Passenger Transportation Analysis Using Smartphone Sensors and Digital Surveys

Arto Perttula [1*], Nhan Nguyen [1], Jussi Collin [1], Jani-Pekka Jokinen [2]

[1] Laboratory of Pervasive Computing, Tampere University of Technology, Tampere, Finland
[2] Department of Computer Science, Aalto University, Finland
[*]arto.perttula@tut.fi

Abstract: Increasing context awareness plays an essential role in developing intelligent transportation systems. In this paper the focus is on the context of the passenger, using smartphone measurements and available weather data as sensory inputs. One case example is to recognize whether the subject is inside or outside the bus. When this is recognized, the differences between bus types (diesel or electric) from the passengers' point of view are studied. This is topical as new electric vehicles are rapidly emerging in public transportation. Modern smartphones contain various sensors such as accelerometers, gyroscopes, magnetometers and barometers. They enable access to useful information regarding the mode of transport. We collected data from public buses with smartphones and simultaneously conducted digital passenger survey to merge passengers' own evaluations for travel conditions to the data. We demonstrate that context recognition using machine learning (ML) algorithms provides useful information for transportation analysis. It can be used together with digital passenger surveys to achieve deeper understanding of dependencies between travel conditions and passenger satisfaction using buses.

1. Introduction

The usage of electric vehicles is increasing at fast pace – also in public transportation. The switch from combustion engines to electric engines opens new research possibilities in the area of passenger experience studies. This is due to the different powertrain which leads to significant change in vibration and sound environment. These changes may become measurable with new sensors on smartphones. Environment changes may also affect to passengers travel experience. However, these are hypotheses that require research to quantify the effect.

Mobility as a Service (MaaS) is a subject undergoing intense study nowadays. It means that the citizens can select the best available transportation modes for each journey. For instance, for easy billing in MaaS system, it is necessary to automatically know which means of transportation people have used. For this, the context recognition of a passenger is very essential. The knowledge of context and travel experience in different transportation modes is one important part in order to design well working MaaS.

Another relevant piece of information is the amount of passengers in a given vehicles. Often this needs to be measured independently from the ticket sale process. For instance, pressure sensors in front of doors [1], [2] have been used previously for this purpose. Camera based systems are also used very widely for passenger detection [3]. In [4], the passengers entering the bus were detected using high-resolution inertial measurement unit. The sensors were located on the floor in front of the first seat row near the front door where passengers enter the bus. The problem of all these above mentioned systems is that they need infrastructure of additional devices. These will be unnecessary, if the sensors integrated to smartphones of the passengers could be used.

It is possible to get more reliable results using multiple different types of sensors for detecting the context. An up to date survey of this kind of context recognition is done by Elhoushi et al. [5]. Driver behaviour has been profiled in [6] using multiple different types of machine learning (ML) algorithms. However, the bad driving features were defined by the researchers – not by real feedback from passengers.

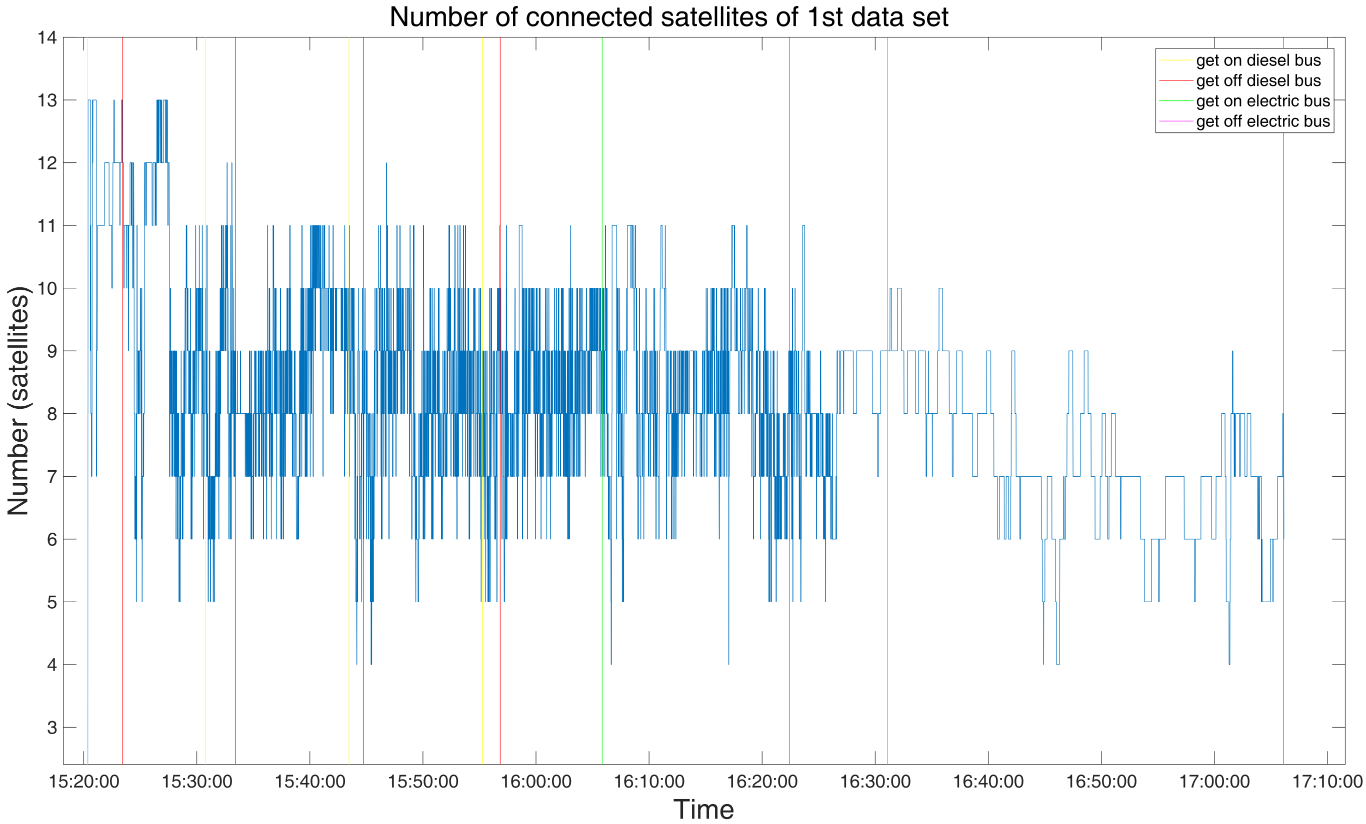

Fig. 1. *Number of connected satellites of 1st data set*

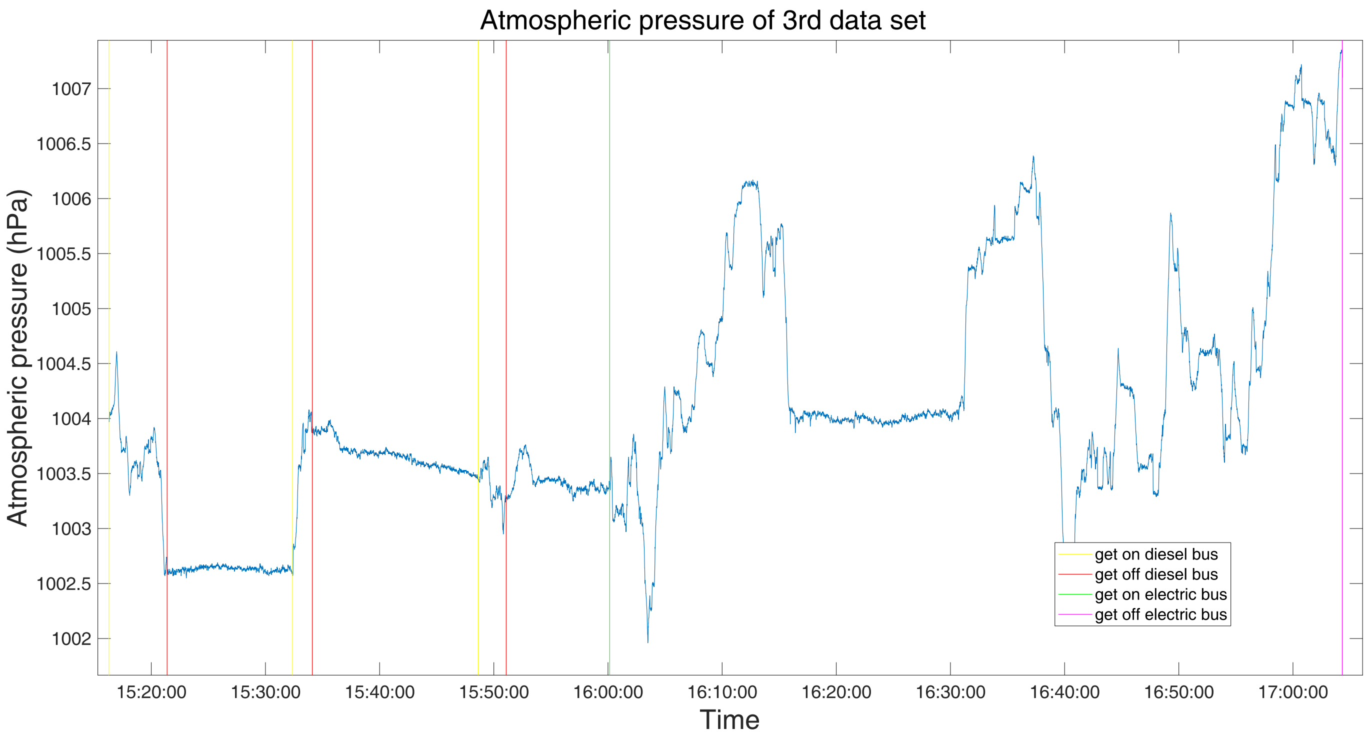

Fig. 2. *Atmospheric pressure of 3rd data set*

The Global Navigation Satellite Systems (GNSS) receivers has been also used to recognize whether the user is onboard or offboard. Carrier to noise ratio is relatively good indicator in this. This approach for onboard/offboard detection has been used earlier for instance in [7] and [8]. Carrier to noise ratio can be measured with Android phones using GnssLogger application [9].

In this paper we studied the usage of Android based smartphone to collect data to recognize the context of a passenger in public city bus. The integrated sensors on smartphone were selected because they can be found from pockets of almost every passenger. Additionally, similar sensors are often attached to vehicles. The data were collected using AndroSensor application. AndroSensor was used to collect the raw data from available sensors in the phone. The first stage was to recognize whether the passenger is inside or outside of the bus. When the passenger was recognized to be on board, the aim was to recognize whether (s)he is traveling with electric or traditional diesel bus. The sensors used for recognition can be found from almost all modern smartphones. Additionally, historical weather data from cloud services were used.

There are two case examples in this paper. The first one is the distinction between diesel and electric bus which is interesting since the assumption is that the travel experience might be different in electric bus than in diesel bus. As mentioned earlier, the powertrain is very different, i.e., vibration profile changes which can be measured with for example with accelerometers and gyroscopes. Torque delivery is different in electric vehicles and this affects the acceleration. Furthermore, the sound environment changes which can be also measured with sensors on smartphone. It is also interesting to link the sensor measurements performed with smartphone to real passenger experiences, thus it is expected that less vibration and more silent environment enhances the traveling comfort. To obtain reference data of real user experience, we conducted simultaneously with sensor measurements a digital survey in which evaluations for smoothness of driving, noise level, and travel experience during the bus trip were asked from the passengers. The other focus is hereby in analyzing the relationship between evaluated smoothness of driving and sensor data with ML algorithms, i.e., the evaluated smoothness according to responses in the survey is the target that is tried to predict using the features based on the sensor measurements.

In the future, the aim is to develop a classifier that can be run in real-time. The idea is that smartphone applications can be developed to utilize the results of this classifier.

The rest of the paper is organized as follows. In Section 2 we describe the data collection procedure and how the data was processed. The results and analysis of the context recognition and user survey can be found from Section 3. Section 4 contains conclusion as well as ideas for future work.

Table 1 Sensor measurements with AndroSensor

Sensor	Dimensions	Unit
Accelerometer	3	m/s^2
Gyroscope	3	rad/s
Light	1	lux
Magnetometer	3	µT
Barometer	1	hPa
Sound	1	dB
GNSS Latitude	1	deg
GNSS Longitude	1	deg
GNSS Altitude	1	m
GNSS Speed	1	km/h
GNSS Accuracy	1	m
GNSS Orientation	1	deg
GNSS Satellites (connected/visible)	1	number

2. Data Collection and Processing Methods

2.1. Data Collection

The data from the smartphone were collected by the field researchers from buses of line 23 in Helsinki – few hours on each day in 15 different days in May 2017. Some buses on that line are electric buses and some are traditional diesel buses, thus we got data from both bus types in the same route. The number of diesel and electric buses are shown in the description of the used data sets in this subsection. The data collection with AndroSensor was done using 2 Hz sampling rate. The collected data is shown in Table 1. The used sampling frequency was quite low. However, the reason for this was that data collection periods were very long and the application was defined as stable with this frequency. The processed results are shown here for 2 days – May 9th and May 10th. The results for the other days are similar to these two days. Each day has 2 data sets, hence, there are four complete data sets. The recording times are the following:

- 1st data set: from 15:00 to 17:00, May 9th 2017. This trip had 4 diesel buses and 2 electric buses.
- 2nd data set: from 18:15 to 20:45, May 9th 2017. This trip had 3 diesel and 3 electric buses.

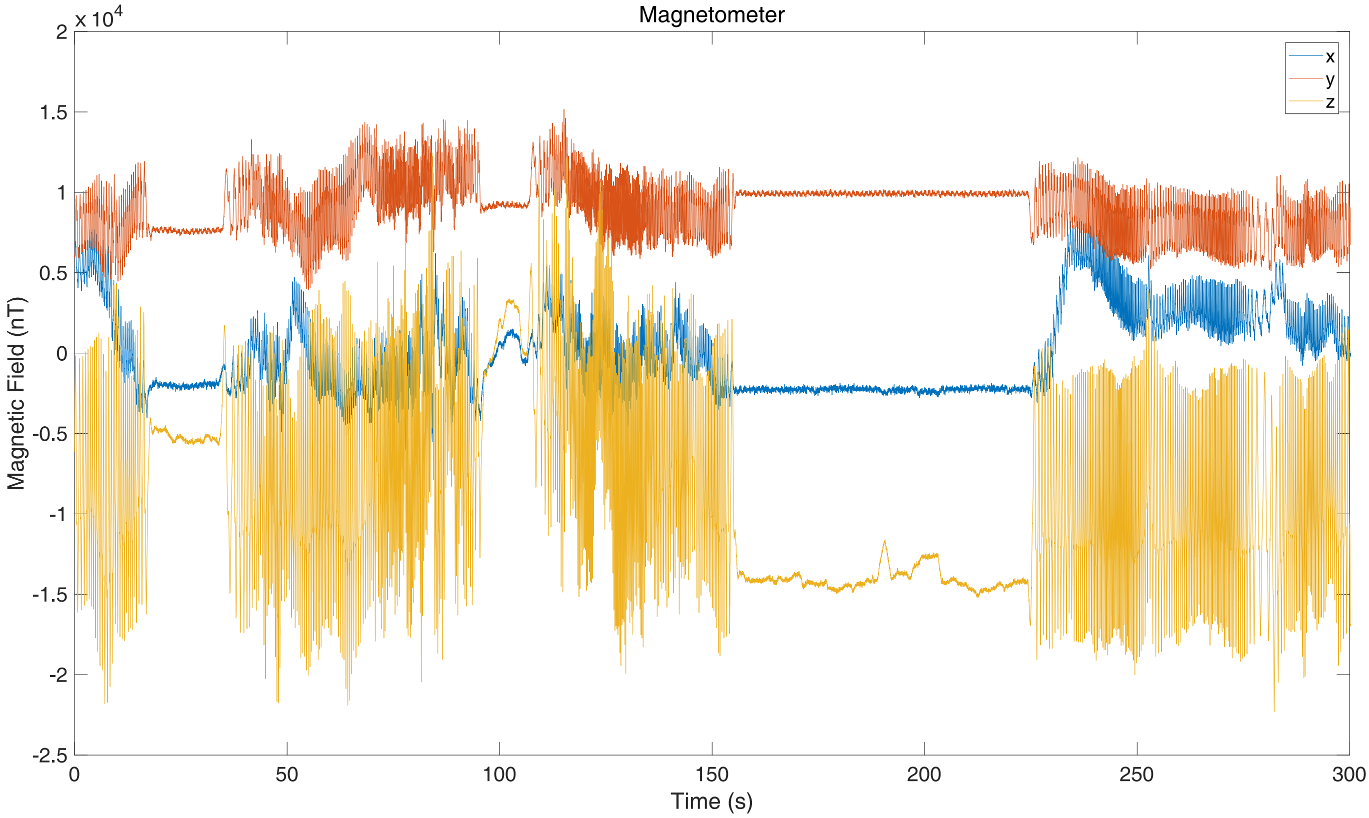

***Fig. 3.** Magnetometer measurements from electric bus*

- 3rd data set: from 15:00 to 17:00, May 10th 2017 with 3 diesel and 1 electric buses.
- 4th data set: from 18:15 to 20:45, May 10th 2017 which had 6 diesel and 1 electric buses.

Since there are data sets from same time interval on different days, the correlation in data at same time interval but on different days can be examined.

There are multiple sensors inside modern smartphones. Raw data of those sensors was analyzed first to find the potential of generating features. For instance, number of connected satellite of 1st data set is shown in Fig. 1 and the atmospheric pressure of 3rd data set in Fig. 2.

The passenger survey was conducted at the same time with the sensor recordings. The field researchers in the buses distributed flyers to the passengers. The flyers included invitation and link to the survey enabling passengers to answer immediately by smartphone or other devices. There were 2500 flyers distributed, 356 answers were received. The response rate was normal compared to other passenger surveys conducted by the Helsinki Region Transport.

In order to classify the context recognition and passengers' experience, Statistics and Machine Learning Toolbox in Matlab was used [10]. The used classifiers were Tree Ensemble AdaBoost (TEA), Tree Ensemble GentleBoost (TEG), Decision Tree (DT), K-nearest neighbors (KNN), Naive Bayes (NB), and Discriminant Analysis (DA). The following section describes the procedure of creating features and responses for each prediction case.

Algorithm 1 Utilizing Discrete Fourier Transform

Input: Raw data from AndroSensor
Output: Relevant variables for prediction

1. Apply Discrete Fourier Transform (DFT) to consecutive raw data samples in 30 seconds
2. Record magnitude of each DFT value
3. The greatest, 2nd greatest, 3rd greatest, the smallest, and 2nd smallest magnitudes as well as the corresponding frequencies of each are chosen as features

Algorithm 2 Temperature Linear Model

Input: Weather data measured at 2 weather stations
Coordinates of each station
Coordinates of the place where to estimate temperature

Output: Temperature of the place in question

1. Create the linear distribution of temperature on the straight line connecting 2 weather stations.
2. Normalize the coordinates of the place onto that line
3. Temperature of the orthogonal point on the line is the temperature of the place in question

Note: The further the place is from the line, the more inaccurate the temperature estimate is

2.2. Creating Features

Before feeding to ML algorithms, raw data was preprocessed to make it more suitable for each purpose (ambient recognition or passengers' experience prediction). Two algorithms were developed for this purpose. Algorithm 1 was used for extracting relevant variables for prediction from raw data utilizing discrete Fourier transform (DFT). The accelerometer and magnetometer measurements were rotated from body frame of reference to inertial frame of reference (IFR) before feeding them to Algorithm 1 so that they are independent of device orientation.

The second algorithm (Algorithm 2) is used as the supplement to create temperature features for ML algorithms.

2.3. Essential Features and Response Data

The data from smartphone sensors was analyzed without any presupposes of the type of the features where differences may occur. Thus, relation of some of the used features to recognition of context is not self-evident. For instance, barometric altitude is calculated from the pressure measurements and the pressure is different inside the bus compared to open air due to ventilation system. The pressure may also change between different buses.

Table 2 Example features and classification results in onboard/offboard classification

max speed	z-acc 2nd best	z-acc avg	altitude std	Onboard
20.08	0.243	10.134	0.560	True
18.51	0.291	10.05	0.629	True
4.72	0.410	9.94	1.568	False
1.96	0.417	9.92	1.695	False

Another sensor, which measurements are surprising, is magnetometer. With short tests using higher sampling frequency (100Hz), we noticed that the amount of throttle affects to magnetic field with electric buses. Fig. 3 shows that there is a lot of variation in the magnetic field when the bus is running compared to periods when it is stopped. The reason for this is probably due to an inverter which was located close to the magnetometer. However, there are still hidden impacts which need further research. The best features to classify the context is chosen in the following subsections even though the impact of all of them cannot be interpreted at the moment.

Holdout method is used to partition the aforementioned data sets for cross-validation [11]. Holdout validation creates a random partition on the observations set. The partition divides the observations into a training set and a test set. In this prediction, 50% random data is used for training and 50% for testing when predicting the same data set. This way we obtained enough data for both – training and testing.

2.3.1 Case: Onboard/Offboard: Features used for recognizing whether the passenger is inside or outside the bus are the following:
1. Absolute value of the difference between 2 consecutive atmospheric pressure samples
2. Standard deviation (std) of successive atmospheric pressure in 30 seconds
3. Maximum speed in the past 5 seconds
4. z-axis linear acceleration in IFR
5 & 6. Apply Algorithm 1 to z-axis acceleration in IFR. Take the second greatest magnitude as 5th feature and greatest magnitude (magnitude of 0Hz) as 6th feature.
7. Difference between Google altitude (elevation data at a coordinate from Google API) [12] and

Table 3 Indoor/Outdoor classification accuracy using different classifiers and test data

Training data	Test data	Type of Classifier	Accuracy of feature sets [%]					
			1st	2nd	3rd	4th	5th	Average
50% of 3rd data set	1st data set	TEA	92.71	95.45	91.33	92.46	95.94	93.58
		KNN	90.48	90.69	66.64	75.12	95.5	83.69
		NB	74.52	73.27	73.11	73.32	97.05	78.25
	2nd data set	TEA	78.14	80.12	80.02	79.23	75.27	78.56
		KNN	88.18	88.15	73.33	76.61	75.69	80.39
		NB	84.71	84.24	84	83.86	77.21	82.8
	50% left of 3rd data set	TEA	97.24	97.59	97.7	96.79	96	97.06
		KNN	99.18	99.25	99.46	99.61	98.24	99.15
		NB	94	93.94	94.97	94.23	96.3	94.69
		Average	88.8	89.19	84.51	85.69	89.69	

barometric altitude calculated by formula (6.19) [13, p. 230]
8. Correlation between Google altitude [12] and barometric altitude in the past 30 seconds
9. Pressure at sea level of each observation after applying the same formula than in feature 7
10. std of feature 7 in 30 seconds

Table 2 contains an example of features and response passed to ML Algorithms.

It is observed that formula (6.19) in [13] which calculates barometric altitude requires surface temperature input. Ordinary smartphones lacks thermometer thus this data needs to be fetched from other sources. Data of two closest weather stations to the buses path were used in Algorithm 2.

2.3.2 Case: Diesel/Electric: The Algorithm 1 is used to extract the features from the sensor data. As an output we obtain five value/frequency pairs, i.e., 10 features as total for each variable. The variables passed to Algorithm 1 to create the features were:

1-10. Magnitude of acceleration norm
11-20. z-axis acceleration in IFR
21-30. z-axis linear acceleration in IFR
31-40. Sound level (in Decibels)
41-50. Number of connected satellites

The other features were:

51. Magnitude of linear acceleration norm
52. Difference between network-based altitude and calculated barometric altitude
53. Correlation between network-based altitude and barometric altitude in 30 seconds
54. Difference in speed of 2 consecutive observations
55. GPS accuracy in meters
56. Maximum speed in the past 5 seconds
57. Fluctuation of connected satellites in 2 minutes
58. Difference in GPS altitude and Google altitude [12]
59. Difference in GPS altitude and barometric altitude

After doing 1st prediction using all 59 features, 5 features predominated the importance according to ML algorithm.

These are:

a. Mean magnitude of acceleration norm
b. 2nd smallest magnitude of DFT of number of connected satellites
c. Frequency of 2nd greatest magnitude of DFT of number of connected satellites
d. Number of fluctuation of connected satellites in 2 minutes
e. Difference in GPS altitude and Google altitude

2.3.3 Case: Passenger Satisfaction: In the survey, passengers were asked to evaluate smoothness of driving on the scale from 1 (very uneven driving) to 5 (very smooth driving), and to evaluate noise level in the bus similarly from 1 (very noisy) to 5 (very silent), and to evaluate travel experience (in general) respectively from 1 (very bad) to 5 (very good). In this article, we focused on the evaluations on the smoothness of driving. The suitability of features in previous section were studied to predict passenger satisfaction. The atmospheric pressure and magnitude of magnetic field were found to be the best features for this purpose.

3. Results and Analysis

3.1. Case: Onboard/Offboard

3.1.1 Results: The prediction results using 50% of 3rd data set as training data can be found from Table 3. The 4th data set is not used here as a test data because it does not bring us any useful information. Historic temperatures of FMI weather stations are used as weather data. The temperature of the closest station is used in the 1st and 3rd prediction sets while temperature of each observation obtained using Algorithm 2 is used in 2nd or 4th prediction sets. 1st and 2nd feature sets in Table 3 use all 10 features. In 3rd and 4th feature sets, features 2, 3, 5, 6, 9, and 10 are used since they play most important role in prediction according to ML algorithms. In 5th feature set only 3 features – 3, 5, and 6 – are used, i.e., there is no need for temperature data. In addition to classifiers presented in Table 3, Decision Tree and Tree Ensemble GentleBoost algorithms were also used but their accuracies were poor compared to other classifiers, thus their results are not presented in the table.

3.1.2 Analysis: From the results, KNN algorithm has highest accuracy when the same data set is used for both – training and testing (50% of the data set for training and 50% of the data set for testing). When using data from the same time span on different days (1st data set for testing), Tree Ensemble AdaBoost appears better. If test data is from different time span on different day (2nd data set for testing), the best choice is Naive Bayes algorithm. The results with 1st and 2nd data sets are more relevant as some overfitting may happen when the same data set is used for training and testing.

There is correlation in data from same time span on different days that helps predicting the context correctly more than 93% by using Tree Ensemble AdaBoost. With test data from different time span than training data, the prediction accuracy result is roughly 85% at best. Usage of more features may not give better prediction results. In Table 3, using 3rd data set as test data to predict context on 1st data set, the results are best with 5th prediction set using any ML algorithm, i.e., with only 3 features (3, 5, and 6). The bottom row shows that 5th prediction set also has higher precision than others on average. However, with 5th prediction set the accuracy varies only slightly between different algorithms.

In general, it is not straightforward which algorithm is the best. Using all 10 features, KNN algorithm can predict correctly up to 88% of time with 2nd data set (different time span and different day in training and testing), and 91% of time with 1st data set (same time span and different day in training and testing). Using Naive Bayes with only 3 features increases speed due to simpler algorithm and reduced number of features. However, with Naive Bayes the prediction accuracy for 2nd data set is only 77%. Clearly, one has to choose between prediction accuracy and prediction speed.

Implementing temperature estimation by linear model (Algorithm 2) is better than using nearest station temperature technique. The effect around 0.29% on average. Even though two nearest weather stations to the bus path are chosen, some observations on the bus path are more than 10km away from the straight line connecting two weather stations. Additionally, the available measurement period of temperature is one hour which is quite infrequent to estimate the temperature for each observation. We tried also using weather data from Foreca where temperature measurements from every 10 minute is available. This increased the accuracies shown in Table 3 on average by 0.12%.

3.2. Case: Diesel/Electric

3.2.1 Results: The results of diesel/electric bus detection are presented in Table 4. The used features for 2nd to 6th feature sets can be found from Table 5. The features *a*-*e* in Table 5 are explained in Section 2.3. In 1st feature set, all 59 aforementioned features were used. Decision Tree and Tree Ensemble AdaBoost classifiers were also used but their accuracies were so poor compared to the other classifiers that their results are not presented in Table 4.

3.2.2 Analysis: In overall, while Discriminant Analysis algorithm provide lower accuracy when predicting using holdout partition (93.81% in average compare to around 99% of other algorithms), it dominates other algorithms when predicting other day or other time span. It also shows that while *c* and *e* features in Table 5 play an important roles to predict, they confuse the algorithms. From the bottom row of Table 4 we can see that predicting gets better without feature *c* (3rd feature set compared to 2nd) or *e* (5th feature set

Table 4 Diesel/Electric bus classification accuracy using different classifiers and test data

Training data	Test data	Type of Classifier	Accuracy of feature sets [%] 1st	2nd	3rd	4th	5th	6th	Average
50% of 1st data set	50% left of 1st data set	TEG	100	99.98	99.93	97.33	96.03	98.5	98.63
		KNN	99.87	99.54	99.64	99.52	94.78	98.52	98.65
		DA	97.3	93.47	94.89	94.65	89.59	92.93	93.81
	2nd data set	TEG	76.9	82.31	83.08	79.46	87.04	87.01	82.63
		KNN	77.47	78.49	78.35	78.44	81.69	89.2	80.60
		DA	73.8	85.77	83.99	78.69	88.87	88.17	83.22
	3rd data set	TEG	86.36	85.86	88.66	88.45	87.78	88.8	87.65
		KNN	81.3	81.55	80.68	81.47	79.89	87.25	82.02
		DA	89.37	95.3	93.8	91.43	88.82	94.86	92.26
	4th data set	TEG	61.51	63.81	70.84	62.24	78.3	78.22	69.15
		KNN	64.97	61.48	61.53	61.22	73.76	84.89	67.98
		DA	71.04	73.64	66.53	53.85	88.33	87.44	73.47
		Average	81.66	83.43	83.49	80.56	86.24	89.65	

Table 5 Feature sets used in Table 4. *a, b, c, d,* and *e* feature sets explained in Section 2.3.2

Feature sets	Features a	b	c	d	e
2nd	✓	✓	✓	✓	✓
3rd	✓	✓		✓	✓
4th		✓		✓	✓
5th		✓		✓	
6th	✓	✓		✓	

compared to 4th and 6th feature set compared to 3rd), worse without feature *a* (6th feature set compared to 5th and 3rd feature set compared to 4th). Once again, although ML algorithms announce that all five features are vital, including all of them confuses the model and causes overfitting thus degrades the prediction accuracy.

In summary, with only 3 features prediction accuracy using Discriminant Analysis algorithm is around 94% when training and testing data are from same data set and 88% when testing data is from different data set than the training data. Both percentages are quite good and the prediction speed of this algorithm is also fairly good compared to KNN algorithm.

3.3. Case: Passenger Satisfaction

The passengers evaluated their satisfaction for the bus ride on scale 1 to 5, i.e., very uneven - very smooth driving. The average grade was 3.80 for all passengers, 3.77 for passengers traveling in diesel buses and 3.85 for electric buses respectively. Thus, the electric buses received better grades, but the difference was not statistically significant. The grades of the diesel buses had higher variance, which was 0.876 for diesel buses and 0.530 for electric buses. The difference between the variances was statistically significant (sig. 0.001 with Levene's test for equality of variances). The average grade for the general travel experience was 4.01 and for the noise level 3.66. In this paper, we focused on analysing the dependence between the satisfaction for the bus ride and the sensor measurements.

Using ML regression algorithms to predict passengers' evaluation, the accuracy is 96.5% when predicting same data set. However, it drops to 60% when changing to different time period. The results of ML regression can be seen in Fig. 4. Two best features according to ML algorithm are atmospheric pressure and magnitude of magnetic field. This is mainly due to the low sampling rate (2Hz) that it is hard to utilize the two obvious important features which are acceleration and angular velocity.

4. Conclusions and Future Work

The measurement arrangements caused some limitations. The largest source for these was the sampling rate of 2 Hz with smartphone measurements which was selected to ensure stable behavior of AndroSensor application. With higher rate, it would be possible to acquire more relevant features for instance from accelerometer and gyroscope data. At best, we were still able to

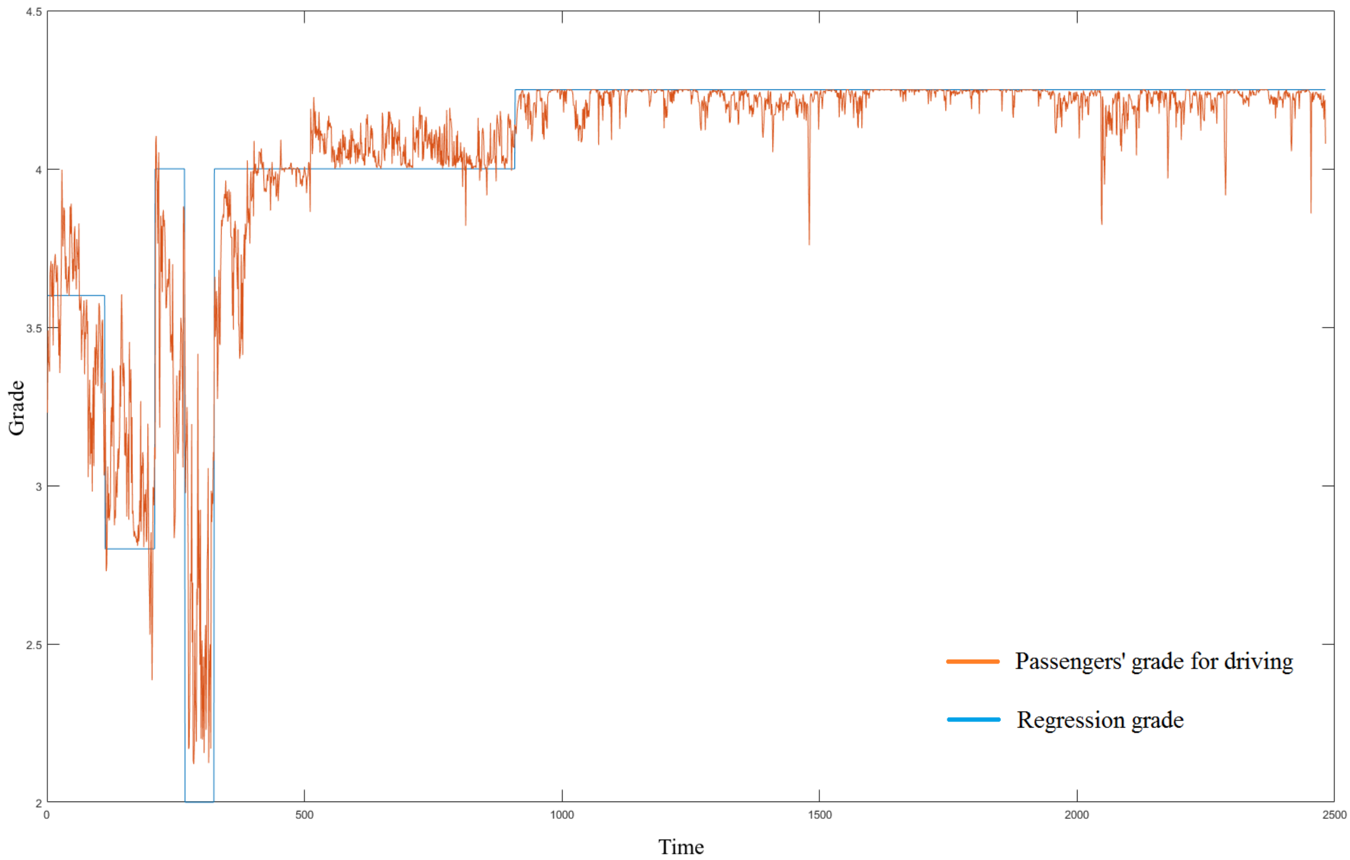

Fig. 4. *Passenger satisfaction regression classification*

achieve over 90% accuracy in onboard/offboard and diesel/electric bus recognition using ML based classifiers. However, the recognition accuracy varies between data sets used for testing.

With a short set of higher rate data, we observed that in public transportation measurements there are components in accelerometer and gyroscope data that are essential to detect the onboard/offboard, diesel/electric bus situation as well as travel experience. Thus, it is expected that the recognition accuracy will improve in the future when high frequency data is used.

Despite of the limitations in measurement arrangements, we managed to recognize the context relatively well which already gives a good basis for real-time applications. This encourages to continue the work with larger data sets and improved data collection methods. The plan is to arrange a new data collection campaign using higher data rate and improved sensor loggers. For example, in our earlier studies carrier to noise values of GNSS have showed promising results in recognizing indoor/outdoor situation. Hence, we are going to use GnssLogger application in addition to AndroSensor in our next measurement campaign. In addition, combining survey software and data logger to same device is studied, as this would reduce manual synchronization efforts.

We have also recorded similar data set with hybrid passenger cars. This set allows to compare the behavior and user experience in public city buses to passenger cars of different quality.

We used in this research five different ML algorithms. However, the suitability of different types of ML methods to our research field needs to be further surveyed. It is possible that there are more suitable methods and algorithms than the currently used ones – especially the ones that take inputs as time series [14].

Acknowledgments

This work was supported by TEKES (Finnish Funding Agency for Innovation) as part of the Living Lab Bus project.

References

[1] Xiang, H. k., Ming, A. t.: 'A Study of the City Bus Passenger Flow Intelligent Statistical Algorithm Based on Motion Vector Tracking', 4th IEEE Int. Conf. Information Science and

Technology, Shenzhen, China, April 2014, pp. 41–44

[2] Zhu, F., Song, J., Yang, R., Gu, J.: 'Research on Counting Method of Bus Passenger Flow Based on Kinematics of Human Body', Int. Conf. Computer Science and Software Engineering, Wuhan, China, December 2008, pp. 201–204

[3] Dollar, P., Wojek, C., Schiele, B., Perona, P.: 'Pedestrian Detection: An Evaluation of the State of the Art', IEEE Transactions on Pattern Analysis and Machine Intelligence, 2012, 34, (4), pp. 743–761

[4] Perttula, A., Parviainen, J., Collin, J.: 'Pedestrian Detection with High Resolution Inertial Measurement Unit', IEEE SENSORS, Orlando, FL, USA, October 2016, pp. 367–369

[5] Elhoushi, M., Georgy, J., Noureldin, A., Korenberg, M. J.: 'A Survey on Approaches of Motion Mode Recognition Using Sensors', IEEE Transactions on Intelligent Transportation Systems, 2017, 18, (7), pp. 1662–1686

[6] Ferreira, J., Carvalho, E., Ferreira, B., de Souza, C., Suhara, Y., Pentland, A., Pessin, G.: 'Driver Behavior Profiling: An Investigation with Different Smartphone Sensors and Machine Learning', PLOS ONE, 2017, 12, (4), pp. 1–16

[7] Collin, J., Kuusniemi, H., Mezentsev, O., MacGougan, G., Lachapelle, G.: 'HSGPS under Heavy Signal Masking - Accuracy and Availability Analysis', Proc. 6th Nordic Radio Navigation Conference and Exhibition, on board between Finland and Sverige, December 2003

[8] Zhou, P., Zheng, Y., Li, Z., Li, M., Shen, G.: 'IODetector: A Generic Service for Indoor Outdoor Detection', Proc. 10th ACM Conf. Embedded Network Sensor Systems, Toronto, Ontario, Canada, November 2012, pp. 113–126

[9] Banville, S., van Diggelen, F.: 'Precise GNSS for Everyone: Precise Positioning Using Raw GPS Measurements from Android Smartphones', GPS World, 2016, 11, pp. 43–48

[10] 'Matlab Statistics and Machine Learning Toolbox', https://se.mathworks.com/products/statistics.html, accessed 13 October 2017

[11] 'Matlab Cross Validation Method', https://se.mathworks.com/help/stats/cvpartition.html, accessed 13 October 2017

[12] 'Google Web Service, Elevation API', https://developers.google.com/maps/documentation/elevation/intro, accessed 18 January 2018

[13] Groves, P.D.: 'Principles of GNSS, Inertial, and Multi-Sensor Integrated Navigation Systems' (Artech House Publishers, 2013, 2nd edn.)

[14] Hallac, D., Vare, S., Boyd, S., Leskovec, J.: 'Toeplitz Inverse Covariance-Based Clustering of Multivariate Time Series Data', Proc. 23rd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining, Halifax, Nova Scotia, Canada, August 2017, pp. 215–223