**Analysis of Activity Recognition Data using IoT Smart-Watch and Personalised Recommendation by Applying Machine Learning Algorithm**

*Abhirup Dey*

*VIT University, Vellore*

*Department of Computer Science and Engineering*

*M.Tech CSE*

*R. Rajkumar*

*Associate Prof.*

*Department of Computer Science and Engineering*

***Abstract-- In this smart emerging world, modern day equipment, like wearable devices, not only provides functionality or advancements in lifestyles but also becoming a trending fashion choice. Most of the devices which are wearable provides basic functionalities like display time or date. But implementation of more smart features like displaying message, phone call or even medical activity recognition can lead the productivity in dense and holds a potential to create a product demanded by huge number of customers. Smart wearable devices connected to internet approaches the methodology and required application and implementation of secure IoT environment and cloud infrastructure. Compared to other internet connected devices wearable devices like smart watches are designed to be capable of monitoring activity for 24 hours a day. Mostly they are designed as durable and water resistance with addition of appropriate sensors for required functionalities and detection. In this paper we are proposing a model for identifying requirements of activity and inactivity recognition by implementing on a secure and smartly designed cloud infrastructure. Here we are also defining a new measurement of heart-rate data applying various machine learning methods.***

***Key Words:*** *IoT, Smart-Wearable-Devices, Cloud, Machine Learning, Web Application, Activity Recognition, Personalized Recommendation*

**1. Introduction**

21st century wearable devices like smart watches not only offer visual display of time and date, but also gives us several other feature rich functionalities which helps to create advancement in day to day human life. Most recognised feature for smart watch is health monitoring. Wearable technology introduced the methodology of continuous monitoring of medical and personal data. This not only gives us productivity and efficiency, but also provide us a better way to live our life- A smart life.

Adding sensors to wearable device enhance the functionalities for collecting data about user activity. By collecting and storing the data into a database or secure storage we can provide a platform of innovation for third party vendors. There is various availability of application of accelerometer and gyroscope application to recognise activity of a particular user. Activity and inactivity recognition of individuals have become a current development scenario for wearable devices. But we can do lot of enhancements and predictive approach by applying machine learning to those data set.

In this paper we will discuss about possible futuristic implementation of personalised activity recommendation based on the data collected through smart watch. Besides we will present the scenario of collecting heart rate data from test cases and predicting upcoming health issues by applying available machine learning model into it. Which does not only add valuable medical functionality, but also gives a boost to the existing scenario and an upgraded technical application.

**2. Related Work**

**2.1 Smart Watch**

In the era of 90’s watches had been designed to be connected to modern computers by integrating light sensors in. After the invention of modern wireless technologies connectivity for smart devices has been redesigned [].

Smart watches are wearable in hands and is placed in a particular place to get stable data. Consisting with screen a smart watch is designed to capable of displaying communication services like SMS, Facebook Feed, WhatsApp and even incoming calls. Some watches are enabled to control music and attached with camera module to capture details in stealth.

Smart watches are classified into two separate genres based on the implementation and design -

*The first type* is known as autarkical watch. The type which comes out with a built-in computational power and dedicated wireless connection for synchronization over the internet. Thus, the processing unit directly process the data with the watch itself. Which undoubtedly demands more power for running [].

*The Second type* consist of a dedicated terminal for event listening and transmitting the records to a hub like a smart-phone device. The heavy work of processing data and transmit back the events to the smart watch. In this way the smart-watch is able to save battery and no extra requirement for computational power [].

**2.2 Activity Recognition**

Research into the field of activity recognition based on the fact of presence of various sensors in a smart-phone, is being conducted for past few years []. But it’s been depicted that a smart-phone is not a good choice for tracking daily activity of a person as they keep their phone inside pockets at idle state. Which places the sensors for tracking near the thigh area and definitely not a suitable place to track activity data from a human body. Mostly people keep their phones aside after usage. In other way, the major functionality of sensor trackers can be void by the inference of other important usage purpose of a mobile phone.

Compared to a smart-phone it’s more suitable to use a smart-wearable device for tracking activity, as it is kept in a particular place of our body and track data for hours without any interruption. Smart-watch is most ideal because it is usually placed in our wrist and an individual mostly use his hand to commit most of the daily works like eating, drinking, writing or even exercise. It’s been studied that a smart-watch is capable of identifying drinking activity by an accuracy of 93.3% compared to a smart-phone with a recorded accuracy of 77.3%[].

**2.3 Activity Unit**

The activity recognition unit identifies the activity mainly based on two sensor units which are accelerometer and gyroscope. Now the accelerometer captures the motion and positioning based on the Cartesian coordinate measurement system. It measures the three axis sensor values as x, y and z which are perpendicular to each other. The sensor values will give a similar output as gravitation force ‘g’ (9.81 m/s2). Which is calculated by the equation –

gl = ∑ (x2 + y2 + z2)0,5

Gravitational force (g) influence every object present on earth. Though the gravitational force is not constant everywhere on earth, keeping that in mind a new algorithm has been developed which defines the mean acceleration of the sensor in the 3D space in each second. This feature was deliberately given a name as “Activity Unit (AU)” and is defined by the international unit for measurement by m/s3. Regardless of the unconditional influence of gravitational force on each and every material, the AU is designed to take note only on the change of acceleration occurrence.

A physical jerk is accompanied to the system for the occurrence of rate of changes [], which can represent the derivative of acceleration w.r.t time. Despite of the consideration of the acceleration being zero, low or high it detects the changes regarded. Contrasting with the jerk, the Activity Unit declare the current level of acceleration which is being occurred. As an example, the jerk will give similar result as zero when the sensor is motion less even when the device is being transferred in a constant acceleration value. In other case the AU value is zero when it is motionless but output different value when the outcome is the case of the device being accelerated in a unchanged condition. We can define the activity unit value by the following equation-

Here the recorded variables are –

* x, y, z are the sensor values in m/s2
* xn, yn, zn are the values received by the sensor in each readout cycle n.
* xmean, ymean, zmean are the mean of the sensor values for each axis.

During the sensors are motionless the gravity is affecting the output of those sensors. After a certain amount of time the algorithm will be unable to detect any force because of the moving average. While the sensors are rotating, they will detect different readings of ‘g’ for each axis. According to previous research implementations, assuming the sensor reading as 32 Hz and sensor rotation by 90 degree, it is required that the force of acceleration on one axis, which had no gravity influence, will be able the acquire the gravity force of one tau, which is about 63.2%, within 2 seconds. This small amount period will provide 64 data triples converted as a usable window frame for activity recognition []. The requirement for time leads to an average of 0.95 as a value for ‘a’ using the following equation, where the provided average is analogous to the values of ymean and zmean.

Here the recorded variables are –

* X(n)= current sensor value
* a= absolute term
* xmean(n)= calculated average
* n=readout cycle
* xmean(n-1) = calculated average for (n-1)

The calculated estimation is being measured up by the consideration of using mean acceleration values for each axis. To determine an estimation for the user activity the summed-up values of AU can depict desired output. The summation of the activity unit values can roughly determine how active the user was throughout the day. The prior can also be useful for the determination of noise for the sensor values.

**3. System Architecture and Cloud Build Model**

Due to low computing power and limit power supply input from a battery around 200mAh the device itself is unable to handle the load of analysis and modelling the intended features of a smart wearable. By implementing the proper utilisation of modern IoT infrastructure we will be able to connect the device to live server and by uploading the data to the server we can analyse and classify user activity. Based on the classification, the system will generate recommendation based on the application of several machine learning model. We will discuss the various models and figure out which model will be the best according to the accuracy.

The cloud architecture is established based on a particular model architecture combining several components. Which is resulting creating a hybrid model for IoT and machine learning by connecting them through a web-based platform. By adding more functionalities, we can expand due to scalability.

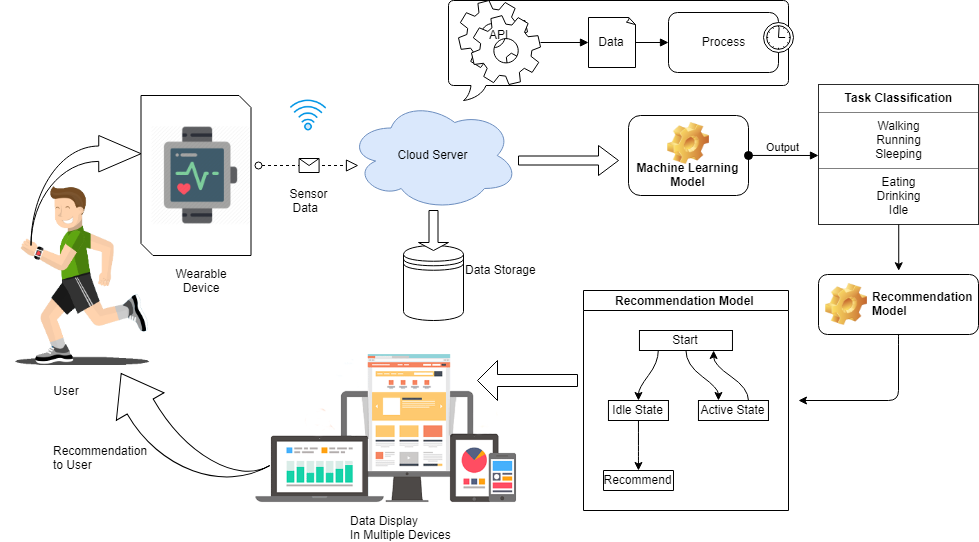
***3.1 Components -***

The whole architecture is can be divided into two parts based on the model architecture. The first is the IoT model which is the user interaction model and the other is the machine learning part, which is the server-side model.

***(a) Device (IoT):***The device which will be attached to the user as a wearable device will be consisting of a hub of sensors collecting data from the user on a time interval set as the server requirements. Due to lack of space or computational power the user data will transfer the data temporarily to smart-phone using Bluetooth. The smart-phone will be able to send the data live to the cloud whenever it is in the range of proper internet or Wi-Fi connection.

***(b) Web Acquisition (API):*** The data which is being transferred to the smart-phone via smart wearable device, can be uploaded to the server via web APIs. Web API services are fast models for accessing and transferring data in a JSON format.

***(c) Storage (DB):*** Uploaded data through API will be stored in a Database. The database can be of two types- SQL or NoSQL. For our system simple MySQL database is implemented to reduce the cost for storage. As NoSQL architectural infrastructure demands more cost for implementation.



***Fig 1: System Architecture***

***3.2 Machine Learning Model Acquisition***-

The activity recognition task and the process model for recommendation will be handled by the backend infrastructure implemented on **Cloud**. So, the cloud is the major component for our system architecture. The process related to machine learning requires heavy computational power, which is unable to implement in a small device like smart-watch and can lead to unbearable expense in development. As well as a result, it will increase the product cost in such a way that customers can lose interest on buying such a product. Cause it is evident that the user always intends to buy an upgraded version. So, if an individual is investing a huge amount of money in a particular product then the person will regret when the next product comes to the market.

Keeping that in mind this architecture is most acceptable one, by uploading the data to cloud. Which not only reduce the cost but also opens up a huge possibility to play with the data. Cloud can provide scalability and security to the data which is difficult to implement in a smart watch.

Activity classification task involves mapping time series data into a single activity collected from both smart-watch and smart-phone. By aggregating the data based on 10 seconds interval, activity is recognised correctly when activity is classified in correct section. Based on previous experiments the activities are grouped into three general categories [] –

TABLE 1. ACTIVITY GROUPING

|  |
| --- |
| 1. Activities not performed by hand   * Walking * Jogging * Running * Standing * Kicking ball * Climbing stairs |
| 2. Activities performed by hand   * Eating * Drinking * Typing * Writing * Playing Computer games * Clapping |
| 3.Activities involved both hand and other parts   * Sleeping * Swimming * Playing * Exercising * Driving |

Several activity recognitions were induced from WEKA [13] labelled training examples for Random Forest (RF) algorithm, Naïve Bayes (NB) algorithm, IB3 instance-based learning algorithm, multi-layered perceptron (MLP) algorithm and j48 Decision Tree algorithm. Mostly the default parameters for the induced algorithms were implemented in the settings, excluding for Naïve Bayes where kernel estimation was enabled, and IB3 algorithm where three nearest neighbours were used. (There are multiple references available for algorithms which are induces, for further details please refer to online documentation referenced in below sections.)

Based on the activities two models are intended to be utilized, models are partitioned into two separate models. The prior one is the personal model where the other one is the impersonal model. It has been found in the previous research that the personal model hugely outperforms the impersonal model [9]. The results were generated based on the collected data from 17 subjects and each test sets were included with 10-cross fold validation. So overall 17 × 10 = 170 models were induced [].

TABLE 2. OVERALL ACCURACY FOR PERSONAL MODEL

|  |  |  |  |
| --- | --- | --- | --- |
| **Algorithm** | **Phone Accel(%)** | **Watch Accel(%)** | **Watch Gyro(%)** |
| RF  NB  IB3  MLP  J48 | 75.3  65.7  67.5  77.3  75.6 | 93.5  86.5  94.3  93.7  95.1 | 79.1  72.9  59.9  80.5  71.0 |
| **AVG** | 72.28 | 92.62 | 72.68 |

TABLE 2. OVERALL ACCURACY FOR IMPERSONAL MODEL

|  |  |  |  |
| --- | --- | --- | --- |
| **Algorithm** | **Phone Accel(%)** | **Watch Accel(%)** | **Watch Gyro(%)** |
| RF  NB  IB3  MLP  J48 | 35.2  24.3  22.1  25.9  20.1 | 70.1  60.1  62.3  63.5  64.2 | 57.3  49.7  50.1  52.9  57.5 |
| **AVG** | 25.52 | 64.04 | 53.5 |

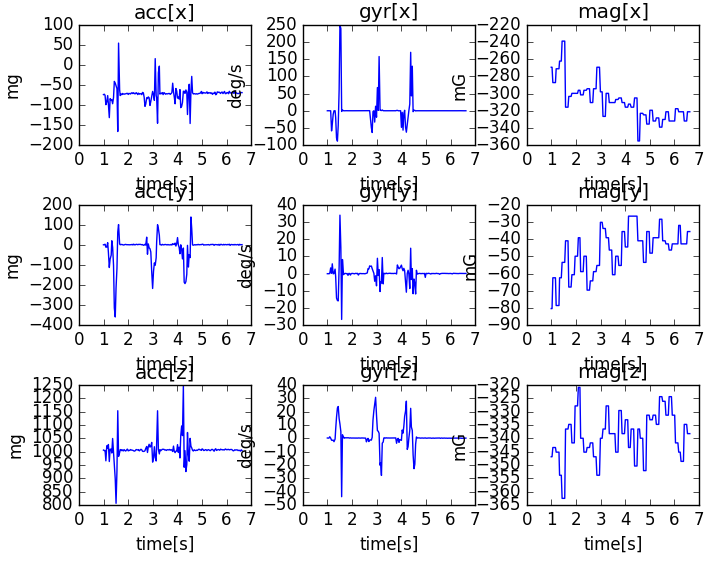
**4. OVERALL EXPERIMENT OVERVIEW**

We are about to discuss the overall process model for data collection and extraction for the full model. The initial process to the end computing the task can be divided into several parts. Based on the activity and analysis –

***4.1 Data Collection –*** Collecting the data from the user is mainly involving the hardware requirements for the system. The smart-watch and smart-phone consisting of multiple sensor hub like accelerometer, gyroscope and gravity sensors also. We were able to collect sample data from the test subject by using third-party applications. From device IoT front we have developed a device which is consisting of an accelerometer, gyroscope sensor and a heart-rate sensor. The heart rate sensor was included as a future enhancement possibility my implementing functionalities based on medical aspects. The Arduino sensors are able to capture the data based on the user activity, which can be collected to the database using API services. The services are platform independent and does not also rely on any particular language. Provides faster, reliable and secure data transfer among the servers.

The useful accessories for tracking and recording for the data collection using Arduino is available on the market based on separate price range and functionalities. By certain research we have found out that two popular model exists for recording Accelerometer and Gyroscope data using Arduino – (1) MPU6050, (2) MPU9250. Now by comparison it’s been depicted that the MPU9250 provides more accurate data from a user.

The data was collected from several users by attaching the module into user body. The accelerometer and gyroscope data are captured on the basis of their axis- x, y & z. The data was able to send through COM port and also by utilizing the functionalities of PyQtgraph and PyQt5, we were able to visualise the data as a live aspect. In Fig -3 we have displayed a sample of such data plotting using python matplot library module.



***Fig 3 – Matplotlib plotting of***

***Accelerometer and Gyroscope data***

***4.2 Dataset-*** Due to lack of user test subjects and available environment for recording huge dataset here we have used existing datasets open source repositories like ‘Kaggle’, for training and testing the machine-learning models. The first dataset which we used was taken from the UCI machine learning repository. The data was recorded from 30 test subjects by average age between from age 19 to age 48 years. During the data collection the users performed six different tasks like-

1. Walking

2. Laying

3. Sitting

4. Standing

5. Walking Upstairs

6. Walking Downstairs

And the data was labelled manually. The data was recorded based on the linear acceleration and angular velocity of accelerometer and gyroscope data in x, y and z axis, captured using Samsung Galaxy S II [14].

***4.3 Data Modification -***

The dataset was split into two separate parts based on subjects where 70% is the train dataset that is the 21 subjects and another is the 30% of test dataset consisting of 9 user record [15] [16].

***4.4 Heart Rate sensor Data -***

5. EXPERIMENTAL RESULTS

Based on the datasets and model acquisitions, we will compute the results and discuss the outcomes we preferably choose to implement on the cloud.

The training dataset loads the data as total acceleration on the x, y and z axis and returns a NumPy array, and prints the shape of the array. The function can be demonstrated by loading all the total acceleration files. The training data comprises with 7352 rows with each window concluding 128 observations. After loading each file as a NumPy array we can easily stitch all three files together. It can be ensured that each and every file is stacked in a way the features a partitioned in another dimension, using dstack() NumPy Function.

Running the sample outputs the shape sand returns NumPy array displaying the sampling and time steps for three consecutive features – x, y and z for the dataset. Its been depicted that the size of windows in test and test datasets matches with the size of output of y in every train and test case replicates the sample number.

A balanced dataset is always easier to model. So to confirm if the dataset is actually reaching the expectations or not, we are using a function *class\_breakdown()* to implement this characteristics. The function first wraps the given NumPy array, then

groups out the class value, and evaluate the size of the every group based on number of rows. The result is being summarised in the following table-

**Train Dataset-**

|  |  |  |
| --- | --- | --- |
| Class | Total | Percentage(%) |
| Class-1 | 1226 | 16.676 |
| Class-2 | 1073 | 14.595 |
| Class-3 | 986 | 13.414 |
| Class-4 | 1286 | 17.491 |
| Class-5 | 1374 | 18.89 |
| Class-6 | 1407 | 19.137 |

**Test Dataset-**

|  |  |  |
| --- | --- | --- |
| Class | Total | Percentage(%) |
| **Class-1** | 1226 | 16.676 |
| **Class-2** | 1073 | 14.595 |
| **Class-3** | 986 | 13.414 |
| **Class-4** | 1286 | 17.491 |
| **Class-5** | 1374 | 18.89 |
| **Class-6** | 1407 | 19.137 |

**Both-**

|  |  |  |
| --- | --- | --- |
| Class | Total | Percentage(%) |
| Class-1 | 1226 | 16.676 |
| Class-2 | 1073 | 14.595 |
| Class-3 | 986 | 13.414 |
| Class-4 | 1286 | 17.491 |
| Class-5 | 1374 | 18.89 |
| Class-6 | 1407 | 19.137 |

REFRENCES

[9] I. H. Witten, and E. Frank - Data Mining: *Practical Machine Learning Tools and Techniques,* 2nd ed., Morgan Kaufmann, 2005.

[13] I. H. Witten, and E. Frank - Data Mining: *Practical Machine Learning Tools and Techniques,* 2nd ed., Morgan Kaufmann, 2005.

[14] Anguita, Davide, Ghio, Alessandro Oneto, Luca Parra Perez, Xavier Reyes Ortiz, Jorge Luis - *A public domain dataset for human activity recognition using smartphones, ISBN978-2-87419-081-0, 2013*

[15] *Human Activity Recognition on Smartphones Using a Multiclass Hardware-Friendly Support Vector Machine* –by Davide AnguitaAlessandro GhioLuca OnetoXavier ParraJorge L. Reyes-Ortiz, IWAAL 2012: Ambient Assisted Living and Home Care pp 216-223

[16] Dataset- Human Activity Recognition with Smartphones Recordings of 30 study participants performing activities of daily living. *https://www.kaggle.com/uciml/human-activity-recognition-with-smartphones*