Pak. J. Engg. Appl. Sci. Vol. 27 July, 2020 (p. 67–77)

Improving the Accuracy of Travel Mode Detection for Low Data Collection Frequencies

Muhammad Awais Shafique[1*], Eiji Hato[2]

1. *Department of Civil Engineering, University of Central Punjab, Lahore, Pakistan*
2. *Department of Civil Engineering, The University of Tokyo, Japan*

* **Corresponding Author:** Email: awais.shafique@ucp.edu.pk

Abstract

Smartphones are a necessary part of our daily lives, and are equipped with a range of sensors. These two characteristics make them a much preferred data collection device, when compared with other wearable sensors and devices. However, due to huge amount of data collected, issues like processing cost and battery consumption are limiting factors. In order to tackle these issues, the current study aims at achieving acceptable detection accuracy while decreasing the data collection frequency, hence reducing processing cost and battery consumption. To begin with, several classification algorithms are compared. Results suggest that boosted decision tree provides the highest accuracy closely followed by random forest; however, random forest is preferred because it requires less processing time. Detection accuracies are calculated at various data collection frequencies, and subsequently improved by addressing the issue of imbalanced data with the introduction of weighted random forest. Further improvement is achieved by applying a two-step post-processing method. Overall accuracy for 0.2 Hz frequency data is improved from 94.98% to 98.78%, whereas for 0.067 Hz frequency, the increase is from 89.16% to 95.40%. Accuracy drop of 3.42% from 0.2 Hz to 0.067 Hz is tolerable because it results in 81.96% decrease in processing time.

Key Words: Classification, Post-Processing; Smartphone, Travel Mode, Weighted Random Forest

1. Introduction

Travel related data is collected by two broad methods. The first method relies on the memory of the respondent wherein the respondent is asked to answer some questions regarding daily travelling. This approach has been in practice for a long time and is still being applied in many countries around the world. With the passage of time, improvements have been incorporated like using travel diaries, telephone and internet for data collection. Despite the widespread usage of this method, it has some inherent drawbacks. The root of the problem is the reliance on memory of the respondent. It leads to incorrect recording of starting and ending times of the individual trips as well as skipping of small trips. Another problem is the low response rate primarily due to the large number of questions to be answered, which is hectic and time-consuming.

To address the drawbacks of conventional data collection method, a second method is widely investigated in which the information is automatically recorded by devices. These devices can either be installed at fixed locations or can be carried around by the respondents. Experiments have been conducted using Global System for Mobile (GSM) communications [1, 3, 23], local area wireless technology (Wi-Fi) [14], Global Positioning System (GPS) [24], accelerometer [2, 19] and smartphones [15, 18]. Smartphones are used recently for collection of travel related data because of the integration of sensors like GPS, accelerometer and gyroscope, and due to its increasingly high penetration rates among countries. Almost the same methodology is followed by all the researchers exploring the scope of smartphones for mode prediction. To start with, sensors' data is collected with the help of smartphones. This raw data is then used to extract meaningful features that are fed to a classification algorithm for training and subsequent testing or prediction Table 1 summarizes some of the past studies that utilized data collected by one or multiple sensors like accelerometer, GPS and gyroscope, for travel mode detection. It is evident from the table that most of the previous studies selected high data collection frequencies yielding huge amounts of data. More the data, better the algorithm will be, but it will result in more processing cost and will also affect the battery time of smartphones. If the same or slightly compromised detection accuracy can be achieved at a lower data collection frequency, the said problems can be reduced. The current study aims to achieve this goal.

Over the years, many classification algorithms have been developed, and many among them, have been applied in the field of travel mode detection. For example, Neural Network [5, 8], Bayesian Network [13, 30], Decision Tree [18, 30], Support Vector Machine [17, 29, 30], Random Forest [19] etc.

The current study can be divided into two parts. The first part compares various popular algorithms reported in literature in order to ascertain the one best suited for travel mode prediction from data collected through smartphones. The comparison is done by taking two criteria into account, accuracy and computational time.

The second part deals with refining the mode detection methodology in order to improve the detection accuracy at low data collection frequency. The objective is to enable the collection of sensors' data at low frequency without compromising the accuracy of the results. This will save battery-time of the smartphone and will yield lesser data points hence reducing the computational cost of the whole process. It can be argued that since smartphones are charged daily, even on the go, there is no need to save battery-time by reducing data collection frequency.

While this is true when respondents are handpicked and given this particular task, on a much larger scale it would be difficult to convince general public to run a data collection application, even for a single day, that affects battery time. Studies with similar objectives have been carried out by other researchers as well [4]. This is the extension of our previous work [20], which only studied the change in data collection frequency.

Table 1: Selected studies using Accelerometer, GPS and Gyroscope

Study	Algorithm(s)	Sampling frequency (Hz)	No. of Participants	Accuracy (%)
[15]	Support Vector Machine	50	4	93.88
[8]	Neural Network	0.25	-	91.23
[26]	Decision tree Naïve Bayes k-Nearest Neighbor Support Vector Machine	36	4	90.6
[16]	Naïve Bayes Support Vector Machine	38	-	97
[18]	Decision Tree followed by discrete Hidden Markov Model	1	16	93.6
[11]	Decision tree Logistic regression Multilayer Perceptron	20	29	91.7
[9]	AdaBoost	60, 100	16	84.9
[28]	Support Vector Machine AdaBoost Decision tree	30	74	92.5
[25]	Support Vector Machine	50	18	96.31
[22]	Rule-based algorithm	1	30	82.05
[31]	Chained Random Forest	-	12	93.8
[7]	Multinomial Logit Model	1	1967	90

2. Data Collection and Preparation

Smartphones were used by 50 participants from Kobe city, Japan, to collect travel data over a period of one month, while using six different modes of transportation namely walk, bicycle, car, bus, train and subway. The collected data consisted of accelerometer and gyroscope readings. It was not compulsory for the participants to collect data for the entire month. Consequently, the number of days for data collection varied among the participants ranging from one day to several days and even the whole month. This explains the relatively small amount of data used in this study, in contrast to the amount expected from a month long collection process. For each trip recorded, the actual mode of transportation used was registered, by the participants, using the smartphone application. The saved data was verified by recall surveys and rechecked with the help of web generated travel maps. After verification, the collected data served as the ground truth for assessing the performance of the classification methodology developed later. The sensors' data was recorded at an average frequency of 14 Hz, which was scaled down to a uniform 10 Hz frequency. Table 2 provides the amount of data and the number of trips for each mode, used in this study.

2.1 Data Frequency

The data frequency was further decreased from 10 Hz in order to check the effect of low frequency on the accuracy of the methodology. The original frequency of 10 Hz (0.1 sec) was reduced to 4 Hz (0.25 sec), 2 Hz (0.5 sec), 1 Hz (1 sec), 0.5 Hz (2 sec), 0.33 Hz (3 sec), 0.25 Hz (4 sec), 0.2 Hz (5 sec), 0.167 Hz (6 sec), 0.143 Hz (7 sec), 0.125 Hz (8 sec), 0.111 Hz (9 sec), 0.1 Hz (10 sec), 0.091 Hz (11 sec), 0.083 Hz (12 sec), 0.077 Hz (13 sec), 0.071 Hz (14 sec) and 0.067 Hz (15 sec). The frequency could not be decreased any further for the given dataset, as the resulting amount of data, available for computation, would be substantially low. In the previous study, the frequency was decreased down to 0.2 Hz only [20]. IPhone 5s was used to understand the strength of relationship between battery usage and data collection frequency. The battery was completely consumed in approximately 3 hr. 23 min. for 10 Hz frequency, whereas it took roughly 11 hr. 32 min. for 0.067 Hz frequency. For the intermediate frequencies, battery time will most likely vary non-linearly with a steep increase in battery consumption near to high frequency end.

2.2 Pre-Processing

From the accelerations along three axes, magnitude of resultant acceleration was calculated [16, 22, 25, 27], using the following equation (1). This was done to incorporate any position in which the smartphone might be stored during travelling.

$$Acc_R = \sqrt{{Acc_X}^2 + {Acc_Y}^2 + {Acc_Z}^2} \qquad (1)$$

Following the result concluded by Shen and Stopher [21] after testing dwell times ranging from 15 seconds to 120 seconds, a dwell time of 60 seconds was used to segregate the data into trips [20]. This was done for all the data collection frequencies. The trip segmentation was not intended to be in strict accordance with the actual trips recorded.

One trip by a single mode might be broken down into two or more individual trips when trip segmentation is applied. The idea was to divide the data into trips covering only a single mode each, when no information is available regarding the actual trip lengths. This is particularly important when the data for the entire day is uploaded onto the server and the resulting Excel CSV file does not contain any information regarding trip segmentation. Note that trip segmentation was done for already known trips. The various trips recorded during the survey were known but they were treated as a continuous stream of readings and trip segmentation was performed to divide the data into probable trips.

Table 2: Amount of data recorded by smartphones

Mode	No. of trips	Total Time (hours)	Amount of data instances
Walk	442	144	5,186,095
Bicycle	10	9	326,500
Car	31	14	500,410
Bus	21	11	381,698
Train	45	18	659,528
Subway	10	7	236,738
Total	559	203	7,290,969

2.3 Feature Extraction

A moving window size of 10 minutes was used to calculate additional features (see [20]). From resultant acceleration, six features were extracted namely maximum resultant acceleration, average resultant acceleration, maximum average resultant acceleration, standard deviation, skewness and kurtosis. Apart from these, pitch and roll recorded by gyroscope were also used. Instead of combining these readings with the accelerometer data, it was expected that the algorithm would develop a relationship for them. Therefore, the gyroscope readings were used directly as features in the algorithm, without any processing done.

2.4 Amount of Learning Data

Stratified bootstrapped sampling was used to divide the collected data into 10 almost equal parts. This means that each part contained 10% of data from each mode randomly selected without replacement. Normally, in a 10-fold cross-validation, nine folds are used to test the remaining one fold on each run. On the contrary, this study applies cross-validation in a way that on each run one part of the data is used to train the algorithm and the remaining nine parts are then tested. The aim was to use only 10% of data for training purpose [20]. This was the reason for dividing the data into ten parts and further using one part instead of nine during cross-validation. At the end of 10 runs, each part has been used once to predict the remaining parts. The average is then calculated and reported as the result. The results reported hereafter are all 10-fold cross-validated.

3. Classification Algorithm

The first task was to determine the appropriate classification algorithm to be used for mode detection. The algorithms to be compared were selected based on their repeated use by other researchers and their good performance in numerous comparative studies [19]. A comprehensive comparison among the algorithms is provided in Table 3. Here it can be seen that boosted decision trees provide the highest prediction accuracy but falls behind in terms of computational time. Although, the accuracy achieved by random forest is slightly lower than by boosted decision trees, the computation is very quick making it a better option, especially when the data is huge. Decision trees are very quick but the prediction is not very accurate. Support Vector Machine (SVM) is the most time-consuming classifier, with accuracy even lower than decision trees. Neural network and Naïve Bayes come last in the list. These results are specific to the dataset used and might change for different datasets. It is clear from the results that for this particular dataset, the performance of tree-based algorithms is much better than the other algorithms. Reasons include robustness, suitability for large datasets and simplicity. They do not require the data to be normalized.

4. Detection Accuracy for Various Data Collection Frequencies

After it was established that random forest was a better algorithm for the collected data, it was employed to calculate the detection accuracies for the range of data collection frequencies discussed earlier. Table 4 provides all the results. It can be seen that for high frequency, the detection accuracy is impressively high but as the frequency is decreased, the accuracy also decreases. High frequency means that the battery of smartphone will be exhausted quickly. In order to make the methodology acceptable, low data collection frequency should be used. Therefore, the methodology was modified to refine the results further for low frequencies, as discussed in subsequent sections.

5. Problem of Imbalanced Data

The collected data is highly imbalanced when it comes to individual modes. More than 70% of the data is associated with only one mode i.e. walk. Due to this imbalance, the predicting ability of random forest becomes bent more towards the majority class. Consequently, the prediction accuracy for that class is relatively higher than other classes, and even worse, is the decrease in accuracy of other classes due to misclassification as the majority class. Two methods were employed to improve the detection accuracy by addressing the problem of imbalanced data.

5.1 Down Sampling

In random forest, a subset of the training data is randomly selected with replacement, to grow each tree. Due to highly imbalanced data, it is likely that the data used for growing a certain number of trees lack representation from all the classes. In some others, the representation by minority classes might not be adequate. All this will result in poor decision trees being formed.

These poor trees will mostly classify the test data in favor of the majority class and will therefore drive the voting system to misclassify the data. The problem of imbalanced data has been tackled by various researchers.

Table 3: Comparison among Classification Algorithms

Mode	Prediction Accuracy (%)					
	Naïve Bayes	**Neural Network**	**Support Vector Machine**	**Decision Tree**	**Boosted Decision Tree**	**Random Forest**
Walk	64.89	95.85	99.29	94.74	98.98	99.25
Bicycle	63.00	58.17	54.18	69.90	80.72	78.07
Car	13.29	31.23	43.70	48.29	70.80	64.68
Bus	75.56	51.65	64.36	62.42	75.07	72.05
Train	1.45	23.18	38.73	46.33	63.43	54.51
Subway	10.45	33.37	43.42	49.20	63.77	54.32
Overall	54.49	79.07	84.57	83.07	90.75	89.16
Time (sec)	15.25	13.64	61.59	0.49	65.71	5.4

In one study, authors selectively down-sized the majority class by using one-sided sampling technique [10]. In another study, over-sampling of the minority class by replication was done, to attain data size comparable with the majority class Ling and Li [12]. Both down-sampling and over-sampling were incorporated in a study to achieve better classification results Chawla, et al. [6].

As it is already mentioned that in random forest, under default conditions, about 63% of the learning data is randomly selected each time a tree is grown. Therefore, if the amount of sampling is reduced for any class, the information is not lost. For instance, if the sampling amount for the majority class is reduced from 63% to 30%, even then the random sampling will be done from the entire dataset, each time a decision tree is grown. Therefore, down-sampling the majority class is a suitable option for random forest. A threshold value of mean (1162 for 15 Hz) was used for down sampling. Any mode having data amount less than the threshold value was 100 % taken for each tree whereas modes having data amounts greater than the threshold value were randomly selected equal to the threshold value.

5.2 Weighted Random Forest

In the second method, Random Forest was modified a little to accommodate, to some extent, the imbalanced data on its own. Random forest was applied to the imbalanced data as usual but the result was not concluded from the usual voting procedure. Rather, the probabilities of each mode were multiplied with the weights, computed for each mode from the distribution of data, to attain the weighted probabilities. The voting was then done using the weighted probabilities and the mode having the maximum probability was concluded as the final prediction.

The weights (W) depend on the data size of each class used to train random forest and were calculated as follows.

$$W_i = 0.5 + \frac{D_{min}}{D_i} \qquad (2)$$

Where

W_i = Weight for class i

D_i = Data size of class i

D_{min} = Minimum data size among all classes

5.3 Detection Results

Table 5 provide the detection results after applying down sampling, and using weighted random forest respectively. It is evident that weighted random forest performs better than down sampling.

6. Post-Processing

A 2-step post-processing technique was introduced. The first step was same as the error correction by voting suggested by Yu, et al. [28]. Rather than using four as the upper bound value, five was found to be a better value for the collected data.

Table 4: Detection Accuracy (%) for various Data Collection Frequencies

Mode	Time interval between readings (sec)								
	0.1	**0.25**	**0.5**	**1**	**2**	**3**	**4**	**5**	**6**
Walk	99.99	99.98	99.96	99.91	99.84	99.76	99.67	99.63	99.58
Bicycle	99.95	99.81	99.52	98.82	97.14	95.39	93.65	89.71	89.56
Car	99.84	99.68	99.04	97.75	95.22	92.33	89.15	85.97	83.88
Bus	99.92	99.73	99.16	97.45	94.28	91.95	87.34	87.21	84.93
Train	99.85	99.51	98.84	96.51	92.2	86.88	83.17	78.94	74.72
Subway	99.72	99.21	98.25	95.01	89.79	84.35	78.42	74.83	71.51
Overall	99.95	99.87	99.68	99.13	98.11	97.01	95.89	94.98	94.19
Time	6975.6	2430.59	982.76	438.65	206.53	136.31	98.9	78.29	65.35
Mode	**Time interval between readings (sec)**								
	7	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
Walk	99.52	99.48	99.38	99.41	99.3	99.32	99.33	99.13	99.25
Bicycle	87.96	85.57	83.28	85.73	81.9	80.82	81.01	81.82	78.07
Car	81.1	78.77	76.44	73.72	72.53	70.23	68.63	68.36	64.68
Bus	82.83	80.84	78.47	78.6	77.06	75.71	74.29	73.54	72.05
Train	71.22	69.13	65.95	62.98	61.06	59.39	57.78	56.51	54.51
Subway	67.58	64.45	61.65	60.61	61.53	58.73	56.86	58.37	54.32
Overall	93.35	92.66	91.85	91.49	90.96	90.46	90.09	89.87	89.16
Time	55.82	48.12	42.12	37.86	33.82	30.84	28.41	26.03	24.15

In the second step, the travel mode predicted for the maximum instances within each trip is assigned to all the instances in that trip. This is done assuming that within each trip, the travel mode is not changed. This way further refinement is induced. In order to apply both steps, the data has to be in the original order. Therefore, the training data was also included as it was randomly selected from the collected data. The post-processing methodology is explained in Table 6 by using an example data of two trips covering only two modes i.e. car and bus. In the example, when predictions from the classification algorithm are compared with the ground truth already available, it is revealed that the algorithm misclassifies four instances in total, two in each trip. During the first step, votes are calculated in a way that for each predicted mode, one vote is added (if total votes for that mode are less than 5) and one vote is subtracted from the other mode (if total votes are greater than 0). The mode with the maximum votes is taken as the corrected prediction at each instance. The misclassifications are reduced from four to two. In the second step, the majority mode in a trip is assigned to all the instances of that trip. Table 7 provides the results for weighted random forest followed by post-processing.

7. Trip-wise Results

The results provided in table 7 demonstrate point accuracy. When translated into trip accuracy, it is observed that out of 639 segmented trips (different from recorded trips - 559) 10 trips were not correctly divided by 60 second dwell time criteria and consequently covered more than one mode.

Table 5: Detection Accuracy (%) after applying Down Sampling, and Weighted Random Forest

Down Sampling											
Mode	**Time interval between readings (sec.)**										
	5	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
Walk	96.23	96.18	95.31	94.63	93.42	93.64	92.73	92.25	92.36	91.81	91.31
Bicycle	95.06	94.88	93.75	93.43	91.77	92.06	90.62	90.78	89.23	90.47	86.54
Car	93.82	92.80	92.01	90.76	89.65	88.30	87.27	87.19	85.45	85.62	83.85
Bus	92.98	91.89	90.63	89.28	87.30	88.00	86.96	85.94	84.57	84.26	83.25
Train	91.00	89.15	87.70	86.27	84.37	82.40	81.62	79.61	79.17	77.71	76.43
Subway	86.33	85.10	81.39	79.51	76.03	76.00	76.22	73.37	74.73	72.27	71.10
Overall	95.07	94.69	93.65	92.81	91.42	91.35	90.46	89.79	89.61	89.06	88.21
Time	66.1	54.5	46.7	40.2	35.5	32.5	29.6	27.9	25.3	22.9	21.4
Weighted Random Forest											
Mode	**Time interval between readings (sec.)**										
	5	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
Walk	98.66	98.59	98.38	98.09	97.79	98.01	97.51	97.5	97.56	97.21	97.18
Bicycle	95.18	94.24	93.48	92.39	90.22	91.13	89.32	88.16	88.31	89.32	85.64
Car	91.59	89.93	87.69	86.31	85.51	82.90	81.28	80.40	78.66	78.70	76.06
Bus	92.12	90.54	89.00	87.97	85.16	86.23	84.02	83.18	80.78	80.57	80.47
Train	84.76	81.76	78.77	76.86	74.26	71.59	68.88	67.80	66.89	63.83	63.20
Subway	87.93	84.79	82.09	79.62	75.91	75.41	75.49	73.84	73.04	70.91	70.60
Overall	96.10	95.45	94.68	94.03	93.17	92.99	92.09	91.78	91.49	90.94	90.51
Time	653.2	466.4	352.3	273.9	226.7	189.6	162.6	140.9	124.3	110.7	99.2

Due to second step of post-processing, these 10 trips were identified to belong to the majority mode and hence some percentage of misclassification was introduced.

Apart from the 10 problematic trips, the rest 629 trips are shown in Table 8. Due to 10-fold cross-validation, the entire data was predicted 10 times, each time using a different 10% training data. Due to this methodology, a slight misclassification can tilt the balance and result in misclassification of the entire trip. Consequently, the table shows the number of folds or the number of cycles the trip was correctly predicted.

A prominent difference between results for 5 second data and 15 second data can be witnessed where the number of trips incorrectly predicted is more in case of 15 second data. Note that a lower number of folds correctly predicted does not necessarily mean that the methodology is faulty or the prediction accuracy is nil. At the same time, it is obvious that point based accuracy alone does not fully reflect the prediction capability of the developed methodology.

Table 6: Example of Post-Processing Method

Trip No.	Ground Truth	Prediction by Algorithm	Step 1			Step 2
			Voting		Correction	
			Car	Bus		
1	Car	Car	1	0	Car	Car
	Car	Car	2	0	Car	Car
	Car	Car	3	0	Car	Car
	Car	Bus	2	1	Car	Car
	Car	Bus	1	2	Bus	Car
	Car	Car	2	1	Car	Car
	Car	Car	3	0	Car	Car
2	Bus	Bus	2	1	Car	Bus
	Bus	Bus	1	2	Bus	Bus
	Bus	Bus	0	3	Bus	Bus
	Bus	Bus	0	4	Bus	Bus
	Bus	Bus	0	5	Bus	Bus
	Bus	Car	1	4	Bus	Bus
	Bus	Car	2	3	Bus	Bus

Table 7: Detection Accuracy (%) using Weighted Random Forest and Post-Processing

Mode	Time interval between readings (sec.)										
	5	6	7	8	9	10	11	12	13	14	15
Walk	99.72	99.73	99.77	99.68	99.81	99.82	99.84	99.84	99.83	99.82	99.83
Bicycle	99.65	99.09	99.21	99.33	97.99	98.67	97.99	98.39	95.72	97.27	97.01
Car	99.26	98.69	98.65	97.57	97.10	96.03	94.80	96.29	94.01	94.88	92.66
Bus	88.81	88.30	86.80	87.95	85.69	86.15	84.99	84.87	81.59	83.16	83.01
Train	96.45	95.33	93.88	92.19	89.14	85.64	81.53	82.30	79.66	75.35	75.01
Subway	98.17	96.73	96.78	93.85	88.96	88.53	87.91	80.07	82.99	79.36	76.44
Overall	98.78	98.55	98.37	98.06	97.52	97.18	96.63	96.57	95.97	95.68	95.40
Time	1030.5	750.3	598.5	473.5	394.4	336.6	291.7	256.0	228.4	205.2	185.9

Further, a limitation of post-processing method is visible from the results. Because of taking the statistical mode for the entire trip that is pre-dominantly misclassified, the percentage of trip correctly predicted is also misclassified. This decreases the overall accuracy. Further it can be observed that lesser amount of data tends to slightly increase the misclassification, making the methodology quite sensitive to the training dataset. It is one of the limitations of this approach and will be addressed in future studies.

8. Conclusion and Future Work

A number of conclusions can be drawn from this study. To start with, selection of an appropriate classification algorithm is vital for accurate classification results. Comparison among various algorithms demonstrate that boosted decision tree provides the highest detection accuracy closely followed by random forest but the computational cost is much high when compared to random forest.

Table 8: Number of Trips correctly predicted

Time interval (sec)	Mode	Number of folds correctly predicted										
		10	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
5	Walk	503	2	2	0	0	0	1	0	0	0	0
	Bicycle	8	0	0	1	0	0	1	0	0	0	0
	Car	26	3	1	1	0	0	0	0	0	0	0
	Bus	17	0	0	0	0	0	0	3	0	0	3
	Train	30	1	1	2	2	1	1	1	0	1	2
	Subway	11	2	2	0	0	0	0	0	0	0	0
	Total	595	8	6	4	2	1	3	4	0	1	5
15	Walk	501	3	2	1	0	0	0	1	0	0	0
	Bicycle	6	2	0	0	0	0	0	0	2	0	0
	Car	15	3	0	1	2	5	1	2	0	0	2
	Bus	9	2	4	0	0	0	0	1	1	3	3
	Train	5	4	0	1	4	5	2	5	4	5	7
	Subway	3	1	1	0	3	2	1	1	1	2	0
	Total	539	15	7	3	9	12	4	10	8	10	12

Consequently, random forest is deemed more feasible. The detection accuracy is greatly affected by the data collection frequency. The accuracy decreased with decrease in frequency. Slight improvement can be achieved for imbalanced data by using weighted random forest instead of random forest.

The two-step post-processing method can further refine the results. Overall accuracy for 0.2 Hz frequency data is improved from 94.98% to 98.78%, whereas for 0.067 Hz frequency, the increase is from 89.16% to 95.40%. Due to decreasing the data frequency from 0.2 Hz to 0.067 Hz, a minute drop of 3.42% in prediction accuracy is observed. This decrease in accuracy is negligible as compared to 81.96% decrease in processing time. Trip-wise analysis suggested that apart from 10 trips incorrectly divided by 60 seconds dwell-time criteria, the mode detection accuracy for entire trips is quite reasonable with only 5 completely misclassified trips for 5 sec data and 12 completely misclassified trips for 15 sec data, out of 629 trips.

The aim of this study was to achieve a considerable amount of accuracy while decreasing the data collection frequency, which in turn would decrease the overall processing cost as well as the battery consumption of the smartphones. The precise effect of data collection frequency on battery time is not investigated in this study. This part should be explored in order to validate the hypothesis formed in this study. Variation in battery consumption in different mobiles should also be investigated. Although, the results are promising for the collected data, the methodology should be tested for other datasets too. Similarly, the developed approach should also be tested against methodologies available in literature, using a common dataset. Battery consumption can be further decreased by employing a clever mechanism to record data at varying frequencies (e.g. higher frequency when movement is detected).

9. References

[1] Anderson, I. and H. Muller (2006), "Practical Activity Recognition using GSM Data," Department of Computer Science, University of Bristol 2000563.

[2] Asakura, Y. and E. Hato (2001), "Behavioral monitoring of public transport users through a mobile communication system," *Journal of advanced transportation,* vol. 35, pp. 289-304.

[3] Asakura, Y. and E. Hato (2004), "Tracking survey for individual travel behaviour using mobile communication instruments," *Transportation Research Part C: Emerging Technologies,* vol. 12, pp. 273-291.

[4] Bieber, G., T. Kirste, and M. Gaede (2014), "Low sampling rate for physical activity recognition," in *Proceedings of the 7th International Conference on PErvasive*

Technologies Related to Assistive Environments, p. 15.

[5] Byon, Y.-J., B. Abdulhai, and A. S. Shalaby (2007), "Impact of sampling rate of GPS-enabled cell phones on mode detection and GIS map matching performance," in *Transportation Research Board 86th Annual Meeting*.

[6] Chawla, N. V., K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer (2002), "SMOTE: synthetic minority over-sampling technique," *Journal of artificial intelligence research,* vol. 16, pp. 321-357.

[7] Dalumpines, R. and D. M. Scott (2017), "Making mode detection transferable: extracting activity and travel episodes from GPS data using the multinomial logit model and Python," *Transportation Planning and Technology,* pp. 1-17.

[8] Gonzalez, P., J. Weinstein, S. Barbeau, M. Labrador, P. Winters, N. L. Georggi*, et al.* (2008), "Automating mode detection using neural networks and assisted GPS data collected using GPS-enabled mobile phones," in *15th World congress on intelligent transportation systems*.

[9] Hemminki, S., P. Nurmi, and S. Tarkoma (2013), "Accelerometer-based transportation mode detection on smartphones," in *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems*, p. 13.

[10] Kubat, M. and S. Matwin (1997), "Addressing the curse of imbalanced data sets: One sided sampling," in *Proc. of the Int'l Conf. on Machine Learning*.

[11] Kwapisz, J. R., G. M. Weiss, and S. A. Moore (2011), "Activity recognition using cell phone accelerometers," *ACM SigKDD Explorations Newsletter,* vol. 12, pp. 74-82.

[12] Ling, C. X. and C. Li (1998), "Data Mining for Direct Marketing: Problems and Solutions," in *KDD*, pp. 73-79.

[13] Moiseeva, A. and H. Timmermans (2010), "Imputing relevant information from multi-day GPS tracers for retail planning and management using data fusion and context-sensitive learning," *Journal of Retailing and Consumer Services,* vol. 17, pp. 189-199.

[14] Mun, M., D. Estrin, J. Burke, and M. Hansen (2008), "Parsimonious Mobility Classification using GSM and WiFi Traces," presented at the Fifth Workshop on Embedded Networked Sensors.

[15] Nham, B., K. Siangliulue, and S. Yeung (2008), "Predicting mode of transport from iphone accelerometer data," *Machine Learning Final Projects, Stanford University*.

[16] Nick, T., E. Coersmeier, J. Geldmacher, and J. Goetze (2010), "Classifying means of transportation using mobile sensor data," in *Neural Networks (IJCNN), The 2010 International Joint Conference on*, pp. 1-6.

[17] Pereira, F., C. Carrion, F. Zhao, C. D. Cottrill, C. Zegras, and M. Ben-Akiva (2013), "The Future Mobility Survey: Overview and Preliminary Evaluation," in *Proceedings of the Eastern Asia Society for Transportation Studies*.

[18] Reddy, S., M. Mun, J. Burke, D. Estrin, M. Hansen, and M. Srivastava (2010), "Using mobile phones to determine transportation modes," *ACM Trans. Sen. Netw.,* vol. 6, pp. 1-27.

[19] Shafique, M. A. and E. Hato (2015), "Use of acceleration data for transportation mode prediction," *Transportation,* vol. 42, pp. 163-188.

[20] Shafique, M. A. and E. Hato (2016), "Travel Mode Detection with varying Smartphone Data Collection Frequencies," *Sensors,* vol. 16, p. 716.

[21] Shen, L. and P. Stopher (2013), "Should we change the rules for trip identification for GPS travel records," in *Proceedings of the 36th Australasian Transport Research Forum ATRF*.

[22] Shin, D., D. Aliaga, B. Tunçer, S. M. Arisona, S. Kim, D. Zünd*, et al.* (2014), "Urban sensing: Using smartphones for transportation mode classification," *Computers, Environment and Urban Systems*.

[23] Sohn, T., A. Varshavsky, A. LaMarca, M. Chen, T. Choudhury, I. Smith*, et al.* (2006), "Mobility Detection Using Everyday GSM Traces," in *UbiComp 2006: Ubiquitous Computing*. vol. 4206, P. Dourish and A. Friday, Eds., ed: Springer Berlin Heidelberg, pp. 212-224.

[24] Tsui, S. Y. A. and A. Shalaby (2006), "Enhanced System for Link and Mode Identification for Personal Travel Surveys Based on Global Positioning Systems,"

Transportation Research Record: Journal of the Transportation Research Board, pp. 38-45.

[25] Xia, H., Y. Qiao, J. Jian, and Y. Chang (2014), "Using smart phone sensors to detect transportation modes," *Sensors,* vol. 14, pp. 20843-20865.

[26] Yang, J. (2009), "Toward physical activity diary: motion recognition using simple acceleration features with mobile phones," in *Proceedings of the 1st international workshop on Interactive multimedia for consumer electronics*, pp. 1-10.

[27] Yu, M.-C., T. Yu, C. Lin, and E. Chang (2013), "Low power and low cost sensor hub for transportation-mode detection," *Studio Engineering, HTC, Tech. Rep*.

[28] Yu, M.-C., T. Yu, S.-C. Wang, D. SC, C.-J. Lin, and E. Y. Chang (2014), "Big data small footprint: The design of a low-power classifier for detecting transportation modes," *Proceedings of the VLDB Endowment,* vol. 7, pp. 1429-1440.

[29] Zhang, L., S. Dalyot, D. Eggert, and M. Sester (2011), "Multi-stage approach to travel-mode segmentation and classification of gps traces," in *ISPRS Workshop on Geospatial Data Infrastructure: from data acquisition and updating to smarter services*.

[30] Zheng, Y., L. Liu, L. Wang, and X. Xie (2008), "Learning transportation mode from raw gps data for geographic applications on the web," presented at the Proceedings of the 17th international conference on World Wide Web, Beijing, China.

[31] Zhou, X., W. Yu, and W. C. Sullivan (2016), "Making pervasive sensing possible: Effective travel mode sensing based on smartphones," *Computers, Environment and Urban Systems,* vol. 58, pp. 52-59.