WEATHER SENSING USING SMART PHONE SENSORS

A Term Paper Final Report

Submitted in partial fulfilment of the

Requirements for the award of the Degree of

Bachelor of Technology

In

Computer Science & Engineering

By

BATCH: 12-CC-04

K.Anusha (12003030) **T.Sudheer (12003038)**

Under the Guidance of

Dr. B. THIRUMALA RAO

(Professor)

CLOUD COMPUTING RESEARCH GROUP

KL UNIVERSITY

Vaddeswaram, Guntur

KL UNIVERSITY

Department of CSE

Declaration

The Term Paper Report entitled "**WEATHER SENSING USING SMART PHONE SENSORS**" is a record of bonafide work of K. ANUSHA (12003030), T.SUDHEER (12003038) submitted in partial fulfilment for the award of B. Tech in Computer Science and Engineering to the K L University. The results embodied in this report have not been copied from any other departments/University/Institute.

K.ANUSHA (12003030)
T. SUDHEER (12003038)

KL UNIVERSITY

Department of CSE

Certificate

This is to certify that the Term Paper Report entitled "**WEATHER SENSING USING SMART PHONE SENSORS**" is being submitted by K.ANUSHA (12003030), T.SUDHEER (12003038)**,** submitted in partial fulfilment for the award of B. Tech in Computer Science and Engineering to the K L University is a record of bonafide work carried out under our guidance and supervision.

The results embodied in this report have not been copied from any other departments/University/Institute.

Under the Guidance of	Head of the Research Group
Dr.B.Thirumala Rao	Dr.B.Thirumala Rao
(Professor)	(Professor)

ACKNOWLEDGEMENTS

The satisfaction that accompanies that the successful completion of any task would be incomplete without the mention of people whose ceaseless cooperation made it possible, whose constant guidance and encouragement crown all efforts with success.

We are grateful to our guide **Dr.B.Thirumala Rao**, Professor, Computer Science & Engineering for the guidance, inspiration and constructive suggestions that helped us in the completion of this project.

At the outset we thank our Head of the Research group **Dr.B.Thirumala Rao,** Professor for the moral support and the excellent facilities provided. I would also like to thank all the teaching and non-teaching staff members of Computer Science department who have extended their full cooperation during course of my term paper.

At the outset we thank our Head of the Department **Dr.V.Srikanth,** Professor for the moral support and the excellent facilities provided. I would also like to thank all the teaching and non-teaching staff members of Computer Science department who have extended their full cooperation during course of my term paper.

We wish to express our warm and grateful thanks to our project coordinator for the guidance and assistance he provided in completing our project successfully.

We thank all our friends who helped us for sharing knowledge and by providing material to complete the term paper in time

K.ANUSHA (12003030)

T.SUDHEER (12003038)

TABLE OF CONTENTS

1. INTRODUCTION

1.1 Topic

This paper titled as “Weather Sensing Using Smart phone Sensors” describes using the smart phones to detect the climatic conditions of the surrounding the smart phone is in which can be done with the help of the built-in sensors. As we all know that the usage of internet has increased from 52.8% in 2012 to about 83% in 2015. With that the number of smart phone users is increasing drastically as it provides more feature than a mobile phone. The advanced feature of the smart phones is that they are equipped with different kinds of sensors increases the capabilities of computer to interact with the environment. Earlier and presently there are different kinds of methodology accepted to detect the climatic conditions such as weather stations and meteorological satellites.

In this paper, we trying to bring up a ever new technique that can be used to predict the future climatic conditions by every individual which can help them in changing plans which can be at risk due to change in the climate. As the usage of smart phones in increasing day-to-day why not use those as our weather prediction stations to take decisions. Our smart phones are a way more smart than we expect they are equipped with different kinds of sensors that helps them to interact with the environment directly. There are many applications in the app store available which are used for weather detections. . These values can’t be directly read from a sensor to a mobile phone but rather we need a mobile application which does acts as an interface between the sensor and the user. This application will not only enable to detect the current temperature but also help in predicting the weather in the near future. The smart phones are equipped with many built-in sensors some of them are accelerometer, gyroscope, magnetometer, barometer, etc.

1.2 Existing System

These applications collect the data from the weather stations or from the meteorological satellites. The areas where the weather stations are set up, the climatic conditions are collected from the equipment installed at the stations. The values are collected and are used in the application. In the places where the weather stations are not present the climatic conditions of those regions is collected from the meteorological satellites. In India the weather stations are located at different places as in Lucknow, New Delhi, Guwahati, Ahmadabad, Kolkata and in South India they are at Bangalore, Mangalore, Coimbatore etc.

There are different satellites which serves the purpose of weather conditions collection, some of the recently launched satellites are MSG 4, FENGYUN 2G, etc. As mentioned above that weather stations and meteorological satellites can be used for collecting the weather conditions why not the smart phones which have become a part of daily lives for most of the individuals be used as a device for weather data collection.

1.3 Proposed System

In near future mostly all the individuals will be having a smart phone with them, thus it can be used to get efficient values for the climatic change for particular location through the sensor available at that location directly. To complete the process of data collection from different places in the country or around the world the smart phones can be used as references that are the devices which are used for data collection. These smart phones can't be directly used as a device of weather data collection but the sensors equipped inside them are capable of fulfilling this functionality. We are all unknown to the fact that "the smart phones are even smarter than we think they are" as they have additional capabilities and characteristics which makes it different from the mobile phones. These smart phones all should be connected to the network through mobile data usage which has an increasing demand in the market as the users of smart phones are drastically increasing. The smart phones sensors can be accessed indirectly through the smart phones network. Once the device is connected to the network the values collected from the sensors can be transferred to the cloud storage connected to the application developed for the purpose of collecting data, after the data is collected not directly it can be used for the making decisions rather they have to be cleaned from different kinds of inconsistency to make it qualified to make decision by applying different kinds of algorithms available for decision-making. Clean data now available in the cloud can be used to make predictions in the near future which can be done by implementing Map-Reduce techniques. The usage of mobile internet is growing day by day. This project implementation will help us to collect data from different locations and store them in the cloud and processing the unnecessary data that gets generated along with the sensor data.

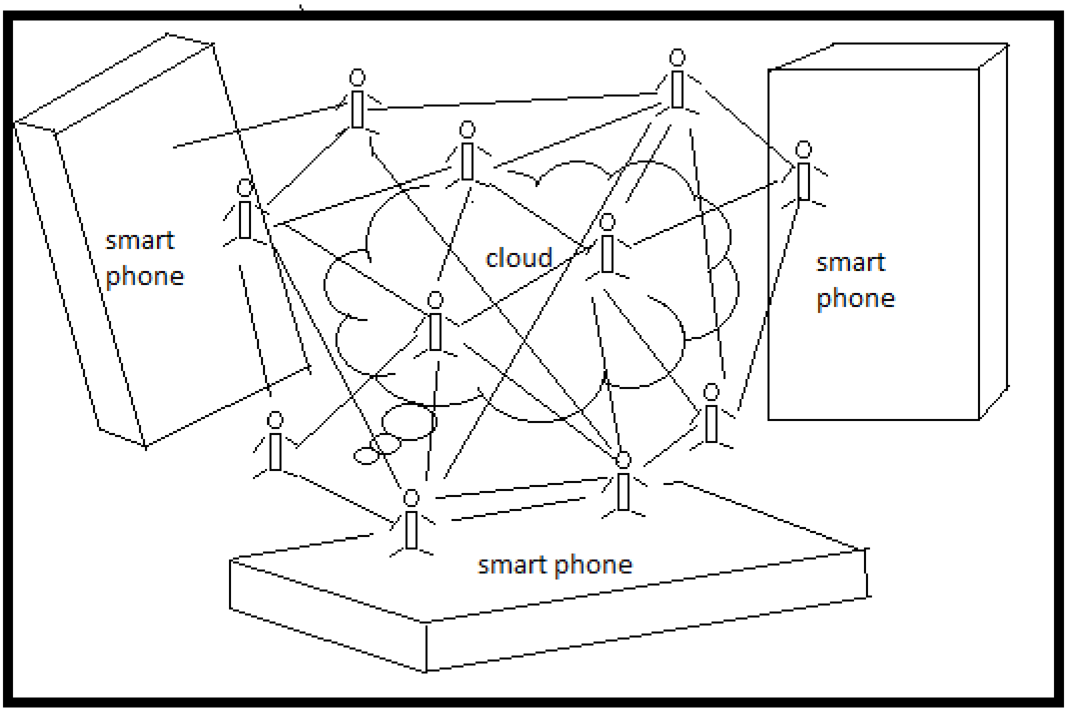

Fig 1.3.1 Using Smart phones as collecting Devices

1.4 Climate Analytics in Cloud Computing

Cloud Computing is a practice of using a network of remote servers hosted on internet to manage, process and to store data instead of a local server or a personal computer. Cloud computing provides services such as PaaS, SaaS, IaaS to the customers for their flexibilities. **PaaS** (Platform as a service) model, a cloud provider like (amazon, aneka) delivers software tools and hardware to the required costumers for their application development and customization. **SaaS** (Software as a Service) is a software distribution model where vendor provides the applications and it the customers who make use of these services. . **IaaS** (Infrastructure as a Service) model these are monitoring, accessing, and managing remote datacenter infrastructures, such as storage and networking services like firewall.

Climate changes play a key role in the environment and its adverse effect on humans. Although these climatic changes are anthropogenic (Human caused). The key factors like humidity, pressure, evaporation, precipitation are responsible for climatic changes. These vary from time to time, day to day. So a large sets of metrological data is required to predict the future changes. These changes in climate and prediction of future climatic changes are considered to be climate science. Climate science is a Big Data Domain which is experiencing enormous growth. NASA climate change repositories alone produce 500 petabytes by 2014. It is obvious that for climatic predictions we require a large amount of data sets which require huge amount of storage area, high computing devices, and servers. So in order to meet these requirements we go for cloud computing.

CAaaS: Climate Analytics as a Service

To meet the Big Data challenges of climate sciences we are moving to a notation of Climate Analytics as a Service. We focus on analytics, because it is the knowledge gained from our interactions with Big Data that ultimately produce societal benefits. When coming to CaaaS which is an extension of PaaS, SaaS, IaaS enabled by cloud computing. CaaaS serves as computation interface for predicting the future climatic changes by using metrological data. A large amount of analytics is required to be done on the scientific data and need to store for future references. For all these computations, storage, hardware and software we need a large infrastructure and maintenance. So, CaaaS serves as a generative technology to meet required needs.

1.5 Additional Information

1.5.1 Accelerometer

These sensors sense the acceleration event of the smart phones. The reading includes the axes whose directions are already defined. The raw data from the sensor is the acceleration of each axis in the units of gravitational force. Timestamp can also be returned together with the three axes readings. The raw data is represented in a set of vectors. $Acc_i = < x_i, y_i, z_i >$, (i =1,2,3,…..)

Accelerometer sensor has been heavily used in smart phones for the purpose of activity recognition.

1.5.2 Compass Sensor

Compass is a traditional tool which is used to detect the direction. In olden days the sailors used the compass to know the direction in which they have to sail to reach the destination. It gives the directions with respect to the north-south pole of the earth. The compass sensor in the smart phones also works with the same functionality. The data reading ranges between $0^0 – 360^0$, where 0^0 is absolute north and the actual reading indicates the angle between current smart phone heading direction and the absolute north in clockwise direction.

1.5.3 Barometer sensor

It is one the latest sensors [2] equipped in some advanced smart phones such as Samsung Galaxy S4, Google Nexus 4/10. This sensor is used to measure the atmospheric pressure of the environment in which the sensor is placed. This sensor can also be used for measuring the altitude.

1.5.4 Ambient Light sensor

In general both the Ambient Light Sensor and Proximity Detection product are a single device. The ALS rounds up the response of human eye to the light intensity and depending on it it adjusts the screen brightness and is done using a variety of attenuation materials.

1.5.5 Gyroscope sensor

It is a sensor that is in the shape of a disc and provides information about the orientation of the smart phone, because of that in the camera application we can find the angle of rotation being mentioned for a better click of the picture. It is also used to tell what constellation you are pointing your phone.

1.5.6 Weather stations in India

S. No	Station Location
1	Bangalore
2	Belgaum (Sambra)
3	Imphal
4	Indore
5	Coimbatore
6	Nagpur
7	Jaipur (Sanganer)
8	Lucknow

Table 1.5.6.1: Some Stations in India

The above table shows some of the places were weather stations are established. These stations have the highest elevation in descending order of the table. There are many more in our country and also around the world.

1.5.7 Meteorological Satellites

S. No	Satellite	Launch Date
1	ASTROCAT	28th September,2015
2	GSAT-6	27th August,2015
3	IRNSS-1D	28th March,2015
4	GSAT-16	7th December,2014
5	IRNSS-1C	10th November,2014
6	IRNSS-1B	4th April,2014
7	GSAT-14	5th January,2014

Table 1.5.7.1: Some of the recently launched satellites.

2. LITERATURE SURVEY

The main technique used for completion of this project Activity Recognition Process plays a major role. The smart phone sensors rapidly produces values all the time but these values are to be captured for the purpose of usage. Once the data is collected that can be used for the activity recognition process. It is a process involving different stages, the data passes stage by stage and is utilised for Activity Recognition. This process has already been used in different fields such as Health Monitoring System, Indoor positioning, etc. In case of Health Monitoring System the sensors which can be used to determine the health condition of a person are utilised for the purpose of analysis. The system id used to monitor wellbeing of high risk cardiac patients. The smart phone works as an ECG data collector and determines whether the person needs any king of support from the doctor externally. It monitors the conditions of the person and can automatically alert the caregivers or call or send a immediate message to the ambulance as an emergency. This was developed by analysing the statistics of cardiovascular disease patients in the Unites States. It was observed that a lot of investment was made in the hospitals by the cardiovascular patients for their treatment as many people suffer from it may be because of their eating habits. This system was built to reduce the cost and anxiety of the people who are well-known to their disease, they can be helped by monitoring them 24X7 and notify them in case of emergencies. For this to be in process the smart phone should be on the patient's body so that he/she can be monitored. The smart phone sensors are used for the purpose of activity recognition, they are used to detect small activities performed by the human beings such as sitting or standing or raising both hands up. Some of the other activities are also observed such as walking, jogging, etc. The smart phones are equipped with biosensors that records the heart signals and transmit it to the centre of monitoring say hospital this data can be examined by the doctor and suggestions or any advice can be given to the patient through a call or a message which can help them prevent an attack.

The other system in which this has been used is Indoor Positioning. . For example, you are standing outside a 10 story building and you want to know the height of each floor. The height of a floor can be practically measured by a measuring tape being in that floor which is to be measured but to determine its height not being in the building becomes challenging. It can be done with the help of barometer sensors by placing them at each floor which will act as reference for data collection. The sensors produce values frequently these have to be captured at regular intervals. These value measured by the sensors place at 2 consecutive floors are compared which results in the height of the lower floor. By repeating the same procedure for the all the pairs of floors height of all the floors can be determined.

To perform any kind of processing on the data collected from the sensors called as the raw sensor data, the data should be cleaned from different kinds of inconsistencies obtained due to different kinds of disturbances. After cleaning the data it has to be transformed in the way the data is required to derive any kind of knowledge from the data being collected for experimenting. This data transformed is classified so that it can be understood better by the person analysing the data to derive different conclusions out of the data. This data now becomes ready to perform the processing. The data is used for activity recognition process that is the data tells about what kind of activity is being done by the sensor, and also about what is to be obtained. The complete process of activity recognition is explained in the next chapter in detail.

2.1 KEY TERMS

Indoor Positioning: An indoor situating framework (IPS) is a framework to find protests or individuals inside a building utilizing radio waves, attractive fields, acoustic signs, or other tactile data gathered by cell phones.

Cloud Computing: Cloud Computing is a practice of using a network of remote servers hosted on internet to manage, process and to store data instead of a local server or a personal computer.

Data Pre-processing: Data pre-processing is an data mining system that includes changing crude information into a justifiable arrangement. Certifiable information is regularly fragmented, conflicting, and/or ailing in specific practices or drifts, and is prone to contain numerous mistakes.

Classification: Classification refers to a supervised learning i.e., when we know the class labels beforehand. While doing the classification we have two kinds of data sets namely learning sets and training set, the former is the data to be classified and the latter is the data to which all the rules required are thought through the process of training and then the class labels are given to the data.

S. No	Paper Title	Problem Solved	Problems yet to be solved
1.	Activity Recognition with Smart phone Sensors, Tsinghua Science and Technology.	The sensors are used as a device for collecting the data and processed for the purpose of activity recognition.	The major challenges that the activity recognition process faces is subject sensitivity, location sensitivity, activity complexity, energy and resource constraints and insufficient training set.
2.	A Health Monitoring System Using Smart Phones and Wearable Sensors, Broadway, Australia.	This paper describes a method of monitoring health conditions of the person regularly. If required it sends advices and also in case of emergency informs the services.	In case of major incidents during the monitoring phase, the values are only recorded but no immediate action is taken.
3.	Using Multiple Barometers to Detect the Floor Location of Smart Phones with Built-in Barometric Sensors for Indoor Positioning.	The barometer sensors can be used to determine the heights of a floor standing outside the building and floor number can also be determined based on the average floor height.	If any kind of disturbances occur to the reference then the values received are noisy and cannot be used to draw conclusions.
4.	The Benefits of Personalized Smartphone-Based Activity Recognition Models, Jeffrey W. Lockhart.	Applications are used to monitor the human activities time to time, the data is collected, transformed and then inducted and experimented for activity recognition task.	The major drawback of this system is that only a single body location can be instrumented.

3. THEORETICAL ANALYSIS

3.1 Activity Recognition Process

Activity recognition [1] is a process of recognizing the goals of the users by making observations on their actions and the environmental conditions. It is an important technology in the area of pervasive computing. This is also used for recognizing simple human activities. For example, a person raising both hands or sitting or standing [3]. These are some of the cases considered by the medical organizations to help the elderly man or woman. The sensors in the smart phones keeps a track of the condition of the person and the information collected through the sensors can be used to send emergency messages to the person. It has a number of applications and is used in many fields of study such as medicine, human interaction and sociology.

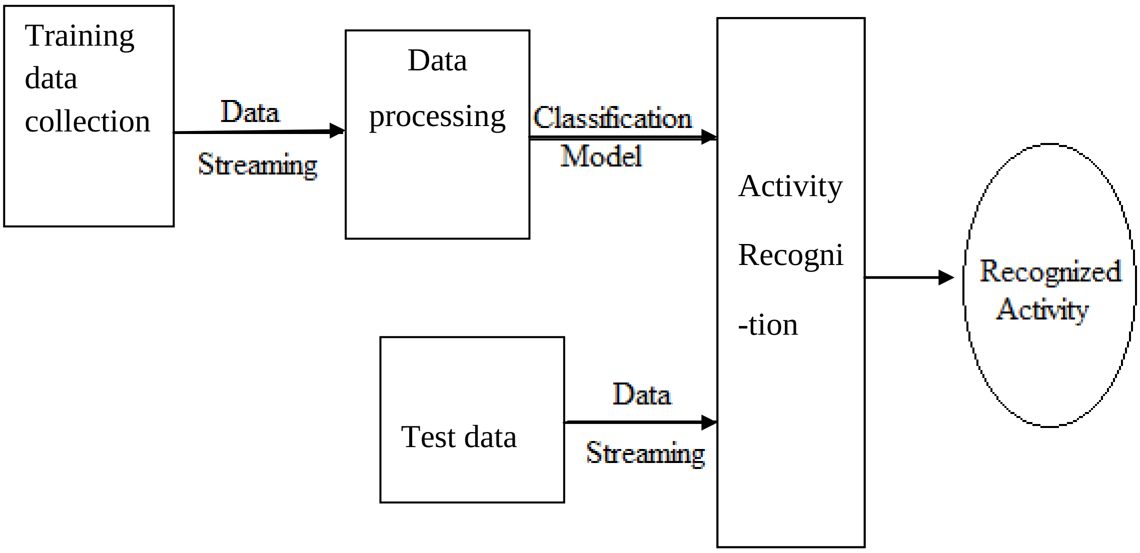

Fig 4.1.1: Activity Recognition Process

The number of users for the smart phones is increasing because of its additional features like interacting with the environment as they are equipped with the built-in sensors. The smart phones have become a major source for mobile internet usage; hence they are used as a main platform of activity recognition in this age of developing technology. Activity recognition process is carried out in number of phases. They are namely:

1. Raw data collection
2. Data pre-processing

3. Feature computation
4. Model training
5. Classification

3.1.1 Raw Data Collection

The term “raw data” refers to the data that is collected directly from different sources used during the process. As per our requirements the source considered is sensors i.e. the built in sensors present in the smart phones. This data is also called as “primary data”. The data is just collected but not subjected to any kind of processing or manipulations [1]. In this stage the major task involved is collecting of data from the sensors as sensors are the source we are considering. In this project the data from the sensors will be collected through the mobile application installed on the smart phone.

The Android sensor framework provides several methods that make it easy for you to determine at runtime which sensors are on a device. The API also provides methods that let you determine the capabilities of each sensor, such as its maximum range, its resolution, and its power requirements. To sense the weather we can use different sensors more specifically barometer sensor which can be used for floor positioning. Previously indoor positioning methods focused on plane locations on a floor and did not provide accurate floor positioning. The values from sensors can’t be directly read from a sensor to a mobile phone but rather we need a mobile application which does acts as an interface between the sensor and the user.

ODK (Open Data Kit) [5] sensors framework reduces the complexity of building sensor-based mobile applications which is to be used for data collection. The sensors API does not automatically store all the values/data produced by the sensors, but we need to use a recording API for that purpose. API is used as follows:

- Lists data sources available.
- Registers listeners to receive raw sensor data.
- Unregisters listeners to stop receiving.

The Android sensor framework [5] provides a package which helps in the interaction with the sensors through the mobile app. A package namely android.hardware package exists which provides with three different classes which helps in interacting with the sensors.

Sensor Manager Class: An instance to this class is used for listing sensors, registering and unregistering sensor event listeners, and acquiring orientation information.

Sensor Class: This class is available in the above mentioned package and is used to create an instance of a specific sensor.

Sensor Event Class: This class is used to create a sensor event object, which provides the information about a sensor event. A sensor event object provides the following information:

- Raw sensor data
- The type of the sensor
- Accuracy of the data
- Timestamp for the event.

This class also provides with an interface called SensorEventListener which provides two call-back methods to monitor raw sensor data, onAccuracyChanged() which gives a reference to the sensor object changed and the new accuracy of that particular sensor and the other method is onSensorChanged() gives a Sensorevent object which contains information about the new sensor data, including its accuracy, the data generated, the timestamp at which it was generated, and the new data of the sensor recorded. The methods used are as follows:

private SensorManager SensorManagername; // *creation of sensor manager object*
private Sensor sensorname; // *creation of a sensor object*

getSystemService(Context.SENSOR_SERVICE); // get service from the sensor to the app
getDefaultSensor(type of the sensor); // returns the default sensor of specified type

registerListener(event object, sensorname,SensorManager.delay); // used to register the listener:

unregisterListener(event object); // used to unregister the listener:

onSensorChanged(SensorEvent event); // is used to take values and store in the local storage

3.1.2 Data Pre-Processing: De-Noising And Segmnetation

Once the data is collected it cannot be directly used for any kind of analysis as it contains many inconsistencies in the data such as missing values, noisy values which on analysis will lead to wrong results. The noisy data refers to meaningless data in some context also termed as "corrupt data". Thus some techniques are applied in order to remove the noisy data as much as possible for getting efficient results on processing. To remove the noisy data, the techniques that are mostly used are binning, band-pass filter and low-pass filter. In binning the data collected is fragmented into bins of equal lengths and in that the noisy values are filled in three ways mostly. Let us discuss them in short.

Binning:

1. Filling all the values with the mean of the values of the bin.
2. Filling all the values with the lower boundary of the bin.
3. Filling all the values with the upper boundary of the bin.

Sorted data for price (in dollars): 4,8,9,15,21,21,24,25,26,28,29,34
* Partition into equal-frequency (equi-depth) bins:
  - Bin 1: 4,8,9,15
  - Bin 2: 21,21,24,25
  - Bin 3: 26,28,29,34
* Smoothing by bin means:
  - Bin 1: 9,9,9,9
  - Bin 2: 23,23,23,23
  - Bin 3: 29,29,29,29
* Smoothing by bin boundaries:
  - Bin 1: 4,4,4,15
  - Bin 2: 21,21,25,25
  - Bin 3: 26,26,26,34

Fig 3.1.2.1: Example for Binning technique

Band-pass filter:

It is used to eliminate the low frequency such as acceleration due to gravity. Here frequency refers to the orientation of the sensor with respect to ground data, and high frequency signal components.

Low-pass filter:

It eliminates the noises generated by the dynamic human motion and preserve low-frequency components.

3.1.3 Feature Computation

In data mining tasks feature extraction is critical for the final recognition performance. For activity recognition, we can extract features both in time and frequency domains.

Time-domain features

The time domain contain the basic statistics of each data segment and those of different segments i.e. finding the values in each segment in each dimension such as,

- Mean
- Max, Min
- Standard deviation
- Correlation
- Single Magnitude Area.

Frequency domain features

Frequency domain features describe the periodicity of the signal, which can be typically calculated based on the Fast Fourier Transform. They are,

- Energy
- Entropy
- Time between peak
- Binned distribution.

3.1.4 Classification

Classification refers to a supervised learning i.e., when we know the class labels beforehand. While doing the classification we have two kinds of data sets namely learning sets and training set, the former is the data to be classified and the latter is the data to which all the rules required are thought through the process of training and then the class labels are given to the data. To perform this task many algorithms exist in the market. It is a form of data analysis which is done to understand the data better when it is large. There are different algorithms such as KNN, J48 and C 4.5 [1], the algorithm we will be implementing in the mobile application is KNN.

Popular classifiers are used in the area of activity recognition. They are base-level classifiers and meta- level classifiers. The base level classifiers are widely used in the activity recognition process such as decision tree and decision table. Decision tree is a kind of hierarchical implementation which has low complexity in implementation. It has excellent interpretation for the 2-staged activity recognition process. A decision tree is a flowchart-like structure in which each interior hub speaks to a "test" on a property (e.g. whether a coin flip comes up heads or tails), every branch speaks to the result of the test and every leaf hub speaks to a class mark (choice taken in the wake of registering all characteristics). The ways from root to leaf speaks to grouping tenets.

In choice investigation a decision tree and the firmly related impact chart are utilized as a visual and diagnostic choice bolster apparatus, where the normal values (or expected utility) of contending choices are figured. The major disadvantage with decision tree is that it is costly to update the model. The decision table is an unlabelled example, it is matched exactly in the table of rules and classes. To build a decision table draw a line for every jolt that will be utilized as a part of the procedure of settling on a choice. Next draw a segment for every conceivable mix of results of those conditions. In the event that there are n conditions and each has a double result, then there will be 2n sections. Keep going, include columns the base of the table for every reaction you need the framework to perform. The data to be trained is compared with the existing class labels and check to which class label it can be classified. If the value to be trained matches the class label it reports the matching class label and it doesn’t respond when no matching class is found.

3.1.5 K-Nearest Neighbour Algorithm:

The k-NN algorithm [4] has been used in many applications for the purpose of pattern recognition, image processing, data mining etc. In this classification the objects are the class members and object is classified into classes based on its K Nearest Neighbours. The K value is a trained set values and usually K is chosen as an odd value to overcome the problem of deadlock, which may occur when the values cannot get classified. The algorithm is we consider each of the characteristics in our training set as a different dimension in some space, and take the value an observation has for this characteristic to be its coordinate in that dimension, so getting a set of points in space.

A training set is a known classification of observation as it trains us what objects of the different objects look alike. We can then consider the similarity of two points to be the distance between them in this space under some appropriate metric.

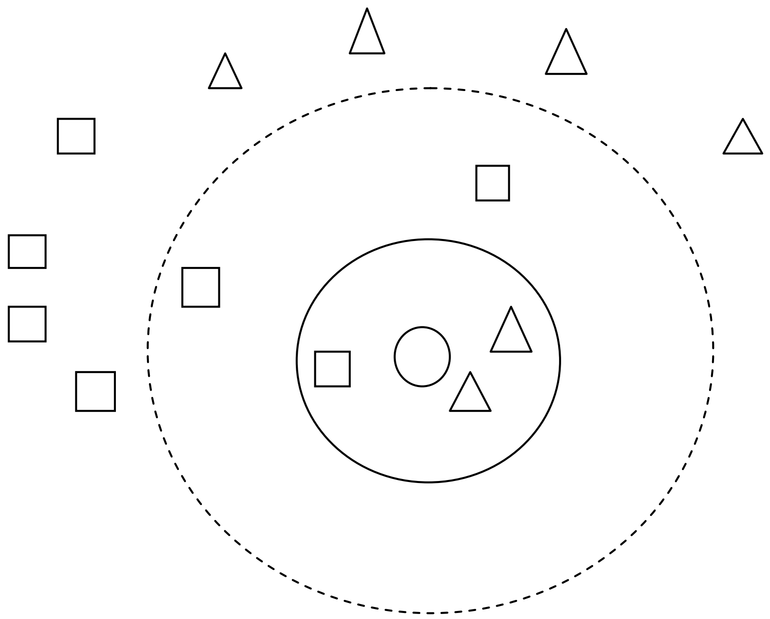

Fig: K nearest neighbours

The way in which the algorithm decides which of the points from the training set are similar enough to be considered while choosing the class to predict for a new observation is to pick the k closest data points from the new observation, and select the most common class among available classes . As this algorithm is based on the k closest data points it is termed as “k Nearest Neighbour”.

KNN algorithm is a kind of lazy learning because no model is learned from the training data. One of the best examples is text classification. The algorithm that KNN follows is as given below:

D- Training dataset d- Test instance

3.1.5.1 Algorithm Steps:

- Compare d with every training example in D to compute distance or similarity between them.
- ‘k’ most similar (closest) examples in D are selected.
- These sets are called as ‘k-nearest’ neighbours of‘d’.
- Then‘d’ takes the most frequent class among the neighbours.
- k=1, is not considered because it contains noise and outliers.

3.1.5.2 Advantages:

- Very flexible
- Can work with any arbitrarily shaped decision boundaries.

3.1.5.3 Disadvantages:

- Time consuming
- Does not produce an understandable model.

4. EXPERIMENTAL ANALYSIS

The Android sensor framework provides several methods that make it easy for you to determine at runtime which sensors are on a device. The API also provides methods that let you determine the capabilities of each sensor, such as its maximum range, its resolution, and its power requirements. The values from sensors can't be directly read from a sensor to a mobile phone but rather we need a mobile application which does acts as an interface between the sensor and the user.

4.1 Android Studio Tool

It is an official studio for android application development based on Intel i J IDEA. The Android Studio is the preferred method because it can directly invoke the tools that we need while developing applications. Through this studio first we will establishing the environmental setup by installation and then creating project setup and development which contains the source code and resource file for our application. The project is build into a debuggable.apk packages that u can install and run on the simulator. In the final stage the project is configured and application is built for the release and distributed among users.

The weather app that is developed in this studio should be taking the values from the inbuilt sensors in the smart phones. The application is communication the sensors through a sensor framework included in the hardware package. The sensor framework provides the application to get accessed with the sensors and collect the raw data. The Android Studio Tool has many features. They are as follows:

1. Translations Editor: Multi-dialect backing is upgraded with the Translations Editor plug-in so you can without much of a stretch change it up of districts to the application's interpretation record. With BCP 47 support, the editorial manager consolidates dialect and district codes into a solitary determination for focused restrictions. Shading codes show whether a district is finished or as yet missing string interpretations.
2. Android Code Samples on GitHub
3. Expanded Template and Form Factor Support
4. Public and Private Resources: Naturally, Android Studio regards all library assets as open: An open library asset is accessible to library customers for use outside the library, and shows up in code finish proposals and other asset references. Android

Studio likewise, be that as it may, bolsters the utilization of private library assets. A private library asset must be utilized inside of the source library, and does not show up in code finish records and other asset references.

5. Cloud connectivity
6. Virtual Devices: To run the application that will be developed we need a platform to execute to test the application. It is possible by installing it on any of the android phone and collects the values and continues the experiment. Instead of using a real mobile, we can utilise the virtual devices provided by the Android Studio Tool.

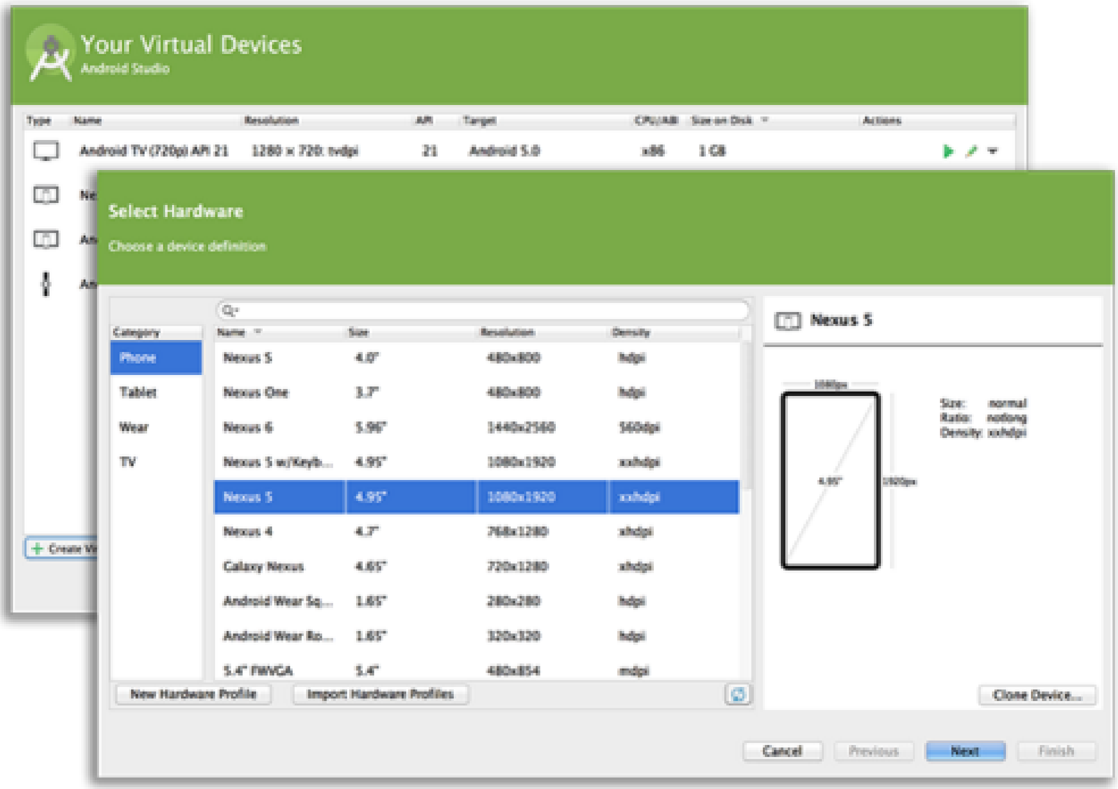

Fig 4.1.1: Virtual devices provided by the Tool.

4.1.1 App Development Workflow

To create applications for Android, you utilize an arrangement of apparatuses that are incorporated into Android Studio. Notwithstanding utilizing the instruments from Android Studio, you can likewise get to a large portion of the SDK apparatuses from the summon line. Creating with Android Studio is the favoured technique on the grounds that it can straightforwardly conjure the instruments that you require while creating applications.

Be that as it may, you may decide to create with another IDE or a basic content manager and conjure the instruments on the order line or with scripts. This is a less streamlined approach to create in light of the fact that you will once in a while need to call order line devices physically, yet you will have entry to the same number of components that you would have in Android Studio.

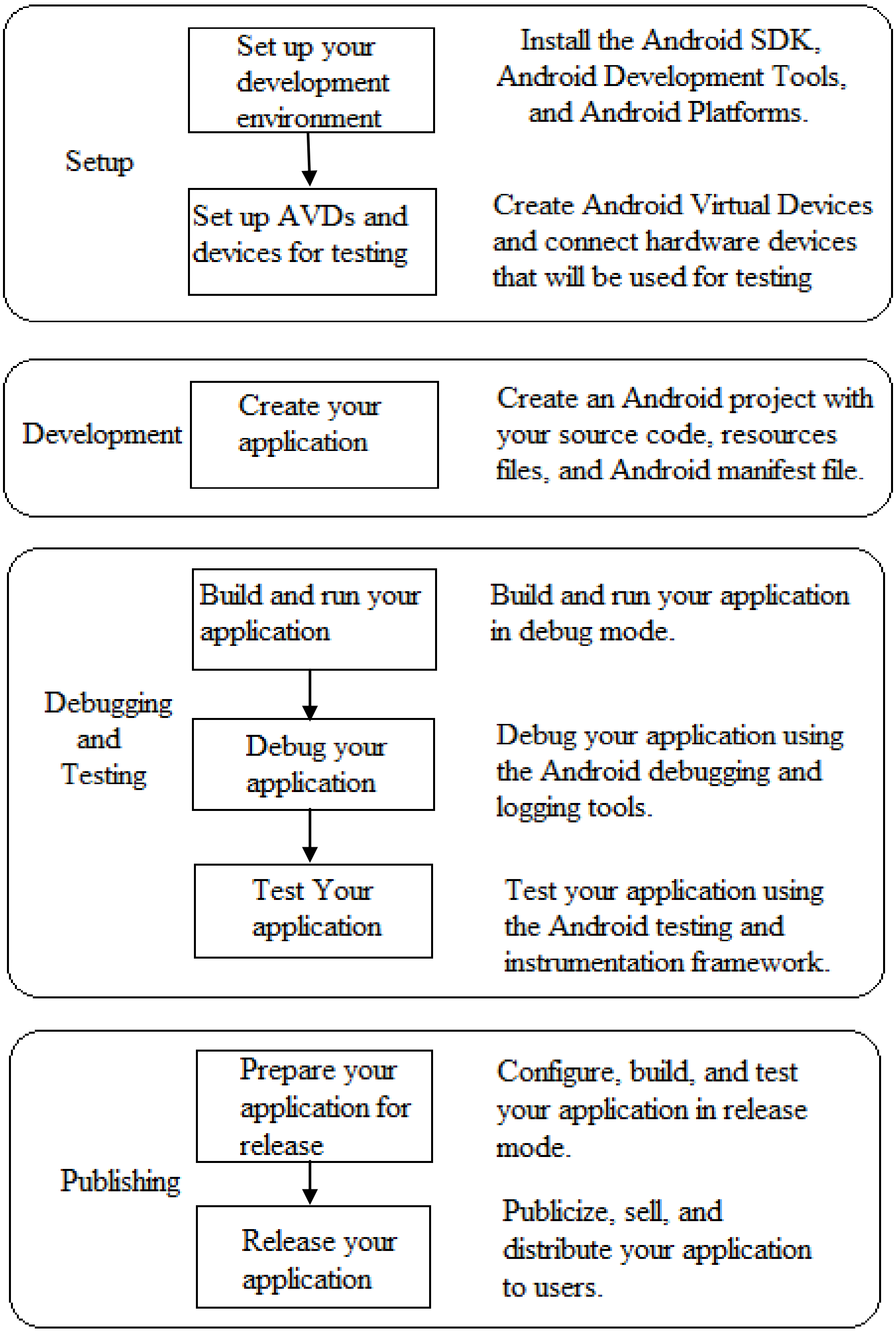

Fig 4.1.1.1: Process of application development

4.2 Android. Hardware Package

The Android Studio Tool provides an Android. Hardware package which contains all the requirements that are required to built an application. Most of the android devices have many built-in sensors that serve the purpose of measurement of different parameters for the environmental conditions. These sensors provide raw data with high precision and accuracy that could be used to different experiments using those values. Most of the Android platforms provide these categories of sensors: motion, environmental and position.

Motion sensors: These sensors measure speeding up strengths and rotational powers along three directions. This classification incorporates accelerometers, gravity sensors, gyrators, and rotational vector sensors.

Environmental sensors: These sensors measure different natural parameters, for example, encompassing air temperature and weight, enlightenment, and dampness. This classification incorporates indicators, photometers, and thermometers.

Position sensors: These sensors measure the physical position of a gadget. This class incorporates introduction sensors and magnetometers.

In this project we will use a package called android. hardware package which provides different interfaces and classes for development of mobile application. The android sensor framework is provided by the android studio tool which helps in accessing the sensors that are equipped in the smart phones through the mobile app which is installed in the smart phone. The sensor framework has the capability to determine the sensors that are available in the smart phone, also the sensor specifications and collect the sensor data. It also supports the task of registering and unregistering the event listeners that are required to signify the sensor changes. The sensor API doesn’t store the sensor reading automatically, thus we use recording API by registering listeners to receive raw sensor data and unregistering listeners to stop. There many types of sensor supported by various android platforms. Some of the examples are TYPE_ACCELEROMETER, TYPE_GRAVITY, TYPE_LIGHT, TYPE_PRESSURE, etc. The sensor framework provides the below classes and interfaces:

Sensor Manager (class): We can utilize this class to make an occurrence of the sensor administration. This class gives different routines to getting to and posting sensors, enlisting and unregistering sensor occasion audience members, and securing introduction data. This class additionally gives a few sensor constants that are utilized to report sensor precision, set information obtaining rates, and align sensors.

Sensor (class): We can utilize this class to make a reference to the type of sensor. It gives different routines to access the sensor specifications.

Sensor Event (class): The framework utilizes this class to make a sensor occasion object, which gives data around a sensor occasion. A sensor occasion item incorporates the accompanying data: the crude sensor information, the sort of sensor that produced the occasion, the exactness of the information, and the timestamp for the occasion.

Sensor Event Listener (interface): It is an interface that performs two basic tasks that receive notifications when the value of sensor varies and also at the instant of accuracy changes. The two tasks involve identifying different sensors available and their abilities and monitoring the sensor events.

```
private SensorManager sensormanagername;
private Sensor sensorname;
...
sensormanagername = (SensorManager) getSystemService(Context.SENSOR_SERVICE);
if (sensormanagername.getDefaultSensor(Sensor.TYPE_OF_SENSOR) != null){
   list<Sesnor> listname = sensormanagername.getSensorList(Sensor.TYPE_OF_SENSOR);
         for(int i=0; i<graySesnors.size();i++){
            if ((graySesnors.get(i).getVendor().contains("Google.Inc.")) &&
                     (graySesnors.get(i).getVersion() == 3)){
                           sensorname = graySesnors.get(i);
                  }
         }
}
else{
   if (sensormanagername.getDefaultSensor(Sensor.TYPE_OF_SENSOR) != null){
            sensorname = sensormanagername.getDefaultSensor(Sensor.TYPE_OF_SENSOR);
         }
         else{
            //do nothing
         }
}
```

Fig 4.2.1: Code to determine the existence of sensor.

4.3 Cloud Connectivity

4.3.1 App Engine Backend with Google Cloud Messaging:

Google Cloud Messaging (GCM) is an administration that permits you to send push notices from your server to your clients' Android gadgets, furthermore to get messages from gadgets on the same association. The GCM administration handles all parts of queueing of messages and conveyance to the objective Android application running on the objective gadget.

This backend layout utilizes Google Cloud Endpoints to characterize a RESTful API which enlists Android gadgets with your GCM server and permits you to send string messages to enrolled gadgets. In the wake of characterizing this API, specifically customer libraries are created naturally and can be called from your Android application, as portrayed underneath.

4.3.2 Adding a Backend in Android Studio:

To add the backend to your current Android application from this backend format, open Android Studio (establishment guidelines) and explore to "Record → New Module..." or right-tap on your task and pick "New → Module". In the "New Module" wizard that shows up, pick "Google Cloud Module". Then pick "Application Engine Backend with Google Cloud Messaging.

4.3.3 Connecting the Android App to the Backend:

The initial step is to obtain the Google cloud messaging API. Open the <backend>/src/main/webapp/WEB-INF/appengine-web.xml file (it should have been opened by default when you generated the backend), and navigate to this link: https://console.developers.google.com/flows/enableapi?apiid=googlecloudmessaging&keyType=SERVER_SIDE&r=0.0.0.0/0.

select a "create new project" option to make another Google Developers Console (or pick a current undertaking, on the off chance that you have one as of now), and snap "Proceed". Perused the Terms of Service and expecting that you agree, check the checkbox to show that you have perused them and click "accept". copy the created API key (in a red rectangle, begins with AIza...) once again into appengine-web.xml record, supplanting

<property name="gcm.api.key"value="YOUR_KEY_HERE"/>with

<property name="gcm.api.key" value="AIza..."/>

At last, backpedal to your venture's overview in Google Developers Console and note down the undertaking number (in red rectangle underneath): this will be your Google Cloud Messaging sender ID in the following step. When you select or make a venture, it will have "Google Cloud Messaging for Android" API empowered consequently. In the configuring arrangement dialog, you can utilize the supplied "0.0.0.0/0" IP address for testing purposes.

4.3.4 Registering Devices with Google Cloud Messaging Backend:

Prior to any messages can be sent from a Google Cloud Messaging backend to the devices, these devices should be enlisted with a GCM backend. When you added this backend module to your task, the required consents, required by Google Cloud Messaging have been included into the Android show of your application, and the required form conditions have been added to your application's build.gradle record. Moreover, a RegistrationEndpoint Cloud Endpoints API has been naturally produced for you, so that you could begin calling this endpoint from your Android application to enlist gadgets with your new Google Cloud Messaging backend. Here is a case code scrap which shows how to make an AsyncTask to enlist the client's device with your new backend

```
class GcmRegistrationAsyncTask extends AsyncTask<Void, Void, String>
{
private static Registration regService = null;
private GoogleCloudMessaging gcm;
private Context context;

private static final String SENDER_ID = "1234567890123";

public GcmRegistrationAsyncTask(Context context) {
this.context = context;
}

@Override
protected String doInBackground(Void... params) {
if (regService == null) {
Registration.Builder builder = new
Registration.Builder(AndroidHttp.newCompatibleTransport(),
new AndroidJsonFactory(), null)
// Need setRootUrl and setGoogleClientRequestInitializer only for
local testing,
```

```
// otherwise they can be skipped
.setRootUrl("http://10.0.2.2:8080/_ah/api/")
.setGoogleClientRequestInitializer(new
GoogleClientRequestInitializer() {
@Override
public         void         initialize(AbstractGoogleClientRequest<?>
abstractGoogleClientRequest)
throws IOException {
abstractGoogleClientRequest.setDisableGZipContent(true);
}
});
// end of optional local run code

regService = builder.build();
}

String msg = "";
try {
if (gcm == null) {
gcm = GoogleCloudMessaging.getInstance(context);
}
String regId = gcm.register(SENDER_ID);
msg = "Device registered, registration ID=" + regId;

// You should send the registration ID to your server over HTTP,
// so it can use GCM/HTTP or CCS to send messages to your app.
// The request to your server should be authenticated if your app
// is using accounts.
regService.register(regId).execute();

} catch (IOException ex) {
ex.printStackTrace();
msg = "Error: " + ex.getMessage();
}
return msg;
}

@Override
protected void onPostExecute(String msg) {
Toast.makeText(context, msg, Toast.LENGTH_LONG).show();
Logger.getLogger("REGISTRATION").log(Level.INFO, msg);
}
}
```

The actual registering can be made to the app, by invoking the asycnctask from one the android activites

4.3.5 Testing the Device Registration in an Emulator:

After adding the GcmRegistrationAsyncTask invocation to the android app as described above you must be ready to test the device registration with the backend locally. Ensure that you can access it via http://localhost:8080. If you can access the backend locally, change the run configuration back to your Android app and run the Android emulator.

4.3.6 Showing the Push Notification from Gcm Backend:

The push notifications are generated from the backend can be shown by adding two more classes into the android client they are GcmIntentService and GcmBroadcastReciever. After adding these classes to your Android application, register them by adding the following snipped into your`AndroidManifest.xml` file

4.3.7 Deploying:

Now the final step is to click the Deploy through which you can monitor the status of your development in the android studio.

After you had deployed your backend to the AppEngine you can connect to your android application my making small modifications which would be understood while using the application and it was all ready to start your application and successfully communicate with your AppEngine Backend.

5. DISCUSSION OF RESULTS

In this experiment we are considering four different sensors to compute the results and carry out the experiment. The four sensors being considered are:

- Type_Magnetic_field
- Type_Ambeint_Temperature
- Type_Pressure
- Type_humidity

The sensors can only detect a small area around it. The range of most of the sensors that are equipped in the smart phones varies from 5mm-11mm. It also includes the area from the smart phone, thus it can sense a very small area.

The pressure sensor uses a mechanical gauge element and computes value based on the displacement. The mostly observed displacement values are 0.004 – 0.012 inches. The unit of measurement by this kind of sensor is psi (pounds per square inch) and the values ranges from 0 – 400 psi.

The most famous temperature sensor available in the smart phones is thermocouple. It is of different forms such as J, K, T, B, E, N, S. The units of measurement is Celsius and the values ranges from 0 – 600 Celsius.

Humidity sensor measures based on relative humidity. It compares the temperature of the sensor and also the humidity of the environment and determines the value. The span is 20%-40% RH (relative humidity).

The magnetic field sensor is used to determine the location of the sensor which is reading the data and value is determined by comparing with the Earth's north and south pole. The values ranges from (+/-) 200 – (+/-) 2000 micro Telsa.

6. FUTURE SCOPE

The Classified [8] values dumped into the cloud for the prediction of the weather conditions for the required intervals of time, that is the data is submitted to the cloud storage maintaining a time window and processed using the Map-Reduce techniques. These predictions are again sent to all the smart phones with weather application using climate Analytics as a service proposed by NASA. This project’s extension is related to Climate Analytics. In this paper we have explained how we can collect values from the smart phone sensors as they have the capability to interact with the environment directly and these values are cleaned, transformed and stored at cloud. The data stored in the cloud are processed using Map-Reduce techniques in order to predict the values or the weather in the near future based on the stored data.

As the users are increasing not only for weather analysis, this methodology can be used for many other purposes which help the people to do their activities with ease and make important decisions. The cloud computing researchers are currently working on the service Climate Analytics as Service to be provided as a service in the cloud.

CAaaS refers to predicting the future weather conditions based on the meteorological data.

7. REFERENCES

1. Activity Recognition With Smart Phone Sensors, Xing Su, Hang Hang Tong, and Ping Ji,TSINGHUA Science and Technology, June 2014.
2. Barometric Sensors, Hao Xia , Xiagang wang ,YanYou Quio, Jun Jian and Vuanfai Chang, 31 march 2015.
3. A Health monitoring Systems using Smart Phones and Wearable Sensors. Valerie Gay, Peter Leijdekkers, University of Technology Sydney, June 2007.
4. KNN Model-Based Approach in Classification, Gongde, Guo Hui Wang, David Bell, Yaxin Bi and Kieran Greer, Queens university Belfast, march 2014.
5. Real-Time Sensing on Android, Yin Yan, Shaun Cosgrove, Ethan Blanton, Cse.edu.buffalo, April 2012.
6. A Study of Mobile Sensing Using Smartphones, Ming Liu, Hindawi Publishing Corporation, 15 January 2015.
7. Y. Liang, X. Zhou, Z. Yu, and B. Guo, Energy-efficient motion related activity recognition on mobile devices for pervasive healthcare, Mobile Networks and Applications, 1-15, 2013.
8. I. Anderson, J. Maitland, S. Sherwood, L. Barkhuus, M. Chalmers, M. Hall, B. Brown, and H. Muller,Shakra: Tracking and sharing daily activity levels with unaugmented mobile phones, Mobile Networks and Applications, vol. 12, no. 2-3, 2007.