Runtime Error: Human Activity Recognition using Smartphones

Tejas Srinivasan
PES1201800110
Department of CSE, PES University
tejas.srinivasan007@gmail.com

Joseph Dominic Cherukara
PES1201800328
Department of CSE, PES University
jd.cherukara@gmail.com

Anirudh S Ayya
PES1201800338
Department of CSE, PES University
anirudhshrikanth65@gmail.com

GBS Akhil
PES1201800188
Department of CSE, PES University
akhilgbs123@gmail.com

***Abstract*—Human activity recognition is the identification of physical activities performed by human subjects using sensor data and is an up and coming topic in the fields of data analytics and machine intelligence. It has various applications in the fields of assistive healthcare, security and surveillance and human-computer interaction (HCR) to mention a few of the many applications. In this paper we have attempted to get an understanding of the sensor data provided to us in the dataset and its source and the conditions that were provided during its collection. We have done a thorough comparison of the models that have been implemented in previous papers through our literature survey and a summary of the results obtained by these papers and the limitations in their approach have been mentioned in the appropriate section as well. We propose a neural network based model to predict the activity undertaken by a subject and plan to compare this model with the previous models for results and tune it to see how it performs under different conditions on different metrics. Various experiments were carried out with a dataset of human activity with smart phones that was obtained from the UCI machine learning repository, which used mobile sensors to show that the proposed approaches can lead to the development of automated real time human activity monitoring for assistive healthcare of the elderly and people with special needs.**

***Index Terms*—Human activity recognition and prediction, Gyrometer and Accelerometer (sensor) data, Deep learning models**

I. Introduction

Since the invention of the mobile phone in the late 20th century, there has been unprecedented growth in their use, with more than 80% of the world owning a mobile phone in 2011. Newer phones are equipped with the ability to multitask and have a plethora of sensors.The integration of these devices into everyday life is growing rapidly, to the extent that they can monitor and track our activities, learn and assist in making decisions. Such technology can be of great use in remote healthcare for the elderly, disabled and those with special needs. However, at the moment, though there is a large capacity for collecting data from these devices, the capacity for automatic decision making and deriving inferences from the data is limited. There is an urgent need for new and efficient data analysis and machine intelligence techniques to be developed to this end.

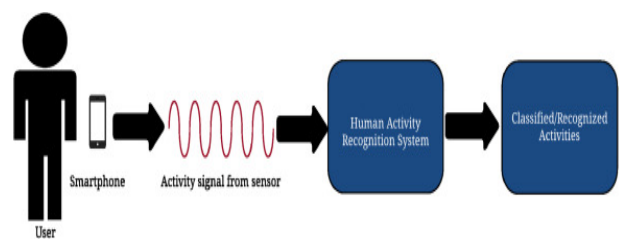

Fig. 1. A depiction of a human activity recognition system

The newest smartphones have 100000 times the processing power of the computer that landed man on the moon 50 years ago. They now come with a number of diverse and powerful sensors, including GPS, audio, proximity, temperature, acceleration, direction and so on. The integration of such sensors in most new devices provides a host of new opportunities for data analysis applications. This can be incredibly helpful for healthcare and remote monitoring of the elderly and people with special needs.

The activity that a user is performing could be inferred by complex analysis of the acceleration and gyroscope data that a smartphone records. One of the main reasons for using such data is to make inferences that are not dependent on the environment that the user is in. The data so obtained can be send across a network for further processing and analysis in real time. This dataset was obtained from orientational data from the accelerometer and gyroscope in a smartphone, while focusing on the classification of 6 user activities (i.e. sitting, standing, laying, walking, walking downstairs and walking upstairs).

II. Related Work

This section summarises the work that has been done using the dataset for various purposes. The assumptions made, a summary of the approach used and the results obtained have been mentioned. Furthermore any lacuna inferred from the following approaches along with any limitations have also been appropriately mentioned.

Canova et.al [1] mainly focus on building classifiers to accurately predict the activities performed by individuals using their smartphones sensor data. The original approach detailed a support vector machine (SVM) was adapted for multi class classification, using computational efficiencies that exploit fixed point arithmetic. Further studies also showed multiple sensors help in recognition because associations in acceleration feature values can help to identify many activities. A section on data visualization is also there in the paper, with an emphasis on two main techniques (PCA and t-SNE) to clearly understand the distinction in the categories of data, an approach that we plan to take as well. The algorithms distinguish between activities where the users are moving(like walking down and up the stairs) and activities where the user is not in motion (sitting, standing etc.) very effectively, Clustering is performed and each of the activities have a different cluster.

Different algorithms such as a baseline multinomial model, linear polynomial and radial based function kernel SVMs. At the end of the paper, the result of the models were shown using the overall misclassification rate as the primary performance metric. Each of the models displayed a similar performance in terms of the train and test error. This indicates that the usage of more complex models for the same task did not improve the performance by as much as seen on the other tasks. The linear kernel SVM was found to have the lowest misclassification rate and the training is computationally efficient. A confusion matrix was constructed for it using the entire train and test data. Moreover, activity specific observations were made (such as sitting being the one with the highest misclassification rate) etc. and t-SNE plots were constructed to explore the variance.

In conclusion the paper confirmed the hypothesis of some activities being classified accurately without subject knowledge while most others required it for training. It was also suggested to use Hidden Markov Models (HMMs) to capture the dependency between activities that involve transitions, an approach that we would love to explore as well. All references used to write the paper were cited in the appropriate section at the end of the paper.

All the latest smartphones are able to identify the inclination and position of the device with the help of tri-axial accelerometers. These accelerometers measure acceleration in the x,y and z spatial dimensions. Readings were recorded for each segment which is 10 seconds long and then features were generated based on the readings. The duration chosen was 10 seconds as 10 seconds was enough to observe the repetitive nature of the graphs that is the repetitive motion of the subject.

Jennifer et al. [2] observed the time between consecutive peaks in the frequency-time graph and concluded the repetitive activites are easy to identify as they are monotonous. To estimate this value, for each example first all peaks are identified and the highest peak for the x, y and z axis was identified using a heuristic method. Depending on the value of the highest peak, a threshold is identified and all other peaks are compared with this threshold value and selected accordingly. If none of the other peaks cross the threshold, the threshold is reduced until at least three peaks are found. We then measure the time between successive peaks and the average is computed.

The forward movement of the leg is recorded by the z-axis and the vertical movements (up and down) were recorded by the y-axis. The horizontal movement is recorded by the x-axis. Sitting and standing do not represent any periodic behavior. Periodic behaviour is shown only by the activities that have a repetitive nature. Activities that involve sudden change in acceleration was comparatively easier to identify as the graph had sudden change in the number of peaks.

Three learning algorithms we used - The J48 decision tree algorithm, logistic regression for multi-class classification and the multi layer neural network. The multi layer perceptron performed the best. Activites such as climbing up and coming down are similar in nature and generate similar graphs with similar height of peaks and hence were hard to identify. The puzzlement due to the above two activities led to many mis-classifications among them which decreased the accuracy by nearly 19.6%.

Girija et al. [3] aimed at developing the best model or scheme to classify various activities performed by users with the help of inertial sensors present in the smartphones.

For the validation of the approach, the dataset used consists of labelled data of 30 people within the age group of 19-48 years. The activity was performed wearing the smartphone having accelerometer and gyroscope sensors used for measuring 3-axial linear acceleration and angular velocity with a rate of 50Hz, it was placed around the waist. The dataset consists of total 561 features.

Information based ranking of features was used to remove some features. Random Forests model was used to reduce high variance and bias by computing the average and balancing the two extremities, due to this it is used to yield an efficient and fast model to solve the problem at hand. Lazy learners such as IBK was also used which are similar to k-nearest neighbour classifier.The models such as Naive bayes and k-mean clustering produced poor accuracies compared to the lazy IBK learner, random forests and ensemble based approaches accuracies.

III. Analysis of the Dataset

The Dataset [4] consists of 561 columns that are being considered as features for the model. A group of 30 volunteers were chosen to carry out the experiment. Each volunteer performed six activities in total (WALKING, SITTING, WALKING_UPSTAIRS, WALKING_DOWNSTAIRS, LAYING, STANDING) wearing a smartphone (Samsung Galaxy S2) on their waist. 3 axial linear acceleration and 3 axial angular velocity were measured at a constant rate by the use of the accelerometer and gyroscope built into the smartphone. The obtained dataset was partitioned into two csv files, 70% training set and 30% test data. A group of noise filters were used to preprocess the sensor readings using a sliding window of fixed length of (128readings/window).

Each activity has a markedly different accelerometer and gyroscope read-out as is explained in [3]. In Fig.2., the frequency of acceleration values on each axis is visibly different. Thus it is evident that by analysis of the physical values, we can predict what activity is being performed by the subject. We intend to use various models to perform these predictions and compare the accuracy of the same.

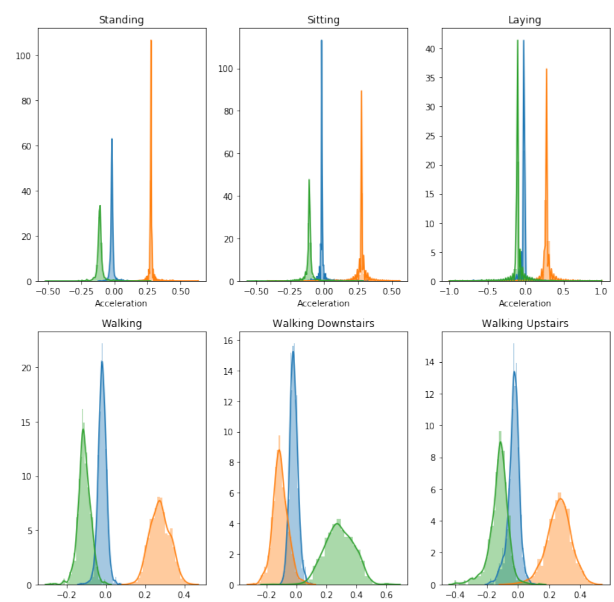

Fig. 2. Frequency of acceleration on different axes for each activity

A. Exploratory Data Analysis and Data Preprocessing

The dataset has 563 columns in total [5]. Out of the 563 entries it was found that 561 were of type float64, 1 was of type int and 1 object which was the column containing activity label. On performing basic data preprocessing, no missing values or duplicate values we found in the dataset. Apart from the activity label associated with each attribute, an identifier was also added to help identify the subject that carried out the experiment.

IV. Proposed Methodology

The dataset contains various physical values as well as labels of what activity the subject was performing when the readings were taken.The dataset has already been divided into testing and training sections. The problem at hand is to be able to predict what the subject is doing based on the given sensor data.

In order to use the values to predict what activity the subject is performing, we intend to use a number of different models, the explanation of the models and the experimental results of the same have been recorded.

A. K-Nearest Neighbour Classifier

The K-Nearest Neighbour Classifier has also been used to classify the activities into 'n' different classes. The main challenge here is to determine the optimal value of K, initially The value of K is determined by hit and trial, for this dataset K value of 7 has shown good results. Euclidean distance is used as a measure for calculating the distance between the k nearest neighbours and the new data point to be classified.

k-fold Cross-validation was used on the KNN model to find the optimal number of neighbors K, which gives the least misclassification error rate. Here the training data is divided into k folds, each time one of the fold is considered to be test data and rest of the k-1 folds are considered as train data, mean of the accuracies are computed. This process is done for certain range of values of K to find the optimal number of neighbors K, graph is plotted between misclassification error rate and number of neighbors K.

B. Support Vector Machine

In our case we have used support vector machines for multi-class classification. A hyperplane is found which exists in the vector space of N-dimensions where N is the total number of features present in the dataset that classifies data points into different classes or categories. To separate the classes, there are many hyperplanes that can be chosen as by the SVM. It finalizes on a plane which is at a maximum distance from the data points of either of the categories/classes .Maximizing the marginal distance gives less scope for misclassification so that when test data points arrive, classification error is the least.

The dataset is split into test and train datasets. They are cleaned and predictors are separated from the outcome variables. The train and test have 561 features and 7352 and 2947 samples respectively. The Y-training and test labels are categorical data and hence encoded into numerical data.

The train and test feature sets are scaled using StandardScaler.

We specify a parameter grid which is passed to the GridSearchCV with five fold cross-validation which consists of the kernels we plan to choose, the C and gamma values. We obtain the best estimator as the estimator with the kernel as rbf (radial basis function) as this had the least misclassification error on the dataset.

C. Artificial Neural Network

Since all the models mentioned above have been explored to some extent in previous work done on similar tasks, we decided to explore a deep learning approach to perform classification of the tasks into the various categories. A multi-layer perceptron neural network model was implemented in Keras (using Tensorflow as the backend) to classify the activities of the subject in the data provided in the train csv file and then testing to see how accurately it classifies the output into the different categories of activities performed by the subject. The model used was a Sequential model implemented in Keras, in which there were 4 dense layers used in the hidden layers with 'relu' chosen as the choice of activation function. BatchNormalization also was used to avoid overfitting on the training set and the final dense layer used a softmax to classify the outputs into different categories. The neural network was trained for 50 epochs and the adam optimizer used and the neural net was evaluated using accuracy as a metric. The results and the insights obtained are mentioned in the appropriate section.

D. KMeans Clustering

KMeans Clustering is an Unsupervised Learning model that is used to cluster data. It works to divide n points into k clusters. Each point is added to the partition with the nearest centroid. The centroids can be initialized in a number of ways. Random initialization is generally used.
The dataset was labelled as having 6 clusters. Thus the model was implemented to partition the data into 6 clusters. 2 different centroid initialization methods were used and the results were compared. The insights from the results were then used to improve the model.

V. Experimental Results

A. K-Nearest Neighbour Classifier

From the graph plotted between mis-classification error and number of neighbors K, it shows that the optimal number of neighbors K is 11. This is obtained by considering the 10-fold cross-validation applied on the training data. The training accuracy is found to be 97.41% and testing accuracy is found to be 90.39%.

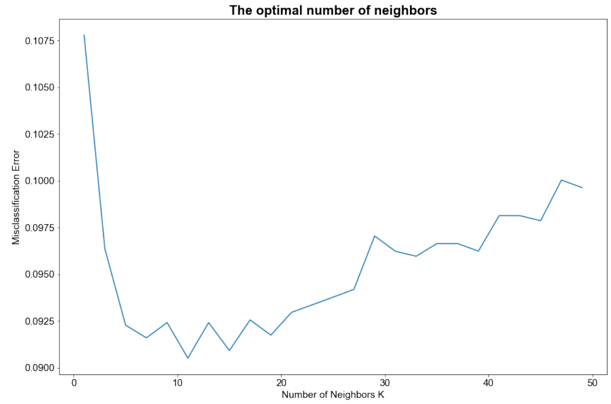

Fig. 3. Finding out the optimum number of neighbours

B. Support Vector Machine

We obtain the best score (Mean cross-validated score of the best_estimator) for training data as 98.59%. Best C is 1000 and the best gamma is 0.001. This was obtained by the rbf kernel. The testing set score for SVM on the rbf kernel is 95.89%. A table of the various activities and the associated metrics have been shown below.

	precision	recall	f1-score	support
LAYING	0.99	1.00	1.00	537
SITTING	0.98	0.89	0.93	491
STANDING	0.92	0.98	0.95	532
WALKING	0.96	0.98	0.97	496
WALKING_DOWNSTAIRS	0.98	0.93	0.95	420
WALKING_UPSTAIRS	0.93	0.96	0.95	471
micro avg	0.96	0.96	0.96	2947
macro avg	0.96	0.96	0.96	2947
weighted avg	0.96	0.96	0.96	2947

```
Training set score for SVM: 1.000000
Testing  set score for SVM: 0.958941
```

Fig. 4. Table of various activities with precision, recall, f1-score and support

The confusion matrix was constructed with the predicted classes on the x-axis and the true classes on the y axis. We observe the majority of the samples are correctly classified and very few are mis-classified by a very close margin.

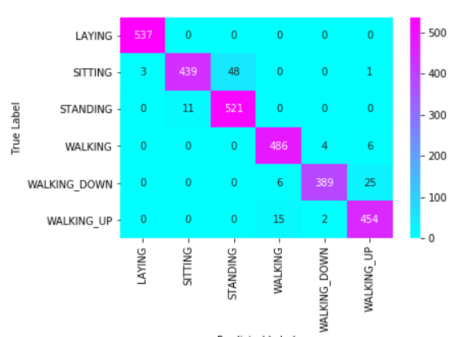

Fig. 5. Confusion matrix depicting actual vs predicted number of samples

C. Artificial Neural Network

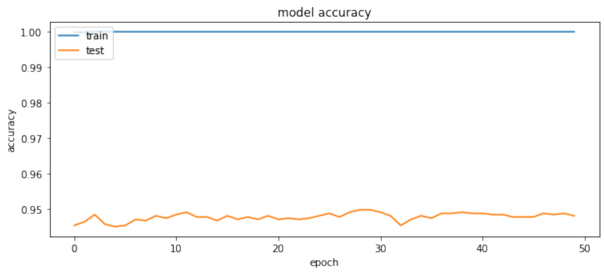

Fig. 6. A plot of the variation of the model's accuracy with number of epochs during training and test times

Hyperparameters like the number of epochs, learning rate, batch size etc. was found by tuning the hyperparameters for the best possible score on metrics such as accuracy,precision,F1 score etc. As we can see from this graph, the train accuracies in constant at 100% and the model does reasonably well with a 95% accuracy on the test dataset.

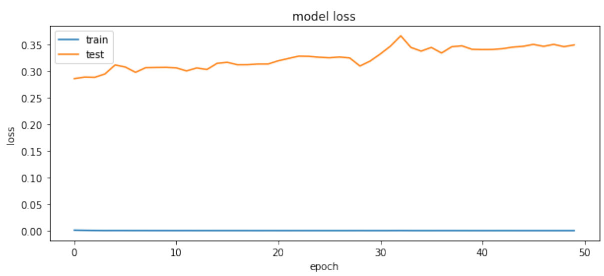

Fig. 7. A plot of the variation of the model's loss with number of epochs during training and test times

The loss of the model decreases a little bit after the first few iterations, after which it remains almost constant at 0.35.

D. KMeans Clustering

The number of clusters was set to 6 and the initialization of the centroids was random. In fig6, t-SNE was used to reduce the data into 2 dimensions for easier visualization. In the graph on the left, the colours represent the clusters that the model generated. The graph on the right shows the distribution of the original labels.

Fig. 8. t-SNE plots of the classified data and the original labels

We can see from the graph that Kmeans is unable to distinguish between a few of the clusters. This is because Kmeans doesn't work well with non-globular shaped clusters. The model by default produces globular clusters as it is distance-based. From the heatmap it is clear that the algorithm is unable to clearly classify the static activities (i.e. laying, sitting and standing) and is unable to clearly classify the dynamic activities (i.e. walking, walking_downstairs and walking_upstairs). However the model is clearly able to distinguish between static and dynamic activities. This is evident from the fact that the data points from static activities form clusters where there are little to no data points from the dynamic activites and vice versa.

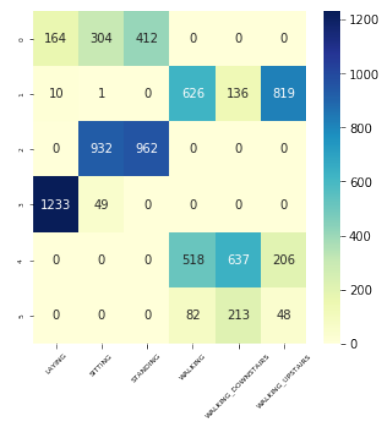

Fig. 9. Heatmap of the output of Kmeans clustering into 6 clusters

The Kmean algorithm suffers from another issue wherein the initialization of the centroids has a major impact on the final clusters that are formed. To avoid this, we tried calculating the centroids of each activity label and used those as the inital centroid values. This however did not have a significant effect in the results and the algorithm performed worse (required more iterations to reach convergence). Hence this result was discarded.

Since the model was unable to classify the 6 activities clearly, Kmeans was done with the number of clusters set to 2. Each cluster would represent the Static Activities and Dynamic Activities.

This model gave us much better results. The algorithm was able to classify between static and dynamic activites accurately. Fig7. shows the results of the model. t-SNE was done for easier visualization. The graph on the left shows the results of the Kmeans clustering model. The graph on the right shows the split of the data into Static and Dynamic activities.

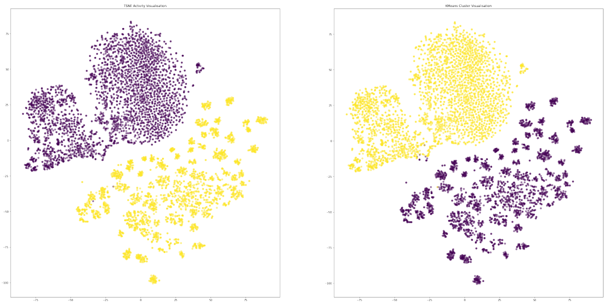

Fig. 10. t-SNE plots of the classified data and the original labels

This is further proved by the fact that the heatmap shows that only 12 data points (out of 7352) were misclassified. Putting it at an accuracy of 99.83%

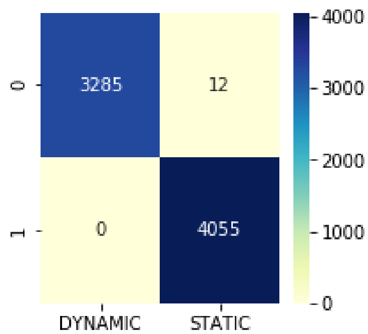

Fig. 11. Heatmap of the output of Kmeans clustering into into clusters

VI. Conclusions

After comparing all the models that we tried for the classification of tasks into different activities, we observed that although most of the models give us very high accuracy on both the testing and training fields, using an artificial neural network to accomplish this task (a sequential model with dense layers as mentioned), gave us the best results and consistently performed well on the cross-validation and test datasets, and hence this model can be deployed for accurate classification of activities which can be very useful for prediction of current activity state and hence can be used for assistive healthcare for the elderly.

In the future, we could also try to implement models like Convolutional Neural Networks(CNNs) for prediction of the activities more accurately by labels within the Static/Dynamic activities classification since CNNs are known to perform very well on labeled image data which is provided. Since the accelerometer data provided is taken in a sequence at different windows of times recorded using the smartphone, we can use sequential learning models like Long-short term memory units (LSTMs) to capture these dependencies or even a hybrid ConvLSTM model of a single dimension to achieve the same. Having also studied Hidden Markov Models, we could try using a variation of those to capture the change in state between activities classified as static and those classified as dynamic.

In conclusion, these models can be used to identify what the user of a mobile device is doing at a given point in time. This information can then used for various purposes such as assistive healthcare, spatio-temporal analysis and for many more applications.

VII. Contributions of Team Members

Each of us was involved in the data preprocessing and exploratory data analysis section. Further for stage 2, each of us worked on a model for classification of the activities, the exact details of which are given below -
Tejas - Artificial Neural Network
Anirudh - Support Vector Machine
Akhil - K-Nearest Neighbour Classifier
Joseph - K-Means Clustering Algorithm
All of the team members were involved in the writing of the report and creation of the video and all decisions regarding the project were taken on call. All the code has been documented and all the references used have been mentioned appropriately.

References

[1] F. S. D. . Nicholas Canova, "Human activity recognition using smartphone sensor data." http://cs229.stanford.edu/proj2016/report/CanovaShemaj-HumanActivityRecognitionUsingSmartphoneData-report.pdf.

[2] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, "Activity recognition using cell phone accelerometers," *ACM SigKDD Explorations Newsletter*, vol. 12, no. 2, pp. 74–82, 2011.

[3] G. Chetty, M. White, and F. Akther, "Smart phone based data mining for human activity recognition," *Procedia Computer Science*, vol. 46, pp. 1181–1187, 2015.

[4] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, "A public domain dataset for human activity recognition using smartphones.," in *Esann*, vol. 3, p. 3, 2013.

[5] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, "Energy efficient smartphone-based activity recognition using fixed-point arithmetic.," *J. UCS*, vol. 19, no. 9, pp. 1295–1314, 2013.