[Team 24] Proj-C2: Terrain Identification with Time Series Data

Ashwini Nayak
Email: aunayak@ncsu.edu

Krishika Shivnani
Email: kmshivna@ncsu.edu

Vignesh Sabesan
Email: vksabesa@ncsu.edu

I. MOTIVATION

The objective of this study was to investigate if the inertial data collected from normal human walk can be used to reveal underlying terrain types. We aim to develop a terrain identification system which collects inertial measurement units from sensors attached to the lower limb of a person. But in this project, we will observe if terrain identification is viable without visual data. We propose a LSTM model to identify the different types of terrain based on time series data.

II. METHODOLOGY

The architecture of our model is shown in Fig 1. The data consists of several sessions from 8 different subjects including the IMU data which was collected from the sensor attached to their lower limb. The accelerometer and gyroscope data were collected at a sampling rate of 40Hz, whereas the labelling for terrains was done at 10Hz.

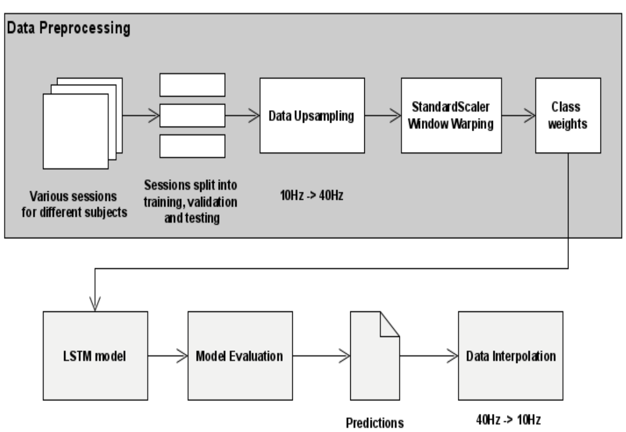

Fig. 1. Model Architecture

A. Data Pre-Processing

First, we separate the dataset files into training, validation and testing respectively. This will ensure that the model learns the data pattern for each subject. To match the sampling rate of X and Y, we upsample the class labels to 40Hz. Using a data augmentation method called window warping, we further reduce the size of the dataset by taking a sliding window of size 30, and running it through the dataset. For each window, we take the most common class label and append it to a new dataframe. A similar process is done for the validation and testing set as well. Since the data is imbalanced, we compute the class weights so that the LSTM model could learn the importance of each class label. The class weights for each label are {0: 0.33, 1: 6.08, 2: 4.65, 3: 1.52}. This preprocessed data is fed into the LSTM model for evaluation.

B. LSTM Model

For this task, we will be using a LSTM model to predict the activity label from the test set data. LSTM network models are a type of recurrent neural network that are able to learn and remember over long sequences of input data. We tried to model our time series classification with a LSTM because it handles the vanishing gradient problem and can provide more stable prediction ability in time series classification. Hence, it will be a good fit for this problem. Pienaar et. al, [2] has proposed a LSTM-RNN model on the human activity recognition dataset provided by WISDM. They were able to achieve an accuracy of over 94% and loss of less than 30% in the first 500 epochs. Zhao et. al, [3] has proposed a Bi-LSTM model to classify human activities based on sensor data. They have used a Bi-LSTM model because it would concatenate the positive time direction and negative time direction to produce a result. The main idea of using this approach is because of the high results obtained in IMU data, and its versatility.

The toolbox and libraries along with the specific functions used are listed below:

TABLE I
LIBRARIES AND FUNCTIONS

Libraries	Function names
pandas	DataFrame, read_csv
keras.model	Sequential
keras.layers	Dense, Dropout, LSTM, Activation, InputLayer
sklearn.metrics	classification_report, confusion_matrix
sklearn.utils	class_weight
sklearn.preprocessing	StandardScaler
keras.optimizers	Adam
keras.utils	to_categorical, np_utils
matplotlib	pyplot
collections	Counter

C. Structure of model

First we start by creating a Sequential model, which consists of a stack of layers where each layer has exactly one input and one output. Next we create an InputLayer with input shape as (30,6) and a LSTM layer with 125 neurons. A Dropout layer is used to reduce overfitting of the model. We then use one Dense layer with ReLU activation and another Dense layer with softmax activation. The last Dense layer is used to transform the size of the dataset to the desired output size. In our task, the output size will be 4 since we have four class labels (0, 1, 2, and 3). The structure of our model is shown in Fig. 2

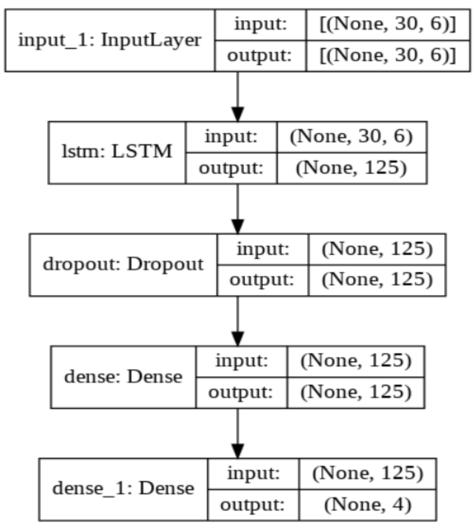

Fig. 2. Model Structure

III. MODEL TRAINING AND SELECTION

D. Model Training

The model was developed using the architecture explained in Section II. To preprocess the data, we first separate the datasets into training, validation and testing. The training set, validation set and testing set had 21, 2, 2 files respectively. The files used for validation were 'subject_007_02__x.csv' and 'subject_002_04__x.csv'. The files used for our own testing are 'subject_001_05__x.csv' and 'subject_005_01__x.csv'. Since X and Y have different sampling rates, upsampling was required to match the sampling rates of X and Y. Each label in Y were multiplied 4 times to match the sampling rate of X. A data augmentation method called window warping was performed to convert the sequential data to a time series data. This would help the model understand the data better, and produce accurate results. While analyzing the data, it was observed that the label 0 (standing or walking in solid ground) had more instances than other labels. Due to the imbalance of data, class weights were computed to give extra importance for the minority class labels.

The final training dataset had a shape of (991762, 30, 6), and the validation dataset had a shape of (98380, 30, 6) because of the window warping method. A LSTM model was defined with a Dropout layer and two Dense layers. This dataset along with the validation dataset were fed into the LSTM model to predict the test set data and compute the evaluation metrics.

E. Model Selection

The hyperparameters that we have considered in this task were the learning rate of the optimizer and the rate of dropout. Dropout is a technique used to prevent a model from overfitting. It works by randomly setting the outgoing edges of hidden units to 0 at each update of training phase.

TABLE II
HYPERPARAMETER TUNING

learning_rate	dropout_rate	Accuracy (%)	macro_avg
0.0001	0.5	91	0.85
0.0001	0.3	86	0.81
0.0001	0.2	92	0.86
0.0005	0.5	92	0.86
0.0005	0.3	93	0.83
0.0005	0.2	92	0.86
0.001	**0.5**	**93**	**0.87**
0.001	0.3	81	0.78
0.001	0.2	93	0.87

We ran each of these models for 8-10 epochs and finally chose the best parameters from these nine combinations. The best chosen model is the one trained with an Adam optimizer with learning rate of 0.001 and which had a dropout rate of 0.5 at the LSTM layer. This model gave an accuracy of 93%. Fig. 3 shows the model accuracy and loss plots for the training and validation dataset.

IV. EVALUATION

The confusion matrix for the prediction looks like:

TABLE III
CONFUSION MATRIX

	Label 0	Label 1	Label 2	Label 3
Label 0	85871	1212	1902	1952
Label 1	126	6046	48	27
Label 2	173	0	8991	7
Label 3	2213	3	54	6507

We know that our dataset has class imbalance with class 3 being the most under-represented. Hence, it is not enough to simply measure accuracy, as our model can just predict the majority class and still get a high accuracy. Therefore, we generate the precision, recall and F1 score values for individual classes along with the macro and weighted average which are specified in the table 4 below. Fig. 3 shows the loss and accuracy plots for the training and validation data on the LSTM model.

After evaluating the model using the confusion matrix and classification report, we load the test datasets which were

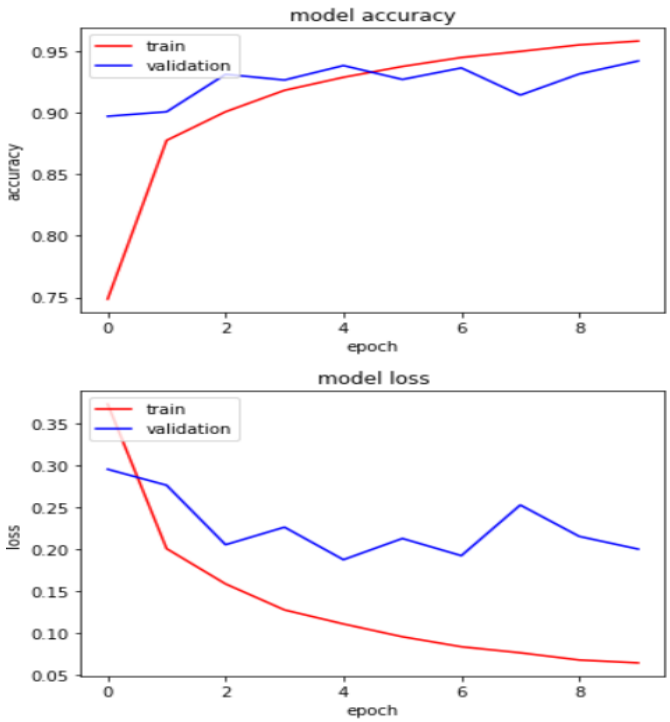

Fig. 3. Model Accuracy and Loss plots

TABLE IV
CLASSIFICATION REPORT

Activity Class	Precision	Recall	F1-score	Support
0.0	0.97	0.94	0.96	90937
1.0	0.83	0.97	0.90	6247
2.0	0.82	0.98	0.89	9171
3.0	0.77	0.74	0.75	8777
accuracy			0.93	115132
macro avg	0.85	0.91	0.87	115132
weighted avg	0.94	0.93	0.93	115132

hidden from the model. These datasets consist of 4 different subjects, out of which the first 2000 prediction labels are shown in Fig. 4, 5, 6 and 7. The F-1 score obtained from our own test set was 0.87 and the F1-score obtained from the test set in gradescope was 0.872.

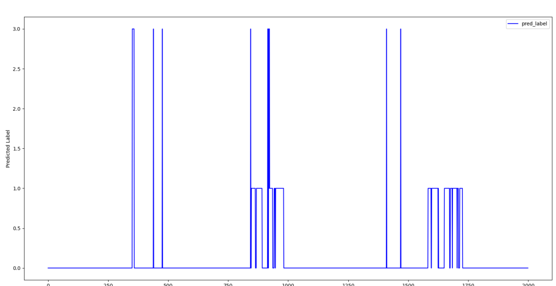

Fig. 4. Subject 9 Predictions

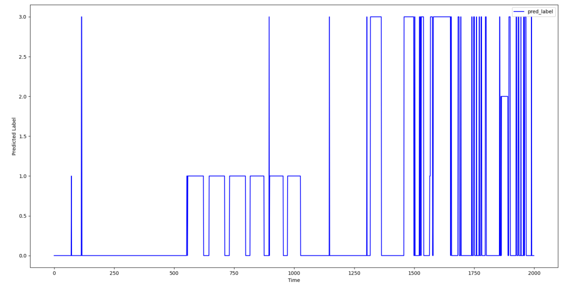

Fig. 5. Subject 10 Predictions

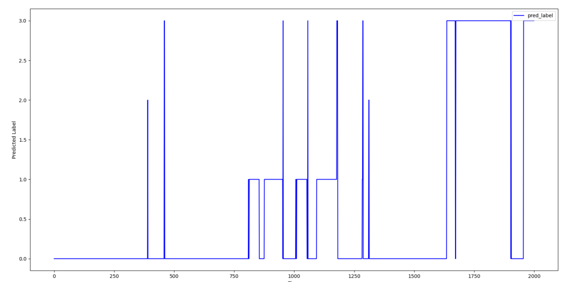

Fig. 6. Subject 11 Predictions

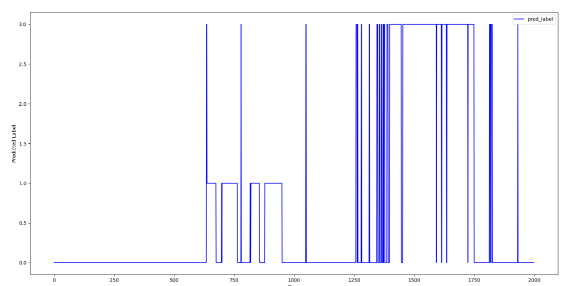

Fig. 7. Subject 12 Predictions

REFERENCES

[1] Hammerla, N. Y., Halloran, S., Plötz, T. (2016). Deep, convolutional, and recurrent models for human activity recognition using wearables. IJCAI International Joint Conference on Artificial Intelligence, 2016-January, 1533–1540.
[2] Mishra, P., Dey, S., Ghosh, S. S., Seal, D. B., Goswami, S. (2019). Human Activity Recognition using Deep Neural Network. 2019 International Conference on Data Science and Engineering, ICDSE 2019, 77–83. https://doi.org/10.1109/ICDSE47409.2019.8971476
[3] Zhao, Y., Yang, R., Chevalier, G., Xu, X., Zhang, Z. (2018). Deep Residual Bidir-LSTM for Human Activity Recognition Using Wearable Sensors. Mathematical Problems in Engineering, 2018. https://doi.org/10.1155/2018/7316954