

Indexed by

Scopus®

PREDICTING THE TYPE OF PHYSICAL ACTIVITY FROM TRI-AXIAL SMARTPHONE ACCELEROMETER DATA

DOAJ
DIRECTORY OF
OPEN ACCESS
JOURNALS

Katarina Pavlović

University of Belgrade,

Faculty of Biology, Belgrade,

Serbia

Crossref

ROAD
DIRECTORY OF OPEN ACCESS
RESEARCH RESOURCES

Key words: machine learning, digital phenotyping, human activity recognition

KoBSON

Cite article:

Katarina, P. [2021]. Predicting the type of physical activity from tri-axial smartphone accelerometer data. *Journal of Applied Engineering Science*, 19(1) 148 - 153. DOI:10.5937/jaes0-27166

SCINDEKS
Srpski citatni indeks

Online access of full paper is available at: www.engineeringscience.rs/browse-issues

Google
Scholar

PREDICTING THE TYPE OF PHYSICAL ACTIVITY FROM TRI-AXIAL SMARTPHONE ACCELEROMETER DATA

Katarina Pavlović*

University of Belgrade, Faculty of Biology, Belgrade, Serbia

Development of various statistical learning methods and their implementation in mobile device software enables moment-by-moment study of human social interactions, behavioral patterns, sleep, as well as their physical mobility and gross motor activity. Recently, through the use of supervised Machine Learning, Human Activity Recognition (HAR) has found numerous applications in biomedical engineering especially in the field of digital phenotyping. Having this in mind, in this research in order to be able to quantify the human movement activity in situ, using data from portable digital devices, we have developed code which uses Random Forest Classifier to predict the type of physical activity from tri-axial smartphone accelerometer data. The code has been written using Python programming language and Anaconda distribution of data-science packages. Raw accelerometer data was collected by using the Beiwe research platform, which is developed by the Onnela Lab at the Harvard T.H. Chan School of Public Health. Tuning has been performed by defining a grid of hyperparameter ranges, using Scikit-Learn's Randomized Search CV method, randomly sampling from the grid and performing K-Fold CV with each combination of tested values. Obtained results will enable development of more robust models for predicting the type of physical activity with more subjects, usage of different hardware, various test situations, and different environments.

Key words: machine learning, digital phenotyping, human activity recognition

INTRODUCTION

In order to demonstrate the usefulness of Human Activity Recognition (HAR) methods based on supervised Machine Learning, the objective of the research was to create a code which predicts the type of physical activity (e.g., walking or climbing stairs) from tri-axial smartphone accelerometer data. It is known that, beside the accelerometer, smartphones contain numerous other sensors, such as gyroscope, GPS, and magnetometer, that can be used to classify movement. All these sensors can be exploited for Human Activity Recognition (HAR) through the use of supervised Machine Learning. As a result, we can use smartphones for various purposes, such as remote health monitoring of disabled and elderly patients while they perform regular activities throughout the day, or just for reminding us to be more active. Furthermore, the use of these devices for monitoring our actions could assist us to better decide about our future activities [1]. Quantification of the behavior in situ using data from portable digital devices is also important for phenomics (the systematic study of phenotypes on a genome-wide scale), since digital phenotyping can be used to correlate mobility and quality of patient's life with spine disease [2], or to capture novel recovery metrics after cancer surgery [3] as well as to predict, diagnose, monitor and develop treatments for brain disorders [4]. A problem of predicting human activity by classifying sequences of accelerometer data using sensors from smartphones and wearable devices has been analyzed by numerous researchers [5-7]. Identifying the actions carried out by one or more subjects through gathering and analyzing context information about the user's state

and his surrounding environment can be performed by the exploitation of environmental and on-body sensors, and distributed computing resources. This might be a challenging problem due to: a large number of observations produced each second, the temporal nature of the observations, and the lack of a clear way to relate accelerometer data to known movements [8]. Some of the Machine Learning models that have been already used for the recognition of activities are Movelet Method [9], Support Vector Machines (SVMs) [10], Decision Trees [11], Naive Bayes [12] and Markov chains [13]. It should be noticed that although Machine Learning models could be fit to training data, they could not be generalized with sufficient accuracy on data from subjects not included in the training set. As a result, an insufficient accuracy of the model can occur, which may be the reason for potential users to mistrust, ignore and remove the considered phone app. Taking this into account, our approach first compares the accuracy of Random Forest Classifier and Multinomial Logistic Regression as models for classification and then further exploits Random Forest Classifier to predict the type of physical activity from tri-axial smartphone accelerometer data.

METHODS

Taking into account that smartphone accelerometers are very precise, as well as that different physical activities give rise to different patterns of acceleration, in order to build a model for predicting the type of the user's physical activity, we have used sensor based HAR. Sensor based HAR collects the motion data from smart sensors such as accelerometers [14], while the model for pre-

dicting the type of the user's physical activity is based on minimizing the discrepancy between user's predicted and true activity.

Suppose we have a set A of predefined activities (in our case those activities are standing, walking, stairs down, stairs up):

$$A = \{A_i\}_{i=1}^m \quad (1)$$

where m denotes the number of activity types. In order to capture activity information we can use a sequence of sensor readings:

$$S = \{d_1, d_2, \dots, d_t, \dots, d_n\} \quad (2)$$

where d_t denotes the sensor reading at time t .

If the true activity sequence is

$$A^* = \{A_j^*\}_{j=1}^n \quad A_j^* \in A \quad (3)$$

where n denotes length of the sequence, than building a model F which predicts the activity sequence based on a sensor readings:

$$\hat{A} = \{\hat{A}_j\}_{j=1}^n = F(s), \quad \hat{A}_j \in A \quad (4)$$

means to learn a model F by minimizing the discrepancy between predicted activity and the truth activity. In order to predict the type of physical activity, we have developed the code using Python programming language and Anaconda distribution of data-science packages. The input data used for training in our research consisted of two files. The first file (`train_time_series.csv`) contained the raw accelerometer data, while the second one (`train_labels.csv`) contained the activity labels, and has been used to train the model. Because the accelerometers are sampled at high frequency, the labels in `train_labels.csv` were only provided for every 10th observation in `train_time_series.csv`.

Raw accelerometer data were collected by using the Beiwe research platform. The Beiwe research platform has been developed by the Onnela Lab at the Harvard T.H. Chan School of Public Health in order to collect and analyze raw sensor and phone use data for biomedical research. It contains three cloud-based components - for collecting data, managing studies and performing data analysis, primarily at the Harvard Medical School.

Raw accelerometer data file has the following format: time stamp, UTC time, accuracy, x, y, z. Since x, y, and z, corresponded to measurements of linear acceleration along each of the three orthogonal axes, we could use them for separating different physical activities, such as standing, walking, going down or going up. Different activities have been numbered with integers (1 = standing, 2 = walking, 3 = stairs down, 4 = stairs up).

First, measurements of linear acceleration (along each of the three orthogonal axes x, y, z) from every 10th observation in `test_time` were appended to the `train_labels` data frame. For predicting the type of physical activity from tri-axial smartphone accelerometer data, first, we compared the accuracy of Random Forest and Multinomial Logistic Regression models for classification of

our type of data. Unlike Multinomial Logistic Regression, which generalizes Logistic Regression to multiclass problems and where the predictions are transformed using the logistic function, random forest is an ensemble learning method for classification and regression which is based on constructing a multitude of decision trees at training time, where the predictions are either the mode of the classes (classification) or mean prediction (regression) of the individual trees.

In order to find out which model, Random Forest or Logistic Regression, is better for our type of classification, the cross-validated accuracy for these two models was calculated. After selecting the better model, tuning of its parameters was performed. Tuning was done by defining a grid of hyperparameter ranges, using Scikit-Learn's `RandomizedSearchCV` method, randomly sampling from the grid and performing K-Fold CV with each combination of values.

Then, in order to test the accuracy of the model before applying it on the final testing dataset, data from previously modified `train_labels` data frame was divided into train and test subsamples. The train subset contained 80% of the data in `train_labels` data frame, while the test subset contained the other 20%. This type of data separation was performed in order to prevent overfitting of the model. Although the intention of our article is to demonstrate the usefulness of the presented method for general use, it should be emphasized that if we had optimized the model for the whole `train_labels` data, then our model would score very well on this set, but would not be able to generalize to new data, used for final testing.

The data used for final testing of the code consisted of two files. The first file, called `test_time_series.csv`, contained the accelerometer data and the goal was to predict the corresponding type of the activity for every 10th observation in this time series. Class predictions were made using measurements of linear acceleration from `test_time_series` as testing data for Random Forest classifier. The predictions were then added to the file called `test_labels.csv`, which consisted of the timestamps corresponding to every 10th observation in `test_time_series.csv`.

The code is presented as a supplementary material to this report (provided at the end of the report).

RESULTS

Physical activity, which is usually defined as "any bodily movement produced by skeletal muscles that requires energy expenditure" [15], offers an attractive target for the application of acceleration sensors, which cover an amplitude range of -12 to 12 g and a frequency range of at least 0 to 20 Hz [16]. The basic mechanism of acceleration measuring can be explained by a mass-spring system, where a mass is displaced when acceleration is applied, generating a force in a spring connected to the mass [17]. Accelerometers can be classified as piezoresistive, piezoelectric and differential capacitive accelerometers, depending on the method of signal transduction

[18]. High accuracy accelerometers, such as ActiGraph (ActiGraph LLC) and StepWatch (modus health llc) have been considered as some of the most promising tools for the assessment of physical activity under free-living conditions. It should be mentioned that accelerometers, when used alone, can produce important errors in the process of motion analysis, since they cannot measure rotation around the vertical direction [17,18]. Activity recognition is an interesting problem for Machine Learning, because each user performs the same activity slightly differently. Therefore, it is important to find a method that can generalize the important features of an activity rather than the specifics of how a particular user performs the activity.

In order to verify which model, Random Forest Classifier or Logistic regression, has better accuracy of predicting the type of physical activity, the ratio of 10-fold Cross Validation Testing results for both models was calculated. If Random forest Classifier has better accuracy, this ratio is greater than 1 and thus testing results will occupy the area above the diagonal line in Figure 1. Comparison of the results of the 10- fold cross-validation testing for both models is shown in Figure 1. It can be noticed that all the testing results are in the area above the diagonal line, which leads to the conclusion that Random Forest Classifier has better accuracy than Logistic Regression. Therefore this model is better suited for predicting the type of physical activity from tri-axial smartphone accelerometer data.

For hyperparameter tuning, we performed many iterations of the entire K-Fold CV process, each time using different model settings. The best parameters obtained this way, are the following:

```
'n_estimators': 300,
'min_samples_split': 10,
'min_samples_leaf': 3,
```

```
'max_features': 'sqrt',
'max_depth': 15,
'bootstrap': True
```

Using the train subset of the train_labels data frame for fitting and using the test subset for testing the model, predicted accuracy was 62,67%. It should be mentioned that one of the factors that influences the accuracy of the model is the distribution of the data in the training dataset. As it can be seen from Figure 2, acceleration data is not represented equally, which implies that some kind of renormalization of the statistical data should be performed. Furthermore, it should be noticed that the train_labels data does not represent all the activities equally. Unequal distribution of different physical activities through time is shown in Figure 3, where, on x-axis, timestamps from train_labels dataset were used as time variable, whereas, on y-axis, types of activities were represented as their corresponding numeric labels (1=standing, 2=walking, 3=stairs down, 4=stairs up). The majority of data (56.8 %) corresponds to walking, while 7.2 % corresponds to standing and about 12.53 % and 23.47 % corresponds to upstairs and downstairs, respectively. This might be the reason why predictions for walking activities occur more often than predictions for activities such as going downstairs, upstairs or standing. This is even more evident when using the whole train_labels data for fitting the Random Forest model and afterwards while making the final predictions for the test_labels dataset. In those cases, the accuracy of the model drops to 45,6%. It should be mentioned that this decrease of the accuracy might also occur due to significant differences between subjects, from whom train_labels and test_labels dataset have been collected.

All this implies that, in order to improve the accuracy of the model, it is important to quantitatively score the model in a way that encourages accuracy of all activities. It should also be noticed that several studies suggested the existence of two major problems with signal validity, obtained by using accelerometer sensors. The first problem is the location of sensors, while the second one

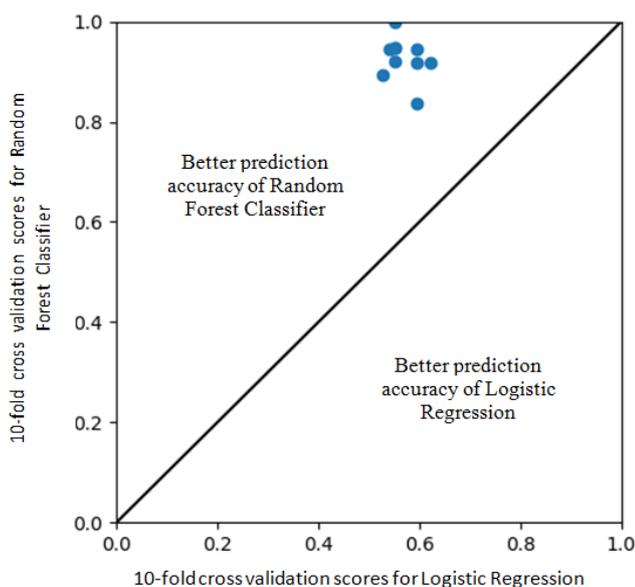


Figure 1: Comparison of the results of cross-validation testing for Logistic Regression and Random Forest Classifier

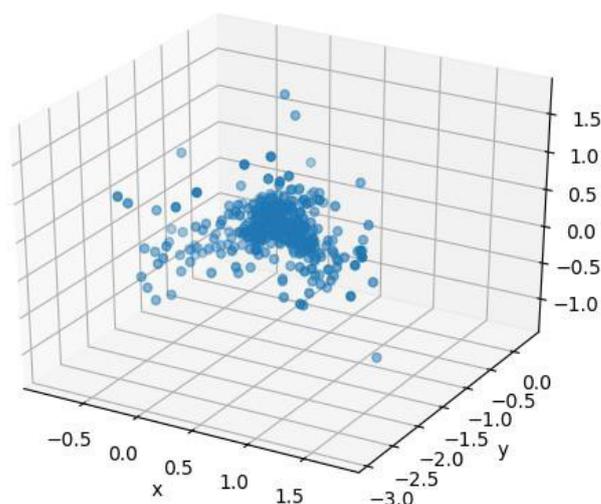


Figure 2: Acceleration data

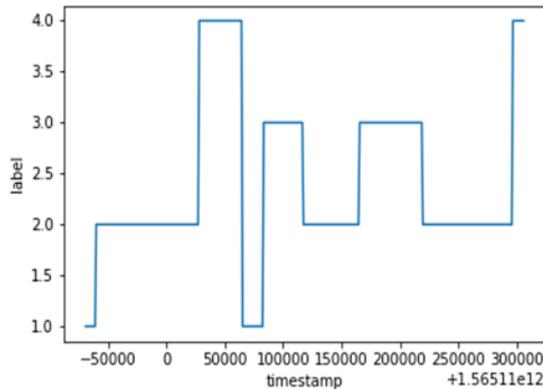


Figure 3: Unequal distribution of different physical activities through time

is reliability of the detected events. As a result, detection of abrupt changes in frequencies or attenuation in amplitude, both of which are characteristic features of freezing, the sensitivity and specificity of the event might be less than satisfactory [19].

Therefore, further improvements of the applied model may be achieved by incorporation of other public datasets to create a more robust model with more subjects, as well as with wider usage of different hardware, different test situations and different environments.

CONCLUSION

In this research, in order to demonstrate the usefulness of supervised Machine Learning methods in the field of digital phenotyping, the code for predicting the type of physical activity from tri-axial smartphone accelerometer data has been created and tested. Comparison of the accuracies of Random Forest and Logistic Regression models pointed out that the use of the Random Forest classifier might be better solution for the HAR applications. Therefore the code, based on application of Random Forest as the method for classification and for predicting four types of physical activity (standing, walking, stairs down, stairs up) is created. Raw accelerometer data was collected by using the Beibe research platform. The results of the final code testing have shown that the accuracy of the model is 45,6%. This accuracy of the code might be the result of unequal distribution of different physical activities in training dataset and also the consequence of the differences between subjects from whom training and testing data have been collected. In order to obtain better predictions than the ones achieved by the presented model, some other deep learning methods such as Recurrent Neural Networks and Convolutional Neural Networks might be suggested.

ACKNOWLEDGEMENT

The author would like to express her gratitude to Harvard University for launching the online learning platform *Using Python for research*, Onnela Lab at the Harvard T.H. Chan School of Public Health for developing open-source version of Beibe research platform and prof. Onnela for his exceptional online tutorials.

REFERENCES

1. Cook, D. J., Das, S. K. (2012),: "Pervasive computing at scale: Transforming the state of the art"; *Pervasive Mobile Computing*; 8 22–35
2. D.J. Cote, I.Barnett, J.P. Onnela, T.R.Smith (2019), *Digital Phenotyping in Patients with Spine Disease: A Novel Approach to Quantifying Mobility and Quality of Life World Neurosurgery* Volume 126, Pages e241-e249
3. N.Panda , I Solsky, E Huang, C Lipsitz, J Pradarelli, D Delisle, J Cusack, M Gadd, C Lubitz, J Mullen, M Qadan, B Smith, M Specht, A Stephen, K Tanabe, A Gawande, JP Onnela, (2020) *A Haynes Utilizing smartphones to capture novel recovery metrics after cancer surgery JAMA Surg.*;155(2):123-129. doi:10.1001/jamasurg.2019.4702
4. C Stroud, JP Onnela* and H Manji (2019) *Harnessing digital technology to predict, diagnose, monitor, and develop treatments for brain disorders Digital Medicine* 2:44
5. D.Anguita, A. Ghio, L. Oneto, X.Parra, J. L. Reyes-Ortiz. (2013) *Energy Efficient Smartphone-Based Activity Recognition using Fixed-Point Arithmetic. Journal of Universal Computer Science. Special Issue in Ambient Assisted Living: Home Care. Vol. 19, Issue 9*
6. D. Anguita, A. Ghio, L. Oneto, X. Parra, J. L. Reyes-Ortiz. *Human (2012), Activity Recognition on Smartphones using a Multiclass Hardware-Friendly Support Vector Machine. 4th International Workshop of Ambient Assited Living, IWAAL 2012, Proceedings. Lecture Notes in Computer Science pp 216-223*
7. J.L. Reyes-Ortiz, A. Ghio, X. Parra-Llanas, D. Anguita, J. Cabestany, A. Català. *Human Activity and Motion Disorder Recognition: Towards Smarter Interactive Cognitive Environments. 21th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, ESANN 2013. Bruges, Belgium 24-26 April 2013.*
8. Jason Brownlee *How to Model Human Activity From Smartphone Data, Mashine Learning Mastery*, <https://machinelearningmastery.com/how-to-model-human-activity-from-smartphone-data/>
9. Emily Huang, Jukka-Pekka Onnela *Activity Classification Using Smartphone Gyroscope and Accelerometer Data Published in ArXiv 2019 Computer Science*
10. Vapnik, V. N.: *The nature of statistical learning theory*; Springer-Verlag New York, 1995
11. Maurer, U., Smailagic, A., Siewiorek, D., Deisher, M.: "Activity recognition and monitoring using multiple sensors on different body positions"; *International-Workshop on Wearable and Implantable Body Sensor Networks*; 2006

12. Jatoba, L. C., Grossmann, U., Kunze, C., Ottenbacher, J., Stork, W.: "Context-aware mobile health monitoring: Evaluation of different pattern recognition methods for classification of physical activity"; International Conference of the IEEE Engineering in Medicine and Biology Society; (2008)
13. Mannini, A., Sabatini, A. M. (2010), Machine learning methods for classifying human physical activity from on-body accelerometers, *Sensors*, 2, 1154–1175
14. J. Wang, Y. Chen, S. Haoc, X. Peng, L. Hu (2018) Deep Learning for Sensor-based Activity Recognition: A Survey *Pattern Recognition Letters* 1-9
15. C. J. Caspersen, K. E. Powell, and G. Christenson, (1985) Physical activity, exercise and physical fitness: definitions and distinctions for health-related research, *PublicHealth Reports*, vol. 100, no. 2, pp. 126–131
16. M. Suzuki, H. Mitoma, M. Yoneyama (2017), Quantitative Analysis of Motor Status in Parkinson's Disease Using Wearable Devices: From Methodological Considerations to Problems in Clinical Applications *Hindawi Parkinson's Disease Article ID 6139716*,
17. J. J. Kavanagh, H. B. Menz, *Accelerometry: a technique for quantifying movement patterns during walking*, *Gait and Posture*, vol. 28, no. 1, pp. 1–15, (2008)
18. K. Aminian, B. Najafi, (2004) Capturing human motion using body-fixed sensors: outdoor measurement and clinical applications, *Computer Animation and Virtual Worlds*, vol. 15, no. 2, 79–94
19. W. Maetzler, J. Domingos, K. Srulijes, J. J. Ferreira, B. R. Bloem, (2013) Quantitative wearable sensors for objective assessment of Parkinson's disease, *Movement Disorders*, vol. 28, no. 12, pp. 1628–1637

SUPPLEMENTARY MATERIAL

```
#importing time module in order to time the running time
of code
import time
start_time= time.process_time()
#importing modules that we are going to use
import numpy as np
import pandas as pd
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score
from sklearn.model_selection import cross_val_score
import matplotlib.pyplot as plt
#This is a "classification problem", since we need
to predict the type of physical activity (e.g., walking,
climbing stairs) We can use either logistic_regression or
forest_classifier as our classifier
logistic_regression=LogisticRegression()
forest_classifier=RandomForestClassifier(max_depth=4)
#Reading in accelerometer data as train_time
train_time=pd.read_csv("train_time_series.csv")
```

```
#Reading in the file which contains the activity labels for
every 10th observation in train_time
train_labels = pd.read_csv("train_labels.csv")
#Reading in files that we are going to use to test the
code
test_time = pd.read_csv("test_time_series.csv")
test_labels = pd.read_csv("test_labels.csv")
#Making 3 empty lists (x, y and z)
x = []
y = []
z = []
#Taking measurements of linear acceleration (along
each of the three orthogonal axes x, y, z) from every
10th observation in train_time and appending them to
corresponding lists above
for i in range(3, len(train_time), 10):
x.append(train_time["x"][i])
y.append(train_time["y"][i])
z.append(train_time["z"][i])
#Adding these lists as columns to the train_labels Data-
Frame
train_labels["x"]= x
train_labels["y"]= y
train_labels["z"]= z
#Printing the first 5 rows of the train_labels DataFrame
to check what we previously did
train_labels.head()
#visualizing train_labels data
plt.plot(train_labels["timestamp"], train_labels["label"])
plt.xlabel("timestamp")
plt.ylabel("label")
plt.savefig('Figure 2.png')
from mpl_toolkits.mplot3d import Axes3D
%matplotlib notebook
fig = plt.figure()
ax = fig.add_subplot(111, projection="3d")
ax.scatter(x, y, z)
plt.xlabel("x")
plt.ylabel("y")
plt.clabel("z")
plt.savefig('Figure 3.png')
#Defining training variables
classification_outcome=train_labels["label"]
acceleration=["x", "y", "z"]
train_predictors=train_labels[acceleration]
def accuracy (estimator, X, y):
estimator.fit(X, y)
predictions = estimator.predict(X)
return (accuracy_score(y, predictions))
#Determining which estimator (logistic_regression
or forest_classification) is better for this classification
problem
#Determine the cross-validated accuracy for logistic
and random forest models.
logistic_regression_scores=cross_val_score(logis-
```

```

tic_regression, train_predictors, classification_outcome,
cv=10, scoring=accuracy)
forest_classification_scores = cross_val_score(for-
est_classifier, train_predictors, classification_outcome,
cv=10, scoring=accuracy)
# Plot Results
plt.axes().set_aspect('equal', 'box')
plt.scatter(logistic_regression_scores, forest_classifica-
tion_scores)
plt.plot((0, 1), (0, 1), 'k-')
plt.xlim(0, 1)
plt.ylim(0, 1)
plt.xlabel("Linear Classification Score")
plt.ylabel("Forest Classification Score")
# Show the plot.
plt.show()
# Save the plot as pdf
plt.savefig('Figure 1.png')
#Tuning in Random Forest
from sklearn.model_selection import RandomizedSearchCV
# Defining number of trees in random forest
n_estimators=[int(x) for x in np.linspace(start=100,
stop=500, num=5)]
# Number of features to consider at every split
max_features=['auto', 'sqrt']
# Maximum number of levels in tree
max_depth = [int(x) for x in np.linspace(3, 15, num=12)]
max_depth.append(None)
# Minimum number of samples required to split a node
min_samples_split=[2, 5, 10]
# Minimum number of samples required at each leaf
node min_samples_leaf=[1, 2, 3]
# Method of selecting samples for training each tree
bootstrap=[True, False]
# Create the random grid
random_grid = {'n_estimators': n_estimators,
'max_features': max_features,
'max_depth': max_depth,
'min_samples_split': min_samples_split,
'min_samples_leaf': min_samples_leaf,
'bootstrap': bootstrap}
print(random_grid)
# Random search of parameters, using 3 fold cross val-
idation, search across 100 different combinations, and
use all available cores
rf_random = RandomizedSearchCV(estimator=forest_
classifier, param_distributions=random_grid, n_iter
=100, cv=3, verbose=2, random_state=1)
# Fit the random search model
rf_random.fit(train_predictors, classification_outcome)
#View the best parameters
rf_random.best_params_
#Defining Random Forest with the best parameters
forest_classifier= RandomForestClassifier(n_estima-

```

```

tors=200, min_samples_split=5, min_samples_leaf=3,-
max_features= 'auto', max_depth= 6, bootstrap= True)
#deviding train_labels dataset into two subsets (training
and testing)
from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split (train_
predictors,classification_outcome, train_size = 0.8,
random_state = 1)
#finding the accuracy of tuned Random Forest on those
subsets
forest_classifier.fit(X_train, Y_train)
predictions1 = forest_classifier.predict(X_test)
from sklearn.metrics import accuracy_score
accuracy_score(Y_test, predictions1)
0.6266666666666667
#Taking measurements of linear acceleration(along
each of the three orthogonal axes x, y, z) from every
10th observation in test_time and appending them to
corresponding lists
x = []
y = []
z = []
for i in range(9, len(test_time), 10): #starting from 9
in order to match timestamps from test_time with ones
provided in the test_labels
x.append(test_time["x"][i])
y.append(test_time["y"][i])
z.append(test_time["z"][i])
#Adding x,y,z columns to the test_labels DataFrame
test_labels["x"]= x
test_labels["y"]= y
test_labels["z"]= z
#Using forest_classifier to predict activity labels for
each row in test_labels
forest_classifier.fit(train_predictors, classification_outcome)
test_predictors = test_labels[acceleration]
predictions=forest_classifier.predict(test_predictors)
#Adding a new column (containing predicted activity
labels) to the test_labels DataFrame
test_labels["label"]= predictions
#saving the file with test timestamps and predictions
test_labels.to_csv("predictions.csv")
#finding the code runtime
end_time=time.process_time()
elapsed_time=end_time - start_time
print(elapsed_time)

```

Paper submitted: 19.06.2020.

Paper accepted: 15.09.2020.

*This is an open access article distributed under the
CC BY 4.0 terms and conditions.*