International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 07 Issue: 05 | May 2020 www.irjet.net p-ISSN: 2395-0072

Human Activity Recognition using Triaxial Accelerometer Sensor Data

Choudhari Pratik[1], Deore Aryan[2], Ganwani Kapil[3], Agarwal Chetan[4]

[1,2,3]*Students, Dept. of Information Technology, Thadomal Shahani Engineering College, Mumbai, India*
[4]*Asst. Professor, Dept. of Information Technology, Thadomal Shahani Engineering College, Mumbai, India*

Abstract - *This paper describes results of different algorithms used to recognize certain types of human activities using data generated accelerometer present in a user's cell phone. The mode was trained with multiple human subjects and tested in real-world conditions. Data is collected through an android app that collects readings from triaxial accelerometer present in the phone and appends the readings to a csv file. We considered three models for our recognition system - Random Forest, Support Vector Machine, RNN – LSTM. The findings show that LSTM shows maximum accuracy.*

***Key Words*: Human Activity Recognition, Accelerometer, Android App Random Forest, Support Vector Machine, RNN, LSTM.**

1. INTRODUCTION

Human activity recognition is defined as an ability to interpret human body gesture or motion with the help of sensors and determine the activity performed. The recognition of human activities can be done in two different ways, namely using external and wearable sensors. In the former, the devices are fixed in a fixed point of interest, so the activities completely depend on the interaction of the subject with the sensors. In the latter, the sensors are attached to the user. The device attached to the user in this case is the android phone in the pocket of the user. External activity tracking devices are a premium investment. An alternate affordable option would be to use a device that everybody already has. Android phones are packed with sensors that are not utilized when the phone is in its idle state. Using the sensors in the android phone, an application can be made to track the activity of the user. The app is allowed to run in the background to collect the data.

A triaxial accelerometer is a sensor that gives an estimate of acceleration along the x, y and z plane. Tis data can be used to estimate velocity and displacement. Activity recognition falls under the category of supervised classification problem, where training data is gathered from an experiment containing human subjects who perform each of the activities. We aim to develop a system that can recognise multiple sets of daily activities performed under real-world conditions, with the data collected by just a single triaxial accelerometer that is built into the user's cell phone (in our study, an Android smartphone).

2. LITERATURE SURVEY

In a review paper distributed by Shian-Ru Ke[1] , he concentrates broadly the advances made till date toward video-based human action acknowledgment. Three viewpoints for human movement acknowledgment are referenced and they are core technology, human activity recognition systems, and applications from a low-level to a high-level representation.

An examination paper distributed by Akram Bayat on "Human Activity Recognition Using Accelerometer Data from Smartphone"[2], depicts the technique to perceive sorts of human physical exercises utilizing the sensor information produced by a client's mobile phone. It proposes a framework wherein another sort of computerized low-pass channel is utilized so as to isolate the part of gravity speeding up from the body increasing speed in the crude information.

In a review paper published by Ong Ching Ann and Lau Bee Theng, Human Activity Recognition-A Review, "a total of thirty-two research papers on sensing technologies used in HAR are studied. The review summarises three area of technologies namely RGB cameras, depth sensors and wearable devices. It also discusses on the pros and cons of each of the mentioned sensing technologies [3].

A survey paper distributed by Oscar D. Lara and Miguel A. Labrador on A Survey on Human Activity Recognition utilizing Wearable Sensors [4] advances a two-level scientific classification for the learning approach (either administered or semi-managed) and the reaction time (either disconnected or on the web). The acknowledgment of human exercises has been drawn nearer in two distinct manners, first utilizing outside and afterward utilizing wearable sensors.

In the review paper published by T. Subetha, Dr. S. Chitrakala on A Survey on Human Activity Recognition from Videos[5], explains the applications of HAR like Content-based Video Analytics, Human-Computer Interaction, Ambient Intelligence, Visual Surveillance, Video Indexing, Robotics, Human fall detection, etc...

We propose using android phones not only to collect the data but also to process it. Android phones are packed with sensors that are not utilized when the phone is in its idle state. Using the sensors in the android phone, an application can be made for tracking the activity of the user. The app can be allowed to run in the background to collect the data. The model could be built right in the phone. This would allow the phone to process the data locally without any internet.

The algorithms under consideration are –

2.1 Random Forest

Random forest is a tree-based algorithm which includes creating multiple decision trees, further aggregating their result to improve generalization ability of the model. Random Forest could be used to solve regression and classification problems. Regression problems have dependant variable as continuous value. Classification problems have dependent variable as categorical. In random forest classifier, the accuracy is increased with number of trees is the forest. Random Forest was considered because it is can solve both regression and classification problems and solve unsupervised ML problems. But it can take longer than expected time to computer a large number of trees.

2.2 Support Vector Machine

The basic guideline behind the working of Support vector machines is straightforward – Create a class isolating hyperplane to isolate whole dataset in classes. SVM finds the focuses that will lie nearest to both the classes. These focuses are known as help vectors. Further, it finds the separation between the isolating plane and the help vectors. This separation between the focuses and the partitioning line is otherwise called edge. The purpose of a SVM calculation is to augment this very edge. The hyperplane turns into the ideal one, when the edge arrives at its most extreme worth. SVM was considered due to the way that the arrangement will consistently be worldwide least not a nearby least, SVM could be utilized for both straightly distinguishable and non-directly divisible information. Directly divisible information has a hard edge though non-straightly detachable has a delicate edge.

2.3 Recurrent Neural Networks

Neural networks are set of algorithms motivated by the working of human brain. They accept a massive set of data, process the data (drawing out the patterns within the data), and returns what it is. The thought behind RNNs is to utilize sequential information. In a conventional neural network, we assume that all inputs (and outputs) are independent of each other. But for many problems that's not the most optimal approach. RNNs are called recurrent because they perform the exact same task for every component of the sequence, with the result being depended on the prior calculations and we know that RNNs have a "memory" which captures information about what has been calculated till the current instance. LSTM's have a tendency to remember information for a longer period of time. Every LSTM cell will have 3 gates namely Forget gate, Input gate, Output gate. Accuracy of Neural Networks increase with increase in data. But they take longer to train and are sensitive to different random weight initializations.

3. METHODOLOGY

The system will be tested on various classification models like Random Forest, Support Vector Machine, RNN – LSTM. The selected models will be training dataset consisting of 1,60,000 tuples divided equally among 4 classes. As shown in Figure 1, the trained model will get aits testing set in the form of a CSV file generated from android phone. The model will return the predicted class label. The system will be divided into various modules like -

3.1 Input Module

As shown in Figure 2, data will be taken as input from an android app. The file generated will have comma separated values. The triaxial-accelerometer reads data in three planes and writes the data along with time stamp in a csv file.

3.2 Processing Module

This csv file is then passed to a pre-trained classification model in the app. Various classification models are tested for accuracy and the model with the highest accuracy is selected. Accuracies of the tested models are mentioned in Table 2.

3.3 Output Module

As shown in Figure 5, the app will return the predicted activity on the prediction screen of the app. Based on the predicted activity and the amount of time an activity is performed, the app will determine the number of calories burnt and return the amount left to burn to reach the goal. Predicted activities include Walking, Sitting, Climbing Upstairs, Going Downstairs.

3.4 Data Info

- Sample Rate – 50Hz.
- Number of participants – 7
- Number of Class Labels – 4
- Accelerometer orientation – Calibrated.
- File Format – Comma Separated Values (CSV).
- Sensor used to collect data - from Tri-axial accelerometer.
- Data Distribution of class labels -
    - Climbing Upstairs - 40,000 tuples
    - Going Downstairs - 40,000 tuples
    - Walking – 40,000 tuples
    - Sitting - 40,000 tuples

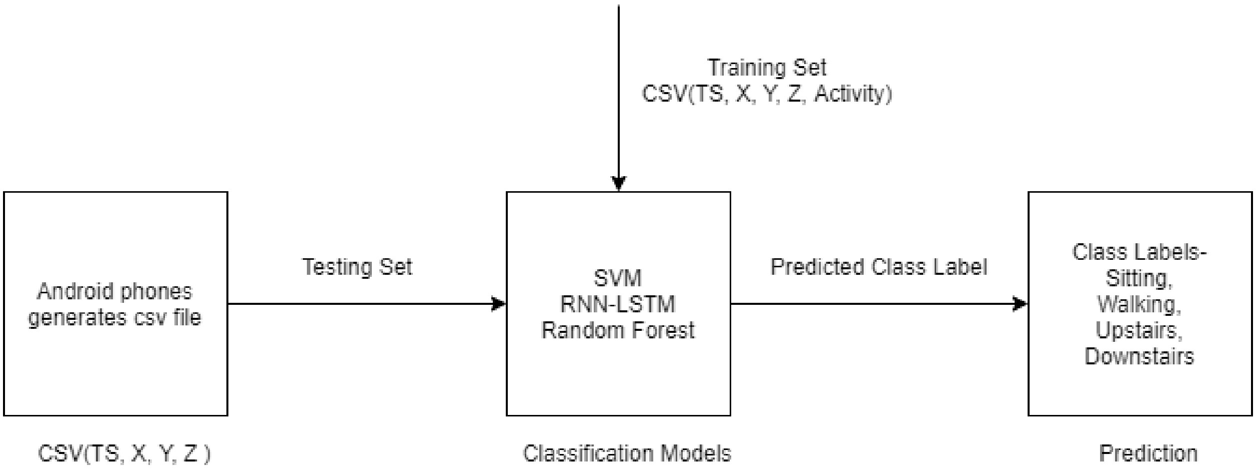

Fig -1: Architecture of proposed HAR system

Table -1: Fields of csv file

Csv File Fields	
Feature Label	Feature info.
X	X coordinate collected from the sensor.
Y	Y coordinate collected from the sensor.
Z	Z coordinate collected from the sensor.
TS	UNIX Timestamp
Activity	Class Label

4. METHODOLOGY

4.1 Data Collection

Data is collected through an android app that collects readings from triaxial accelerometer present in the phone and appends the readings to a csv file as shown in Figure 2. Zhen-Yu He[6] found that the best place to mount the accelerometer is in the jeans pocket. Instead, study by Uwe Maurer[7] suggest that the accelerometer should be carried in a bag wore by the user whereas research by Emmanuel Munguia Tapia[8] suggests that the accelerometer could be strapped on to the dominant wrist. In the end, the optimal position to place the accelerometer is subjective and depends on the application and the type of activities to be recognized.

In our case, the best orientation of the phone happens to be in the trousers pocket. The front facing camera should be on top and the screen should face forward. The csv file is saved in the android phone locally with the path /storage/emulated/0/Documents/TSEC_HAR/_file_name.csv

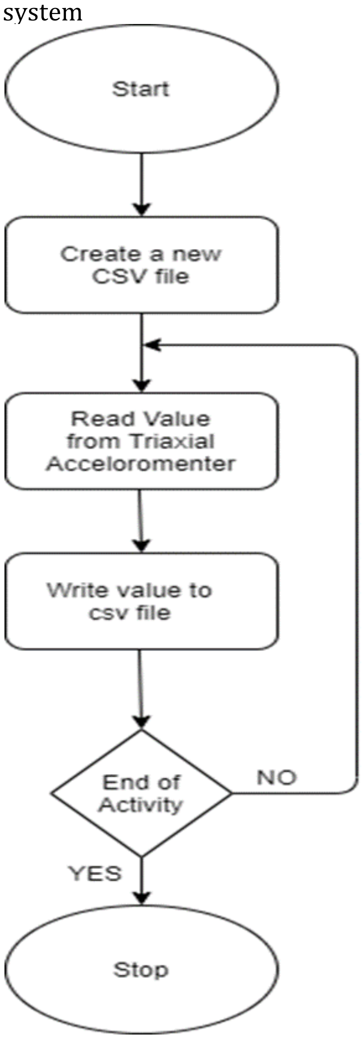

Fig -2: Input module of the app

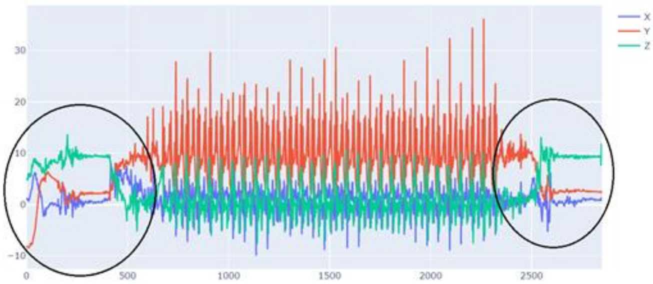

Fig -3 – Impure data

4.2 Data Cleaning

The tuples generated during the time (3 sec) it takes to start recording and placing the phone in the pocket generated impure data. Similarly, the tuples generated during the time (3 sec) it takes to take the phone out of the pocket and stopping the recording generated impure data. This data must be removed before feeding the data to the model.

A more precise way to clean the data would be to visualise the data and drop the tuples which are not in the general pattern of the activity performed. For example, a sample of impure data is displayed in Figure 3. The tuples circled don't follow the general pattern of the activity performed, hence they need to be trimmed. The trimmed(clean) data is shown in Figure 4.

4.3 Data Organisation

The cleaned data from all the participants is appended in to a single file. Tuples of Walking from all the participants is first appended followed by tuples of Sitting from all the participants followed by tuples of Upstairs from all the participants followed by tuples of Downstairs from all the participants. The final csv is then used to train the model.

4.4 Model Training

There are three models under consideration –Random Forest, Support Vector Machine, RNN – LSTM.

Random forest algorithm was considered because it is good at handling binary features, categorical features, and numerical features. A very little pre-processing is required. The data need not to be transformed or rescaled.

SVM algorithm was considered because it uses L2 Regularization. So, this prevents SVM form over fitting due to good generalization capabilities.

RNN algorithm was considered because it remembers things learnt from prior input(s) while generating output(s).

Based on Research done by Junyoung Chung[9] we decided to implement the gated version - LSTM or GRU over the traditional tanh implementation of RNN. We implemented LSTM as it has three gates output, input and forget gate compared to GRU which has two gates update and reset gate.

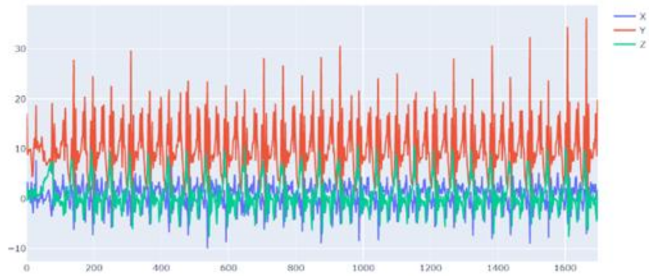

Fig -4 – Clean data

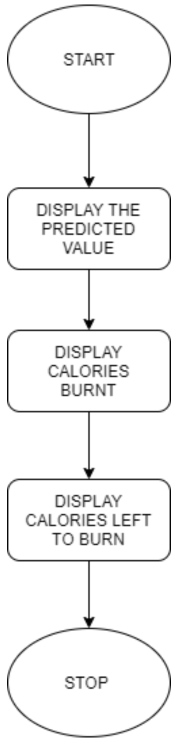

Fig -5 – Output module of the app

5. RESULTS

The observed accuracies of various models are –

Table -2: Accuracies of various classification models

Accuracies of models	
Model	Accuracy
LSTM	95
SVM	85
Random Forest	66.4

We observe that LSTM performs the best in terms of accuracy. It also takes the shortest amount of time ie. 5 minutes to train the model. Random Forest performs poorly but takes a decent 8 minutes of time. SVM takes the longest to train with 30 minutes and return an accuracy of 85 percent.

6. CONCLUSIONS

The model with the highest accuracy was chosen for the implementation of current model. Support Vector Machine shows a significant boost in accuracy compared to Random forests. LSTM was recorded to have the highest accuracy of 95 percent, which is 11.7 % more than SVM and 43% more than Random Forest. LSTM emerged as the clear winner not only because of its higher accuracy, but also due to its ability to improve its accuracy with increase in training data. Future work could include –

- adding a bias of 1 to the forget gate of every LSTM as done by Rafal Jozefowicz[10].
- implementation of model in cloud.
- using camera-based input for HAR.

7. REFERENCES

[1] S. R. Ke, H. L. U. Thuc, Y. J. Lee, J. N. Hwang, J. H. Yoo, and K. H. Choi, A review on video-based human activity recognition, vol. 2, no. 2. 2013.

[2] A. Bayat, M. Pomplun, and D. A. Tran, “A study on human activity recognition using accelerometer data from smartphones,” Procedia Comput. Sci., vol. 34, pp. 450–457, 2014, doi: 10.1016/j.procs.2014.07.009.

[3] O. C. Ann and L. B. Theng, “Human activity recognition: A review,” Proc. - 4th IEEE Int. Conf. Control Syst. Comput. Eng. ICCSCE 2014, no. April 2018, pp. 389–393, 2014, doi: 10.1109/ICCSCE.2014.7072750.

[4] Ó. D. Lara and M. A. Labrador, “A survey on human activity recognition using wearable sensors,” IEEE Commun. Surv. Tutorials, vol. 15, no. 3, pp. 1192–1209, 2013, doi: 10.1109/SURV.2012.110112.00192.

[5] T. Subetha and S. Chitrakala, “A survey on human activity recognition from videos,” 2016 Int. Conf. Inf. Commun. Embed. Syst. ICICES 2016, no. July 2018, 2016, doi: 10.1109/ICICES.2016.7518920.

[6] Z. Y. He and L. W. Jin, “Activity recognition from acceleration data using AR model representation and SVM,” Proc. 7th Int. Conf. Mach. Learn. Cybern. ICMLC, vol. 4, no. July, pp. 2245–2250, 2008, doi: 10.1109/ICMLC.2008.4620779.

[7] U. Maurer, A. Smailagic, D. P. Siewiorek, and M. Deisher, “Activity recognition and monitoring using multiple sensors on different body positions,” Proc. - BSN 2006 Int. Work. Wearable Implant. Body Sens. Networks, vol. 2006, no. May 2006, pp. 113–116, 2006, doi: 10.1109/BSN.2006.6.

[8] E. M. Tapia et al., “Real-time recognition of physical activities and their intensities using wireless accelerometers and a heart rate monitor,” Proc. - Int. Symp. Wearable Comput. ISWC, pp. 37–40, 2007, doi: 10.1109/ISWC.2007.4373774.

[9] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, “Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling,” pp. 1–9, 2014, [Online]. Available: http://arxiv.org/abs/1412.3555.

[10] R. Jozefowicz, W. Zaremba, and I. Sutskever, “An empirical exploration of Recurrent Network architectures,” 32nd Int. Conf. Mach. Learn. ICML 2015, vol. 3, pp. 2332–2340, 2015.