Terrain Identification from Time Series Data for Competition Project *

Srilekha Gudipati
[sngudipa]

Sai Krishna Teja Varma Manthena
[smanthe]

Aditi vakeel
[avakeel]

I. Background

Navigation through different terrains for lower-leg amputees can be a challenging task. Such people seek the assistance of leg robots for movement support and comfort. These leg robots or prosthetics can be trained with intelligence to not only support their users but also predict diverse terrains and aid in wayfinding [1].

The most common prediction techniques adopt image sets and try to estimate the terrain type in natural settings. However, that does not prove to be reliable since the properties of the terrain alter with the environmental conditions. This kind of constraint presents a necessity for sensory data for accurate predictions. This project focuses on the sensor data, i.e., the IMU (inertial measurement units) streams collected from the lower limb. Random Forest Classifier [2] is adopted to study the surfaces.

II. Dataset and Intuition

The dataset is made up of several sessions of IMU(inertial measurement unit) values obtained from sensors on each subject's leg. The labels are derived from annotations of terrain type in a synchronized data stream. The study's labels, 1, 2, and 3—which correspond to walking on grass, upstairs, and downstairs, respectively—indicate these three different activities. "Fig. 1" shows the imbalance in the dataset.

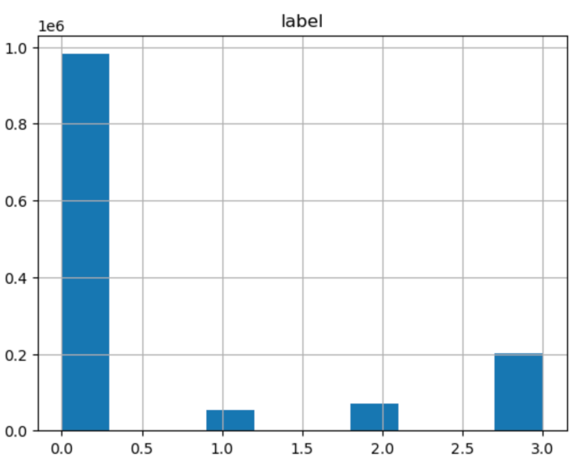

Fig. 1. Dataset before SMOTE.

The data set contains multiple files that include measurements like gyroscopes and accelerations of lower limbs(in all three dimensions, hence 6 values). The frequency of attribute files, _x.csv and _x_time.csv, and label files, _y.csv and _y_time.csv, are in the ratio 4:1 . Thus, we have opted to interpolate the combined data files. After the data pre-processing, we used SMOTE which is a Synthetic Minority Oversampling Technique to compensate for the imbalanced data shown in "Fig. 2". We used Random Forest to train and test the data, which produced 92% accuracy.

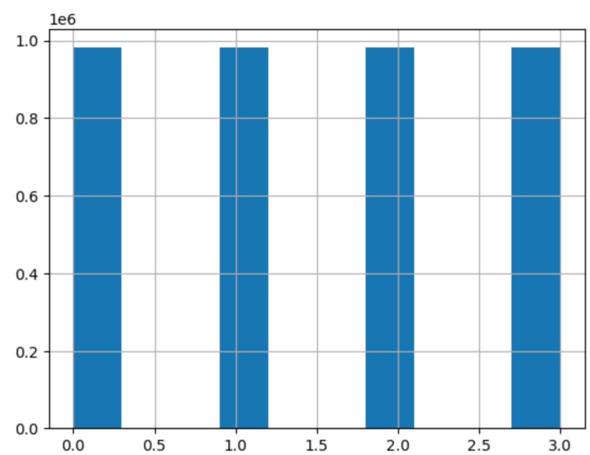

Fig. 2. Dataset after SMOTE.

III. Preprocessing

The following steps describe in detail the steps followed to pre-process and prepare the model for the prediction:

1) As a first step the different files are combined to prepare a proper data set to feed the model. This includes combining _x.csv with its time stamp file,_x_time.csv; _y.csv with its corresponding time stamp file _y_time.csv. This produces a single file with accelerometer and gyroscope measurements and another single file with labels and their respective timestamps.
2) Secondly, files obtained from the above step are further joined based on timestamps to produce a proper data set that consists of all measurements of each subject of study. However, due to the difference in the sampling rate of _x.csv and _y.csv files it produces data with missing labels.

3) To solve the problem described above, we have interpolated the data to produce final data frames using the 'pad' method from the pandas library.
4) Next comes the problem of data imbalance which is solved by sampling. We have used the 'SMOTE' technique to sample the data and maintain the balance of the data. SMOTE works by selecting pair of minority class observations and then creating a synthetic point that lies on the line connecting these two. It is pretty liberal about selecting the minority points and may end up picking up minority points that are outliers.

IV. Model Training and Selection

After the completion of the series of steps described above, we then perform model selection, training and testing of the data. This is followed by the hyper-parameter tuning and observation of results in the form of an accuracy and classification report.

A. Model Training

The model chosen for this study is the Random Forest Classifier. As part of this step, initially, the data is divided into a training and testing set in the ratio of 80:20. The model is trained with 80% of the data and 100 trees as a parameter.

B. Model Selection

The testing set is used for the prediction and evaluation of the model's performance. We have performed hyper-parameter tuning on the number of trees to produce better results by a method of trial and error. We've considered overfitting from the point where the number of trees increased but the f1-score and accuracy have dropped. This hyper-parameter tuning improved the model's performance on a moderate scale.

```
Accuracy: 0.9220053510001274
```

Fig. 3. Accuracy of the Random Forest Classifier.

V. Evaluations

The final model produced an accuracy of 92%, which is highly impressive. However, this could be misleading since we used the SMOTE method to perform sampling, which could bias the model. To better understand the results, we have used the classification report,"Fig. **??**" which consists of precision, recall, and the f1-score of the model's performance. Studying the various metrics, "Fig. 4", led to the conclusion that the selected model was showing decent performance on the given dataset. However, we would like to explore more complex models like CNN to observe and understand the trend of the data for better performance.

References

[1] C. Kertész, "Rigidity-Based Surface Recognition for a Domestic Legged Robot," in IEEE Robotics and Automation Letters, vol. 1, no. 1, pp. 309-315, Jan. 2016, doi: 10.1109/LRA.2016.2519949.

[2] S. Bhattacharya et al., "Surface-Property Recognition With Force Sensors for Stable Walking of Humanoid Robot," in IEEE Access, vol. 7, pp. 146443-146456, 2019, doi: 10.1109/ACCESS.2019.2945983.


```
=== Classification Report ===
              precision    recall  f1-score   support

         0.0       0.90      0.81      0.85    195360
         1.0       0.95      0.98      0.97    197032
         2.0       0.96      0.98      0.97    196372
         3.0       0.87      0.92      0.89    196136

    accuracy                           0.92    784900
   macro avg       0.92      0.92      0.92    784900
weighted avg       0.92      0.92      0.92    784900
```

Fig. 4. Classification report.