Tackling the SHL Recognition Challenge with Phone Position Detection and Nearest Neighbour Smoothing

Peter Widhalm
Austrian Institute of Technology
Vienna, Austria
peter.widhalm@ait.ac.at

Philipp Merz
Austrian Institute of Technology
Vienna, Austria
philipp.merz@ait.ac.at

Liviu Coconu
Austrian Institute of Technology
Vienna, Austria
liviu.coconu@ait.ac.at

Norbert Brändle
Austrian Institute of Technology
Vienna, Austria
norbert.braendle@ait.ac.at

ABSTRACT

We present the solution of team *MDCA* to the Sussex-Huawei Locomotion-Transportation (SHL) recognition challenge 2020. The task is to recognize the mode of transportation from 5-second frames of smartphone sensor data from two users, who wore the phone in a constant but unknown position. The training data were collected by a different user with four phones simultaneously worn at four different positions. Only a small labelled dataset from the two "target" users was provided. Our solution consists of three steps: 1) detecting the phone wearing position, 2) selecting training data to create a user and position-specific classification model, and 3) "smoothing" the predictions by identifying groups of similar data frames in the test set, which probably belong to the same class. We demonstrate the effectiveness of the processing pipeline by comparison to baseline models. Using 4-fold cross-validation our approach achieves an average F1 score of 75.3%.

CCS CONCEPTS

• **Information systems → Data stream mining**; • **Human-centered computing → Ubiquitous and mobile computing systems and tools**.

KEYWORDS

activity recognition, transport mode recognition, signal processing, neural networks.

ACM Reference Format:
Peter Widhalm, Philipp Merz, Liviu Coconu, and Norbert Brändle. 2020. Tackling the SHL Recognition Challenge with Phone Position Detection and Nearest Neighbour Smoothing . In *Adjunct Proceedings of the 2020 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2020 ACM International Symposium on Wearable Computers (UbiComp/ISWC '20 Adjunct), September 12–16, 2020, Virtual Event, Mexico.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3410530.3414344

UbiComp/ISWC '20 Adjunct, September 12–16, 2020, Virtual Event, Mexico

ACM ISBN 978-1-4503-8076-8/20/09…$15.00
https://doi.org/10.1145/3410530.3414344

1 INTRODUCTION

After two successful competitions in the previous years, the third Sussex-Huawei Locomotion-Transportation (SHL) recognition challenge in 2020 continues this emerging tradition. Again the goal is to recognize 8 different means of transport and locomotion using sensor data recorded with smartphones. The results of the past two competitions [3, 6] have shown that the patterns in the sensor data vary between users and also depend on the position where the smartphone is worn. Therefore, the performance of a classification model is higher if it has been trained specifically for a particular user and a defined wearing position. In addition, the detection accuracy also depends on the length of the observation period, i.e the length of the data frames. In the previous competitions, using Hidden Markov Model smoothing as a post-processing step has proven to be an effective way to leverage the additional information contained in longer time series [6]. First, the data is segmented with a sliding time window of a few seconds length to generate input vectors with fixed dimension for the classification model. In a second step the classifier output of multiple consecutive time frames is smoothed by defining (or learning) transition probabilities and applying the Viterbi algorithm. This years' competition introduces additional challenges: the modes of transportation have to be inferred from data windows that are only 5 seconds long. The training data come from only one user and from four different wearing positions: Hand, Bag, Hips, and Torso. The test data, however, were collected by 2 other users and come from an unknown wearing position. The data frames in the test set have been created by segmenting the sensor time series using a 5-seconds sliding window. The resulting frames were shuffled to remove their temporal ordering. Moreover, the sliding window had a step size larger than the window size, which means that the data frames do not connect seamlessly and there are gaps between the data frames. This makes it very difficult - or even impossible - to restore their original ordering. In addition to the training and test datasets, a small validation dataset was provided. This dataset contains data from all four phone positions and from the same two users as the test data.

We tackle this task using a similar classification model as in the 2018 competition [7]. However, based on the learnings of the previous SHL recognition challenges we identified the following key approaches to address the particular difficulties of this years' competition:

(1) finding out the phone wearing position of the test data and training a classifier specifically for that position;
(2) using data from the two "target" users in the test data to improve classification performance specifically for these users;
(3) "smoothing": finding a way to average over groups of related data frames instead of classifying solitary data frames.

We present a processing pipeline that implements each of these points, and we evaluate its performance using the validation dataset provided for the SHL recognition challenge.

The remainder of this paper is organized as follows: in Section 2 we define the particular task of the 2020 SHL recognition challenge and describe the data we used for our submission. We specify the features extracted from the sensor data and present details about our method in Section 3. Finally, we report our experimental results in Section 4.

2 DATA AND TASK DESCRIPTION

The data used for the challenge is a subset of the Sussex-Huawei Locomotion-Transportation (SHL) Dataset [2, 5]. All data was collected with a HUAWEI Mate 9 smartphone and a specific Android application [1]. The datasets contain readings from the following sensors: 3D accelerometer, gyroscope, magnetometer, linear accelerometer, gravity, orientation, ambient air pressure. The sensor data were segmented into data frames with 5 seconds length, each containing 500 values, corresponding to a sampling frequency of 100Hz. The data comprises 59 days (196072 frames) of training data, 6 days (28789 frames) of validation data, and 40 days (57573 frames of test data. In our processing pipeline and experimental evaluation we use parts of the validation dataset for model training. To avoid confusions, we refer to the training dataset as T, the validation data as V, and to the test data as X in the following sections.

The goal of the 2020 SHL recognition challenge is to recognize 8 modes of transportation in a user and phone position independent manner, which is reflected by the provided data. The training data T contains data from user 1 and from four different phone positions: Hand, Hips, Torso, and Bag. The validation set V contains data from users 2 and 3, and the same four phone positions. Both T and V include the correct activity class labels. The data frames in T and V were generated by segmenting the whole sensor time series with a non-overlapping sliding window and are consecutive in time. The test data X contains data from users 2 and 3, but from only one *undisclosed* phone position. The data frames were again generated with a non-overlapping sliding window, but here the jumping size was larger than the window size so that there are gaps between the data frames. In addition the samples in X are shuffled and therefore *not* consecutive in time. The class labels for X are held back by the organizers of the challenge, because this is the data on which the submissions must make their predictions.

3 PROCESSING PIPELINE

Our data processing pipeline consists of the following steps:

(1) **Extracting features** from the raw sensor data;
(2) **Phone Position Recognition:** training a classification model for phone position recognition using datasets T and V and detecting the phone position in dataset X;
(3) **Phone Position and User-specific Transport Mode Classification:** training a classification model using dataset V (all phone positions) and dataset T (only the phone position detected in the previous step);
(4) **"Nearest Neighbour smoothing":** identifying groups of similar data frames in X and inferring a common class label for the group of data frames by combining their class posteriors.

In the following we will explain each of these steps in detail.

3.1 Feature Extraction

We use two different sets of features: one for classification and the second for defining similarity among data frames in the test set X.

For classification we use the same features as in [8] including the following sensor modalities: 3D accelerometer, gravity sensor, gyroscope, magnetometer, and barometric sensor. From the 3D sensors we compute the magnitude of the (x, y, z)-vectors to obtain rotation invariant values. We use statistics such as mean value, standard deviation, minimum and maximum value. In addition we derive features from the Fourier transform and the autocorrelation function of the sensor time series. We will denote the classification feature vectors of a data frame d by $\mathbf{f}_d$.

To introduce a measure of "similarity" between data frames (which we will use for identification of Nearest Neighbours in Sect. 3.4) we use data of the 3D accelerometer, the orientation sensor, and the barometer. The feature vector includes the mean value of the orientation vector, the mean ambient air pressure, and the autocorrelation of the acceleration magnitude for time lags from 10ms to 100ms. The feature vectors defining the similarity between data frames will be denoted by $\mathbf{s}_d$.

3.2 Phone Position Recognition

In order to recognize the phone wearing position of dataset X we train a classifier using datasets V and T (all phone positions) as training data. Instead of using just the phone position as class labels we generate tuples (m, p) which combine a particular transport mode label m and a phone position p. Since there are 8 different modes of transportation and 4 different phone positions, this results in $8 \times 4 = 32$ distinct class labels. The classification model is a Multi-Layer Perceptron (MLP) with two hidden layers (20 and 12 units, respectively). For each feature vector $\mathbf{f}_d$ the trained MLP computes posteriors $P(m, p|\mathbf{f}_d)$ from which we can easily derive $P(p|\mathbf{f}_d) = \sum_m P(m, p|\mathbf{f}_d)$. The phone wearing position of X is detected by maximizing $P(p|X)$, ie.

$$p_X = \underset{p}{\operatorname{argmax}} \prod_{d \in X} P(p|\mathbf{f}_d).$$

Note that this approach assumes that all data in X are from the same phone wearing position!

For the test dataset X our approach identified the phone position "Hips". This result is extremely important as the following step in the processing pipeline builds upon it and will produce poor results if the inferred phone position is wrong.

3.3 Phone Position and User-specific Transport Mode Classification

Relying on the result of the previous step we train a phone position-specific transport mode classification model. In order to do so, we remove all data from T which were not collected by the "Hips" phone and keep only the resulting subset T_{Hips}. Since the patterns in the data also depend on the user, we additionally use dataset V for model training. In our experiments we observed that keeping the data from *all* phone positions in V improves the model performance. The reason might be that V is very small, and the benefit of adding more user-specific data outweighs the negative impact of mixed phone positions.

3.4 Nearest Neighbour Smoothing

The data frames generated by segmenting sensor time series (e.g. data of a single journey) with a sliding window are often very similar to each other, in particular when the time between the data frames is short. This similarity can be explained by the fact that many factors influencing the patterns in the sensor signals remain constant or change slowly or very rarely: the user, the phone wearing position, the orientation of the phone, the transport mode, the particular vehicle, the pavement of the road, the driving (or walking) speed, etc. Contrarily, if some of these factors change, the similarity between data frames decreases, sometimes dramatically. This is the reason why user and phone position-specific models perform better than user and phone position-independent models. It also explains why shuffling the data frames before splitting them into a training and validation dataset can introduce a strong upward scoring bias to model validation results [9].

On the other hand, the inter-dependencies between data frames can also be used to improve classification performance. If the data frames are consecutive in time, Hidden Markov Model (HMM) smoothing can be applied as a post-processing step [7]. The transition probabilities reflect the dependencies between neighbouring frames and control how frequently the mode of transport can change. Unfortunately, in the present task the data frames in X are not consecutive in time. Restoring the original time series in the correct temporal order is not possible, because the step size of the sliding window used to generate the data frames is larger than the window size. We therefore introduce a method to leverage the dependencies among the data frames in X without relying on their temporal ordering. To explain our approach we will rephrase the findings above without referring to temporal proximity.

Supervised classification learning (in particular k-Nearest Neighbour classification) assumes that the instances in the test set have the same class label as the most similar instances in the training set. In practice, this assumption is often wrong, because of a "distributional shift" that arises when some of the factors influencing the distribution of the data differ between the training and the test data. Models relying too much on this assumption are said to overfit the training data. However, the assumption that the instances in a particular dataset have the same class label as the most similar instances in the *same* dataset often holds, because there are often groups of instances sharing many of the influencing factors (phone orientation, vehicle, pavement, speed, etc), one of which is the mode of transport. Therefore, we can assume that "smoothing" over neighbouring instances in X can increase classification performance.

To predict the label of a data frame $x_0 \in X$ we identify the 10 Nearest Neighbours, ie. 10 data frames $x_1, ..., x_{10} \in X$ with the smallest distance to x_0. The distance between two data frames a and b is computed as the *standardized Euclidean distance* between similarity feature vectors $\mathbf{s}_a$ and $\mathbf{s}_b$:

$$d(\mathbf{s}_a, \mathbf{s}_b) = \sqrt{\sum_i \frac{(s_{a,i} - s_{b,i})^2}{\sigma_i^2}},$$

where σ_i^2 is the sample variance of feature s_i in X. We then combine the class posteriors of the 10 nearest neighbors computed by the MLP as follows:

$$P(m|\mathbf{f}_x) = \prod_{k=0,...,10} P(m|\mathbf{f}_{x_k})$$

and identify the class label as

$$\hat{m}_x = \underset{m}{\operatorname{argmax}}\, P(m|\mathbf{f}_x).$$

Note that we used features $\mathbf{s}_x$ instead of $\mathbf{f}_x$ to compute similarity between data frames. We chose features we expected to be strongly governed by external factors which slowly or rarely change over time: the mean ambient air pressure depends on altitude and weather, the mean phone orientation depends on the wearing position and pose of the user, and the autocorrelation function of the acceleration signal is influenced by current speed, properties of the vehicle (engine, tires, seats) and the road surface. These features are therefore suitable for identifying groups of related data frames in X. However, due to their sensitivity to external factors we do not expect these features to be suitable for transport mode recognition across different datasets.

It is also worth mentioning that Nearest Neighbour smoothing is based on similar ideas as *graph-based semi-supervised learning* (SSL) [10], where a nearest neighbour graph is used to approximate manifolds in the feature space, which are assumed to consist of points having the same class label. However, there are some important differences to our method: graph-based SSL propagates information from instances with *given* labels to unlabelled instances. The class-labels are iteratively propagated through the entire graph, attempting to find a globally optimal and consistent solution. Nearest Neighbour smoothing, on the other hand, combines class-posteriors within a local neighbourhood of each instance to improve classification accuracy.

4 RESULTS

In this section we report the experimental results of the proposed classification model and test the effectiveness of the individual steps in the processing pipeline.

All experiments were conducted on a computer with Intel(R) Core(TM) i7-8650U CPU (8 cores, 1.9GHz) and 16GB RAM. We implemented the algorithms in Java using proprietary libraries for MLP training developed at our institute. The implementation is single-threaded, thus using only one CPU core at a time, and does not use GPU acceleration. Model training took about 2 minutes (time for reading the training data from disk and extracting features

not included) and storing the model on disk requires about 246kB. Labelling the test dataset took about 6 seconds (0.5s for feature extraction, 2.5 seconds nearest neighbour search, 3 seconds MLP predictions).

To validate our approach to phone position recognition we trained a classifier using only dataset T and predicted the phone positions of the subsets V_{Hand}, V_{Hips}, V_{Bag}, and V_{Torso}. In each of these experiments the phone position was correctly identified.

We tested the performance of the user and phone position-specific MLP by comparison with several baseline models. The first baseline model was trained using the entire dataset T (all phone positions) as training data. This approach is obvious since T was made available for the purpose of model training. However, the resulting model is specific to user 1 and contains no information about the characteristics of users 2 and 3 in dataset X. The specific phone position of the data in X is also contained in T, albeit mixed with three other phone positions. This baseline model achieves an average F1 score of 56.0 %. A second baseline model was trained using the entire dataset V, which contains data from the same users as X. Since we had to use V for both model training and validation, we estimated the classification performance by 4-fold cross-validation (without shuffling the data frames): in each iteration we excluded 25% of V from model training and used it for validation. With an average F1 score of 56.2% this user-specific model performs only slightly better than the first baseline. However, V is very small and since we used 4-fold crossvalidation only 75% of V could actually be used for training in each crossvalidation iteration. Assuming that a larger dataset would have allowed for a greater improvement, we combined T and V to train another classification model, which further increased the average F1 score to 56.8%. A phone position-specific model trained with only data from the Hips phone in T and V allowed a clear improvement and achieved an average F1 score of 59.9%. However, the proportion of data from set V is still very small. Combining the Hips data of T with the data from *all* phone positions in V clearly outperformed all previous models, achieving an average F1 score of 63.3%. An overview of the results is provided in Table 1.

The last step in the processing pipeline, Nearest Neighbour smoothing, was also validated by 4-fold cross-validation. The results are detailed in Table 2. In this experiment the average F1 score of our model was 75.3%. Similar to the results of the previous SHL recognition challenges, there are confusions between *Car* and *Bus*, and between *Train* and *Subway*. This indicates that these transport modes generate similar patterns in the sensor data. Interestingly, some transport modes are frequently predicted to be *Still*, in particular *Bus* and *Car*. The reason might be that the data frames are only 5 seconds long and the mislabelled frames correspond to short stops at traffic lights. Confusions between *Walk*, *Run*, and *Bike* might indicate that these classes are more strongly affected by user-characteristics (e.g. age, size, gait) and phone position than the other modes of transport.

Table 1: 4-fold cross-validation results of MLPs trained with different datasets.

training data	avg F1
T	56.0%
V	56.2%
$T + V$	56.8%
$T_{hips} + V_{hips}$	59.9%
$T_{hips} + V$	63.3%

Table 2: 4-fold cross validation results of the proposed classification model.

		predicted activity							
	(%)	Still	Walking	Run	Bike	Car	Bus	Train	Subway
actual activity	Still	18.8	0.6	0.0	0.0	0.2	0.5	0.3	0.2
	Walk	1.7	14.0	0.4	1.5	0.3	0.0	0.0	0.1
	Run	0.0	0.1	1.6	0.2	0.0	0.0	0.0	0.0
	Bike	0.4	0.1	0.0	7.3	0.1	0.4	0.0	0.0
	Car	2.3	0.0	0.0	0.0	7.9	1.9	2.0	0.1
	Bus	1.1	0.1	0.0	0.0	1.3	3.2	0.6	0.1
	Train	1.1	0.1	0.0	0.0	0.2	0.4	11.5	1.8
	Subway	0.2	0.0	0.0	0.0	0.1	0.0	2.1	12.7
	Recall:	90.7	77.2	81.6	87.2	55.8	50.9	76.0	84.1
	Precision:	73.5	93.1	80.0	79.6	77.6	49.9	69.7	84.7
avg. Recall:		**75.5**							
avg. F1:		**75.3**							

5 CONCLUSION

In practical applications, using smartphone sensor data to accurately and robustly recognize transport modes is a difficult task, because the characteristic patterns in the sensor data depend on physiological properties of the user and the phone wearing position (among other factors). The problem is further complicated when the transport mode has to be recognized in real-time, i.e. using only a few seconds of data. The SHL recognition challenge 2020 attempts to address this problem by using test data from an undisclosed phone wearing position and users that were not included in the training data. The data frames in the test data are only 5 seconds long and not consecutive in time. Restoring the temporal order by "data stitching" is prevented by introducing gaps between the frames.

However, we presented a processing pipeline tailored to the particular task of the 2020 SHL recognition challenge that does *not* meet the requirements of practical real-time applications. Instead of processing individual data frames as they would arrive in an online stream, we applied an offline batch process. This allowed us to leverage dependencies between the data frames in the test set. Moreover, knowing that all test data are from the *same* phone position we could detect the phone position and apply a position-specific classification model. The provided validation data, albeit small, allowed adapting the model to characteristics of the users in the test set. We showed empirically, that these steps can improve the classification performance drastically.

The recognition result for the testing dataset will be presented in the summary paper of the challenge [4].

REFERENCES

[1] Mathias Ciliberto, Francisco Javier Ordoñez Morales, Hristijan Gjoreski, Daniel Roggen, Sami Mekki, and Stefan Valentin. 2017. High reliability Android application for multidevice multimodal mobile data acquisition and annotation. In *Proceedings of the 15th ACM Conference on Embedded Network Sensor Systems*. ACM, 62.

[2] Hristijan Gjoreski, Mathias Ciliberto, Lin Wang, Francisco Javier Ordonez Morales, Sami Mekki, Stefan Valentin, and Daniel Roggen. 2018. The University of Sussex-Huawei Locomotion and Transportation Dataset for Multimodal Analytics with Mobile Devices. *IEEE Access* 6 (2018), 42592–42604. https://doi.org/10.1109/ACCESS.2018.2858933

[3] L. Wang, H. Gjoreski, M. Ciliberto, P. Lago, K. Murao, T. Okita, and D. Roggen. 2019. Summary of the Sussex-Huawei locomotion-transportation recognition challenge 2019. In *Proceedings of the 2019 ACM International Joint Conference and 2019 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers*. 849–856.

[4] L. Wang, H. Gjoreski, M. Ciliberto, P. Lago, K. Murao, T. Okita, and D. Roggen. 2020. Summary of the Sussex-Huawei locomotion-transportation recognition challenge 2020. In *Proceedings of the 2020 ACM international joint conference and 2020 international symposium on pervasive and ubiquitous computing and wearable computers*.

[5] Lin Wang, Hristijan Gjoreski, Mathias Ciliberto, Sami Mekki, Stefan Valentin, and Daniel Roggen. 2019. Enabling reproducible research in sensor-based transportation mode recognition with the Sussex-Huawei dataset. *IEEE Access* 7 (2019), 10870–10891.

[6] Lin Wang, Hristijan Gjoreski, Kazuya Murao, Tsuyoshi Okita, and Daniel Roggen. 2018. Summary of the sussex-huawei locomotion-transportation recognition challenge. In *Proceedings of the 2018 ACM international joint conference and 2018 international symposium on pervasive and ubiquitous computing and wearable computers*. 1521–1530.

[7] Peter Widhalm, Maximilian Leodolter, and Norbert Brändle. 2018. Top in the Lab, Flop in the Field?: Evaluation of a Sensor-based Travel Activity Classifier with the SHL Dataset. In *Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers*. ACM, 1479–1487.

[8] Peter Widhalm, Maximilian Leodolter, and Norbert Brändle. 2019. Ensemble-based domain adaptation for transport mode recognition with mobile sensors. In *Adjunct Proceedings of the 2019 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2019 ACM International Symposium on Wearable Computers*. 857–861.

[9] Peter Widhalm, Maximilian Leodolter, and Norbert Brändle. 2019. Into the Wild—Avoiding Pitfalls in the Evaluation of Travel Activity Classifiers. In *Human Activity Sensing*. Springer, 197–211.

[10] Xiaojin Zhu, John Lafferty, and Ronald Rosenfeld. 2005. *Semi-supervised learning with graphs*. Ph.D. Dissertation. Carnegie Mellon University, language technologies institute.