Smartphone Transportation Recognition using Machine Learning

Elena Merdzanovska, Stefan Kalabakov, Ivona Milanova, Hristijan Gjoreski

Faculty of Electrical Engineering and Information Technologies,

University of Ss. Cyril and Methodius in Skopje, Macedonia

elenamerdzanovska@yahoo.com, kalabakovs@gmail.com, ivonamilanova221@gmail.com , hristijang@feit.ukim.edu.mk

*Abstract*—Activity recognition is a rich field aiming at differentiating between various user activities, non-motorized and motorized alike. In the past this task has relied heavily on using GPS, but it's high power consumption and need for an unobstructed view of a satellite have made it a suboptimal solution. This paper aims to solve this classification problem using raw data from inertial and ambient sensors as input to a multi-class classifier. Since the issue is such that requires a lightweight solution which can function well on portable devices, a one minute window is chosen as the prediction interval, providing a compromise between recognition speed, accuracy, and the need to store large amounts of data. Furthermore, in order to minimize the number of features needed to get an accurate prediction we implement two feature selection techniques. As a last step, we explore classical machine learning methods as options for solving the classification issue and perform an eight-fold validation, with which we achieved f1-score of 0.83.

Keywords—transport; activity recognition; sensors; smartphone; machine learning model; feature analysis

# Introduction

The vast amount of sensor data that has been made available by smartphones in the past several years, have made activity recognition and transport mode recognition a thriving field. Producing such models could have a profound impact on several areas such as trajectory tracking [1], parking spot detection [2], safe driving [3], improving transport systems [4] and many more. To be able to achieve these goals on a broader level, there is a need of using an extensive and rich dataset, as well as efficient feature extraction and selection methods.

# Related Work

Most of the studies in this field, particularly in activity recognition tasks focus on using GPS in order to detect the location of the device. Using the GPS speed, acceleration and trajectory of the trip, they differentiate between different user activities [16]-[25]. We analyze the state of art from four different aspects: type of classifier, different classifier parameters and window size.

Among many works it can be seen that when doing an activity recognition they focus on recognizing between three activities like Walking, Car and Train [11] up to ten including Still, Walk, Run, Car, Bus, Subway, Train, Motorcycle, High Speed Rail and Bike [12]. Nevertheless the most frequently used are Still, Walk, Run, Bike, Car, Bus, Train and Subway. One thing that is common for activity recognition tasks is that as the number of classes increases it becomes more difficult tell apart the activities.

Depending on the problem, different window sizes are used. Smaller window sizes allow a faster activity recognition also reduced resources and require less energy. Large data windows on the other hand are considered suitable for recognition of more complex activities. Experiments with changing the windows sizes are done in order to get the best trade-of between recognition speed and accuracy as well as memory size [14], [15]. The window size varies from one study to another. Location based approaches usually use a longer window from several minutes up to tens of minutes. On the contrary, hybrid approaches that include inertial and GPS sensors prefer a shorter window.

For the recognition itself, many different classifiers are used. Decision tree (DT), K-nearest neighbor (KNN), support vector machine (SVM) and naive Bayesian (NB). In order to improve the overall accuracy in predicting the activity many studies are using more advanced techniques such as ensable classifiers, multi-layer classifiers and post processing. Some of the studies also proposed a solution for adapting the classifier to a person using a Multi-Class Adaptive Training [26]. To improve the performance some other techniques used a combination of machine learning methods such as Discrete Hidden Marov Model [27] and Bootstrap aggregating [28]. A technique that has attracted a lot of attention in the machine learning community lately is deep learning was applied to the transportation mode recognition tasks [13], [19]. Two most widely used techniques for evaluating the performance are F1-score and recognition accuracy obtain­ed from the confusion matrix.

# Dataset

The SHL (University of Sussex-Huawei Locomotion) Dataset [5] is a highly versatile and precisely annotated large-scale dataset of smartphone sensor data for multimodal locomotion and transportation analytics of mobile users. The dataset comprises 7 months of measurements, collected from all sensors of four smartphones carried at typical body locations, including the images of a body-worn camera, while three participants used eight different modes of transportation in the southeast of the United Kingdom, including in London. It contains 750 hours of labelled locomotion data sorted into the following classes: Car, Bus, Train, Subway, Walk, Run, Bike and Still.

Specifically, the work in this paper is done on a smaller part of the dataset, which consists 271 hours of training data. The data is organized in frames where each frame represents one minute and comprises of 6000 samples. This raw data was collected using the following sensors on a Huawei P9: accelerometer, gyroscope, gravity sensor, linear acceleration sensor, magnetometer, orientation and pressure sensor.

# Feature Extraction

As suggested in various papers regarding this or similar topics it is beneficial to extract statistical, time domain and frequency domain features which have proven especially useful in extracting information about high frequency motion such as that from the vehicle's engine [6]. Furthermore, the low frequency components of the acceleration data represent the influence of gravity and are thus useful in determining the user's posture [7].

Figure 1, for example, illustrates a clear distinction between user activities involving motion of the body when observing the distribution of the spectral entropy of the accelerometer signal along the x-axis.

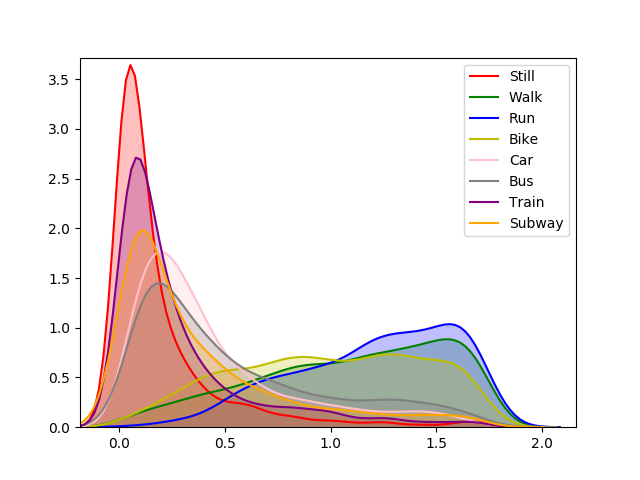


Figure 1: Distribution of the spectral entropy for the x-axis of the accelerometer

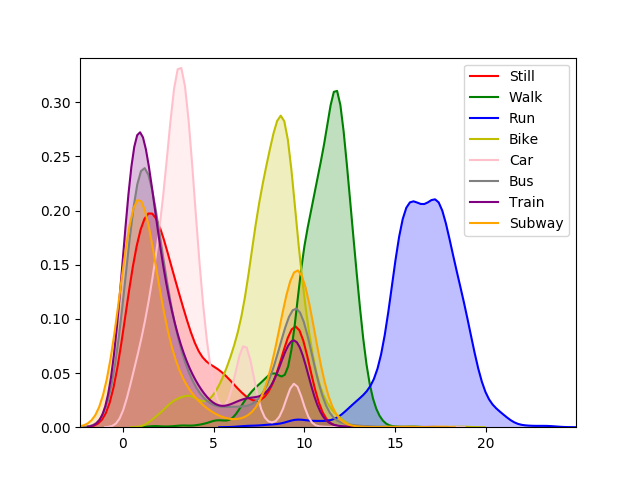
The spectral entropy of the signal, useful when differentiating between states that produce signals with similar energy was computed as follows:

The full set of features that were computed and have proven to be relatively standard across several related works can be found in the table below.

TABLE 1 Set of used features

|  |  |
| --- | --- |
| **Domain** | **Features** |
| Statistical | Mean, STD, Variance, Median, Min  Max, Range, Interquartile range  Kurtosis, Skewness, RMS |
| Time | Integral, Mean Crossing Rate  Auto-Correlation, Peak to Average Ratio |
| Frequency | FFT DC,1,2,3,4,5 Hz  Spectral Energy, Spectral Entropy,  Wavelet Entropy, Wavelet Magnitude |

For all but the accelerometer and the linear acceleration sensor only the statistical domain, features were taken into consideration. Figure 2 also presents a distinction between the classes by looking at the distribution of the computed RMS (Root Mean Square).

  
Figure 2: Distribution of RMS of the y component of the acceleration

# Feature Selection

The total number of features we extracted from our data was 448. Due to the high correlation between some of the features we believed the same classification score could be achieved using only a part of them. In order to decrease feature calculation time we decided to choose the most representative subset of the total 448 features and use it in the final classifier. After a review of the existing work on this topic we noticed that SFS [8] was often used and that it would be a good option for us and decided to use it in a combination with mRMR feature filtering method, because it has shown good performance in eliminating similar features. An overview of the two different algorithms is given in the sections below.

## mRMR (Maximum Relevance Minimum Redundancy)

This method was used to reduce the correlation inside the feature set (minimize redundancy), while maintaining high correlation with the target class (maximizing relevance) [9]. In other words, it was desired to achieve a good classification score and at the same time to exclude less useful features. In this approach, the main criterion for choosing the features is their mutual information. More specifically, let us assume that M is the set of all features and F contains all already selected features. When selecting the x i feature from M to include in F the goal is to maximize the relevance between the feature and the target class (i.e., *xi , c*), which is shown with the equation:

and to simultaneously minimize the redundancy between that feature and the already selected features (i.e., *xi , xj*), given with:

To combine the previous two conditions, in our case MIQ (Mutual Information Quotient) [10] was used as a parameter in the mRMR algorithm. Including that, the final formula for choosing the next feature is given with:

* *MI(x, y)* - Mutual information of x and yThe feature to be examined
* *xi* –a previously selected feature
* *M* - Set of all features
* *F* - Set of the selected features
* *c* - Target class

Hence, using mRMR all the extracted features were ranked to choose the most useful ones; the top 65 features were selected out of the 448 features long set.

## SFS (Sequential Feature Selector)

Sequential feature selection algorithms (SFAs) are a family of greedy search algorithms that are used to reduce an initial d-dimensional feature space to a k-dimensional feature subspace where k < d. Concisely, SFAs remove or add one feature at the time based on the classifier performance until a feature subset of the desired size k is reached [3]. These algorithms and the features selected are classifier-specific, since they are chosen based on the scores obtained by training many instances of the same classifier on different feature sets. There are four different types of SFAs:

#### Sequential Forward Selection (SFS)

#### Sequential Backward Selection (SBS)

#### Sequential Forward Floating Selection (SFFS)

#### Sequential Backward Floating Selection (SBFS)

The floating variants, SFFS and SBFS, can be considered as extensions to the simpler SFS and SBS algorithms. The floating algorithms have an additional exclusion or inclusion step to remove features once they were included (or excluded), so that a larger number of feature subset combinations can be sampled. In our case, Sequential Forward Selection (SFS) was used for a number of different classifiers, but only on the 65 most relevant features selected with rMRM in the first phase of feature selection.

# Evaluation Metrics and Techniques

## Performance metric

The evaluation metric for the SHL Challenge, as well as for our project is the F1-score. In general, the main reason for using F-Score instead of accuracy is its better handling of uneven class distribution. Accuracy is weighted towards majority class performance, whereas F-measure is useful for measuring the performance on minority classes as well. The formula for calculating the F1-score is given below:

* *Pi*- Precision of class *i*
* *Ri*- Recall of class *i*
* *n* - Number of classes

## Validation technique

In any real-world application, a classification model is trained on historical data and used to predict new data. In our case that means that it does not make sense to have samples from the same day both in the train and in the test dataset, because that would never happen in real life. In order to perform appropriate validation of the models we had to recreate the same conditions. In other words, the train and test data subset used for building the models always have to include samples from entirely different days (preferably different weeks, to keep an approximately uniform distribution of the classes). Having this in mind, because the train data consists of a around 15 weeks, a K-fold cross-validation with 8 folds was performed on each of our models, with around two weeks in each fold. Since the data was originally given in a random order, to achieve this kind of validation, we had to sort the train data so that the samples are back in the order in which they happened, using the additional file train\_order.txt.

# Experimental results

In the model selection part, out of the numerous classifiers that exist, we decided to try the following ones:

#### Naive Bayes

#### Decision Tree

#### SVM (Support Vector Machine)

Since SVMs are only capable of two-class classification, this algorithm can be used for multi-class classification by building different classifiers for each class and then combining them, also known as one-vs-rest classification.

#### Logistic Regression

A Multinomial Logistic Regression for our multi-class problem was used, which means that the cost function, which is minimized, is the multinomial loss fit across the entire probability distribution.

#### Random Forest

An algorithm that fits a number of decision tree classifiers on various sub-samples of the dataset and use averaging to improve the predictive accuracy and control over-fitting.

#### MLP (Multi-layer Perceptron) Classifier

A type of feedforward artificial neural network that consists of at least three layers of nodes. This model optimizes the log-loss function, in our case using stochastic gradient descent and as an activation function for the hidden layer the rectified linear unit function is used.

We trained each of these models on a reduced feature set specifically selected for each classifier by SFS algorithm. In the following table the mean F1-score and its variance are given, after performing the K-fold cross validation described earlier.

Analyzing the mean F1-score and its variance for each different classifier it was concluded that the classifier that produced the best result was the Random Forrest Classifier.

In pursuance of getting the best result of the algorithm some parameter tuning was done. More specifically the parameters that were being tuned were the maximum depth of the individual trees created by the Random Forrest Classifier and the number of estimators. The results in Graph 1 represent the F1 macro score, which can be interpreted as an average of the F1-score for each class, such that this score reaches its best value at one and worst score at zero.



Figure 3 Results from different combinations of parameters for the Random Forest Classifier

### Observing the data a conclusion can be made that adding more estimators to the classifiers produces better results until a certain limit is reached where overfitting starts happening. It can also be noticed that increasing the maximum depth of the trees actually causes overfitting and gives a lower F1-score. Nevertheless having really small amount of estimators produces a lower score with a significantly high variance meaning there are inconsistencies in the data and the end result when running the algorithm on different test set will vary every time and cannot be easily predicted.

### As a final step in order to fully describe the performance of the classification model a confusion matrix as shown in Table 2 was made. It represents the sum of all the matrices obtained by the different train/test splits.

TABLE 2 Confusion matrix from the Random Forrest Classifier

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Still | Walk | Run | Bike | Car | Bus | Train | Subway |
| Still | 1655 | 8 | 1 | 17 | 4 | 18 | 78 | 28 |
| Walk | 17 | 1749 | 4 | 23 | 0 | 2 | 0 | 3 |
| Run | 0 | 17 | 395 | 0 | 0 | 0 | 0 | 0 |
| Bike | 8 | 12 | 2 | 1643 | 2 | 9 | 4 | 1 |
| Car | 56 | 8 | 3 | 14 | 1631 | 642 | 80 | 16 |
| Bus | 31 | 22 | 2 | 15 | 198 | 1333 | 33 | 26 |
| Train | 106 | 5 | 0 | 9 | 37 | 30 | 1302 | 262 |
| Subway | 17 | 16 | 0 | 0 | 7 | 8 | 449 | 829 |

### Analyzing the matrix several conclusions can be drawn. The algorithms differentiates quite well between Still, Walk, Run, Walk and Bike and other activities as not many samples from each of these classes was classified in the other classes. However we can notice that there are a lot of misclassified samples between Car/Bus, as well as Train/Subway. This bring us to a conclusion that in order to fully distinguish between the classes that are hard to classify another more advanced method needs to be used.

# Conclusion

We analyzed the results specifically the F1-score obtained from different classifiers and feature selection in order to determine the best combination for a more accurate activity recognition. Using a combination of features including statistical features, time domain features, and frequency domain features extracted from four primary sensors including tri-axial accelerometer, gyroscope, magnetometer and a pressure sensor a recognition score of up to 83% was achieved. The good thing about the approach we used was that it does not rely on any other external signal sources and priori information, which makes it universal. In order to choose the most relevant features we relied on two different types of feature selection algorithms. The first one was MRMR, which gave us a list of features ordered by the most relevant and least redundant, while the second was SFS which using a classifier we supplied implementing a greedy search algorithm selected the subset of features most relevant to the problem. The results of the study show that the classifier with the best F1-score was the Random Forrest Classifier using 500 estimators and 20 max depth of the trees. Analyzing the confusion matrix it can be seen that the algorithm lacks the ability to do a clear differentiation between the class pairs car/bus and train/subway as they are very similar regarding the overall position of the phone as well as their movement. From here, we can define the issue for further work that would be developing a hybrid hierarchical classification model, which would make it possible to set apart the more hardly distinguishable classes.

##### References

1. S. Bhattacharya, H. Blunck, M. B. Kjærgaard and P. Nurmi, "Robust and Energy-Efficient Trajectory Tracking for Mobile Devices" IEEE Transactions on Mobile Computing, vol. 14, no. 2, pp. 430-443, 2015.
2. J. K. Suhr and H. G. Jung, "Automatic Parking Space Detection and Tracking for Underground and Indoor Environments" IEEE Transactions on Industrial Electronics, vol. 63, no. 9, pp. 5687-5698, 2016.
3. M. Fazeen, B. Gozick, R. Dantu, M. Bhukhiya, and M. C. González,. "Safe driving using mobile phones" IEEE Transactions on Intelligent Transportation Systems, vol. 13, no. 3, pp. 1462-1468, 2012.
4. J. Engelbrecht, M. J. Booysen, G. van Rooyen, and F. J. Bruwer, "Survey of smartphone-based sensing in vehicles for intelligent transportation system applications," IET Intelligent Transport Systems, vol. 9, no. 10, pp. 924-935, 2015.
5. H. Gjoreski, M. Ciliberto, F. J. Ordoñez Morales, D. Roggen, S. Mekki, S. Valentin. “A versatile annotated dataset for multimodal locomotion analytics with mobile devices.” In Proc. ACM Conference on Embedded Networked Sensor Systems. ACM, 2017.
6. S. Hemminki, P. Nurmi, S. Tarkoma, “Accelerometer-based transportation mode detection on smartphones” ACM Conference on Embedded Networked Sensor Systems, ACM 2013:1.
7. P.H. Veltink, H.J. Bussmann, W. de Vries, W.J. Martens, R.C Van Lummel, “Detection of static and dynamic activities using uniaxial accelerometers”, IEEE Trans. Rehab. Eng., 1996, 4, 375–385.
8. A. Jahangiri and H.A. Rakha, “Applying machine learning techniques to transportation mode recognition using mobile phone sensor data,” IEEE Transactions on Intelligent Transportation Systems, vol. 16, no. 5, pp. 2406–2417, 2015
9. H. Peng, F. Long, and C. Ding, “Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 27, no. 8, pp. 1226–1238, Aug. 2005.
10. G. Gulgezen, Z. Cataltepe, and L. Yu, “Stable and Accurate Feature Selection,” Proc. European Conf. Machine Learning and Knowledge Discovery in Databases: Part I (ECML PKDD ’09), vol. 5781, 2009
11. T. Nick, E. Coersmeier, J. Geldmacher, and J. Goetze, “Classifying means of transportation using mobile sensor data,” in Proc. of Internetional Joint Conference on Neural Networks, Barcelona, Spain, 2010, pp. 1–6.
12. S. H. Fang, H. H. Liao, Y. X. Fei, K. H. Chen, J. W. Huang, Y. D. Lu, and Y. Tsao, “Transportation modes classification using sensors on smartphones,” Sensors, vol. 16, no. 8, pp. 1324–1339, Aug. 2016.
13. S. H. Fang, Y. X. Fei, Z. Xu, and Y. Tsao, “Learning transportation modes from smartphone sensors based on deep neural network,” IEEE Sensors Journal, vol. 17, no. 18, pp. 6111–6118, Aug. 2017
14. M. C. Yu, T. Yu, S. C. Wang, C. J. Lin, and E. Y. Chang, “Big data small footprint: The design of a low-power classifier for detecting transportation modes,” in Proc. of Very Large Data Base Endowment, Hangzhou, China, 2014, pp. 1429–1440.
15. X. Su, H. Caceres, H. Tong, and Q. He, “Online travel mode identification using smartphones with battery saving considerations,” IEEE Transactions on Intelligent Transportation Systems, vol. 17, no. 10, pp. 2921–2934, Mar. 2016
16. Y. Zheng, Y. Chen, Q. Li, X. Xie, and W. Y. Ma, “Understanding transportation modes based on GPS data for web applications,” ACM Transactions on the Web, vol. 4, no. 1, pp. 1–36, Jan. 2010.
17. Y. Zheng, Q. Li, Y. Chen, X. Xie, and W. Y. Ma, “Understanding mobility based on GPS data,” in Proc. of the International Conference on Ubiquitous Computing, Seoul, Korea, 2008, pp. 312–321.
18. Z. Xiao, Y. Wang, K. Fu, and F. Wu, “Identifying different transportation modes from trajectory data using tree-based ensemble classifiers,” ISPRS International Journal of Geo-Information, vol. 6, no. 2, pp. 57–79, Feb. 2017.
19. Y. Endo, H. Toda, K. Nishida, and J. Ikedo, “Classifying spatial trajectories using representation learning,” International Journal of Data Science and Analytics, vol. 2, no. 3-4, pp. 107–117, Dec. 2016.
20. G. Xiao, Z. Juan, and C. Zhang, “Travel mode detection based on GPS track data and bayesian networks,” Computers, Environment and Urban Systems, vol. 54, pp. 14–22, Nov. 2015.
21. L. Stenneth, O. Wolfson, P. S. Yu, and B. Xu, “Transportation mode detection using mobile phones and GIS information,” in Proc. of the ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, Chicago, Illinois, 2011, pp. 54–63.
22. I. Semanjski, S. Gautama, R. Ahas, and F. Witlox, “Spatial context mining approach for transport mode recognition from mobile sensed big data,” Computers, Environment and Urban Systems, vol. 66, pp. 38–52, Nov. 2017.
23. A. Bolbol, T. Cheng, I. Tsapakis, and J. Haworth, “Inferring hybrid transportation modes from sparse GPS data using a moving window svm classification,” Computers, Environment and Urban Systems, vol. 36, no. 6, pp. 526–537, Nov. 2012.
24. H. Gong, C. Chen, E. Bialostozky, and C. T. Lawson, “A GPS/GIS method for travel mode detection in new york city,” Computers, Environment and Urban Systems, vol. 36, no. 2, pp. 131–139, Mar. 2012.
25. P. A. Gonzalez, J. S. Weinstein, S. J. Barbeau, M. A. Labrador, P. L. Winters, N. L. Georggi, and R. Perez, “Automating mode detection for travel behaviour analysis by using global positioning systems-enabled mobile phones and neural networks,” IET Intelligent Transport Systems, vol. 4, no. 1, pp. 37–49, Mar. 2010
26. B. Cvetkovic, , B. Kaluža, M. Gams and M. Luštrek, “Adapting Activity Recognition to a Person with Multi-Classifier Adaptive Training,” Journal of Ambient Intelligence and Smart Environments vol. 7, no. 2, pp. 171-185, Mar . 2015
27. Reddy, S., et al., Using Mobile Phones to Determine Transportation Modes. Acm Transactions on 32 Sensor Networks, 2010. 6(2).
28. Zheng, Y., et al., Understanding transportation modes based on GPS data for web applications. ACM Transactions on the Web (TWEB), 2010. 4(1): p. 1.