Cross-Location Transfer Learning for The Sussex-Huawei Locomotion Recognition Challenge

Vito Janko
vito.janko@ijs.si
Jožef Stefan Institute
Ljubljana, Slovenia

Martin Gjoreski
martin.gjoreski@ijs.si
Jožef Stefan Institute
Ljubljana, Slovenia

Carlo Maria De Masi
carlo.maria.demasi@ijs.si
Jožef Stefan Institute
Ljubljana, Slovenia

Nina Reščič
nina.rescic@ijs.si
Jožef Stefan Institute
Ljubljana, Slovenia

Mitja Luštrek
mitja.lustrek@ijs.si
Jožef Stefan Institute
Ljubljana, Slovenia

Matjaž Gams
matjaž.gams@ijs.si
Jožef Stefan Institute
Ljubljana, Slovenia

ABSTRACT

The Sussex-Huawei Locomotion Challenge 2019 was an open competition in activity recognition where the participants were tasked with recognizing eight different modes of locomotion and transportation. The main difficulty of the challenge is that the training data was recorded with a smartphone that was placed in a different body location than the test data. Only a small validation set with all locations was provided to enable transfer learning. This paper describes our (team JSI First) approach, in which we first derived additional sensor streams from the existing ones and on them calculated a large body of features. We then used cross-location transfer learning via specialized feature selection, and performed two-step classification. Finally, we used Hidden Markov Models to alter the predictions in order to take their temporal dependencies into account. Internal tests using this methodology yielded an accuracy of 83%.

CCS CONCEPTS

• **Computing methodologies → Transfer learning**; *Supervised learning*.

KEYWORDS

Activity recognition, machine learning, feature extraction, competition, smartphone, transfer learning

ACM Reference Format:
Vito Janko, Martin Gjoreski, Carlo Maria De Masi, Nina Reščič, Mitja Luštrek, and Matjaž Gams. 2019. Cross-Location Transfer Learning for The Sussex-Huawei Locomotion Recognition Challenge. In *Adjunct Proceedings of the 2019 ACM International Joint Conference on Pervasive and Ubiquitous Computing and the 2019 International Symposium on Wearable Computers (UbiComp/ISWC '19 Adjunct), September 9–13, 2019, London, United Kingdom.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3341162.3344856

1 INTRODUCTION

Smartphones, smart watches and other wearables have become ubiquitous. By analyzing sensor data acquired via such devices we can reason about the user's context which in turn enables personalized context-sensitive services. One of the most exploited types of context information is the user's activity. For this reason, activity recognition with wearable devices is a research topic studied by many researchers [8][13].

The typical approach to activity recognition with wearable devices is by applying machine learning to inertial sensor data. An important, but often neglected challenge, is that machine-learning models are normally location-dependent: a model that was trained to recognize the user's activity on data from inertial sensors in one body location (e.g., the wrist) performs quite poorly when used on data from another body location (e.g., a pocket on the hip). For example, a significant decrease in the performance was observed when a model trained using a smart watch worn on the left wrist was tested on data from the right wrist [6]. The issue is even more severe with smartphones, since they can be carried in many different locations. It can be tackled by location-specific models [4], but to train such models it is desirable to exploit cross-location data.

The Sussex-Huawei Locomotion-Transportation (SHL) recognition challenge 2019 [5][14] addressed exactly this problem. The goal of the challenge was to recognize eight modes of locomotion and transportation activities from inertial sensor data of a smartphone in a location-independent manner. More precisely, the goal was to recognize the user's activity from the data from a smartphone placed on the user's hand, but most of the provided training data was collected from smartphones on a torso, hips and in a bag.

The main machine-learning technique for dealing with such problems is transfer learning. In general, transfer learning focuses on storing knowledge gained while solving one problem and applying it to a different but related problem [16]. In the activity recognition domain, the most utilized transfer-learning techniques are transferring instances, transferring features and feature representations, transferring model parameters and transferring relational knowledge [2] [12]. Recently, with the advancement of end-to-end deep learning, transferring Convolution Neural Network (CNN) filters has also become an established transfer-learning technique. Although transfer learning was demonstrated to be feasible [1][10], it still remains a challenging task. For example, Morales et al. [11]

UbiComp/ISWC '19 Adjunct, September 9–13, 2019, London, United Kingdom
© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-6869-8/19/09…$15.00
https://doi.org/10.1145/3341162.3344856

Table 1: Summary of the provided datasets. "X" marks if the data is available for a particular smarthphone location or if the data includes the activity label.

	Bag	Torso	Hips	Hand	Labels	Days
SHL-Training	X	X	X		X	59
SHL-Validation	X	X	X	X	X	3
SHL-Test				X		20

used the DeepConvLSTM deep-learning model to transfer CNN layers across mobile AR datasets, sensor modalities and sensor locations. In their study, transferring filters across datasets was in most cases outperformed by domain-specific baseline models.

In our study we tested transferring instances, transferring features and transferring CNN filters from locations with a lot of training data to the target location with only a small amount of training data. While naive approaches to these types of transfer did not outperform the target-location model, the transfer did prove successful in the end. This can be attributed to the fact that the data from the target location (hand) was quite scarce compared to the data from the rest of the locations (torso, hips and bag). In this paper we will describe the approach that yielded the best performance based on our internal experimental results and was submitted for the competition (team JSI First). The approach consists of selecting features that perform best across locations (feature transfer), and combining instances from the target and other locations (instance transfer). Classical machine learning was used, since it yielded better results than deep learning, even when transfer of CNN filters was employed. An important element was also exploiting the temporal information by smoothing the predictions with a Hidden Markov Model (HMM).

2 SHL CHALLENGE DATA

The goal of the SHL challenge was to recognize eight modes of locomotion and transportation – *Car*, *Bus*, *Train*, *Subway*, *Walk*, *Run*, *Bike*, and *Still* – using inertial sensor data of a smartphone. The data was originally recorded using four smartphones worn at different on-body locations (*Hips*, *Torso*, *Bag*, *Hand*); however only a subset of all the data was provided by the challenge organizers.

The provided data came in three sets. The *SHL-Test* set (the SHL prefix distinguishes it from test data for specific experiments discussed later on) contained only *Hand* location and was unlabeled – correctly labeling it was the competition's goal. The *SHL-Training* set was the largest set, but it only contained data from non-hand locations. Finally, in the *SHL-validation* set, a little validation data was provided from all four locations including the *Hand* phone location. Overall, the challenge data comprised of 3 x 59 days of *SHL-Training* data (59 days of data for each of the three locations), 4 x 3 days of *SHL-Validation* data and 20 days of *SHL-Test* data – as summarized in Table 1.

The raw sensor data was sampled at a frequency of 100 Hz and it included data from the following sensors: acceleration (x, y and z), gravity (x, y and z), gyroscope (x, y and z), linear acceleration (x, y and z), magnetic field (x, y and z), orientation (x, y, z and w) and pressure. Notably, data from all sensors that could be used to identify the location of the user (e.g. GPS, Wi-Fi) was omitted. The data was segmented using 5-second windows and labels were provided per-sample. The distribution of the activities for the *SHL-Training* and *SHL-Validation* data is presented in Figure 1. The blue color represents the distribution of the *SHL-Training* labels. The light-green color represents the distribution of the *SHL-Validation* labels and the dark-green color is the intersection between the two datasets. From the figure it can be seen that the label distribution is quite similar between the two datasets. The biggest difference is in the distribution of the *Run*, *Bike*, *Train* and *Subway* classes.

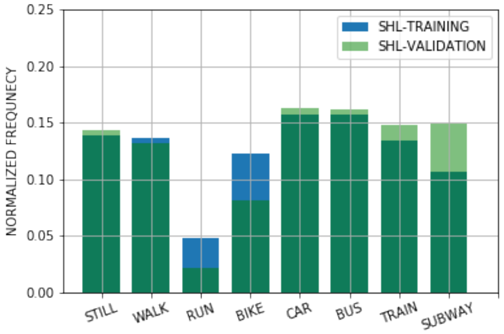


Figure 1: Distribution of the labels for the SHL-Training and SHL-Validation datasets.

3 PRE-PROCESSING AND FEATURES

In this work we opted for the use of classical machine learning, as it yielded better results then the deep learning. In order to employ it, we pre-processed the data and then from it calculated a large body of features.

3.1 Data ordering and split

The 5-second segments of the *SHL-Training* dataset were provided in the correct order, but the segments of the *SHL-Validation* and *SHL-Test* set were not – they were shuffled by the competition organizers. We assume that this was done in order to enforce that the competitors use a classification window of 5 seconds or less.

Shuffled validation data, however, presents a problem for any kind of training and testing that would be performed on this set. As an example: if we just split the *SHL-Validation* data in half to form an internal training and test set, many consecutive instances (that are very similar) would be one in the training and the other in the test set – leading to overfitting and a high reported accuracy, which would not translate to the *SHL-Test* set.

To remedy the issue, we designed an algorithm that can order the shuffled dataset based on the similarity of the segments. This ordering created clusters of data, that we believed were originally sequential, with no information on how the clusters follow each other. Note that this ordering was by no means flawless, but it was good enough to both split the validation data into two equally sized sets (*Validation1* and *Validation2*) and to apply the Hidden Markov Model smoothing on the data.

3.2 Deriving data streams

SHL dataset provides 20 different sensor streams, if we are individually counting each axis of the 7 provided sensors. From these original sensor streams it is possible to derive additional sensor streams that are useful for the AR. The subsequent steps treat these derived sensor streams like any of the original ones.

First derived sensor stream is the magnitude of the data. It was calculated for all the data that is coming from tree-axis sensors (acceleration, linear acceleration, gravity, magnetic field and angular velocity).

Second, additional sensor streams were created by rotating the accelerometer and magnetomer data from the phone's coordinate system to the "world" (North-East-Down) coordinate system. This could be useful for determining, for example, if the magnetic field is coming from above or below, as the same axis is always pointed upwards. In addition, they could be useful to match and compare the data from different phone locations, as these phones all point in the same direction after the rotation. This transformation was done by multiplying the current values with the coordinate system change matrix (details in [7]), using quarternions to determine the current orientation [9].

3.3 Features

In order to use classical machine learning, features had to be calculated from each five-second window of data. This window size was chosen as it was the largest possible given the limitations imposed by the nature of the competition and because our previous experience [7] on a similar problem showed that larger windows outperform the smaller ones – presumably due to infrequent activity transitions. Labels were calculated for each window as the most frequent label in that window.

Calculated features can be roughly categorized as being frequency-domain or time-domain and the following two subsections describe each category respectively. Altogether 858 features were calculated.

3.3.1 Frequency-domain features. These features were calculated using the power spectral density (PSD) of the signal, which is based on the fast Fourier transform (FFT). PSD characterizes the frequency content of a given signal and can be estimated using several techniques. The simplest one is to use a periodogram, which is obtained by taking the squared-magnitude of the FFT components. An alternative to a simple periodogram is the Welch's method, which is also widely used and commonly considered superior to periodogram. In our work, we opted to use the Welch's method to obtain the PSD. We have implemented the same frequency-domain features as in the last-years competition [7]: Three largest magnitudes of the FFT components, entropy of the the normalized FFT components and their energy.

3.3.2 Time-domain features. We have used time-domain features, that have proven themselves in our previous work [3][4] and previously won competitions [7][8]. These features were designed for accelerometer data and most of them were calculated only on the acceleration (and its derived) data streams. Some of the features were also calculated on the gyroscope data streams, however, some features such as *linear velocity* were left out as they have no semantic interpretation when calculated on non-acceleration data.

A description and analysis of the expert features can be found in our previous paper [4]. In summary, the magnitude data stream provided the information on the intensity of the activity, while the individual axes provided the information on the orientation of the device and subsequently on the position of the user. Some features come from statistics and describe the intensity and "shape" of the signal: the mean, variance, Pearson's correlation between axes, their covariance, skewness, kurtosis, quartile values and range between them. Others have a more physics-based interpretation, such as velocity and kinetic energy. The rest came from expert knowledge of the domain: the number and height of peaks in the window, signal's mean, its sum and squared sum, and the number of times the signal crosses its mean value.

4 METHOD

The main difference and at the same time the main problem of this challenge compared to the one previous year is that we were required to classify data from the *Hand* location, while having relatively small labeled training data for that location. We thus had to heavily rely on data recorded on different locations (*Torso*, *Hips*, *Bag*).

We first investigated to which degree is the data transferable from one location to another, and then based on that designed a two step classification pipeline that we used to create our predictions.

4.1 Cross-location training

Ideally, we would use all of the training and all of the validation data for training the final model. Such model could then generalize across all locations. In practice, however, this naive approach did not generate good results (Figure 5).

To investigate the issue we compared both the raw data and the features of the same time instance across different locations. Pressure data – as expectedly – closely matched. Raw data was loosely matching for the axis rotated into the universal coordinate system (Section 3.2). Most of the calculated (non-pressure) features, however, on average displayed a difference from one location to another of more than 20% of their value range. This persisted even if the values were first normalized into a [0 − 1] interval. This explained the poor performance of the models that were tested and trained on different locations.

The differences between feature values were not uniform – dynamic activities (*run*, *walk*, *bike*) displayed much greater disagreements than the static activities (*Bus*, *Car*, *Subway*, *Train*, *Still*). Focusing on the *Hand* location during static activities – the differences were greatest when the hand was rapidly moving, but this accounted for only roughly 3% of the data. These results were not entirely surprising, as the smartphone worn in hand creates completely different trajectories during dynamic activities than the smartphone worn in a trouser pocket. On the other hand, when resting in a vehicle, all body locations are subject to similar vehicle vibrations.

4.2 Proposed pipeline

Test dataset contained only data for the *Hand* location, and we thus focused on optimizing the pipeline for only this location. The main idea is that *SHL-Training* set is not suitable for learning the dynamic activities for the *Hand* location, however it can be helpful when learning the static activities (as described in Section 4.1). The resulting pipeline is schematically illustrated on Figure 2 and described in the following steps.

- Create a Random Forest classifier c_1 using *Hand* location in the *SHL-Validation* set using all features.
- Use c_1 to classify all instances in the *SHL-Test* set.
- Use HMM smoothing (Section 4.4) to alter the predictions, taking into account their sequence.
- Create a Random Forest classifier c_2 using *Hand* location in the *SHL-Validation* set and a non-hand location in the *SHL-Train* set. Only features selected by the feature selection as described in Section 4.3 were used.
- Use c_2 to re-classify all instances that were previously classified as either *Bus*, *Car*, *Subway*, *Train*, or *Still*.
- Use HMM smoothing for the second time, generating the final predictions.

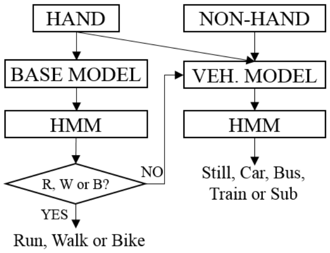


Figure 2: The proposed pipeline.

We expect the c_1 classifier to perform well when classifying dynamic activities (e.g. running, walking), as they are usually easy to learn even with a smaller dataset, especially if both the train and test data came from the recording of the same user. Nonetheless, the first HMM smoothing step is performed in order to further increase the accuracy, as the correct division between the static and dynamic activities is key to the correct re-classification.

Random Forest classifier was used as it gave good results in the last year competition, and was the most accurate in this year's internal tests. Note, that when we internally tested the pipeline, *Validation1* and *Validation2* sets were used for training and testing.

Feature selection is especially crucial when creating the c_2 classifier as some features are similar between locations, but most are not. We suspect that a different feature selection procedure for the c_1 would also be beneficial, but was skipped due to the time constraints of the competition.

4.3 Feature selection

Since a relatively high number of features was computed, feature selection was used to remove the ones that do not contribute to the accuracy of the model in order to reduce overfitting and speed up the training process. This process also removed the features that are very different from one location to another. Both the data in the *SHL-Train* set and in *SHL-Validation* set were used for the feature selection process. From the *SHL-Validation* set we always used the *Hand* location, while from the *SHL-Train* we used one of the non-hand locations, repeating the process for each one. In all cases only the data from the static activities was used, as the selected features were meant to be used only for the second step of the proposed pipeline (Section 4.2).

The feature selection procedure consists of three steps. In the first step, the mutual information between each feature and the label was estimated, where larger mutual information means higher dependency between the feature and the label.

After the features were sorted according to this value, correlated features were removed based on the Pearson correlation coefficient. This has shown that roughly half of the features are redundant, which was expected due to the number of features and similar data streams. To make the process computationally feasible, only 100 features were taken at a time, starting with those with the highest mutual information with the label. Correlation was then calculated for each pair. If the correlation was higher than a certain threshold (experimentally determined as 0.8), the feature with lower mutual information was discarded. After that, next 100 features were added and the correlation between each pair was calculated again.

In the final step, features were selected using a greedy "wrapper" algorithm. A Random Forest classifier was first trained using only the best scoring feature on the *SHL-Training* set. The trained model was used to predict labels for the *Validation1* set and the prediction accuracy was calculated (schematically in Figure 3). Then the second-best feature was added and the model was trained again. If the accuracy on the *Validation1* set was higher than without using this feature, the feature was kept. This procedure was repeated for all remaining features. This strict selection initially led to overfitting to the validation set (accuracy was much higher compared to the test set), so the condition for keeping a feature was made less strict: the feature was kept if the accuracy did not decrease by more than an experimentally set threshold. Using this rule, overfitting to the validation set was reduced.

Validation1 set was used instead of the whole validation set, so we could use *Validation2* set to verify that the model did not overfit. After the feature selections procedure was completed we switched the *Validation1* and *Validation2* set and repeated it. We then kept only features that were selected by both iterations.

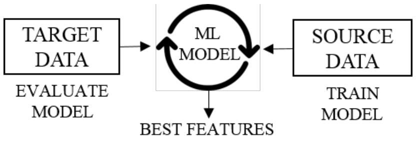


Figure 3: Cross-location feature matching.

4.4 Hidden Markov Model

Using only classical classification, all the windows are classified independently from one another. This approach discards all the information on temporal dependencies between them. If a user is currently on a train, for example, but the next window is classified as *Bus*, followed by another *Train* classification, it is far more likely for *Bus* to be a misclassification than a vehicle switch (Figure 4).

This motivated us to use an extra step after each classification, where the temporal information was taken into account. This was

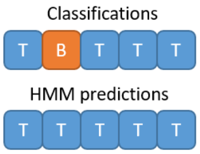
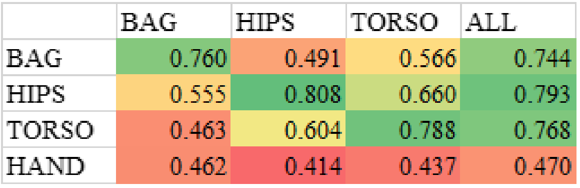


Figure 4: Top row shows a sequence of *Train* and *Bus* classifications. They are corrected using HMM smoothing into a sequence of only *Train* activities shown below.



	BAG	HIPS	TORSO	ALL
BAG	0.760	0.491	0.566	0.744
HIPS	0.555	0.808	0.660	0.793
TORSO	0.463	0.604	0.788	0.768
HAND	0.462	0.414	0.437	0.470

Figure 5: Accuracy of models trained on the *SHL-Training* set (different locations in columns) and tested on the *SHL-Validation* set (different locations in rows).

done using an HMM model. In this model the hidden states represent the actual activity, while the visible output represents the classified activities. The parameters of this model are the transition probabilities between the states and the probabilities of observed emissions in each state. The former can be estimated from the transition matrix of the *SHL-Training* set (matrix of probabilities that one activity is followed by another), while the latter from the confusion matrix on either the *Validation1* or *Validation2* set.

5 EXPERIMENTAL RESULTS

5.1 Cross-location training

We started with the naive approach of training a model on the *SHL-Training* set using data of individual phone locations, or merging the data from all three locations into one training set. The models were trained with the Random Forest algorithm. They were tested on all four locations of the *SHL-Validation* set. The results are shown in Figure 5. One can immediately see that the accuracy was the highest when the same location was used for training and testing, which was to be expected. It was quite low when different locations were used, which is also not a surprise. The most interesting set of results is for the model trained on all the *SHL-Training* data, including *Bag*, *Hips* and *Torso*, but not *Hand*. On all the locations but *Hand*, this model performed close to the location-specific models, which suggests that location-independent models are possible. Unfortunately it was not useful for the competition, though, since the result on *Hand* was very poor. We speculate that the reason is both the lack of *Hand* training data and the fact that the *Hand* location is more different from the others than the others are between themselves.

5.2 Proposed pipeline

We started our pipeline with the base model that classified all the activities. This model was trained using the Random Forest algorithm on all the features. During internal experiments, it was trained on both halves of the *SHL-Validation* set: when it was tasted on *Validation1*, *Validation2* was used for testing, and vice versa. The reported results are the average of both runs. The model that was eventually used to classify the *SHL-Test* set was trained on the whole *SHL-Validation* set.

Table 2: Accuracy of the classification after each step of the proposed pipeline. H - Hand, B - Bag, P - Hips

Location	H	T+H	B+H	P+H	P+T+B+H	Voting
Base	64					64
Smoothing 1	81					81
Vehicle	63	68	66	67	68	69
Smoothing 2	82	86	78	84	80	83

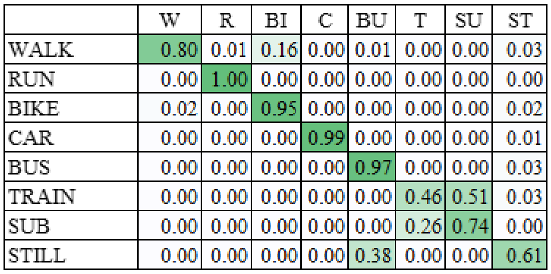
The accuracy of the base model is shown in the first row of Table 2. It is fairly low – only 64%. However, after HMM smoothing, in increases to respectable 81%, which is shown in the second row of Table 2. The confusion matrix for the smoothed model can be seen in Figure 6. The model did very well on *Walk*, *Run*, *Bike* and *Car*. The most confused classes were *Train* and *Subway*, which are indeed very similar. *Bus* was often confused with *Still*.

Since distinguishing the vehicles and *Still* proved problematic with a model trained on the small *SHL-Validation* set, we trained another set of models to distinguish the vehicle classes. Four models were built: three trained on each of the three locations in the *SHL-Training* set, and one trained on *Hand* data in the *SHL-Validation* set. The first three used features chosen with the cross-location feature selection in which half of the *SHL-Validation* set was used as the validation set in the wrapper-based feature selection. *Hand* data was also added to the training sets for these models. The feature selection and training was repeated twice, once on *Validation1* and once on *Validation2* set. The models that were eventually used to classify the *SHL-Test* set used the union of both feature sets. We also trained a model on the data from all three *SHL-Training* locations combined with *SHL-Validation Hand* data, and a majority-voting model aggregating the predictions of other models.

The results for the smoothed based model combined with unsmoothed vehicle models are shown in the third row of Table 2. The absence of smoothing of the vehicle predictions caused the accuracy to drop substantially compared to the previous row, but it was better than the accuracy of the base classifier. The *Hand*-only model performed the worst, whereas the *Torso* + *Hand* model performed the best. Apparently the motion of the torso is closest to the motion of the hand, although the differences were not large. The final row of Table 2 shows the results smoothed for the second time with a HMM model. This step confirmed the benefit of the vehicle mode, since the results are substantially better than after the first smoothing. *Torso* + *Hand* remained the best location, performing the same as the voting model. To classify the *SHL-Test* set we used a majority-voting ensemble of the models that were equally good or better than the *Hand* model: *Hand*, *Torso* + *Hand* and *Torso* + *Hips*.

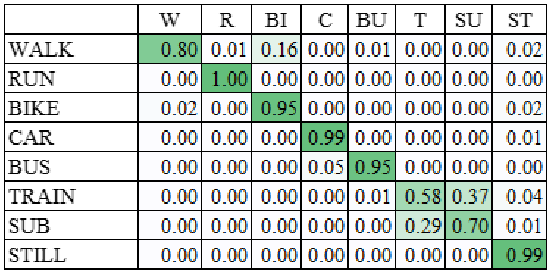
5.3 Feature selection

For the *Bag* location, 37 features were kept; 32 were kept for the *Torso* location and 40 for the *Hips* location. This presents only a small fraction of all features – most of the kept features came from



	W	R	BI	C	BU	T	SU	ST
WALK	0.80	0.01	0.16	0.00	0.01	0.00	0.00	0.03
RUN	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00
BIKE	0.02	0.00	0.95	0.00	0.00	0.00	0.00	0.02
CAR	0.00	0.00	0.00	0.99	0.00	0.00	0.00	0.01
BUS	0.00	0.00	0.00	0.00	0.97	0.00	0.00	0.03
TRAIN	0.00	0.00	0.00	0.00	0.00	0.46	0.51	0.03
SUB	0.00	0.00	0.00	0.00	0.00	0.26	0.74	0.00
STILL	0.00	0.00	0.00	0.00	0.38	0.00	0.00	0.61

Figure 6: Normalized confusion matrix for the base model with HMM smoothing. 2-fold cross-validation was used on the *SHL-Validation* data.



	W	R	BI	C	BU	T	SU	ST
WALK	0.80	0.01	0.16	0.00	0.01	0.00	0.00	0.02
RUN	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00
BIKE	0.02	0.00	0.95	0.00	0.00	0.00	0.00	0.02
CAR	0.00	0.00	0.00	0.99	0.00	0.00	0.00	0.01
BUS	0.00	0.00	0.00	0.05	0.95	0.00	0.00	0.00
TRAIN	0.00	0.00	0.00	0.00	0.01	0.58	0.37	0.04
SUB	0.00	0.00	0.00	0.00	0.00	0.29	0.70	0.01
STILL	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.99

Figure 7: Normalized confusion matrix for the final model with HMM smoothing. 2-fold cross-validation was used on the *SHL-Validation* data.

axis without the orientation – either magnitudes or de-rotated sensor streams. Most features were generated from the data from magnetometer sensor, with the acceleration data closely following.

6 CONCLUSION

Following the SHL Challenge 2018, which was fairly straightforward activity recognition – albeit with somewhat atypical (transportation) activities – the 2019 edition brought an interesting new twist. In the era of increasing availability of large datasets, transfer learning is a hot topic, so it is quite appropriate it was chosen for the challenge. The challenge also addressed a quite practical problem of the smartphone being carried in a variety of locations on the body. However, it was probably not entirely successful at achieving the stated objective of developing an activity-recognition model that is independent of the phone location. The reason is that the evaluation was done on a specific location – *Hand* – so the competitors presumably built models for that location (at least we did). Our experiments indicated that location-independent models are not very accurate, particularly on the Hand location which seems to be most different from others.

Our approach relied on the observation that the data is more similar across different locations if the user is stationary (either still or in a vehicle) than if the user is walking, running or biking. This encouraged us to use a two-step classification method that uses data from multiple locations when the activity is assumed stationary, and only hand data from hand otherwise.

The Hidden Markov Model smoothing had a great impact on the recognition quality, increasing the accuracy by roughly 15%. We thus encourage the use of this approach in similar domains where the activities are long on average. Given that we created an imperfect ordering of the *SHL-Test*, we expect the HMM to display a worse performance, slightly decreasing the overall accuracy. However, given the relatively high internal accuracy of 83% we still hope the results are good enough for a high placement at the competition. The recognition result for the testing dataset will be presented in the summary paper of the challenge [15].

REFERENCES

[1] Alberto Calatroni, Daniel Roggen, and Gerhard Tröster. 2011. Automatic transfer of activity recognition capabilities between body-worn motion sensors: Training newcomers to recognize locomotion. In *Eighth international conference on networked sensing systems (INSS'11)*. Eighth International Conference on Networked Sensing Systems (INSS'11).

[2] Diane Cook, Kyle D Feuz, and Narayanan C Krishnan. 2013. Transfer learning for activity recognition: A survey. *Knowledge and information systems* 36, 3 (2013), 537–556.

[3] Božidara Cvetković, Vito Janko, and Mitja Luštrek. 2015. Demo abstract: Activity recognition and human energy expenditure estimation with a smartphone. In *2015 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops)*. IEEE, 193–195.

[4] Božidara Cvetković, Robert Szeklicki, Vito Janko, Przemyslaw Lutomski, and Mitja Luštrek. 2018. Real-time activity monitoring with a wristband and a smartphone. *Information Fusion* 43 (2018), 77–93.

[5] 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.

[6] Martin Gjoreski, Hristijan Gjoreski, Mitja Luštrek, and Matjaž Gams. 2016. How accurately can your wrist device recognize daily activities and detect falls? *Sensors* 16, 6 (2016), 800.

[7] Vito Janko, Nina Reščiç, Miha Mlakar, Vid Drobnič, Matjaž Gams, Gašper Slapničar, Martin Gjoreski, Jani Bizjak, Matej Marinko, and Mitja Luštrek. 2018. A New Frontier for Activity Recognition: The Sussex-Huawei Locomotion Challenge. In *Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers*. ACM, 1511–1520.

[8] Simon Kozina, Hristijan Gjoreski, Matjaž Gams, and Mitja Luštrek. 2013. Efficient activity recognition and fall detection using accelerometers. In *International Competition on Evaluating AAL Systems through Competitive Benchmarking*. Springer, 13–23.

[9] Jack B Kuipers et al. 1999. *Quaternions and rotation sequences*. Vol. 66. Princeton university press Princeton.

[10] Marc Kurz, Gerold Hölzl, Alois Ferscha, Alberto Calatroni, Daniel Roggen, and Gerhard Tröster. 2011. Real-time transfer and evaluation of activity recognition capabilities in an opportunistic system. *machine learning* 1, 7 (2011), 8.

[11] Francisco Javier Ordóñez Morales and Daniel Roggen. 2016. Deep convolutional feature transfer across mobile activity recognition domains, sensor modalities and locations. In *Proceedings of the 2016 ACM International Symposium on Wearable Computers*. ACM, 92–99.

[12] Sinno Jialin Pan and Qiang Yang. 2009. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering* 22, 10 (2009), 1345–1359.

[13] Daniel Roggen, Alberto Calatroni, Mirco Rossi, Thomas Holleczek, Kilian Förster, Gerhard Tröster, Paul Lukowicz, David Bannach, Gerald Pirkl, Alois Ferscha, et al. 2010. Collecting complex activity datasets in highly rich networked sensor environments. In *2010 Seventh international conference on networked sensing systems (INSS)*. IEEE, 233–240.

[14] 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.

[15] Lin Wang, Hristijan Gjoreski, Ciliberto Mathias, Paula Lago, Kazuya Murao, Tsuyoshi Okita, and Daniel 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*. ACM.

[16] Jeremy West, Dan Ventura, and Sean Warnick. 2007. Spring research presentation: A theoretical foundation for inductive transfer. *Brigham Young University, College of Physical and Mathematical Sciences* 1 (2007), 32.