HAR-CTGAN: A Mobile Sensor Data Generation Tool for Human Activity Recognition Supplementary Material

Joshua DeOliveira[1,2], Walter Gerych[1], Aruzhan Koshkarova[1,2], Elke Rundensteiner[1], and Emmanuel Agu[2]

[1]Data Science Program, Worcester Polytechnic Institute, Worcester, MA
[2]Department of Computer Science, Worcester Polytechnic Institute, Worcester, MA

***Abstract*—This document contains supplementary materials that were omitted from the paper ”HAR-CTGAN: A Mobile Sensor Data Generation Tool for Human Activity Recognition” relating to the data cleaning and feature engineering process for the data set used.**

Data Cleaning

A. User Aggregation

The featurized *ExtraSensory* dataset consists of 60 unique users with their recorded mobile sensor data streams aggregated by disjoint 1-minute intervals of time series data along with the discrete data recorded from device sensors and manual user input. Each instance of data is accompanied by metadata relating from the user the data came from (anonymized by a randomly generated IDs) and timestamps corresponding to when the datum was recorded. In order to pool all the data together, the user IDs and timestamps were stripped from each instance and populated into one master dataset. Since each user recorded their data from different devices that can have different sensors available than other devices, the lowest common denominator of sensors used needed to be established. Also, poorly labeled data (whether it be missing discrete contexts or missing labels in the instance) needed to be removed from the master dataset in order to fully utilize well defined, trustworthy data. To clean the data, any singular instance that had more than 15% of its features having missing values or didn't have a mutually exclusive label for the activity were removed. For data instances that remained after the filtering that did still have missing values, the unknown entries were labeled with a 0. This trims the dataset down into clean, well populated instances consisting of features extracted from commonly used sensors. After cleaning, the *ExtraSensory* dataset was reduced from 377,346 instances to 141,508.

Feature Engineering

B. Feature Engineering

Since the *ExtraSensory* mobile sensor data set consists of such a wide and diverse set of continuous and discrete features, it is important to properly extract the most meaningful features for a downstream HAR classification. Doing so will guide our experiments as to what features are the most valuable for up-sampling with more synthetic instances. In order to find these important features, we use random forest (RF) feature importance [1]. When utilized in the context of RF, the feature importance algorithm first quantifies how much each feature contributes to the final prediction of a decision tree, and then determines the average values of these contributions across the forest. More specifically, the RF feature importance [2] is computed by measuring the degree to which each feature reduces the Gini Index, defined as:

$$\text{Gini}(w) = \sum_{k=1}^{K} w_k \left(1 - w_k\right) = 1 - \sum_{k=1}^{K} w_k^2 \tag{1}$$

where K refers to the total number of features considered, and w_k represents sample weights. Moreover, within a single tree's internal node m, the feature importance γ of x is

$$\gamma_{jm}^{(Gini)} = GI_m - GI_l - GI_r \tag{2}$$

where GI_l and GI_r are the Gini Index of the two child nodes after a split respectively. Given that a feature x appears in a decision tree i in nodes M, γ of x in the i-th tree is defined

$$\gamma_{ij}^{(Gini)} = \sum_{m \in M} \gamma_{jm}^{(Gini)} \tag{3}$$

Furthermore, given that there are n trees in the forest

$$\gamma_{j}^{(Gini)} = \sum_{i=1}^{n} \gamma_{ij}^{(Gini)} \tag{4}$$

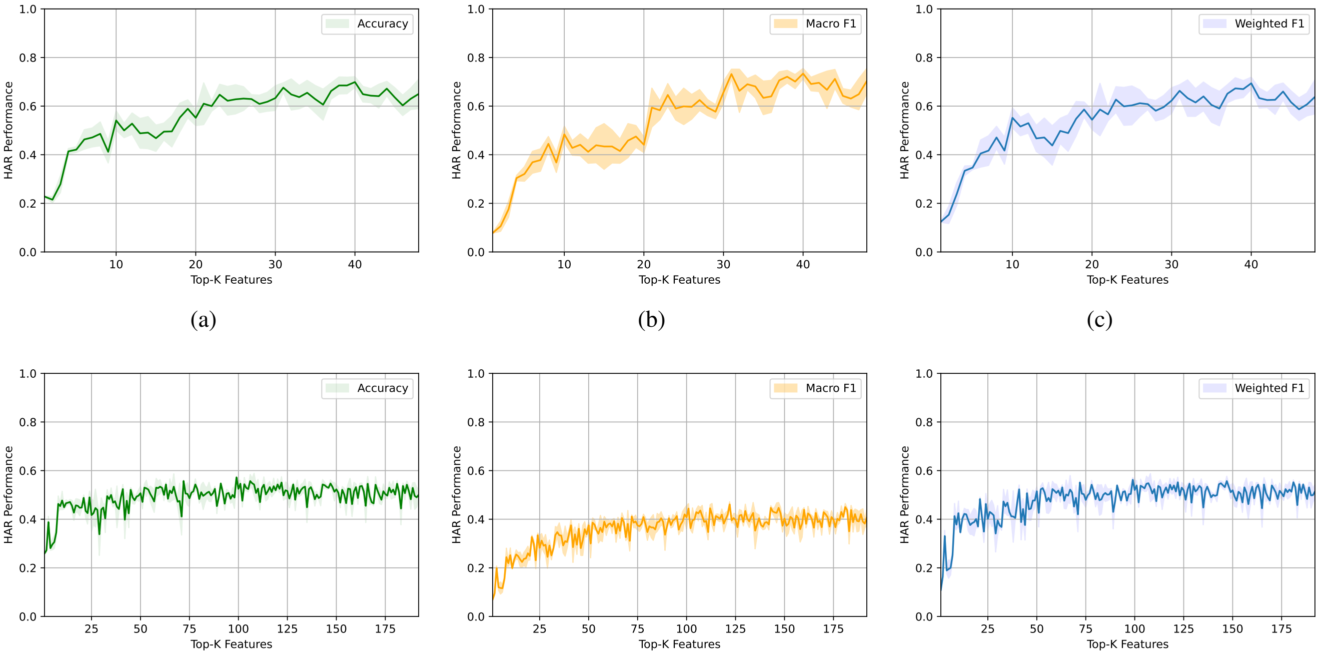
Finally, we normalize the values of γ by dividing a feature's importance by the total sum of all feature importance values.

$$\gamma_j = \frac{\gamma_j}{\sum_{i=1}^{c} \gamma_i} \tag{5}$$

To find the most important continuous and discrete features separately, we ran a set of stratified random forest feature importance tests for each of the 7 HAR activities. For a single HAR activity, the score for a feature would be

$$Importance(x_d | \mathbb{A} = a) = \zeta \tag{6}$$

for a discrete feature and



(a) (b) (c)

(d) (e) (f)

Fig. 1:
Plots a-c. From left to right: accuracy, macro-f1, and weighted-f1. HAR model performance when employing stepwise selection of discrete features. When using 30 discrete features, weighted-f1, macro-f1, and accuracy begin to converge to just above 0.6 when incorporating additional features.
Plots d-f. From left to right: accuracy, macro-f1, and weighted-f1. HAR model performance when employing stepwise selection of continuous features. When using 100 continuous features, macro-f1 peaks and converges around 0.4, while accuracy and weighted-f1 oscillate between 0.4 and 0.6 when additional features are added into a classification model.

$$Importance(x_c|\mathbb{A} = a) = \beta \tag{7}$$

for a continuous feature, where ζ and β are the positive modular additive inverse of its rank. The feature importance are then computed as shown below:

$$\zeta + Rank(\gamma_j) = 0 \mod \|\mathbb{D}\| \tag{8}$$

$$\beta + Rank(\gamma_j) = 0 \mod \|\mathbb{C}\| \tag{9}$$

For example, for a discrete feature that received the 3^{rd} greatest feature importance according to the random forest out of 48 discrete features, that feature would receive a feature score of 45. These scores for the top discrete and continuous features were then aggregated respectively across the 7 activities and given their final rankings for their feature importance.

$$Importance(x) = \sum_{a}^{|\mathbb{A}|} Importance(x|\mathbb{A} = a) \tag{10}$$

From there, we perform stepwise feature selection to determine how many of the $top-k$ most important discrete features and the $top-j$ most important continuous features (based on descending order of importance score) to consider for our final baseline feature input for a hypothetical downstream HAR model. Thus, we trained 48 HAR classifiers each tasked with classifying the activity performed only using the $top-m$ discrete features for $m = 1, ..., 48$. Similarly, we trained 192 HAR classifiers performing the same task, but only using the $top-n$ continuous features for $n = 1, ..., 192$. By finding the minimum number of continuous and discrete features necessary until reaching asymptotic performance in classification when using real data, we can know what features are most important for up-sampling with synthetic generation as well as input for HAR classifiers in our experimental study. As seen in Figure 1, we choose to consider 30 discrete features and 100 continuous features for later use.

References

[1] J. Yu, C. Xia, J. Xie, and H. Zhang, “Research on feature importance of gait mechanomyography signal based on random forest,” in *2020 International Conference on Computer Vision, Image and Deep Learning (CVIDL)*, IEEE, 2020.

[2] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, *et al.*, “Scikit-learn: Machine learning in python,” *Journal of machine learning research*, vol. 12, no. Oct, 2011.