Data Science Report

Energy Science and Technology M-Tech

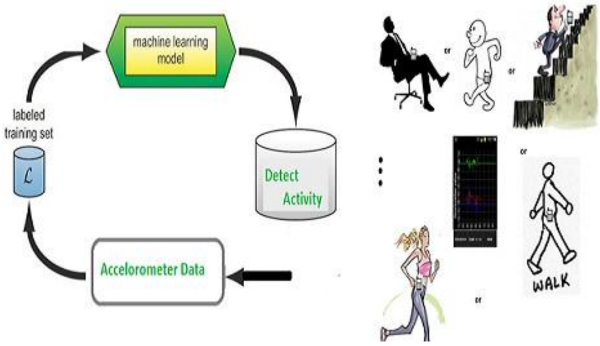
S Sandhya Rani (ET22MTECH11005)

Maurya Akshaykumar R. (ET22MTECH11002)

Human Activity Recognition Using Smartphones

Domain Background:

Since 1870 a large growth in human life expectancy has been observed in Europe. This growth has expanded the whole world principally due to the great achievements in the healthcare field. As a result, the proportion of elderly people is rapidly increasing. Aging people in general lives in isolated conditions. In addition to that some of them cannot live normally and take advantage of health care facilities services. Building remote monitoring systems for elderly patients who live alone or without permanent caretaking will improve their quality of life. For better decision-making, these remote monitoring systems need some regular and trustful patient information. The goal of this project is to build a machine learning model capable of collecting signals by using smartphone inertial sensors (accelerometer and gyroscope) HAR and HAPT datasets will be used as inputs of a machine learning model capable of recognizing some of human daily activities (sitting, walking …) included in the dataset. The final model could be used as a good source of information about patients' daily activities needed by remote monitoring systems.



The intuition behind the dataset:

The experiments have been carried out with a group of 30 volunteers within an age bracket of 19-48 years. Each person performed six activities (WALKING, WALKING_UPSTAIRS, WALKING_DOWNSTAIRS, SITTING, STANDING, LAYING) wearing a smartphone (Samsung Galaxy S II) on the waist. Using its embedded accelerometer and gyroscope, we captured 3-axial linear acceleration and 3-axial angular velocity at a constant rate of 50Hz. The experiments have been video-recorded to label the data manually. The obtained dataset has been randomly partitioned into two sets, where 70% of the volunteers were selected for generating the training data and 30% for the test data.

The sensor signals (accelerometer and gyroscope) were pre-processed by applying noise filters and then sampled in fixed-width sliding windows of 2.56 sec and 50% overlap (128 readings/window). The sensor acceleration signal with gravitational and body motion components was separated using a Butterworth low-pass filter into body acceleration and gravity. The gravitational force is assumed to have only low-frequency components, therefore a filter with a 0.3 Hz cut-off frequency was used. From each window, a vector of features was obtained by calculating variables from the time and frequency domain.

Y-Labels(Encoded)

In the dataset, Y-labels are represented as numbers from 1 to 6 as their identifiers.

- WALKING as 1
- WALKING_UPSTAIRS as 2
- WALKING_DOWNSTAIRS as 3
- SITTING as 4
- STANDING as 5
- LAYING as 6

Problem Statement:

Raw triaxial signals cannot be fed directly to machine learning models. A signal processing pipeline should be built to filter noise and extract useful and clean signals splitting them into windows. From each window, a vector of features is generated to obtain a classical dataset. The target column will contain activity labels associated with each vector of features(window). Recognizing the activity associated with each vector is a multiclass classification problem. To solve it, I intend to use some supervised learning classifiers (since each vector has an activity ID label), and deep learning (LSTM) and then compare predictions to see which model is working best for this problem.

It is a challenging time series classification task. It involves predicting the movement of a person based on sensor data and traditionally involves deep domain expertise and methods from signal processing to correctly engineer features from the raw data in order to fit a machine learning model. In other words, you can call, it a multiclass classification problem, for given a new data point we have to predict Human Activity corresponds to one of the 6 Activities.

Solution Statement:

We have a fairly small data set of Human Activity Recognition that has been labelled as "Walking", "Walking Upstairs", "Walking Downstairs", "Standing", "Sitting" and "Lying". We downloaded HAR Dataset from the UCI repository and we know that the data set is defined in two parts first is the RAW data set and second is pre-engineered by a domain or signal expert engineer. So first, we use a pre-engineered dataset with classical machine learning (ML) to learn from the data and predict human Activity. Second, we could then use the RAW dataset with a Deep learning model to learn from the data and predict human Activity.

What is the best performance metric for this Problem?

Accuracy: For any model we have printed the overall accuracy with this simple "Accuracy" metric. Accuracy is the proportion of correct predictions made by the model. While accuracy is a simple and intuitive metric, it can be misleading in imbalanced datasets where the classes are not equally represented

Confusion Matrix: The very important thing is that the confusion Matrix told us what types of errors and what types of confusion are happening. Simply for understanding this metric for this project view, we know that we have 6 class labels, and often times it could so happen that our Model will be confused between sitting or standing, and walking upstairs or walking downstairs. So, the confusion Matrix is a very-very important way of understanding which class your Algorithm or ML model is doing very well or for which classes your Algorithm or

ML model is getting confused. The best performance metric for this problem would depend on the specific goals and requirements of the application. However, some commonly used performance metrics for multiclass classification problems like human activity recognition are:

Precision: The proportion of true positive predictions among all positive predictions. Precision measures how often the model correctly predicts a class label when it is predicted.

Recall: The proportion of true positive predictions among all actual positive instances. Recall measures how often the model correctly identifies a class label when it is present in the data.

F1 Score: The harmonic mean of precision and recall. The F1 score provides a balanced measure of both precision and recall.

In summary, the best performance metric for this problem depends on the specific requirements of the application and the trade-offs between precision and recall. The confusion matrix is an important tool for understanding the model's performance and identifying areas for improvement.

Machine Learning Models Which We applied are:

Logistic Regression

Logistic regression utilizes a sigmoid function to model probabilities and is a linear classification model. The sigmoid function outputs a value between 0 and 1, making it well-suited for classification tasks. Overfitting occurs when a model learns training data too closely, leading to poor predictions on new observations. Ridge (L2) regularization addresses this by penalizing large predictors, reducing variance and preventing overfitting.

Hyperparameter Tuning-To optimize model parameters such as regularization factor and stopping criteria, cross-validation is an effective technique. The process involves holding out a validation set from the training data for each fold and training the model on the remaining data. This is repeated for a set number of folds, and the parameters with the best evaluation score are chosen as the optimal parameters for the model.

Linear SVC

A Linear Support Vector Classifier (Linear SVC) aims to identify the optimal hyperplane that can best separate and categorize our data. Once the hyperplane is obtained, the model can be used to classify new data based on the features provided.

Kernal SVM

SVM algorithms rely on a set of mathematical functions, known as kernels, to transform input data into the necessary form. There are various types of kernel functions available, which can differ in their structure and properties.

Decision Tree

Random Forest Classifier is a widely used predictive model in pattern recognition and machine learning. It is based on decision trees, which are hierarchically structured models that can predict responses from input data. In a Random Forest Classifier, multiple decision trees are constructed and their outputs are combined to produce more accurate predictions.

Random Forest Classifier

The Random Forest Classifier is a popular ensemble method that involves constructing multiple decision trees using a bootstrapping algorithm. Each tree is built independently and based on randomly chosen feature vectors. By combining the outputs of these trees, Random Forest can provide better predictions than a single decision tree classifier.

Deep Learning Models

For the HAR time series classification problem, various machine learning and deep learning models have been explored, with LSTM being one of them. LSTM is a Recurrent Neural Network model that can learn order dependence in sequence prediction tasks. It is well-suited for this problem as it can remember values over arbitrary intervals. In my implementation, I experimented with both a single-layer LSTM model and a two-layer LSTM model with hyperparameter tuning.

Benchmark Model:

For this problem, the benchmark model is Linear SVC or rbf SVM classifier or Logistic Regression as our best model while applying ML Classical Model. When we talk about LSTM Model, here with LSTM we are using simple RAW data (in the ML model we are using Single engineered data made by an expert), but we can see the result without any FE data, LSTM performs very-very well and got highest 91% accuracy with 2_layer LSTM with hyperparameter Tunning and also we can clearly when we are increasing LSTM layer and Hyperparameter Tunning the cross-entropy value is decreasing and Accuracy is increasing.

Results & Conclusion:

ML Model Accuracy score

Model Name	Hyperparameter Tuning	Accuracy Score
Logistic Regression	Done	**95.83%**
Linear SVC	Done	**96.47%**
RBF SVM classifier	Done	**96.27%**
Decision Tree	Done	**86.46%**
Random Forest	Done	**92.07%**

Deep Learning LSTM Model

Model Name	Hyperparameter Tuning	Cross entropy	Accuracy Score
LSTM With 1_Layer(neurons:32)	Done	**0.47**	**90.0%**
LSTM With 2_Layer (neurons:48, neurons:32)	Done	**0.39**	**90.0%**
LSTM With 2_Layer (neurons:64, neurons:48)	Done	**0.27**	**91.0%**

Reference:

1. **https://www.researchgate.net/publication/260003860_Human_Activity_Recognition_using_Smartphone/link/0c960536bbc554d6b4000000/download**
2. **https://link.springer.com/article/10.1007/s40860-021-00147-0**
3. **https://www.nature.com/articles/s41746-021-00514-4**

GitHub link:

https://github.com/sandhyarani08/Data-science-project.git