21979	21986	8
veh08	Motorcycle	cell02	22918	22925	8
veh08	Motorcycle	cell03	22926	22934	9
veh08	Motorcycle	cell04	22935	22943	9
veh08	Motorcycle	cell05	22944	22952	9

Figure 3: Example of cell dwelled time calculation from simulation

Vehicle ID	Class	Input Feature: Cell Dwelled Time (sec)					
	Vehicle Type	Cell00	Cell01	Cell02	Cell03	Cell04	Cell05
veh00	Car	10	8	7	9	8	9
veh01	Car	10	9	7	8	9	8
veh02	Car	10	9	7	9	8	9
veh03	Car	10	9	7	8	9	8
veh04	Skytrain	7	8	7	8	8	12
veh05	Skytrain	8	8	7	8	7	12
veh06	Skytrain	8	7	8	8	7	10
veh07	Skytrain	8	7	8	7	8	7
veh08	Motorcycle	9	9	8	9	9	9
veh09	Motorcycle	61	10	9	9	10	10
veh10	Motorcycle	10	9	9	9	10	9
veh11	Motorcycle	10	8	8	9	10	9

Figure 4: Example of input features and output label from simulation

Figure 4 shows the data structure which included vehicle IDs, output label, and input features, reconstructed from calculated cell dwelled time at each cell tower, which is prepared for classification. To evaluate the vehicle types classification, the standard random forest, k-nearest neighbor, and support vector machine algorithms are chosen. With the cell dwelled time at every cell tower (C_i for all i) as the input feature, and the vehicle type as the output label, the problem of classifying vehicle types is here formed. To evaluate the classification algorithms, the train-test split method has been used with the splitting ratio of 80% for training and 20% for testing.

3 Results and Discussion

3.1 Vehicle Type Classification on One-dimensional Road Network

Table 2: Summary of main traffic parameters as defined in 6 congestion levels

	Green Time (sec)	**Red Time (sec)**	**Total Flow Rate**	**Congestion Level**
case 1	150	80	1400	0.072
case 2	100	130	1400	0.116
case 3	150	80	2800	0.143
case 4	50	180	1400	0.161
case 5	150	80	4200	0.215
case 6	100	130	2800	0.233

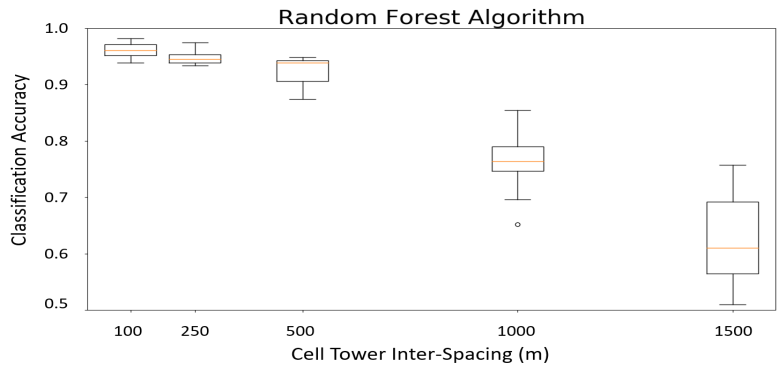
Table 2 shows the 6 cases of congestion levels of the motorcycle and car which are composed of the length of green and red time intervals with 230 seconds of cycle time and the number of vehicles per hour, including for both motorcycles and cars. However, the rate of Skytrain is constant and set to 10 Skytrains per hour. The congestion level can be calculated by:

$$\text{Congestion Level} = \frac{\text{Red Time}}{\text{Cycle Time}} \times \frac{\text{Total Flow Rate}}{\text{Saturation Flow Rate}} \tag{1}$$

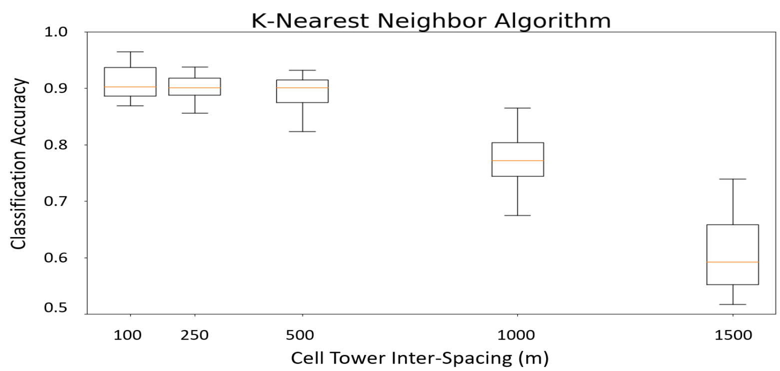
where the cycle time is 230 seconds and the saturation flow rate is 6800 vehicles per hour.

SUMO simulation has been carried for a week and its TraCI outputs have been processed by Python classification codes. Figure 5 shows the relationship between cell tower inter-spacing and classification accuracy. Each box represents the distribution of the simulation in each case of congestion level, as shown in Table 2 with varying traffic light signal inter-spacing are varied from 500, 1000, and 1500 meters. For the cell tower inter-spacing as the x-axis, the lengths of 100, 250, 500, 1000, and 1500 meters indicate the resolution of each classification algorithm. Because the cell tower inter-spacing implies the precision of vehicle location, the cell tower inter-spacing can be considered as the resolution of mobile phone location data availability. The location of the individual vehicle could be estimated by using the cell tower inter-spacing length together with the timestamp that each mobile phone connects with each cell tower.

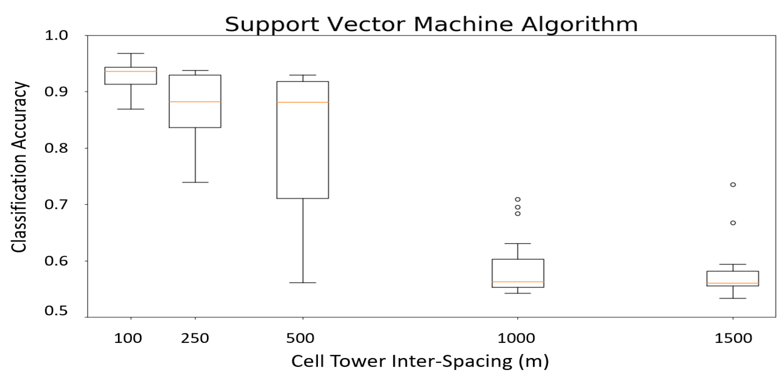
From Figure 5, the classification accuracy is compared to each cell tower inter-spacing value to observe its effects. The graph of each algorithm shows that the classification accuracy is better in the shorter, hence higher in resolution, cell tower inter-spacing than a longer inter-spacing case. This is because the low value of cell tower inter-spacing indicates a high precision data for classification. And unsurprisingly, all the algorithms show this similar trend.



(a)



(b)



(c)

Figure 5: Boxplot of classification accuracy for each cell tower inter-spacing from each algorithm

In Figure 5(a), the random forest algorithm is employed to analyze the data for calculating the classification accuracy. Among all of the algorithms in this paper, the random forest is the best algorithm that predicts the highest overall classification accuracy. This algorithm provides high classification accuracy with the cell tower inter-spacing lower than 500 meters which is in the magnitude of design lengths of the installed cell tower in city urban areas. This observation is also obtainable from Figure 5(b), for the k-nearest neighbor algorithm. However, the overall classification accuracy of k-nearest neighbor algorithm is lower than that of random forest algorithm. The classification accuracy from the support vector machine is shown in Figure 5(c). The support vector machine algorithm provides high classification accuracy just in the case of 100 meters cell tower inter-spacing. In that case, the cell tower should be installed in large numbers in order to achieve good classification accuracy comparable to the random forest and k-nearest neighbor algorithms.

Table 3: Confusion matrix showing percentages of classification outcomes for different algorithms

		Predicted Outcomes								
		Random Forest Algorithm			K-Nearest Neighbor Algorithm			Support Vector Machine Algorithm		
		Motorcycle	Car	Skytrain	Motorcycle	Car	Skytrain	Motorcycle	Car	Skytrain
Actual Class	Motorcycle	36.583	7.230	0.011	36.243	7.561	0.022	26.512	17.347	0.014
	Car	8.226	47.405	0.004	10.670	44.954	0.012	6.941	48.653	0.006
	Skytrain	0.050	0.026	0.462	0.044	0.027	0.0468	0.048	0.112	0.367

Table 3 shows the confusion matrix of overall classification accuracy for each algorithm. From all of the algorithms, the Skytrain is the vehicle type that is most convenient to classify with the lowest misclassification. This is because of the regularly scheduled and patternable behavior which is independent of the traffic on the road. However, the vehicle types which affect the classification accuracy by majority are the motorcycle and car. The overall classification accuracies are 84.45%, 81.66%, and 75.53% for the random forest, k-nearest neighbor, and support vector machine algorithm respectively. These classification accuracies are in overall good enough for many applications. And, as demonstrated in Figures 5(a)-5(c), the accuracies can be improved when the cell tower inter-spacing is shortened.

3.2 Vehicle Type Classification on Chula-SSS

To evaluate the classification accuracy in a more realistic road, instead of the one-dimensional road, Chula-SSS, included the calibrated cars and traffic light signal, is chosen. The motorcycles are added to Chula-SSS by the ratio of cars and motorcycles, based on the actual vehicle types of vehicles in Bangkok, and the sublane model also set in Chula-SSS for motorcycles. The Skytrains and the railways are added to Chula-SSS with the same schedule and location as in Bangkok. By adding the motorcycles and Skytrains, Chula-SSS is simulated for 30 times by using the SUMO simulator, and the data is collected by using TraCI in the interested area as shown in Figure 6.

The collected data are converted to cell dwelled time by using the same logic as in a one-dimensional road. The cell tower inter-spacing is varied from 100, 250, and 500 meters, and located along the considered street. However, with the two-directional vehicle flows, the input features are different for each vehicle flow direction, there are two input features for one cell tower represented each vehicle flow direction. After that, random forest, k-nearest neighbor,

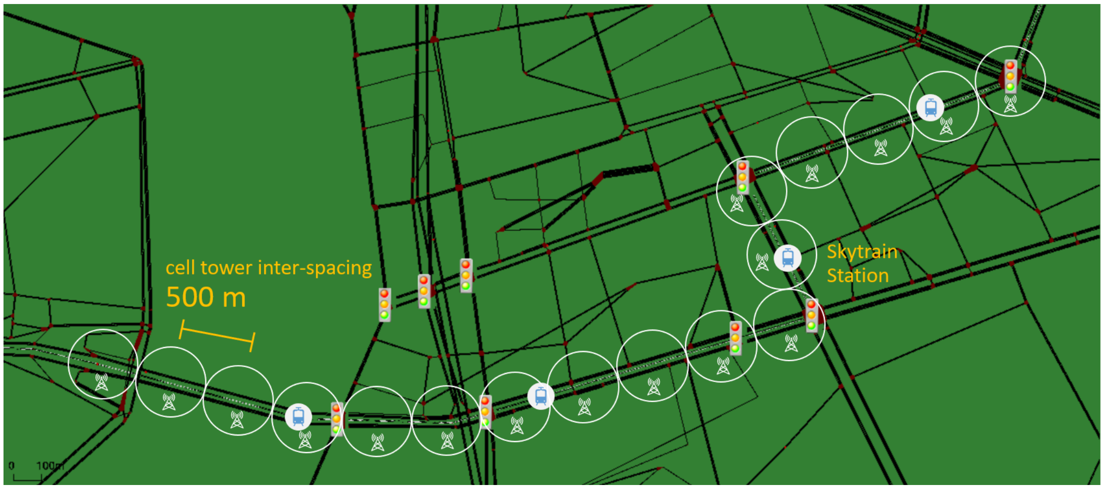


Figure 6: Chula-SSS with cell tower inter-spacing 500 meters

and support vector machine algorithms perform for vehicle type classification with the tuned hyperparameters from grid search with 5-fold cross-validation. For model evaluation, the train-test split method has been used with the splitting ratio of 80% for training data and 20% for testing data.

Table 4: Hyperparameters from brute-force optimally searching

Algorithms	Hyperparameters	Cell Tower Inter-Spacing (m)		
		100	250	500
Random Forest Algorithm	n_estimators	100	100	90
	max_features	sqrt	0.1	sqrt
K-Nearest Neighbor Algorithm	n_neighbors	7	17	17
Support Vector Machine Algorithm	C	1	1	1

The tuned hyperparameters from grid search with 5-fold cross-validation of Chula-SSS simulated data, shown in Table 4, are determined to be the hyperparameters for each algorithm for vehicle type classification. For the random forest algorithm, n_estimators is the number of trees in the forest, and max_features is the maximum number of features, bootstrapped for each tree. For the k-nearest neighbor algorithm, n_neighbors is the number of the closet neighbors using for classifying. For the support vector machine algorithm, C is the regularization parameter from a squared ridge regression.

Figure 7 shows the boxplot of each algorithm at each cell tower inter-spacing. For the random forest and k-nearest neighbor algorithms, the classification accuracy is more than 80% with the cell tower inter-spacing less than or equal to 500 meters. However, for the higher the cell tower inter-spacing value, the larger the variance of classification accuracy. For the support vector machine algorithm, with the cell tower inter-spacing less than or equal to 500 meters, the classification accuracy is in the range between 70% and 80%. From 3 considered algorithms, the random forest algorithm achieves the highest classification accuracy.

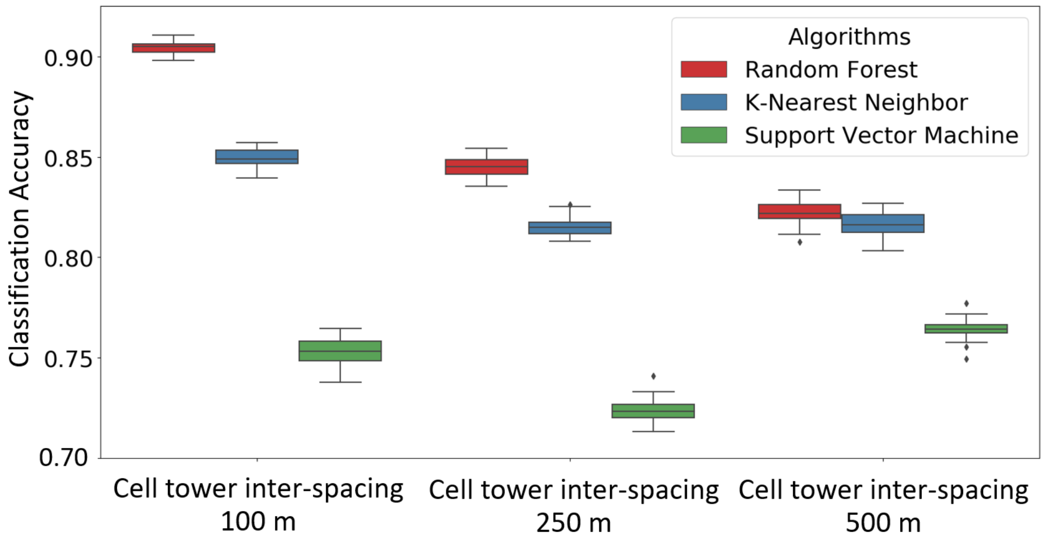


Figure 7: Boxplot of classification accuracy for each cell tower inter-spacing from each algorithm from Chula-SSS

The random forest algorithm is the most complex algorithm among the considered algorithms in this paper, with many bootstrapped trees. So, the complex relationship of the input features can be detected and used as decisions for classification. For the worst algorithm, the support vector machine algorithm with a linear kernel has the simplest logic to classify. By using the hyperplane, maximizing the distance between the hyperplane and data points in a high-dimensional axis, the linearly separated data points could be classified accurately. However, mobile phone location data in this paper are not linearly separated. So, the classification accuracy from the support vector machine algorithm with a linear kernel is worse than the others. For the k-nearest neighbor algorithm, k nearest neighbor data points are used for a majority vote to classify. With the more complex logic than the support vector machine algorithm with a linear kernel, the classification accuracy from the k-nearest neighbor algorithm is higher. Compared to the random forest algorithm, the classification accuracy is lower, because the random forest algorithm can capture more complex information from the input features.

4 Conclusion

This paper is concerned with the computer simulation study in an attempt to classify the vehicle types of car, motorcycle, and Skytrain. The location data of mobile phones simulated from SUMO are processed and submitted to 3 standard machine learning algorithms i.e. random forest, k-nearest neighbor, and support vector machine algorithms. In Bangkok traffic congestion evaluation problems, the Skytrain could be reckoned as the data noise because Skytrains do not affect the on-surface road traffic states to be observed. This paper shows that such noises can be distinguishable with ease. The results show also that the classification accuracy is highly influenced by the cell tower inter-spacing. Particularly, the lower inter-spacing, the higher classification accuracy. Based on the simple one-dimensional road network setting, the results suggest that the random forest and k-nearest neighbor algorithms could be effectively utilized with the cell tower inter-spacing as large as 500 meters. On the other hand, the support vector machine algorithm can be applied accurately with the cell tower inter-spacing that must be as short as 100 meters. In addition, based on Chula-SSS [1] dataset, the classification accuracy is lower than the one-dimensional road network. The random forest, k-nearest neighbor, and support vector machine with a linear kernel algorithms are descendingly ordered by the complexity. Due to the finding that the more complex the algorithm, the higher the classification accuracy, the random forest, k-nearest neighbor, and support vector machine with a linear kernel algorithms are descendingly ordered in classification accuracy. In practice, the existing cell tower inter-spacing is typically in the order of 500 meters for the current 4G mobile phone technology. And the cell tower inter-spacing can be designed as low as 100 meters for the upcoming 5G technology. Hence, this SUMO-based finding suggests a clear future potential of using the ubiquitously available mobile phone data for the vehicle type classification and other subsequent travel information extraction analysis in the future. To explore more detail about our work, all source codes and datasets are publicly available a `https://github.com/IoTcloudServe/smart-mobility-chula`.

5 Acknowledgements

This research is financially supported by the Asi@Connect's Data-Centric IoT-Cloud Service Platform for Smart Communities (IoTcloudServe@TEIN) project with grant contract ACA 2016/376-562 under the umbrella of Smart-Mobility@Chula demonstration site. And the authors would like to thank for technical helps from Mr. Meechai Homchan.

References

[1] C. Aswakul, S. Watarakitpaisarn, P. Komolkiti, C. Krisanachantara, and K. Techakittiroj, "Chula-sss: Developmental framework for signal actuated logics on sumo platform in over-saturated sathorn road network scenario," *EPiC Series in Engineering*, vol. 2, pp. 67–81, 2018.

[2] M. G. Demissie, G. H. de Almeida Correia, and C. Bento, "Intelligent road traffic status detection system through cellular networks handover information: An exploratory study," *Transportation research part C: emerging technologies*, vol. 32, pp. 76–88, 2013.

[3] J. Xing, Z. Liu, C. Wu, and S. Chen, "Traffic volume estimation in multimodal urban networks using cell phone location data," *IEEE Intelligent Transportation Systems Magazine*, vol. 11, no. 3, pp. 93–104, 2019.

[4] M. Lewandowski, B. Płaczek, M. Bernas, and P. Szymała, "Road traffic monitoring system based on mobile devices and bluetooth low energy beacons," *Wireless Communications and Mobile Computing*, vol. 2018, 2018.

[5] M. Lv, L. Chen, X. Wu, and G. Chen, "A road congestion detection system using undedicated mobile phones," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 6, pp. 3060–3072, 2015.

[6] N. Caceres, L. M. Romero, F. G. Benitez, and J. M. del Castillo, "Traffic flow estimation models using cellular phone data," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 3, pp. 1430–1441, 2012.

[7] A. Janecek, D. Valerio, K. A. Hummel, F. Ricciato, and H. Hlavacs, "The cellular network as a sensor: From mobile phone data to real-time road traffic monitoring," *IEEE transactions on intelligent transportation systems*, vol. 16, no. 5, pp. 2551–2572, 2015.

[8] S. Jiang, J. Ferreira, and M. C. Gonzalez, "Activity-based human mobility patterns inferred from mobile phone data: A case study of singapore," *IEEE Transactions on Big Data*, vol. 3, no. 2, pp. 208–219, 2017.

[9] J. Martín, E. J. Khatib, P. Lázaro, and R. Barco, "Traffic monitoring via mobile device location," *Sensors*, vol. 19, no. 20, p. 4505, 2019.

[10] P. A. Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y.-P. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, and E. Wießner, "Microscopic traffic simulation using sumo," in *The 21st IEEE International Conference on Intelligent Transportation Systems*, IEEE, 2018.

[11] M. Semrau and J. Erdmann, "Simulation framework for testing adas in chinese traffic situations," *SUMO 2016–Traffic, Mobility, and Logistics*, vol. 30, pp. 103–115, 2016.

[12] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[13] D. W. Aha, D. Kibler, and M. K. Albert, "Instance-based learning algorithms," *Machine learning*, vol. 6, no. 1, pp. 37–66, 1991.

[14] B. Schölkopf, A. J. Smola, R. C. Williamson, and P. L. Bartlett, "New support vector algorithms," *Neural computation*, vol. 12, no. 5, pp. 1207–1245, 2000.

[15] X. Ge, H. Cheng, M. Guizani, and T. Han, "5g wireless backhaul networks: challenges and research advances," *IEEE Network*, vol. 28, no. 6, pp. 6–11, 2014.

[16] S. K. Singh, P. Komolkiti, and C. Aswakul, "Impact analysis of start-up lost time at major intersections on sathorn road using a synchro optimization and a microscopic sumo traffic simulation," *IEEE Access*, vol. 6, pp. 6327–6340, 2017.

[17] "Simulation/SublaneModel." `https://sumo.dlr.de/docs/Simulation/SublaneModel.html/`. Accessed: 2020-01-01.

[18] "Statistics of Vehicles in Bangkok." `https://web.dlt.go.th/statistics/`. Accessed: 2019-11-11.

[19] D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker, "Recent development and applications of SUMO - Simulation of Urban MObility," *International Journal On Advances in Systems and Measurements*, vol. 5, pp. 128–138, December 2012.