Team 36: Proj-C: Terrain Identification from Time Series Data

Sai Naga Vamshi Chidara
schidar@ncsu.edu

Venkata Pranav Singaraju
pvsingar@ncsu.edu

Krishna Saurabh Vankadaru
kvankad@ncsu.edu

I. Methodology

For the Terrain Identification task at hand, first we pre-processed data following the procedure described in the Data Pre-processing section below. But, the training data we have, contains class imbalance. To counter that, we used SMOTE(Synthetic Minority Oversampling Technique)[1] which synthesizes new data samples from existing data samples. So, after pre processing the data, we applied SMOTE and then we trained different classifiers with different configurations.

A. Data Preprocessing

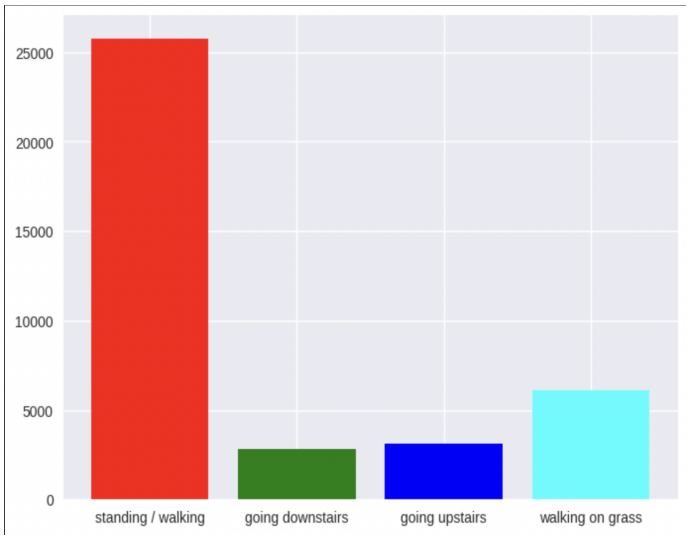
The training dataset for terrain identification task contains data for 8 subjects (Subject001 - Subject008) and the test dataset contains data for 4 subjects (Subject009 - Subject012). Each subject has the files subject_00*_x, subject_00*_x_time, subject_00*_y, subject_00*_y_time.

The file _x_time contains the timestamps for the accelerometer and gyroscope measurements where as the file _y_time contains the timestamps for the labels. The units are in seconds and the sampling rate is 40 Hz for _x_time and 10 Hz for _y_time. The files _x contains the accelerometer and gyroscope measurements from the lower limb and files _y contains labels 0 to 3 where label 0 indicates standing or walking in solid ground, 1 indicates going down the stairs, 2 indicates going up the stairs, and 3 indicates walking on grass.

First, for each subject, we joined the _x file and its corresponding _x_time file into one file. Similarly, we joined the _y file and its corresponding _y_time file into one single file. So now, for each subject, we will have the accelerometer and gyroscope readings with the corresponding times in one single file and the labels with the corresponding times in another single file.

Then, for each subject, we merge the file containing accelerometer and gyroscope readings with the file containing labels using the ”time” column (outer join). This resulting file will contain all the attributes corresponding to one subject in one single file. Here while merging these two files, we have to account for the difference in the sampling rate for _x (40 Hz) and _y (10 Hz). We have labels for only a quarter of the x samples because of this difference in sampling rate.

To fill the missing labels, we use interpolation technique with padding, where the missing label for a sample is the label for the sample before it. As a result, for each subject, we have labels for all its samples. In the next stage of data preprocessing, we merge all subject files into one single file which will be used for training and validation.



We plotted this data as above and identified that there is severe imbalance in the dataset with majority of data samples having 0 as their class label. So, we used SMOTE to synthesize data samples for other class labels to balance the data set.

II. Model Training and Selection

A. Model Training

In the training data files, we have 8 subjects and the xyz accelerometers and xyz gyroscope measurements recorded for each subject for different sessions. For training the model, we have to split the data into training and validation sets. But, as the data is a time series data, we can’t split randomly as we might lose the information in the previous values in the sequential data. So, we thought of a different approach to split the data. We explored the subject-leave-out and used it to split training samples into training dataset and validation set.

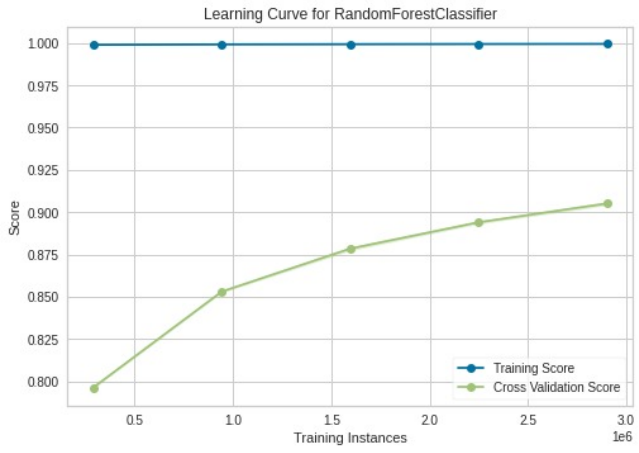
1) Subject-Leave-Out: In this approach, we selected Subjects 5 and 7 to be the validation data set. The remaining subjects (1,2,3,4,6 and 8) are considered as training data set.

B. Model Selection

For the phase 1 of the project we used the Random Forest classifier[2] which is a classical machine learning technique. Random Forest is an ensemble learning technique which constructs multiple decision trees at training time and output the class that is the mode of the classification output of the individual trees. Each tree in the random forest is constructed based on the random bootstrap samples of the training data. The main motive of selecting Random Forest is that it offers low variance keeping the bias unchanged.

Important hyper-parameters[3] of Random Forest while training are n_estimators, max_features, max_depth, min_samples_leaf and criteria among others. For criteria, we have selected Gini-Impurity over Entropy[4]. Since, Entropy ranges from 0-1 and Gini ranges from 0-0.5, Gini Impurity is computationally faster and helps in selecting the features. Also, Gini Impurity favours larger partitions and is easy to implement.

Apart from the current xyz accelerometers and xyz gyroscope measurements, the label of the class also depends on the previous time stamp values. Hence, the xyz current values are not the only features governing the current label. But, mostly the current values will dominate the class label at that time and hence we are using Random Forest algorithm.



One of the important hyper-parameters is the n_estimators. It is nothing but the number of trees in the Random Forest. We have trained different Random Forest Classifiers with n_estimators={20,40,60} and calculated the scores on the validation set. Out of the three Random Forest Classifiers, we observed that the validation set score is high for the model with n_estimators=20. Above is the graph corresponding to the Random Forest Classifier with n_estimators=20.

III. Evaluation

Based on the validation set scores, we have considered Random Forest Classifier (n_estimators=20,'Gini Impurity', min_samples_leaf=1, min_samples_split=2, boot_strap=True). Following tables show the metrics of the model along with the average (macro) scores of this model on different classes of validation data set.

Metric	Score-Values
Accuracy	0.91
Precision	0.90
Recall	0.91
F-1 score	0.90

class	precision	recall	f1-score
0	0.88	0.80	0.84
1	0.95	0.97	0.96
2	0.95	0.98	0.97
3	0.86	0.89	0.88

IV. Scope for Improvement

In the first phase of the project, we have used classical machine learning technique called Random Forest Classifier to predict the different terrains from time series data. To improve the accuracy and F1 score, we would like to consider the following improvements for the next phase of the project.

1. We will be considering deep learning architectures(LSTM[5] or bi-directional-LSTM[6]) as the data is sequential data i.e. Time-Series data.
2. In place of Subject-Leave-Out, we will explore Session-Leave-Out to split the data into training set and validation set.
3. Instead of SMOTE technique, we will modify loss function of the deep learning architecture similar to weighted cross entropy loss function[7] to compensate class imbalance. Even though SMOTE duplicates the instances of minority classes, deep neural networks will over-fit the minority class data points.

References

[1] "https://www.analyticsvidhya.com/blog/2020/10/overcoming-class-imbalance-using-smote-techniques". In.

[2] "https://machinelearningmastery.com/random-forest-for-time-series-forecasting". In.

[3] Philipp Probst, Marvin N. Wright, and Anne-Laure Boulesteix. "Hyperparameters and tuning strategies for random forest". In: *WIREs Data Mining and Knowledge Discovery* 9.3 (Jan. 2019). DOI: 10.1002/widm.1301. URL: https://doi.org/10.1002%2Fwidm.1301.

[4] Arjun Saud, Subarna Shakya, and Bindu Neupane. “Analysis of Depth of Entropy and GINI Index Based Decision Trees for Predicting Diabetes”. In: *Indian Journal of Computer Science* 6 (Jan. 2022), pp. 19–28. DOI: 10.17010/ijcs/2021/v6/i6/167641.

[5] Yang Guo, Zhenyu Wu, and Yang Ji. “A Hybrid Deep Representation Learning Model for Time Series Classification and Prediction”. In: *2017 3rd International Conference on Big Data Computing and Communications (BIGCOM)*. 2017, pp. 226–231. DOI: 10.1109/BIGCOM.2017.13.

[6] Mehak Khan et al. “Bidirectional LSTM-RNN-based hybrid deep learning frameworks for univariate time series classification”. In: *The Journal of Supercomputing* 77 (July 2021), pp. 1–25. DOI: 10.1007/s11227-020-03560-z.

[7] Sheng Lu et al. “Dynamic Weighted Cross Entropy for Semantic Segmentation with Extremely Imbalanced Data”. In: *2019 International Conference on Artificial Intelligence and Advanced Manufacturing (AIAM)*. 2019, pp. 230–233. DOI: 10.1109/AIAM48774.2019.00053.