Multiclass Classification of Aerobic Activities from EndoMondo User Data

CSE 258 Assignment 2

Jeffrey Makings
CSE MS Candidate
UC San Diego
San Diego, California
jmakings@ucsd.edu

1 FitRec EndoMondo User Dataset

When searching for a dataset to study, we found Dr. McAuley's FitRec datasets[1] from EndoMondo User Data to be of interest. This was due to the vast number of workout samples available (167,783 in the filtered version for workout route prediction task) and several features available to choose from for a potential model. These included geographic information (latitude and longitude), user ID, altitude, user heart rate, gender, sport, and timestamp.

Below are pie charts with a breakdown of activities on each individual workout, and the gender make up of each workout entry. The gender breakdown is highly unbalanced, men make up about 95% of the workouts in the dataset.

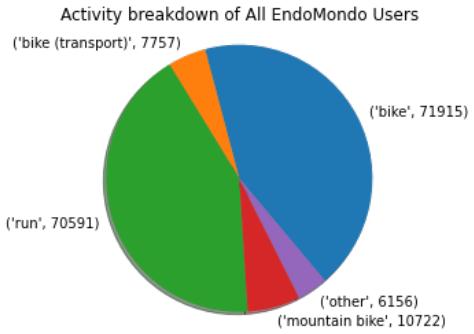

Figure 1: Sport Breakdown of EndoMondo workouts. 'Other' category includes orienteering, indoor cycling, skating, cross-country skiing, core stability training, walk, hiking, fitness walking, and roller skating

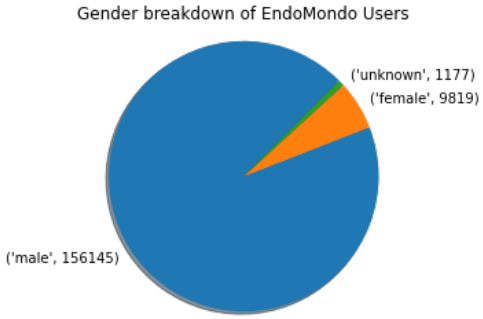

Figure 2: Gender Breakdown of EndoMondo Users

Certain potential features within the dataset were sparse, for example, we initially believed the mean user speed and mean user maximum speed for each activity to be interesting data. However, only 31,662 entries (less than 19%) had speed data to use.

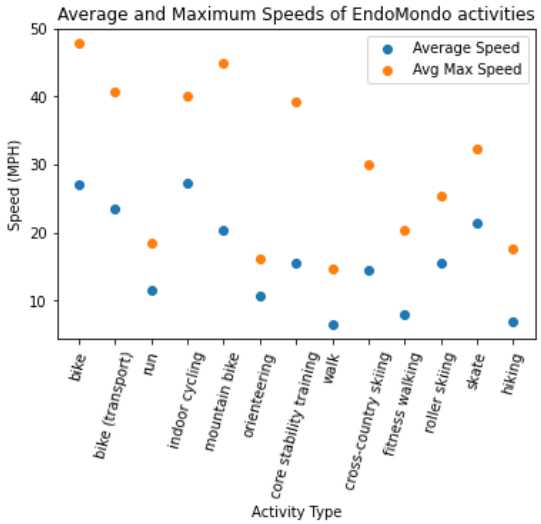

Figure 3: Average speed and average max speed for each activity

Many other features, however, including average and maximum heart rates, latitude and longitude data, and altitude data, were all complete for each entry. For each of these potential features, we computed the mean for each activity group. To get the mean maximum heart rate, we took the fastest recorded heart rate for each entry, summed this over the activity group, and divided this by the number of samples in that activity group.

To get a better understanding of the latitude, longitude, and altitude data, the maximum absolute difference in the variable for each workout was computed. For example, in the longitude data for each workout, the maximum longitude was subtracted from the minimum longitude, and the mean of these values for each activity group were computed. This was done because many workouts started and ended in the same location, making it challenging to find the total distance traveled/altitude change per workout.

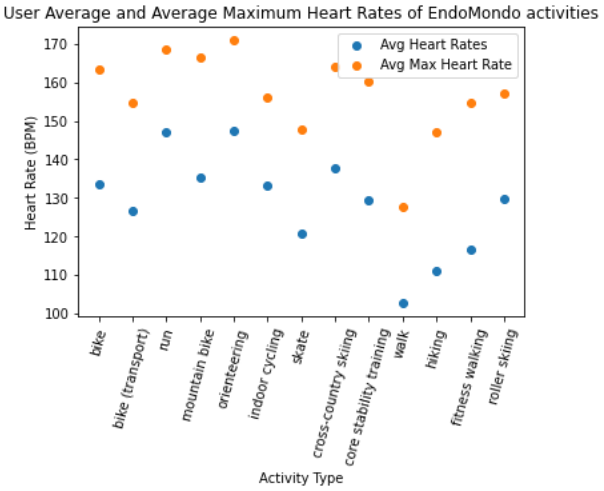

Figure 4: Mean Average and Maximum Heart Rates for activities

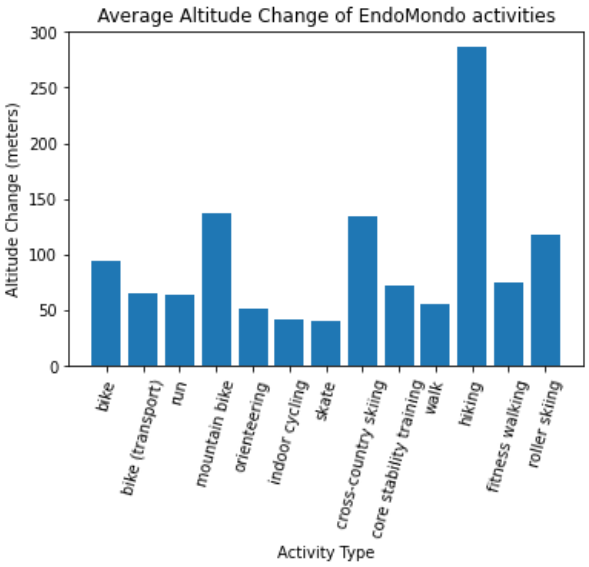

Figure 5: Altitude change in meters for each activity

Before identifying a predictive task and creating models to fit this aim, the data was filtered further by activity type. Prior to filtering, there were 43 activity types, 30 of which had less than 100 workout samples (many with as few as 1 sample). These 30 were removed, decreasing the dataset size from 167,373 workouts to 167,141 among 13 different sporting activities.

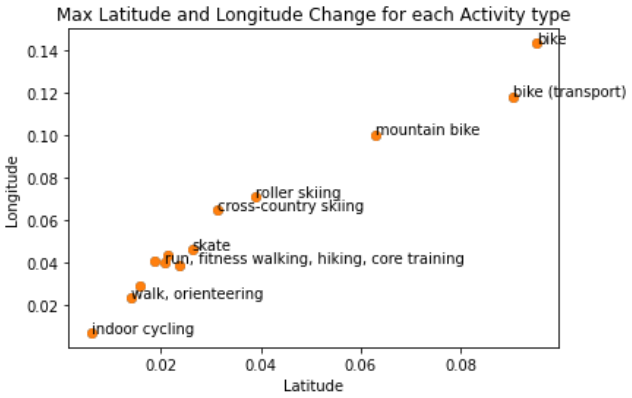

Figure 6: Geographic change (longitude and latitude) in degrees for each activity

2 Classification Predictive Task

With the above explanation of the dataset and exploratory analysis of the data, we believed that multiclass classification of the sport/activity would be an appropriate predictive task for this assignment. We enjoyed the classification tasks early in the course, however we always wanted to expand the knowledge to a multiclass model and try out models beyond logistic regression classifiers.

The EndoMondo activity data presents a formidable challenge, as there are 13 possible classes instead of the previous 2, and the data is highly unbalanced towards 'run' and 'bike' classifications.

Previously when using binary classifiers, balanced error rate was used on the test set to evaluate a model. However, in this multiclass classification, there is no false positive/negative predictions, only correct and incorrect predictions. So instead, we decided to evaluate the classifier by the overall accuracy rate for both the training and test sets. This is computed from simply taking the # of correct classifications / total # of samples. The accuracy rate of the test set is trivially used to evaluate the classifier's accuracy; however the training set accuracy rate is used to prevent overfitting to the training set. Minimizing the difference between the training set and test set accuracy rates is critical to preventing an overfit model.

$$CR = \frac{C}{A}$$

CR – The correct rate;

C – The number of sample recognized correctly;

A – The number of all sample;

For baseline comparison, the most trivial model possible is a classifier that predicts only the most common classification in the training set for every test sample. For this particular dataset, the most common activity was 'bike', so the baseline classifier predicts 'bike' every time. Another baseline classifier was a logistic regressor with the features being only the bias term and the max

speed during that sampled workout. Although this feature was useful for the samples with speed data, few samples included speed data, and this classifier ended up being just better than a trivial predictor.

After discovering that max speed and average speed were not good features, we went to work on finding features that would fit better. As stated in part 1, the features that became most useful were altitude, latitude and longitude, and mean workout heart rate.

To obtain change in altitude data, for each sample, the maximum altitude was subtracted from the minimum altitude. This number was then added to a dictionary with the workout ID as its key for later use in the model, and into a dictionary with altitude data for each sport for general statistical purposes (Figure 5).

```
#sumAlt for computing change in altitude per sport
sumAlt = defaultdict(int)
#altUser for storing altitude change for each workout
altUser = {}
for d in filterData:
    change = max(d['altitude']) - min(d['altitude'])
    altUser[d['id']] = change
    sumAlt[d['sport']] += change
```

Figure 7: Code snippet for computing altitude change

The same process is used to compute the change in latitude and longitude for each workout. For heart rate data, the maximum and mean heart rates were computed for each workout. The average heart rate was used as a feature due to its greater variance between different activities in comparison to max heart rate.

As a final feature, a one hot encoding of gender information was included. Although there were many more men than women in the dataset, women had distinct workout preferences in comparison to men, typically preferring running over biking while biking dominated the men's dataset.

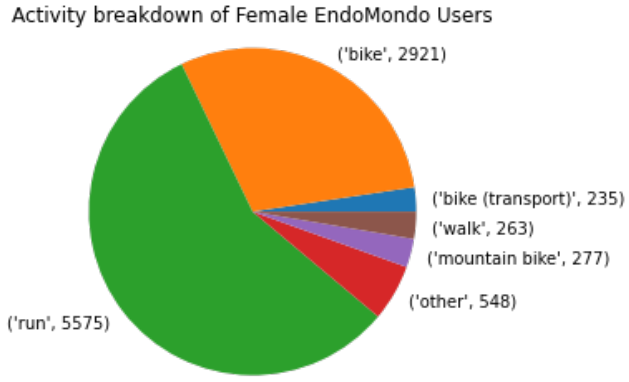

Figure 8: Pie Chart of female activities

Finally, different specific classifiers were tested with various feature combinations and ablations, ranging from simple to more complex models. The different model types tested were logistic regressors, random forest classifiers, extra trees classifiers, and support vector machines.

3 Simple to Complex Classification Models

The models tested included various logistic regressors, random forest classifiers, extra trees classifiers, and support vector machines. All of these models are provided curtesy of the Scikit-learn python library. In the end, the extra trees classifiers typically performed better and consistently obtained the highest accuracy scores in comparison to the other classifiers.

The Extremely Randomized Trees classifier (Extra Trees) is a tree-based ensemble classification method, first described in Geurts et al. (2006) [2]. Similar to the Random Forest classifier in many ways, Extra Trees fits a large number of decision trees from the whole training dataset, randomly sampling features at each split point in the tree.

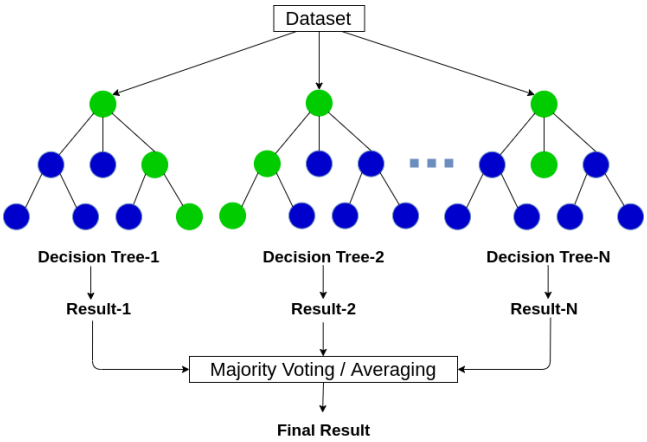

Figure 9: Random Forest/Extra Trees Classifier Diagram[3]

This model was selected because it produced consistent results across various training/validation/test splits, it achieved the highest accuracy scores among the models tested, and it allowed for effective weighting of the various unbalanced classifiers. On top of this, it had a reasonable run-time.

The model was optimized by balancing the class weights so they were inversely proportional to the class frequency, which allowed for effective identification of less frequently occurring classifications. Also, various numbers of trees in each model were tested to ensure maximum accuracy.

Initially, overfitting was a big worry for us because after testing on the training set, the classifier was highly accurate. However, after cross validation on the validation/testing data, it appeared the model produced similar accuracy results regardless of the tested set. This is one of the advantages of an Extra Trees classifier, due to the many decision trees overfitting is less likely.

Other models tested included logistic regression, Support Vector Machines, and random forest classifiers. Tried and true logistic regression performed the worst out of all tested models, although the runtime was the fastest due to having the simplest time complexity.

Although we initially had high hopes for Support Vector Machines, it underperformed when compared to tree algorithms, and the run time was far too high for this model to be practical. SVM is useful for binary classification, however when predicting between 13 different options this involves building binary SVMs between 13 different classes.

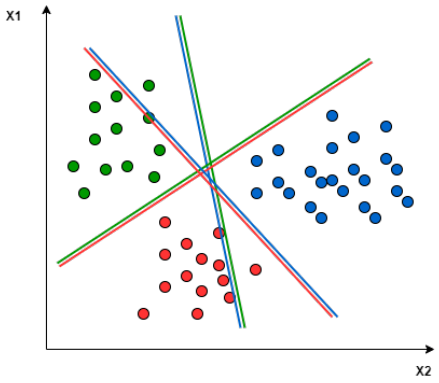

Figure 10: Simplified Multiclass SVM hyperplane example. This figure includes 3 classes, however this data included 13 classes, making the SVM much more complex

Finally, Random Forest Classifiers were tested and had similar results to the Extra Trees Classifier. Due to the many similarities between these two models, this is expected. The major difference between these models is Random Forest "bootstraps" the training data (samples without replacement) and selects the best decision split while Extra Trees draws from the whole dataset and applies splits randomly. For this particular problem Extra Trees produced a slightly higher accuracy score.

4 Associated Literature

The dataset in this project came from Jianmo, Muhlstein, McAuley, "Modeling heart rate and activity data for personalized fitness recommendation."[1] In their study, workout data from wearable fitness devices was collected from endomondo.com. This is the dataset we chose to use for this study.

This study produced *FitRec*, which is a Long Short-Term Memory model, and one of the applications was predicting how workout measurements (e.g. heart rate) will change across a workout. In this assignment, we decided to instead do the opposite, and try to characterize the activity from certain measurements. Other functions of the FitRec model include identifying features that affect workout performance and recommending alternative workout routes to individuals to achieve a target heart rate profile.

In Parkka et al (2006)[4], one of the earlier papers written on activity classification from sensor data, the researchers used a custom decision tree to classify human activities. This approach was similar to our model, and they yielded a similar result: 82% total classification accuracy. However, their dataset differed in that they had only 16 test subjects, but with 35 different channels of data per person as opposed to the 6 features we used.

In Mekruksavanich S., Jitpattanakul A. (2021) [5], the authors claim to have created a 4-layer hybrid CNN-LSTM model that outperforms previous state of the art activity classifiers by 2.24% accuracy. They used data collected by users' cell phones, and classified more mundane activities such as sitting, standing, and walking up and down stairs. Another key difference they describe is using deep learning for automated feature extraction as opposed to handcrafting features in a more conventional machine learning process.

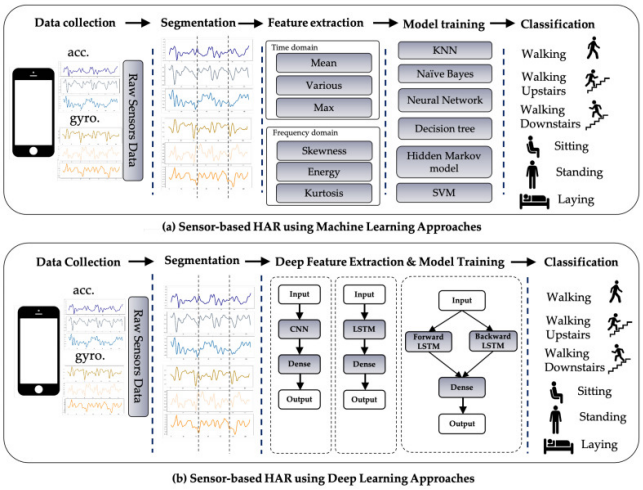

Figure 11: Mekruksavanich/Jitpattanakul's state of the art CNN-LSTM model for activity classification

5 Results and Conclusions

In conclusion, various models were tested with different sets of features. Described below were the models tested:

Guess Only Bike Model (Baseline): model that guesses 'bike' exclusively. Trivial model. 42.98% accurate

Max Speed Logistic Regressor: Logistic Regressor that used maximum speed as the only feature. If no max speed present, global max speed used as feature. Failed because less than 20% of the workouts included speed data. 51.48% accurate

Average Heart Rate Logistic Regressor: Logistic Regressor using average heart rate during workout as the only feature. Failed because it could not account for the less frequent classes. 58.01% accurate.

Latitude and Longitude Regressor: Logistic Regressor using maximum latitude/longitude – minimum latitude/longitude as feature. This feature ended up being much more useful in distinguishing between 'bike' and 'run', the two largest classes. 73.19% accurate

Latitude, Longitude, and Heart Rate Regressor: Combination of the previous two sets of features into one regressor. Proved that

the average heart rate was a less informative feature than geographic data. 74.23% accurate

Latitude, Longitude, and Altitude Radial Basis Function Support Vector Machine: This classifier is a support vector machine with the decision hyperplane defined by the function exp[-(*gamma*)||x-x'||2]where *gamma* is a hyperparameter to tune. This classifier showed promise, however the long run time to create one-to-one classifiers between each class made it impractical and too challenging to tune. 75.80% accuracy

Latitude and Longitude Random Forest Classifier: From here, we left logistic regression behind and tested a random forest classifier with latitude and longitude features. This resulted in an improvement over the logistic regressor, largely because this was the first classifier to begin to include the less frequent classifiers. 78.99% accurate

Latitude, Longitude, and Altitude Random Forest Classifier: This classifier is an extension of the previous classifier but expanded to include maximum altitude – minimum altitude data as a feature. 80.79% accurate

Latitude, Longitude, and Altitude Extra Trees Classifier: This classifier uses the same parameters as the above function, however instead an extra trees classifier was implemented. 81.09% accurate

And finally,

Latitude, Longitude, Altitude, and Gender Extra Trees Classifier: This classifier added a one hot encoding of gender to the previous classifier. Although the gender only provided minor improvement due to the small proportion of women in the dataset, the difference between men and women's activity preferences resulted in some improvement.

The Extra Trees Classifier worked well for this dataset because of the many different classes that needed to be identified and the large number of continuous features available to work with. This allowed for the creation of large numbers of decision trees that would ensemble to create a single classification and greatly decrease the possibility of overfitting the data. These many trees, along with weighting the classifier, allowed for classification of the less frequent activities in the dataset as well.

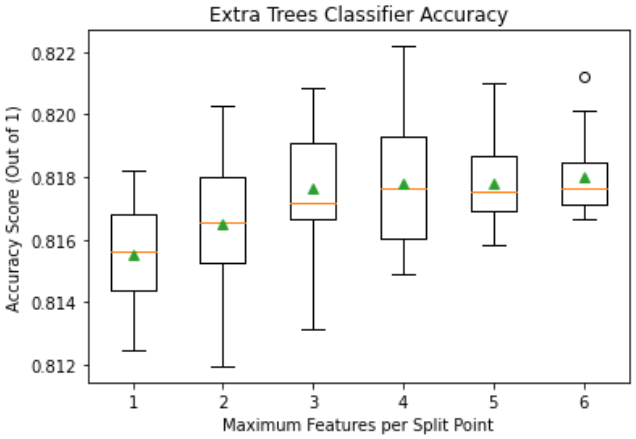

Figure 12: Classifier Accuracy comparing maximum allowed numbers of features before decision tree splitting

To tune hyperparameters in this model further, different numbers of decision trees, maximum features before split point, and minimum samples before a split point were tested. To summarize hyperparameters, giving the model more options is generally better for maximizing the accuracy score. The most important hyperparameter is the number of decision trees. As shown in Figure 13, increasing the number of decision trees results in a steady increase in accuracy score, leveling off after 100 decision trees. Although we wanted to test numbers exceeding 250 decision trees, the memory required and the run time to fit this model became challenging for the computer to handle.

The other hyperparameters were maximum features allowed per split point and minimum number of samples before split. See Figures 12 and 14, which express that allowing the model to pick the maximum number of features before splitting if possible and allowing the minimum number of samples (2) before splitting generally leads to a more accurate model.

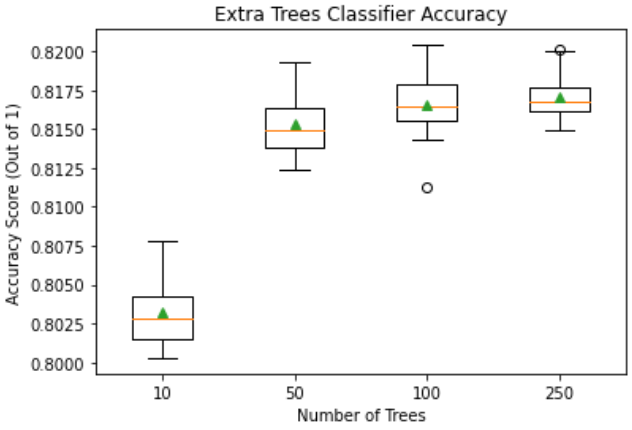

Figure 13: Classifier Accuracy with different numbers of decision trees. Having a sufficient number of decision trees is crucial to maximizing accuracy

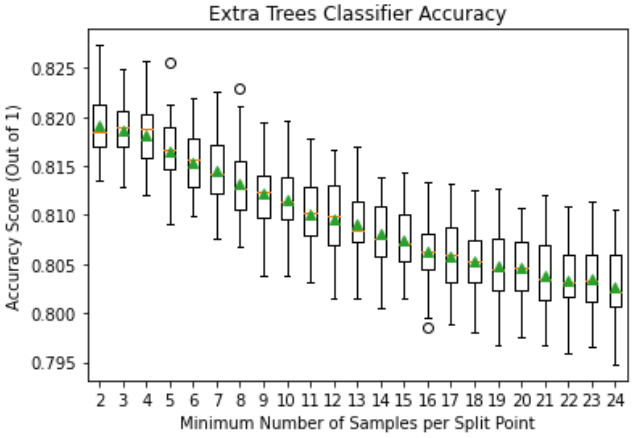

Figure 14: Accuracy comparing minimum number of samples between split points

Altogether, the best set of hyperparameters in this model was 250 decision trees, the minimum number of samples between split points (2) and the maximum number of allowed features before tree splitting (6 for this model). This model was then evaluated with a 5-fold cross validation of the data set that was repeated 3 times each. The mean accuracy was evaluated to be 81.87% correct classification.

Classification problems like this are now being solved by increasingly complex neural networks and other deep learning techniques, however this tree-based multiclass classification model still works well even for a dataset with hard to distinguish differences between classes.

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
0	0.815065	0.819552	0.819493	0.818296	0.819552
1	0.820270	0.817877	0.819911	0.818176	0.817399
2	0.817578	0.815843	0.819134	0.822604	0.819253

Figure 15: Cross-Validated Accuracy Score Results. Rows represent repeated modeling from the same fold

ACKNOWLEDGMENTS

I'd like to thank Dr. McAuley for being the chilliest, most interesting, and best piano-playing professor while also presenting an awesome class that allows students to both be creative and learn subjects like recommender systems which are incredible interesting and ever-present in our digital lives. I'd also like to thank all the TAs in this course (especially the TA grading this right now) for working so hard to grade the assignments and exams for close to 1000 people, you guys are the real MVPs. I'd also like to thank my cat Ivy for emotional support during this quarter.

REFERENCES

[1] Jianmo Ni, Larry Muhlstein, Julian McAuley, "Modeling heart rate and activity data for personalized fitness recommendation", in Proc. Of the 2019 World Wide Web Conference (WWW'19), San Franscisco, US, May 2019

[2] Guerts, P., Ernst, D. & Wehenkel, L. "Extremely randomized trees." Mach Learn 63, 3-42 (2006). Httpes://doi.org/10.1007/s10994-006-6226-1

[3] "Decision Tree vs Random Forest- Which Algorithm Should You Use?" Analytics Vidhya, 12 May 2020, https://www.analyticsvidhya.com/blog/2020/05/decision-tree-vs-random-forest-algorithm/.

[4] J. Parkka, M. Ermes, P. Korpipaa, J. Mantyjarvi, J. Peltola and I. Korhonen, "Activity classification using realistic data from wearable sensors," in IEEE Transactions on Infomration Technology in Biomedicine, vol. 10, no. 1, pp. 119-128, Jan. 2006, doi: 10.1109/TITB.2005.856863

[5] Mekruksavanich S, Jitpattanakul A. LSTM Networks Using Smartphone Data for Sensor-Based Activity Recognition in Smart Homes. Sensors (Basel). 2021 Feb 26;21(5):1636. Doi: 10.3390/s2105636. PMID: 33652697; PMCID: PMC7956629.