# Short Segment Random Forest with Post Processing using Label Constraint for SHL Recognition Challenge

**Hitoshi Matsuyama**
Graduate School of Engineering,
Nagoya University
hitoshi@ucl.nuee.nagoya-u.ac.jp

**Katsuhiko Kaji**
Faculty of Information Science,
Aichi Institute of Technology
kaji@aitech.ac.jp

**Kenta Urano**
Graduate School of Engineering,
Nagoya University
vrano@ucl.nuee.nagoya-u.ac.jp

**Nobuo Kawaguchi**
Institutes of Innovation for Future
Society, Nagoya University
kawaguti@nagoya-u.jp

**Kei Hiroi**
Graduate School of Engineering,
Nagoya University
k.hiroi@ucl.nuee.nagoya-u.ac.jp


*UbiComp/ISWC'18 Adjunct,*, October 8–12, 2018, Singapore, Singapore
ACM 978-1-4503-5966-5/18/10.
https://doi.org/10.1145/3267305.3267532

## Abstract

The bases of the approaches of UCLab(submission 1) towards SHL recognition challenge are using Random Forest and letting it select important features. Using accelerometer, gyroscope, magnetometer, gravity and pressure sensor as input data, features such as mean, variance, max, difference of max and min, and main frequency are calculated. We find that activities of Still, Train, and Subway are highly similar and hard to distinguish. To achieve robust recognition, we make predictions for every segment of 3 seconds and produce final prediction based on these predictions. Moreover, to deal with the case that one line contains two or more activities, we use a rule-based post processing to predict these activity labels. As a result, using the lines of last 20% in training dataset as validation set, predictions for 3-second segments have around 0.879 of F1-score and predictions for lines have around 0.942.

## Author Keywords

Random Forest;Signal Processing;Activity Recognition

## ACM Classification Keywords

I.5.4 [The ACM Computing Classification System (1998))]: Signal processing

## Introduction

Owing to the development of the downsizing of computers and the rapid spread of smartphones, we can easily record the information of our daily activities with application services like life logs. To realize automatic activity recording, we need to perform signal processing on sensor data and classify it into pre-defined activity classes.

Machine learning methods can be used for such purposes. There are many machine learning methods which can be used for activity recognition, e.g. kNN, k-means, Support Vector Machines[1, 2, 3] or Decision Tree. In machine learning, however, small number of training data sometimes leads to incorrect predictions. However, in this situation where we can acquire a lot of sensor data through our smartphones, vast amount of training data is available to train a classifier. Thus, to recognize activity class with higher accuracy using a number of sensor data, this Sussex-Huawei Locomotion-Transportation recognition challenge[1] is a good opportunity.

In this paper, we —UCLab(submission 1)— use Random Forest classifier[4], which effieicently learns with a large size of data[5]. It is also compatible with the multi-class classification problem. Moreover, we can easily check the contribution of each feature value. In this challenge, our laboratory make two submissions with different approaches, one is based on Random Forest which we describe in this paper, and the other is based on deep neural network[6]. Latest deep neural approach is discussed for solving classification problems[7, 8], however, our challenge in this paper is to fully utilize a traditional-machine-learning method that can compete with deep neural approach.

We propose a method which adopts two-stage strategy. First stage is learning process with shorter segments, 3-second in detail, than 1-minute line. Second stage is voting process using the classification results of 3-second segments to classify the corresponding 1-minute line. In this challenge, classes such as Train and Subway are similar to each other. However, these classes do not appear alternately in every 3 seconds. The second stage arranges these classes to produce robust results. In addition, for test dataset, rule-based post processing is used as second stage, to classify the lines containing two activities. Proposed method achieve 0.942 of F1 score when predicting the last 20 percent of lines of training data.

[1] http://www.shl-dataset.org/activity-recognition-challenge/

## SHL Dataset and Task Description

In this challenge, all of the participant teams are given the same datasets[9, 10, 11] and struggle to accomplish the task below.

### *SHL Dataset*

Each data in the given dataset is recorded by a single participant by a Huawei Mate 9 smartphone. In the time of record, the participant was performing on a daily basis, with the smartphone worn inside the front right pocket. The smartphone was logging the sensor data and give data of accelerometer, gyroscope, magnetometer, linear acceleration, gravity, orientation and ambient pressure.

The dataset is divided into two parts: training data and test data. The training data is composed of 271 hours and the test data consists of 95 hours. Both the training data and test data is composed of frames, created by segmenting the original data with a 1-minute length sliding window without overlapping. All the training data files contain a matrix of size 16310 lines and 6000 columns, and the test data files include a matrix of size 5698 lines and 6000 columns. Also, the activity classes are labeled to every value in each

**Table 1:** The duration of each activity class in training dataset

| Class | Duration (seconds) |
|---|---|
| Still | 137984 |
| Walk | 131209 |
| Run | 41146 |
| Bike | 125855 |
| Car | 148649 |
| Bus | 125296 |
| Train | 151023 |
| Subway | 117437 |

column in training data. There are 8 activity classes: Car, Bus, Train, Subway, Walk, Run, Bike, and Still. The numbers of lines for each activity class vary (shown in Table 1). Table 1 shows the numbers of lines for each class. We look at the top label of every line to get the numbers in Table 1. However, there are lines which contain two or more activity classes.

*Task*
Our purpose of this challenge is to recognize the 8 activity classes from the training data and make a prediction of activities of the test data. We have to develop an algorithm and a model which processes the training data and outputs the result of recognition for test data.

## Data Processing Overview

Overview of proposed method is shown in Figure 1. Before processing sensor data, we remove the lines which include more than one class from training data. Next, we divide one line into 20 blocks. The length of each block is 3-second. After that, each block is procesed to extract feature values. Thereafter feacute vectors are formed as compositions of features. In training, the labels are also processed into 20 blocks and used to train a Random Forest classifier.

After training, the classifier can output labels for 3-second blocks. For the test dataset, 3-second labels are sometimes not consistent. For example, Car and Bus appear alternately, which cannnot be considered as real activity. Therefore we perfom a rule-based post processing to the output labels (stated in *Post Processing using Label Constraint*).

*Line Division*
We divide each line into 20 blocks for the following reasons: in order to increase the number of training data and to grasp characteristics of short duration. 1-minute duration is too long to obtain characteristics of activities considering the signal cycles of activities such as Walk, Run, and Bike.

*Random Forest Classifier*
In this challenge, we use a Random Forest classifier. Random Forest classifier is an ensemble learning algorithm which has a multi-class classifier with multiple decision tree structures. In a Random Forest algorithm, independent decision trees are constructed for the bootstrap samples and identification is performed by integrating the results output from each decision tree.

We use 11 values from Acceleration data, 5 values from Angular velocity, 4 values from Magnetism, 3 values from Gravity and 2 values from Pressure (detailed explanation of each feature is described in *Feature Extraction*.) We set these 25 values as feature values for Random Forest Classifier.

When training a classifier, to achieve higher accuracy, we perform a grid-search using following parameters. Other parameters not listed here are set to the default values of the library (Scikit-learn's Random Forest). After gird-searching, the best parameters are: n_estimators = 28, criterion = gini, max_features = sqrt.

- n_estimators: 10, 11, 12, ..., 28
- criterion: gini, entropy
- max_features: srqt, log2, None[2]

## Feature Extraction

As the feature vectors, we use Acceleration, Angular velocity, Magnetism, Gravity and Pressure data. To process

[2]Setting None means max_features is equal to n_features which is the number of features in input data

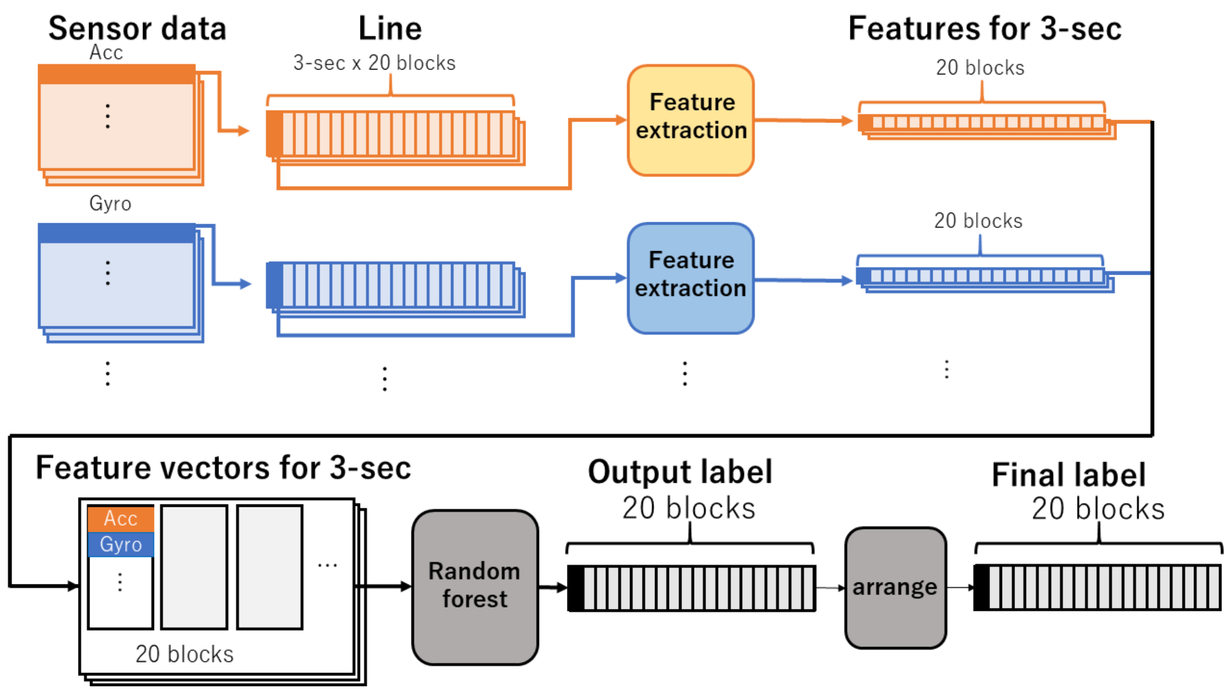


**Figure 1:** Data processing flow

the data effectively, we extract features stated below. Explanations about some special operations (normalization and fundamental frequency) follow the list.

- Acceleration:
  - mean of each axis after normalization
  - variance of each axis
  - variance of the norm
  - maximum value of the norm
  - fundamental frequency of the norm
  - mean of the norm after normalization
  - variance of the means of norm which is divided into 20 blocks (0.15 second x 20blocks)
- Angular velocity:
  - variance of each axis
  - variance of the norm

  - maximum value of the norm

- Magnetism:
  - variance of each axis
  - variance of the means of norm which is divided into 20 blocks (0.15 second x 20blocks)

- Gravity:
  - mean after normalization

- Pressure:
  - difference between maximum and minimum values
  - variance

*Normalization*
We divide acceleration data by $9.8$, the approximation of gravity-acceleration. This operation shrinks values to smaller ones and is expected to work as removal the influence of gravity-acceleration.

*Fundamental Frequency*
This approach aims to acquire the fundamental frequency of activities. Activities such as Walk, Run, Bike are assumed to have different frequency. Fast fourier transform (FFT) is performed to each 3-second blocks to acquire amplitude in frequency domain with the range of 0 Hz to 50 Hz. After that, the frequency which has the maximum amplitude value is used as the fundamental frequency.

## Data Processing for Test Dataset

When making predictions for test dataset, basically same operations as *Line Division* and *Feature Extraction*. However, the following operations are performed to deal with invalid values and to produce consistent results.

*Missing Values in Test Dataset*
Test dataset contains NaN values. Before procesing test dataset into features, we replace NaN values to the average value of before/after NaN.

*Post Processing using Label Constraint*
Test dataset has some lines with multiple activities. Thus using the labels of 3-second blocks leads to lower accuracy. We add a rule-based label arrangement. The rules which are applied to the 20 labels for 3-second blocks are shown below:

1. If the nuber of the most common label is $\geq$ 15, the most common label is adoped.
2. If top 2 most common labels are both in "Bike, Car, Bus, Train, Subway", the most common label is adoped.
3. If top 2 most common labels are both in "Bike, Walk", the most common label is adoped.
4. If none of above conditions applies, the most common label overwrites the labels after 3rd position.

## Result

*Evaluation*
In this SHL recognition challenge, evaluation metric is F1-score. To evaluate the F1-score of proposed method, we split the given training dataset into two parts: the first 80 percent for training, and the latter 20 percent for validation. We first train the classifier with the training data and then make a prediction for validation data. The parameters used in training is the best ones described in *Random Forest Classifier*. The confusion matrix for the valivation data is shown in Table 2.

**Table 2:** Confusion matrix of the classifier for train data

| | | Predicted | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | still | walk | run | bike | car | bus | train | subway |
| Actual | still | 440 | 4 | 1 | 8 | 1 | 8 | 5 | 6 |
| | walk | 7 | 410 | 0 | 1 | 1 | 0 | 1 | 1 |
| | run | 0 | 0 | 150 | 0 | 1 | 0 | 0 | 0 |
| | bike | 10 | 6 | 0 | 400 | 0 | 1 | 0 | 1 |
| | car | 0 | 7 | 1 | 2 | 480 | 5 | 1 | 1 |
| | bus | 6 | 7 | 1 | 2 | 14 | 380 | 3 | 0 |
| | train | 5 | 2 | 0 | 2 | 2 | 2 | 430 | 24 |
| | subway | 4 | 5 | 0 | 0 | 0 | 3 | 40 | 370 |

When we evaluate the prediction of validation data with 20 predicted labels per a line, we get F1-score of 0.879. After the post processing, F1-score improves to 0.942.

As shown in Table 2, most activities are classified correctly. Its F1-score is 0.942. Looking at each result, Walk and Run are classified almost perfectly. However, activities with small movement (Still, Car, Bus, Train, and Subway) are not classified correctly. Particularly, classifications for Train-against-Subway is difficult.

*Discussion*
In Random Forest, we can get the feature importance[12] of a classifier. Looking at the feature importance of our classifier, features from sensor axis-y mainly contributed for the classification. That is because the direction of the smartphone is mostly vertical and sensor axis-y reflects the characteristics of each movement the most. Axis-x did not contribute to the classification. It is because the direction of sensor axis-x is almost horizontal and awkward to get information about movements. When the participant is not in a move, features from magnetism and angular velocity are helpful for the classification. They tell the classifier the characteristics about the changes of environments and the movements of vehicles such as a subway starts running.

According to the feature importance, the fundamental frequency was not important for the classification. Using the fundamental frequency as a feature is besed on the idea that each activity has a particular frequency and it appears as the fundamental frequency. However, this approach did not play an important role in the classification. The possible reasons are the followings. (1) 3-second block is not enough to grasp the characteristics of activities. (2) The fundamental frequencies of Walk, Bike, and Run are similar to each other and confuse the classifier. For the second reason, the same thing can be argued to the fundamental frequencies of Still, Car, Bus, Train, and Subway.

## Computational Resources
We used a workstation with the following specs: AMD Ryzen Threadripper 1950X (3.4GHz, 16cores/32threads), 64GB DDR4 RAM. The software used for implementation was Scikit-learn 0.19.1 with Python 3.6.4. It took a few minutes to train 1 model and a few hours to perform a grid-search.

## Conclusion
In this paper, we developed a method to classify 8 activities: Still, Walk, Run, Bike, Car, Bus, Train, Subway with Random Forest classifier. At first, we removed the multi-labeled lines from the training data. We divide 1-minute lines into 3-second blocks before training a classifier. We extracted features from acceleration, angular velocity, magnetism, gravity, and pressure. After predicting the labels for 3-second blocks, post processing using label constraint is performed. This process is based on the intution such as Car and Bus do not appear alternately in every second. To evaluate our method, we split the given training dataset into two parts: the first 80 percent for training data, and the latter 20 percent for validation data. Using the proposed method, we

got the F1-score 0.942 for the validation data. However, we have to improve the method to classify activities which are hard to classify, especially Train-against-Subway, to raise our score. The recognition result for the testing dataset will be presented in the summary paper of the challenge[13].

## ACKNOWLEDGEMENT

This work was partially supported by JSPS KAKENHI Grant Number JP17H01762.

## REFERENCES

1. V. N. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, 1995.
2. V. N. Vapnik. An Overview of Statistical Learning Theory. *IEEE Transactions on Neural Networks*, 10(5):988–999, 1999.
3. V. N. Vapnik. *The Support Vector Method of Function Estimation*, pages 55–85. Springer US, 1998.
4. L. Breiman. Random Forests. *Machine Learning*, 2(1):5–32, Oct 2001.
5. R. Genuer, J.-M. Poggi, C. Tuleau-Malot, and N. Villa-Vialaneix. Random Forests for Big Data. *Big Data Research*, 9:28–46, 2017.
6. Y. Yuki, J. Nozaki, K. Hiroi, K. Kaji, and N. Kawaguchi. Activity Recognition using Dual-ConvLSTM Extracting Local and Global Features for SHL Recognition Challenge. In *International Workshop on Human Activity Sensing Corpus and Its Applications (HASCA2018)*. ACM, 2018.
7. D. E. Rumelhart, G. E. Hinton, and R. J. Williams. *Learning Representations by Back-propagating Errors*, pages 533–536. Nature Publishing Group, 10 1986.
8. Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation Applied to Handwritten Zip Code Recognition. *Neural Computation*, 1(4):541–551, Dec 1989.
9. H. Gjoreski, M. Ciliberto, L. Wang, F. J. Ordóñez Morales, S.Mekki, S.Valentin, and D. Roggen. The University of Sussex-Huawei Locomotion and Transportation Dataset for Multimodal Analytics with Mobile Devices. *IEEE Access*, pages 1–1, 2018.
10. H. Gjoreski, M. Ciliberto, F. J. Ordoñez Morales, D. Roggen, S. Mekki, and S. Valentin. A Versatile Annotated Dataset for Multimodal Locomotion Analytics with Mobile Devices. In *Proceedings of the 15th ACM Conference on Embedded Network Sensor Systems*, volume 4 of *SenSys '17*, pages 61:1–61:2, 2017.
11. M. Ciliberto, F. J. Ordoñez Morales, H. Gjoreski, D. Roggen, S. Mekki, and S. Valentin. High Reliability Android Application for Multidevice Multimodal Mobile Data Acquisition and Annotation. In *Proceedings of the 15th ACM Conference on Embedded Network Sensor Systems*, volume 5 of *SenSys '17*, pages 62:1–62:2, 2017.
12. L. Breiman. Using Models to Infer Machanisms, 2002. `http://oz.berkeley.edu/users/breiman/wald2002-2.pdf`.
13. L. Wang, H. Gjoreski, K. Murao, T. Okita, and D. Roggen. Summary of the Sussex-Huawei Locomotion-Transportation Recognition Challenge. In *International Workshop on Human Activity Sensing Corpus and Applications (HASCA2018)*, 2018.