View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by SJSU ScholarWorks

San Jose State University
SJSU ScholarWorks

Master's Projects | Master's Theses and Graduate Research

Spring 5-20-2019

Sensor - Based Human Activity Recognition Using Smartphones

Mustafa Badshah
San Jose State University

Follow this and additional works at: https://scholarworks.sjsu.edu/etd_projects

Part of the Artificial Intelligence and Robotics Commons

Recommended Citation
Badshah, Mustafa, "Sensor - Based Human Activity Recognition Using Smartphones" (2019). *Master's Projects*. 677.
DOI: https://doi.org/10.31979/etd.8fjc-drpn
https://scholarworks.sjsu.edu/etd_projects/677

This Master's Project is brought to you for free and open access by the Master's Theses and Graduate Research at SJSU ScholarWorks. It has been accepted for inclusion in Master's Projects by an authorized administrator of SJSU ScholarWorks. For more information, please contact scholarworks@sjsu.edu.

Sensor - Based Human Activity Recognition Using Smartphones

A Project Report

Presented to

The Faculty of the Department of Computer Science

San José State University

In Partial Fulfillment

Of the Requirements for the Class

CS 298

By

Mustafa Badshah

May 2019

© 2019

Mustafa Badshah

ALL RIGHTS RESERVED

The Designated Project Committee Approves the Project Titled

Sensor – Based Human Activity Recognition Using Smartphones

by

Mustafa Badshah

APPROVED FOR THE DEPARTMENT OF COMPUTER SCIENCE

San José State University

May 2019

Dr. Robert Chun	Department of Computer Science
Dr. Nada Attar	Department of Computer Science
Mr. Adil Khan	Software Engineer, Adobe

ABSTRACT

It is a significant technical and computational task to provide precise information regarding the activity performed by a human and find patterns of their behavior. Countless applications can be molded and various problems in domains of virtual reality, health and medical, entertainment and security can be solved with advancements in human activity recognition (HAR) systems. HAR is an active field for research for more than a decade, but certain aspects need to be addressed to improve the system and revolutionize the way humans interact with smartphones. This research provides a holistic view of human activity recognition system architecture and discusses various problems associated with the design aspects. It further attempts to showcase the reduction in computational cost and significant achievement in accuracy by methods of feature selection. It also attempts to introduce the use of recurrent neural networks to learn features from the long sequences of time series data, which can contribute towards improving accuracy and reducing dependency on domain knowledge for feature extraction and engineering.

***Index Terms* – Human activity recognition, machine learning, mobile sensors, accelerometer, gyroscope, feature selection, RNN.**

ACKNOWLEDGEMENTS

I want to thank my mentor Dr. Robert Chun for his constant support throughout this research and for encouraging me to do my best. I would also like to thank my committee members, Dr. Nada Attar and Mr. Adil Khan for having taken interest in my research and having agreed to help me as needed.

Last but not least, I would like to thank my parents for their invaluable support and for making it possible for me to chase my dreams.

Table of Contents

1 Introduction

Mobile devices have become an integral part of our daily life. This is due to an increase in the sophisticated development and the integration of quality sensors, high computational power, large storage capacity and continuous connectivity in mobile devices. People constantly interact with their low cost, small-sized smartphones in their day to day activities, which has led to the rise in the research of extracting knowledge from data acquired by pervasive sensors in mobile devices [1]. Attention towards creating lifelogs, which refers to the use of technology to capture and document large amounts of a user's life through mobile devices, has increased considerably. A good example of lifelogging is capturing the number of steps walked each day using a smartphone. Lifelogs can be used to document simple physical activities such as walking, running, sitting, etc. or complex activities such as eating, working, exercising, etc. This has a wide variety of application in various fields of research such as medicine, augmented reality, computer-human interaction, security and targeted advertising. A lifelog can be used to mine knowledge and give insights about the lifestyle of a user and help improve the quality of life by providing personalized recommendations and services. Creating context-aware applications and services with low-cost consumer hardware will be a significant step towards solving more complex problems.

The biggest issue faced in creating a detailed lifelog is the collection of activity data through various wearable sensors and the accurate classification of human activity based on the collected data. Mobile devices come integrated with powerful sensors, such as accelerometers, gyroscopes, GPS, magnetometer, proximity and ambient light detectors. With the use of smartphones, researchers have an opportunity to collect sensor data easily with the use of minimal infrastructure. Modern machine learning techniques can be used to distinguish

and recognize human activities based on the data collected. A simple smartphone could help solve the problem of documenting a detailed history of a user's daily activity. Advancements in deep learning and methods for feature selection along with the inclusion of a variety of sensors can push the boundaries of human activity recognition on deeper ontological levels.

2 Problem Definition

Human Activity Recognition (HAR) is the problem of identifying a physical activity carried out by an individual dependent on a trace of movement within a certain environment. Activities such as walking, laying, sitting, standing, and climbing stairs are classified as regular physical movements and form our class of activity which is to be recognized. To record movement or change in movement, sensors such as triaxial accelerometer and gyroscopes, capture data while the activity is being performed. A triaxial accelerometer data detects acceleration or movement along the three axes and a gyroscope measures rotation along the three axes to determine direction. Data recorded is along three dimensions of the X, Y and Z axis at the specified frequency. For example, a frequency of 20Hz would indicate that 20 data points are recorded each second of the action. Various other physiological signals such as heartbeat, respiration, etc. and environmental signals such as temperature, time, humidity, etc. can further augment the recognition process. Activity recognition can be achieved by exploiting the information retrieved from these sensors [26].

The challenge arises as there is no explicit approach to deduce human actions from sensor information in a general manner. The large volume of data produced from the sensors and use of these features to develop heuristics introduces the technical challenge. Storage, communication, computation, energy efficiency, and system flexibility are some of the aspects which need to be analyzed in detail to build a robust activity recognition system. Conventional pattern recognition methods have made tremendous progress in discovering significant information from scores of low-level readings. But such recognition models are successful for data collected in controlled environments, and for few activities only. Complex HAR tasks are hindered due to the naïve feature extraction techniques and limitation in domain knowledge.

The shallow features extracted degrades the performance of unsupervised learning algorithms and connected activities. Deep learning models have the capabilities to learn features of the higher order [27]. Advancement in such models makes it conceivable to learn and improve the performance of the predictive models and find deeper knowledge from human activities.

3 Architecture, Sensors and Design Issues

3.1 General Architecture

The architecture for HAR systems has a generalized architecture composing of two main phases – (1) Data Acquisition, (2) Activity Recognition. Data acquisition mainly deals with the collection and storage of the sensor data while the activity recognition deals with machine learning models used for predictive analytics. The data acquisition system has a standard structure as shown in Figure 1. The main component of the data acquisition phase is the sensors which measure the various attributes such as acceleration, location, audio, temperature, etc. The other components are the integration device, communication network, and remote application server. The integration device is used prominently for collecting and preprocessing the raw sensor signal. The data can also be sent to a remote application server with the use of networking protocols such as TCP/IP or UDP, for real-time analysis and visualization [2], [20]. Not all the components are required and implemented in each data acquisition system. In [4]-[6], the sensors are integrated within a device itself and carry out the analytical processing on it. Other systems require an external wearable device which communicates with an integration device such as a laptop, cellphone, etc. Different applications and requirements reflect on these differences in the general architecture.

Activity recognition component relies heavily on machine learning models and is built on the training and testing stages. As described in [11], the training stage requires a large dataset of the collected features to train the model. Varied processes such as data cleaning, feature extraction, dimensionality reduction, and feature selection take place in the training stage. Similarly, the authors in [11] describe the testing stage. This stage has a smaller dataset and undergoes the same data processing. It is then used to test the machine's predictions and evaluate

the training.

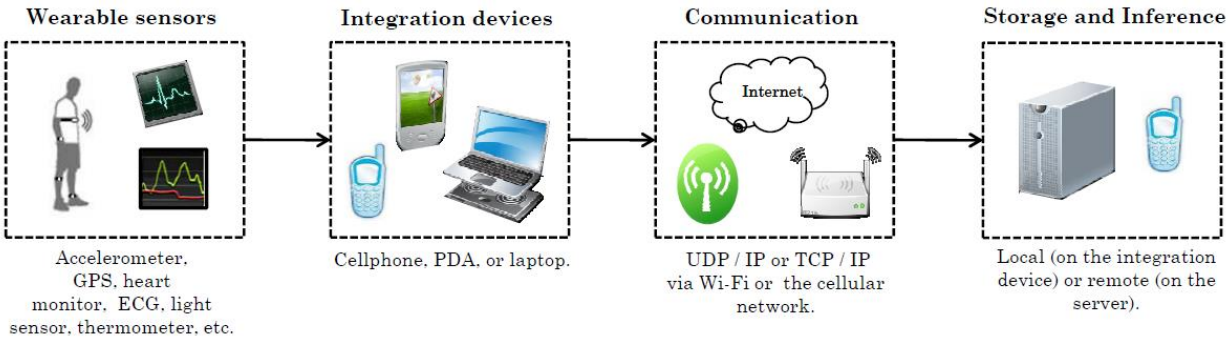

Fig 1. Data acquisition structure for HAR system

3.2 Sensor Modality

According to [27], sensor modalities are classified into three basic types:

- Body-worn sensors
- Object sensors
- Ambient sensors

3.2.1 Body-Worn Sensors:

Sensors such as an accelerometer, gyroscope, and GPS which are embedded within devices such as smartphones, watches, glasses, caps, etc. come under the category of body-worn sensors. These sensors are attached at various positions to the human body to trace and recognize different activities. Among the work surveyed, the accelerometer is the highly adopted sensor, due to its capabilities of recording change in acceleration of the human body. A gyroscope or a magnetometer is paired with the accelerometer recognize daily physical activities or sports activities.

3.2.2 Object Sensors:

Object sensors are sensors which are placed in an environment or on certain objects to detect motion of a specific object. RFID, WIFI, Bluetooth can record information on a deeper

level to recognize complex activities. Such sensors are not worn on the body of the person but are placed in the environment to detect movement from a different perspective. Use of such sensors is very rare as they are expensive and difficult to install. The pairing of object sensors with body-worn sensors in controlled environments can help advance the research of deeper ontological level human activity recognition.

3.2.3 Ambient Sensors:

Ambient sensors capture the variables of the environment such as temperature, sound, pressure, humidity, etc. They do not specifically capture the change in movement of a human but capture data related to change in the environment. They provide knowledge holistically about a person's surrounding and the environment in which the action is being carried out. Such sensors are found in smart home environments.

3.3 Design Issues

There are various design issues which need to be taken into account to develop an efficient HAR system. The design issues impact heavily on the usage of the system and the accuracy of the prediction.

3.3.1 Selection of Sensor Attributes

A variety of sensors are present and can be broadly classified into the following four groups:

3.3.1.1 Acceleration Signal:

Triaxial accelerometers are low energy consuming, cheap sensors which are extensively used to recognize activities such as walking, sleeping, sitting, etc. Most of the ambulatory activities can be recognized with the use of a triaxial accelerometer [7] – [9]. The position and

placement of the accelerometer play a significant role in the prediction of the activity and accuracy achieved. According to He et al. [9], the best place to keep the accelerometer is in the pant pocket, but this conclusion varies with the type of activity to be recognized. Accelerometers are also not useful to deduce more meaningful activities such as working at the desk or eating at the dinner table. The same body motion for multiple activities is confusing to recognize from the accelerometer's point of view [15].

3.3.1.2 Location Signals:

Global Positioning Systems provide location data with the use of satellites. All current smartphones are equipped with GPS sensors and the data can be used to provide context-aware recognition or infer activity based on ontological reasoning [5]. However, Reddy et al. [19] report that the biggest issues faced in GPS sensors are that they perform poorly indoors, are high energy consuming sensors and are associated with privacy issues. This puts a limit on the usage of GPS data for real-time applications. Riboni et al. [5] suggests that to overcome the poor performance indoors, GPS sensors should be paired with accelerometers.

3.3.1.3 Environmental Attributes and Physiological Signals:

For better contextual information, attributes like audio, temperature, light intensity, time, microphones, etc. are used. They provide information about the environmental setting of an individual to infer activities. The authors in [15] and [17] analyze that individually, the environmental sensors do not contribute sufficiently towards the recognition, and are also easily affected due to weather conditions, external artificial illumination and loud noise levels. Physiological signals such as heart or respiration rate, ECG, body temperature can be considered as vital signs and have been used in a few HAR systems. Tapia et al. [16] combined data collected from a heart monitor and accelerometer for activity recognition and concluded that the

heart rate signal is not useful. It misclassified activities while increasing cost and energy with the use of additional obtrusive sensors.

3.3.2 Energy Consumption

Individual sensors and sensors embedded within mobile devices are heavily constrained by battery life. Many applications require critical data to be delivered from the sensors which make efficient energy consuming sensors highly desirable. Battery life can be extended by limiting the communication with the sensors as it is a very expensive operation. Short-range communication, data filtering, and compression techniques should also be utilized to save energy. Riboni et al. in [5] discuss how analytics and classification should be performed over the integrated device itself to lower the communication with an external server.

3.3.3 Processing Techniques

Processing the data on the server or within the integrated device itself is an important design decision. Lara et al. in [11] deduce that the design aspect depends on the application, whether fast real-time results or passive results are required. A HAR system deployed over a mobile device is a more scalable design as it reduces the communication and alleviates the server load by locally computing the classification. It would also help overcome unreliable communication systems and become a highly responsive application. However, mobile devices fall short on high processing power and large storage needs. In [12] - [14] the researchers discuss the shortcomings of a mobile HAR system in terms of storage, energy consumption, and computational power. In [13], Williams and Matthew describe a scenario where the application requires data from a group of users, and how it is beneficial to compute the classification over a central server rather than on individual mobile devices. This trade-off needs to be carefully

analyzed with the needs of the application.

3.3.4 Obtrusiveness

Many HAR systems require data to be collected from many sensors for a deeper inference. This requires individuals to wear or carry more sensors which can become uncomfortable, expensive and also invasive. Unobtrusive systems are highly desirable and the authors in [4], [15] and [19] propose systems which collect data from embedded sensors within a smartphone or a watch.

3.3.5 Flexibility

The design of an activity recognition model is under heavy scrutiny and debate as some studies suggest that the recognition model should be specific to an individual [23] and some emphasize the model should be flexible enough for a generalized group [20]. The analysis shows that it is not suitable nor efficient to train the same model for different users if there are too many activities to train the model for or if the individual is unable to perform certain activities (e.g. swimming). However, recognizing activities without considering the individual characteristics would lead to a decrease in accuracy and in the training efficiency.

4 Related Work

In this section, we analyze previous HAR systems that rely on supervised learning and sensor data and separate them into online and offline systems. Supervised learning required labeled data to learn from and give an output. Online systems provide immediate feedback while offline systems need more time to recognize activities due to high computational demands.

4.1 Online Systems:

Vigilante is a mobile application built by the authors of [24] for real-time human activity recognition. The authors measured the acceleration and physiological signals such as heart and respiration rate, breath waveform and skin temperature through external sensors. Using the Mobile Evaluation of Classification Algorithms (MECLA) library, the authors implemented the C4.5 decision tree classifier to recognize three ambulation activities. The classifier achieved an overall accuracy of 92.6%. The application has a fast response time and is trained with different users with diverse characteristics to ensure a more flexible system is built. This reduced the need to retrain the model for new users and was considered a moderate energy efficient system.

Berchtold et al. in [4] proposed the ActiServ platform which used a cellphone to capture the acceleration signal. The authors developed an efficient and portable fuzzy inference system to classify ambulation activities. The accuracy achieved varied between 71% and 97%. If the algorithm is meant to meet a real-time response then the accuracy drops down to 71%, and if the algorithm is allowed to train to its full capacity, which takes an order of days, it reaches an improved accuracy of 97%. A subject dependent analysis boosted the accuracy to 90%.

Riboni et al. [5] presented COSAR, a framework for context-aware activity recognition. Through COSAR, it was possible to recognize activities such as brushing, writing on a blackboard, strolling, etc. The authors used the combination of two accelerometers and a GPS

sensor within a cellphone to collect data. The authors introduced the concepts of potential activity matrix and statistical historic classification to filter out activity prediction which was not suitable. The overall accuracy achieved was 93%.

4.2 Offline Systems:

Parkka et al. [17] utilized twenty-two signals which include acceleration, vital signs and environmental variables to classify seven ambulatory activities. Sensors were positioned at different points over the body and the individual required to carry along a compact computer in a bag pack to collect the sensor data. This system was considered highly obtrusive and had high privacy concerns as well. The authors built three classification models like the auto-generated decision tree, custom decision tree, and an artificial neural network. The researchers achieved the highest accuracy of 86 % from the first classifier. These models required high computational capabilities and hence were classified as offline systems.

Zhu and Sheng [25] proposed a system which utilized the combination of two classifiers to recognize activities. The system architecture was highly obtrusive as sensor data were collected on to a PDA device and then the signals were transferred to a computer. Classification made use of the acceleration signals and worked in 2 phases. The first phase required an artificial neural network to classify the activity as stationary or non-stationary. The output is then inputted to a Hidden Markov Model (HMM) model for a specific activity prediction.

Lara et al. [20] used a single smartphone and a sensor device to collect acceleration and vital sign information and create a portable and unobtrusive real-time platform, named Centinela. Signals acquired from the sensors were processed to extract time and frequency features. Centinela programmed to recognize five ambulatory activities and achieved an accuracy of 95.7% after being evaluated over eight different classifiers. Certain activities

achieved an accuracy of 100 %. As the proposed system depended on an ensemble of classifiers, it carried a high computational cost, and hence considered as an offline system. To overcome the issues faced in this system, the authors proposed Vigilante.

5 Data, Algorithms, and Framework

5.1 Data:

"Activity Recognition Using Smartphone" dataset was prepared and made publicly available by Davide Anguita et al. [26] and can be freely downloaded from the UCI Machine Learning repository. The raw data is not available, but the preprocessed version of the dataset is made publicly available to carry out experiments. The smartphone sensor data was collected from the experiments conducted on a group of 30 volunteers who were within the age bracket of 19 - 48 years. The set of physical activities focused by the authors are walking, sitting, standing, laying, walking upstairs and walking downstairs. The authors attached a Samsung Galaxy SII smartphone to each subject to capture sensor data. Each subject was instructed to perform each of the six activities twice. On the first trial, the smartphone was firmly attached on the left side of the waist of the subject but in the second trial, the subject was given the opportunity to place the smartphone as they preferred. This insured there is variation in data based on the position of the phone for the same activity. The authors video recorded the action performed by each subject which assisted them to manually label the signals captured by the sensors.

Signals produced by the accelerometer and gyroscope, embedded within the Samsung Galaxy SII, are captured through a smartphone app. An accelerometer, as the name suggests, is used to measure the acceleration of the device. Values along the X, Y and Z axis are utilized to detect motions such as swinging, tilting, vibration, etc. Figure 2 shows the orientation of the axis of a triaxial accelerometer with respect to the device. Values provided over the three axes also include the gravitational acceleration of the earth ($g = 9.81\ m/s^2$). If the mobile device is at rest, it would only show the gravitational acceleration over one of the axes based on the

orientation. A gyroscope, on the other hand, makes use of angular velocity to calculate the rotation or twist in a smartphone device. The rate of rotation is measured in rad/s along the three axes. While an accelerometer detects directional movement, a gyroscope detects the lateral orientation of the device. Both the sensors are used to measure the rate of change, but for different things.

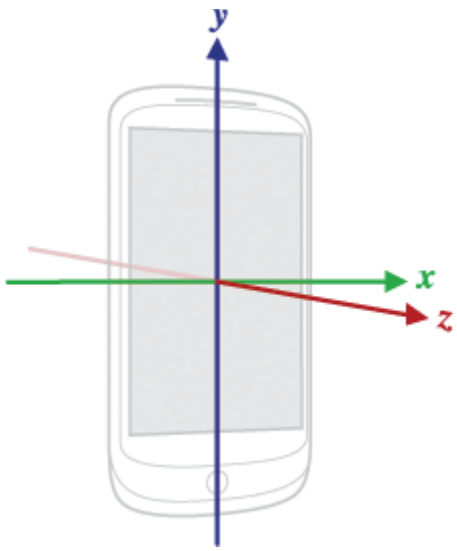

Fig 2. Axis orientation of a smartphone device

Anguita et al. captured the sensor signals at a constant rate of 50Hz and were subsequently preprocessed to reduce noise. The signals were preprocessed for noise reduction with a median filter and a 3rd order low-pass Butterworth filter with a 20Hz cutoff frequency. The Butterworth filter was used to separate the acceleration signal into body acceleration and gravitational acceleration. The processed signals were sampled into a fixed window of length 2.56 seconds with a 50% overlap. Each window had 128 data points for each of the original features recorded, which are body acceleration, body gyroscope and gravity acceleration over X, Y and Z axis. The windowed inertial signals were feature engineered and several, time and frequency, features were extracted from each window. Feature engineering resulted in a feature vector of 561 elements. The authors randomly split the dataset in a 70:30 ratio which created a distribution of

21 subjects for training and 9 subjects for testing. The training dataset has a total of 7352 windows of data while the testing data set has 2947 windows of data. The HAR process becomes clear where a window of activity, specifically 2.56 seconds of activity, is predicted for a new user by a model trained over a trace of activity from known subjects.

5.2 Feature Selection

Feature selection is an important concept in machine learning which is applied as part of the pipeline. It is the concept of automatically or manually selecting a set of features which contribute to improving the model and the prediction output. This step is undertaken as it immensely impacts the performance of the model in terms of the build time as well as the accuracy. Irrelevant features in the dataset can negatively influence the training as it makes the model train on data which do not contribute to reaching the predictive output. The impact is quite often seen on the accuracy as the irrelevant data acts just as noise. Feature selection provides benefits by reducing overfitting, improving accuracy and reducing training time. By applying feature selection, we reduce the dimensions of the dataset. This is often misunderstood as dimensionality reduction which is not the case. Dimensionality reduction techniques often combine features together to reduce the dimensions, while feature selection techniques eliminate attributes without affecting the rest. Two techniques of feature selection are discussed in the following sections.

5.2.1 Tree-Based Feature Selection:

Feature importance is an important property which comes along with tree-based classifiers such as decision trees or random forests. The property of the model gives you the importance of each feature with the dataset and enables us to determine the influence of the

feature on the prediction. Higher the score of the feature relates to a greater influence on the positive output. The unimportant features are discarded based on a threshold value.

5.2.2 L1 Based Feature Selection:

L1 based feature selection utilizes the coefficients of regression models for the selection and interpretation of features. The idea behind this technique is that features which are uncorrelated to the prediction variable will have their coefficients close to or equal to zero and important features would high coefficients. Linear models which are penalized with the L1 norm produce sparse solutions. As each non-zero coefficient contributes to the penalty, the L1 regularization forces weak features to zero and are discarded. For classification Logistic Regression or LinearSVC models are used while for regression Lasso model is used. The parameters C or alpha control the sparsity or the number of features selected.

5.3 Machine Learning Algorithms:

The following section discusses the various machine learning used in the experiments.

5.3.1 Decision Tree Classifier:

A popular machine learning model, decision trees uses a tree-like structure to represent decisions. They are constructed in a top-down structure with the use of metrics such as Gini impurity and information. It calculates the importance of each feature and uses it to split the elements into homogenous subsets. The nodes represent the condition of the split and the leaf nodes represent the decision or the predicted output. The branches or the edges of the tree direct to one of the output variables. Decision trees are modeled for both, classification and regression problems. Though the decision tree is easy to understand it tends to overfit as it continues to split on attributes and trains critically on the training data. To avoid overfitting the decision tree is generally pruned to stop it from growing too deep.

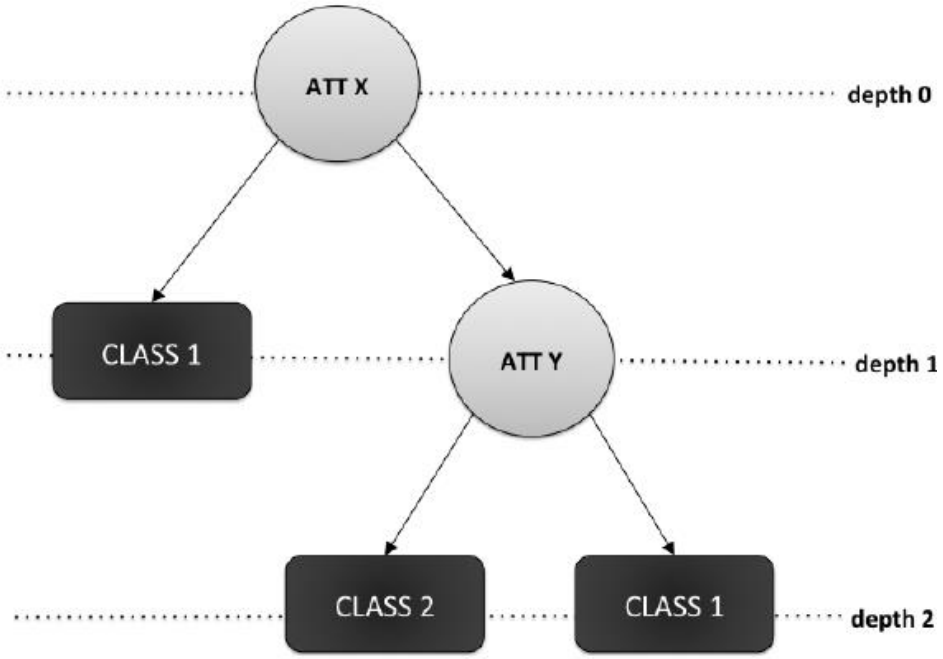

Fig 3. General decision tree structure for classification

5.3.2 Random Forrest Classifier:

Random Forest Classifier builds a forest which is an ensemble of decision trees. It creates a set of decision trees from a randomly selected subset of the training data and aggregates the decision from all the trees to decide the final output. This technique is robust as it prevents the noisy output of some trees affecting the final decision and avoids overfitting. It cancels out the bias by averaging all the predictions. Random forest is differentiated from decision trees as it does not search for the best feature while splitting the node. It instead searches for the most appropriate feature from a subset of features. This provides diversity and randomness to the algorithm. The algorithm can be easily modeled to both, classification and regression problems.

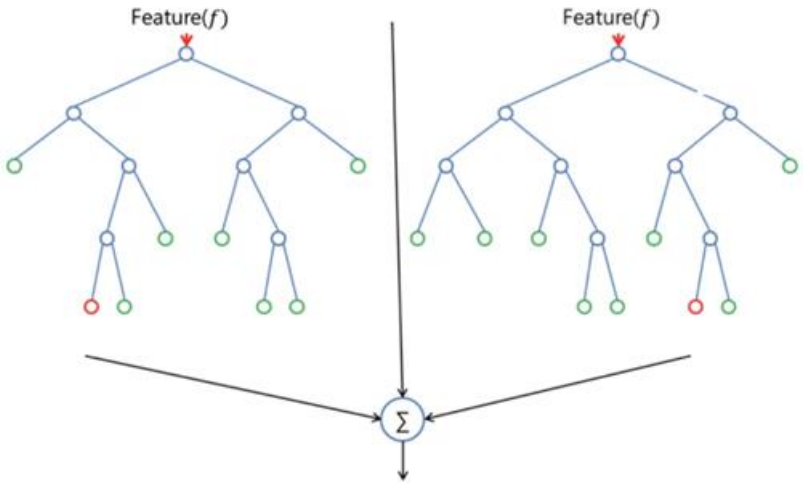

Fig 4. General random forest structure for classification

5.3.3 Artificial Neural Network:

Artificial Neural Network is a system inspired by the biological neuron structure of the brain. The structure can be defined as a set of connected neurons organized in consecutive layers. The input layer acts as the first layer which brings in the data into the network. The hidden layer consists of artificial neurons which take in a set of weighted inputs and apply an activation function to produce an output. There could be multiple hidden layers in a network which makes it capable to solve complex problems. The output of a layer of neurons is passed on as the input to the successive layer and is termed as a feed forward network. The output layer provides the final predictive output.

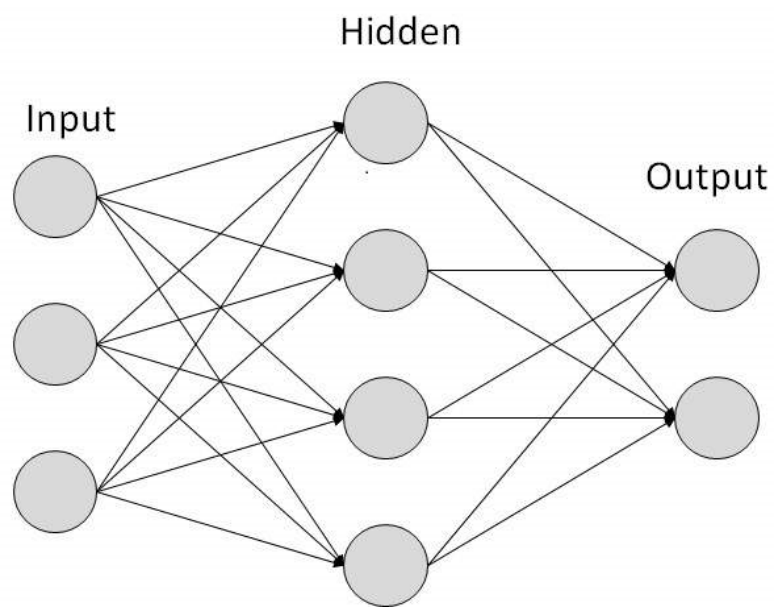

Fig 5. General Structure of Artificial Neural Network

Activation functions core logic of the neural networks and defines the output of the neuron given an input or a set of inputs. Following are the different activation functions:

- **Sigmoid:**

  The sigmoid function has a characteristic ‘S’ shaped curved and is widely used in binary classification. The function generates a probability output between 0 and 1 for a set of input.

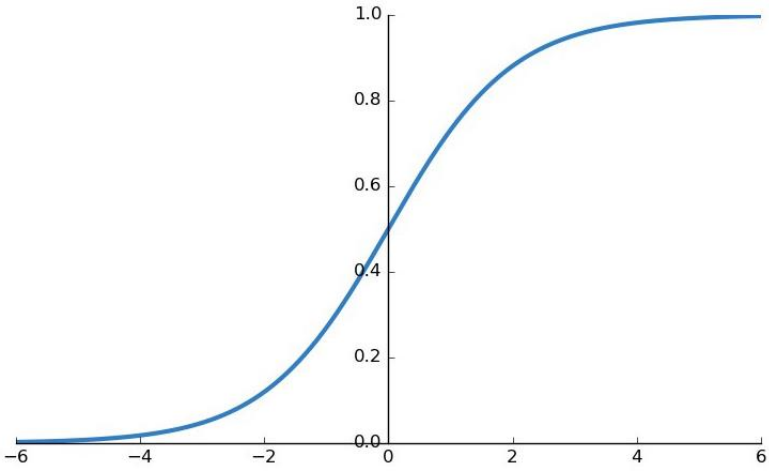

Fig 6. Sigmoid Curve

- **ReLU:**

Rectified Linear Units are most commonly used in the hidden layers of the artificial neural networks. The function is such that if the input is less than zero the output is 0 and if the input is greater than zero it gives the input itself as the output.

$$ReLU(x) = \max(0, x)$$

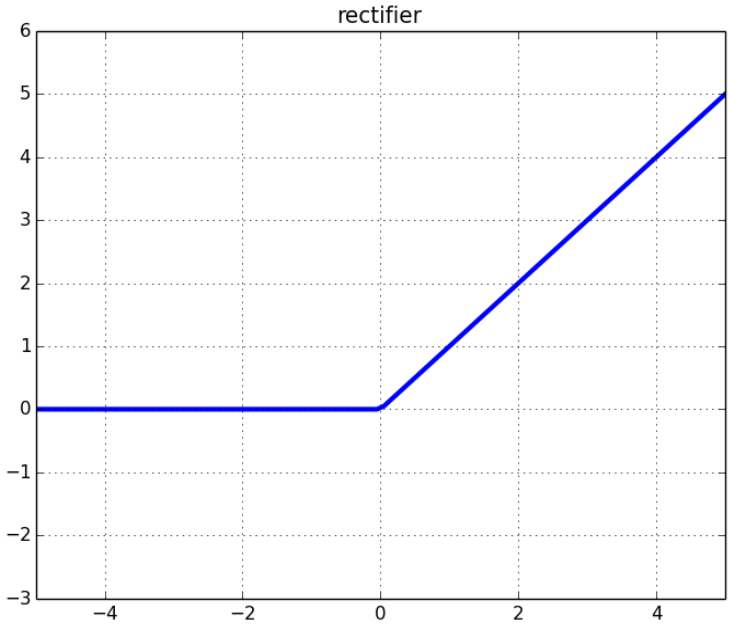

Fig 7. ReLU Curve

- **Softmax:**

The softmax activation function is used for multi-class classification. It calculates the probability distribution of each class over all possible target classes. Based on the calculated probabilities it determines the output for a given set of inputs.

$$Softmax(x) = \frac{e^j}{\sum_i \ e^i}$$

5.3.4 Recurrent Neural Network:

Recurrent Neural Networks are the only networks with internal memory which makes them robust and powerful. The network can be precise about the next prediction as they have the ability to remember significant bits of the input due to the internal memory. This makes the model highly preferable to train sequential data like text, audio, video and time series. A feed-forward network does have any memory of the previous input and works only on the current input. In a recurrent neural network, the current is considered along with the past learnings. Weights are applied to both the current input and the looping back output and are adjusted through gradient descent or backpropagation.

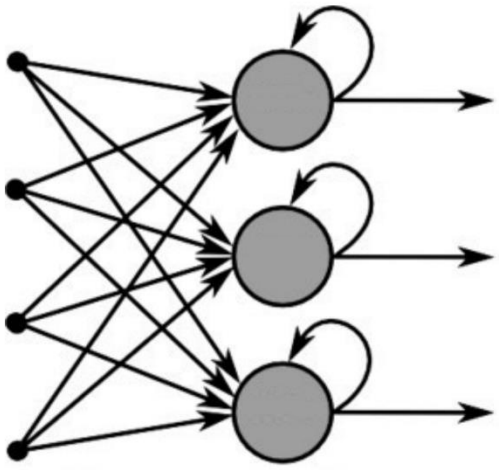

Fig 8. General Structure for Recurrent Neural Network

- **Long Short Term Memory (LSTM):**

Recurrent Neural Networks usually have a short memory and are extended by LSTM units to extend the memory of the network. It enables the network to remember input over a longer period of time which makes it an essential unit in the layers of the recurrent neural network. It provides the capabilities to absorb more information from even longer sequences of data. This helps to boost the precision of the prediction by taking into account more data.

5.4 Frameworks

5.4.1 Pandas:

Pandas, an open source Python library, provides a framework to construct data into a tabular form and perform a row, column and cell transformations. It is a useful tool load, analyze and mine data for insights and also structure it in a form to be consumed by machine learning algorithms.

5.4.2 NumPy:

NumPy, an open source Python library, is used along with the pandas library to handle multidimensional data and perform complex scientific and mathematical operations on the data.

5.4.3 Scikit-learn:

Scikit – learn is Pythons open source machine learning library. It provides the capabilities to easily build various regression, classification and clustering algorithms. It allows to create pipelines and validate output with a variety of evaluation metrics. The library also always to customize the algorithms by hyper tuning the parameters of the models. It also contains algorithms which are used to preprocess data, extract features and reduce dimensions.

5.4.4 Keras:

Keras is an open source neural network API in Python which can run on top Theano, CNTK, and TensorFlow. It provides a layer of abstraction to the complexities of creating a neural network and which augments the process of fast experimentation. It is well suited to create recurrent networks and convolutional networks.

6 Experiments and Results

6.1 Data Analysis:

The dataset was converted in a CSV file and imported into a pandas data frame. The only preprocessing required was to combine the subjects and activity columns to their respective windowed features. Upon exploration, it was found that the dataset did not have any missing values. The data set was explored further to understand the various features and their effects on the activities. The training dataset has a total of 7352 observations or windows of data. It has a total of 561 time and frequency features where each observation corresponds to one of the 6 ambulatory class activities. The 6 activities are as follows:

- Class 1 - Walking
- Class 2 - Walking Upstairs
- Class 3 - Walking Downstairs
- Class 4 - Sitting
- Class 5 - Standing
- Class 6 – Laying

The dataset was investigated to check the balance between the activity's observations performed by the 30 subjects. The count and graph analysis showed that the distribution of the classes ranged 13% and 19% for both, the training and test data. Though the data distribution is not equal for all activities, they are closely balanced. Table 1 gives us the distribution of each class activity for the training and test data. Figure 9. plots the count of activities for the training data set and Figure 10. displays the count of activities recorded for the test data. The graphs clearly indicate that the activity 'laying' has the maximum number of observations recorded and 'walking downstairs' has the minimum. This observation can similarly be seen in the training dataset too.

Training Data		
Class	**Count**	**Percentage**
Walking	1226	16.676
Walking Upstairs	1073	14.595
Walking Downstairs	986	13.411
Sitting	1286	17.492
Standing	1374	18.689
Laying	1407	19.138

Testing Data		
Class	**Count**	**Percentage**
Walking	496	16.831
Walking Upstairs	471	15.982
Walking Downstairs	420	14.252
Sitting	491	16.661
Standing	532	18.052
Laying	537	18.222

Training and Testing Data		
Class	**Count**	**Percentage**
Walking	1722	16.72
Walking Upstairs	1544	14.992
Walking Downstairs	1406	13.652
Sitting	1777	17.254
Standing	1906	18.507
Laying	1944	18.876

Table 1. Percentage Distribution of Activities

Though the observations for each activity are not exactly equal the data set overall provides a well-balanced distribution of the activity observations. Even after the separation of the data into the training and testing dataset, the balance in observation holds true. The count of observations for certain activities such as walking up the stairs or downstairs helps us understand that the subjects did not carry out the task for a specific time but instead for a fixed distance. Different walking styles has lead to a recording a different count of observations for each subject.

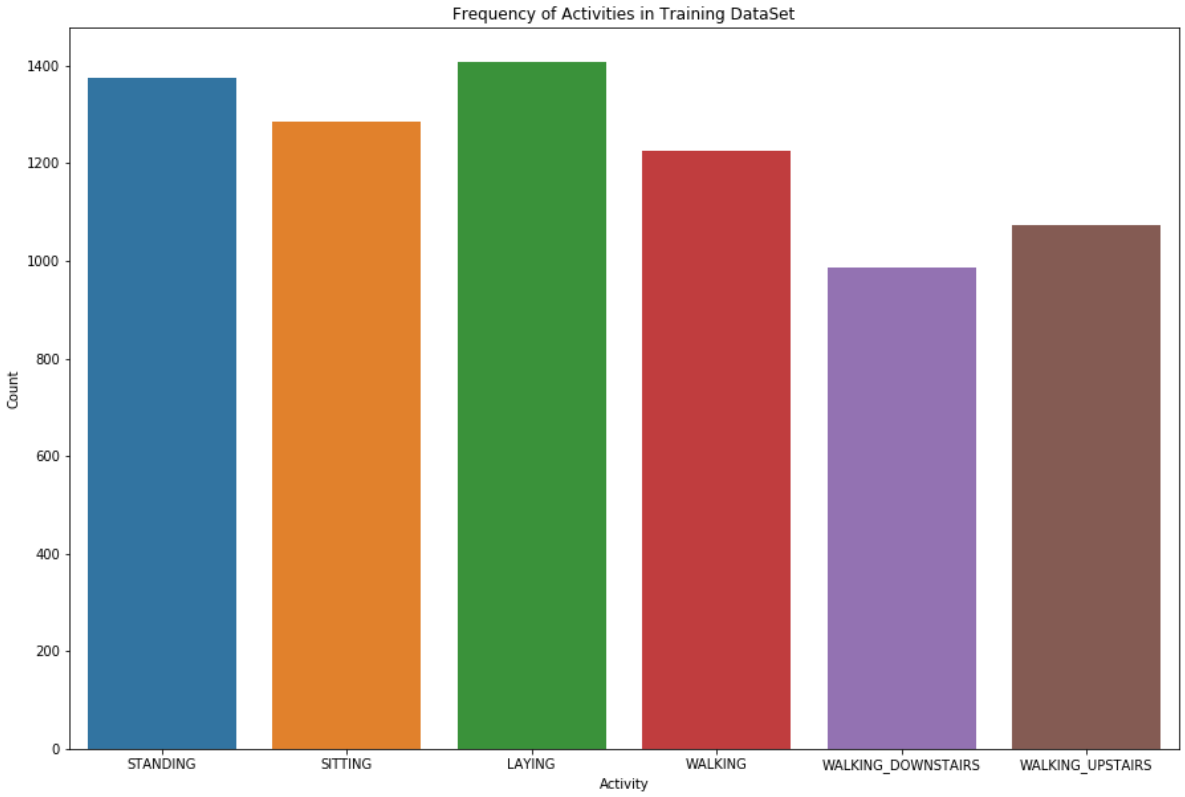

Fig 9. Frequency of Activities in Training Dataset

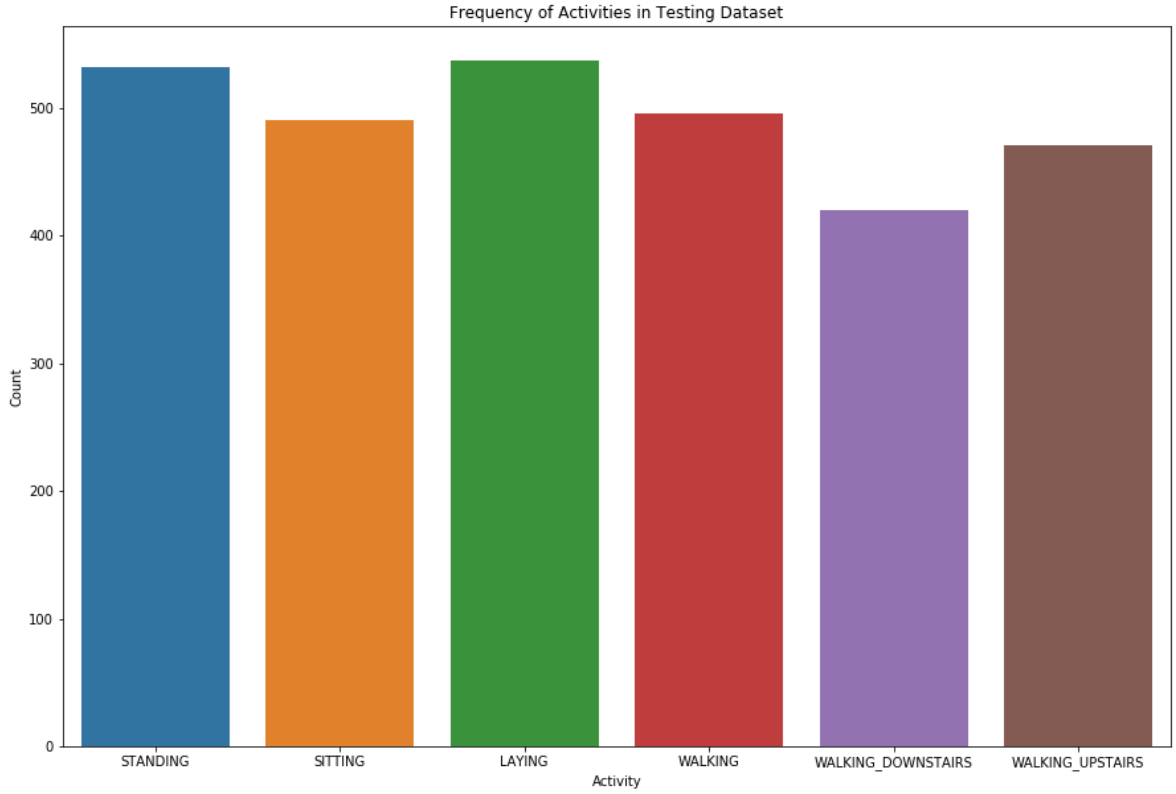

Fig 10. Frequency of Activities in Testing Dataset

The next set of analysis is carried out to understand the variation in data for each activity. The experiments were carried out on the data recorded only for subject 15. The feature of 'Mean Body Acceleration' along the X, Y, Z axis was documented as a scatter graph. The graph in Figure 11 and Figure 12 indicates that the mean value of the body acceleration is more variable for the activities of walking, walking upstairs and walking downstairs than the passive activities of sitting, standing and laying.

Fig 11. Mean Body Acceleration – X Axis Scatter Plot

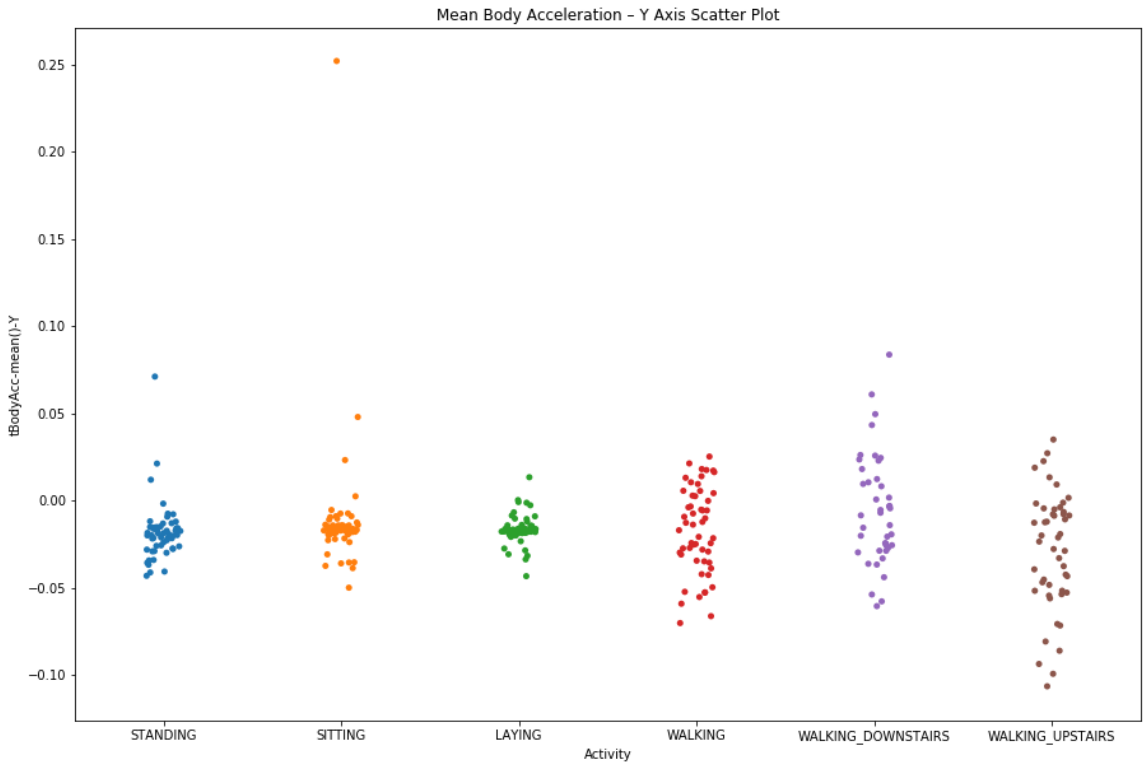

Fig 12. Mean Body Acceleration – Y Axis Scatter Plot

The maximum acceleration feature along the X, Y and Z axis for subject 15 is plotted and analyzed next. Analysis indicates that there is a clear distinction in the maximum values between the passive and active activities as all the passive activities fall below the active ones. The analysis from Figure 13 reveals that values along the X-axis can help us differentiate between walking, walking upstairs and walking downstairs but does not provide any insights into the passive activities. This provides a certain indication that the acceleration alone is not sufficient enough for ambulatory activity recognition but data from a different sensor such as a gyroscope would help in differentiating among the passive activities. Figure 14 is the graph plotted for feature 'Angle(X, GravityMean)'. The plot shows a clear distinction for the 'Laying' activity from the other classes. In a similar way, other features engineered provide important insights into recognizing human activities.

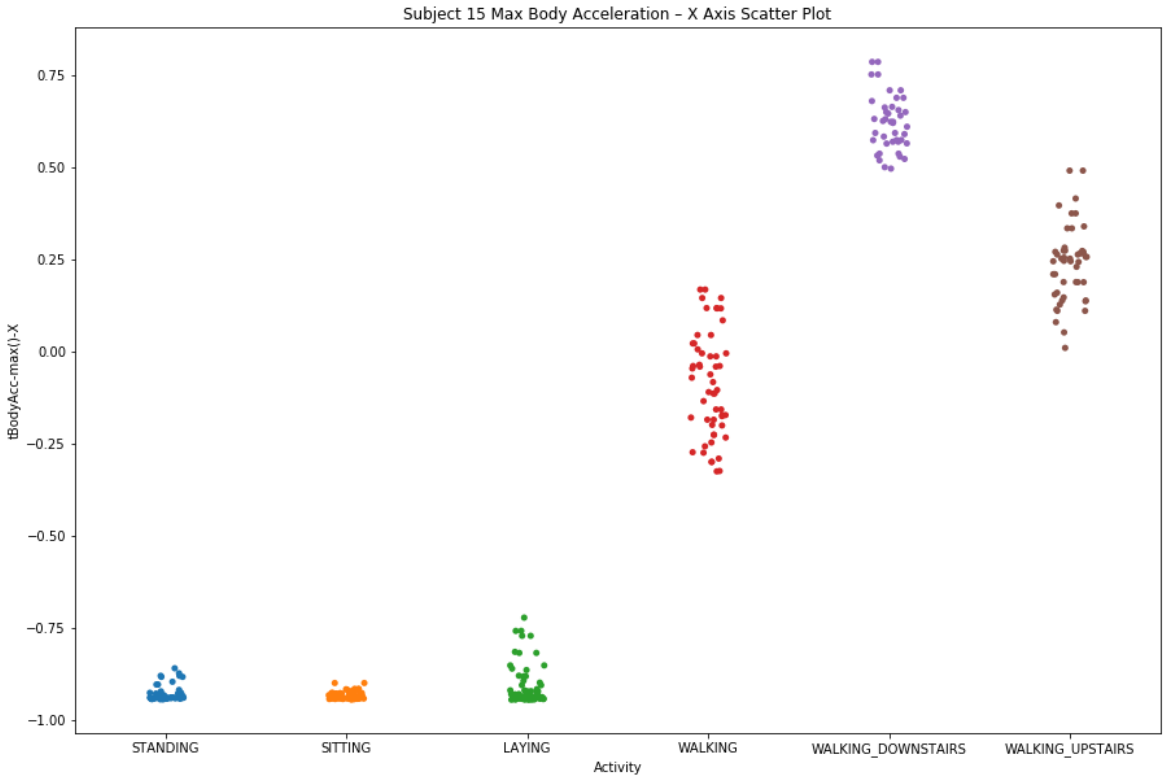

Fig 13. Max Body Acceleration – X Axis Scatter Plot

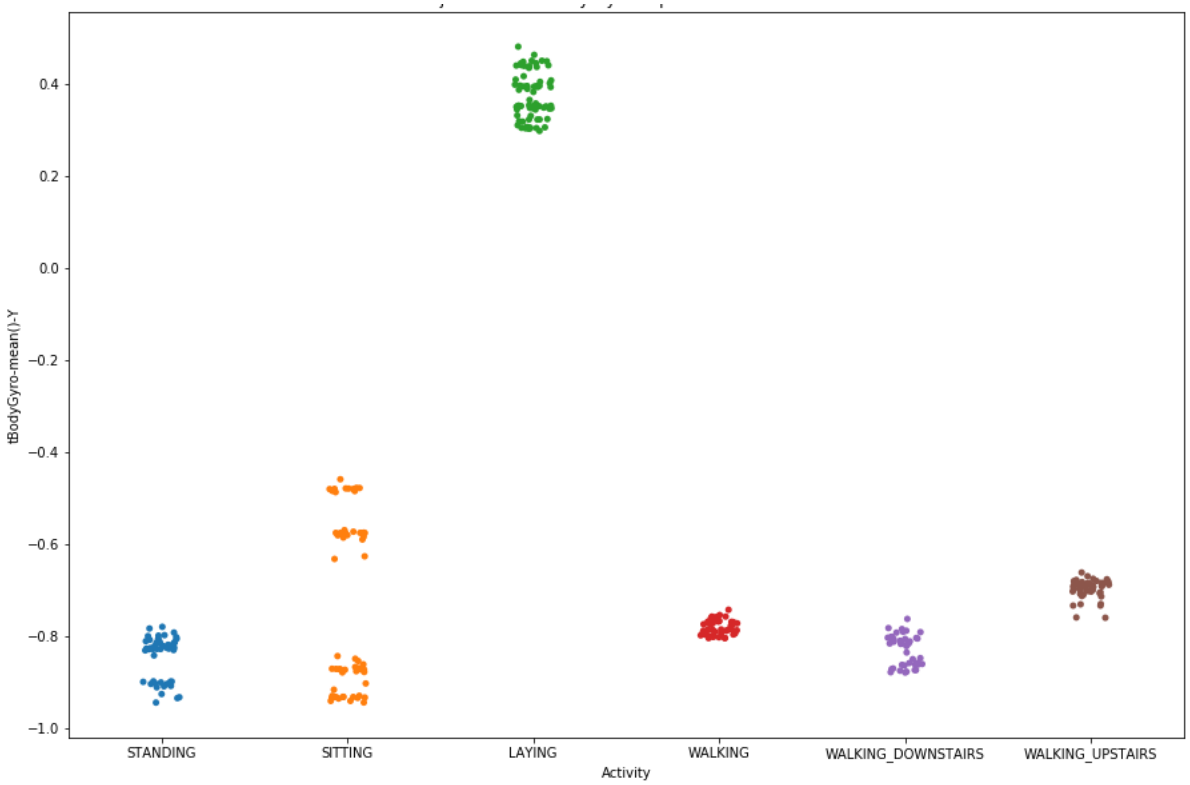

Fig 14. Angle (X, GravityMean) Scatter Plot

6.2 Machine Learning Model Evaluation

6.2.1 Feature Selection and Classification Model Evaluation

A comparative analysis was carried on four classifiers to understand the improvement in the model after the feature selection. The four classification algorithms experimented on are the Decision Tree Classifier, Random Forrest Classifier, Gradient Boosting Classifier, and the Artificial Neural Network. The results were compared to the metric of the time taken to build and train the model and the accuracy of the model. We use the tree-based feature selection and L1 based feature selection methods to reduce the dimensions of the feature dataset. From the 561 features from the original dataset, 91 features were selected by the Tree-Based feature selection method and 108 features were selected by the L1 based feature selection method. Table 2 lists the method of feature selection and the resulting feature dimensions.

Feature Selection Method	Features Selected
Tree-Based Feature Selection	Dataset shape (7352, 91)
L1 - Based Feature Selection	Dataset shape (7352, 108)

Table 2. Feature Selection Method and Resulting Dimensions

The L1 - Based feature selection keeps more features than the tree-based method and the execution time is far less than the tree-based feature selection method. For the L1 based feature selection, variable C had a value of 0.01. Tuning this parameter further could have enabled to select more distinct and important features.

To build the models of the Random Forrest Classifier and Gradient Boosting Classifier a value of 200 was set for the variable ‘n_estimators’. Upon experimenting with values of 100, 200, 300, 400 and 500 for the variable ‘n_estimators’ it was noticed that the accuracy did not increase after the value of 200 and hence it was considered as the base case for a set of

experiments. The artificial neural network was built using the Keras library using Tensorflow as its backend. The network was configured to have two hidden layers with 40 hidden units in each with the "ReLU" activation function for each neuron. The output layer was configured to utilize a 'Softmax' activation and the 'Adam' optimizer was used to boost accuracy. The model was compiled to run over 500 epochs with a batch size of 20 using 'Categorical Cross Entropy' as its loss function. Table 3 summarizes the accuracy of each model and Table 4 summarizes the build and prediction time for each classification model.

Model Accuracy Summarization			
	No Feature Selection	**Tree-Based Feature Selection**	**L1-Based Feature Selection**
Decision Tree Classifier	85.78%	82.91%	84.08%
Random Forrest Classifier	92.80%	89.71%	90.22%
Gradient Boosting Classifier	94.06%	92.29%	92.60%
Artificial Neural Network	94.77%	92.63%	93.41%

Table 3. Model and Accuracy Summarization

Model Build Time Summarization			
	No Feature Selection	**Tree-Based Feature Selection**	**L1-Based Feature Selection**
Decision Tree Classifier	4.279s	0.724s	0.759s
Random Forrest Classifier	21.44s	8.019s	8.733s
Gradient Boosting Classifier	252.89s	51.39s	56.10s
Artificial Neural Network	33.74s	22.79s	16.65s

Table 4. Build and Predict Time Summarization

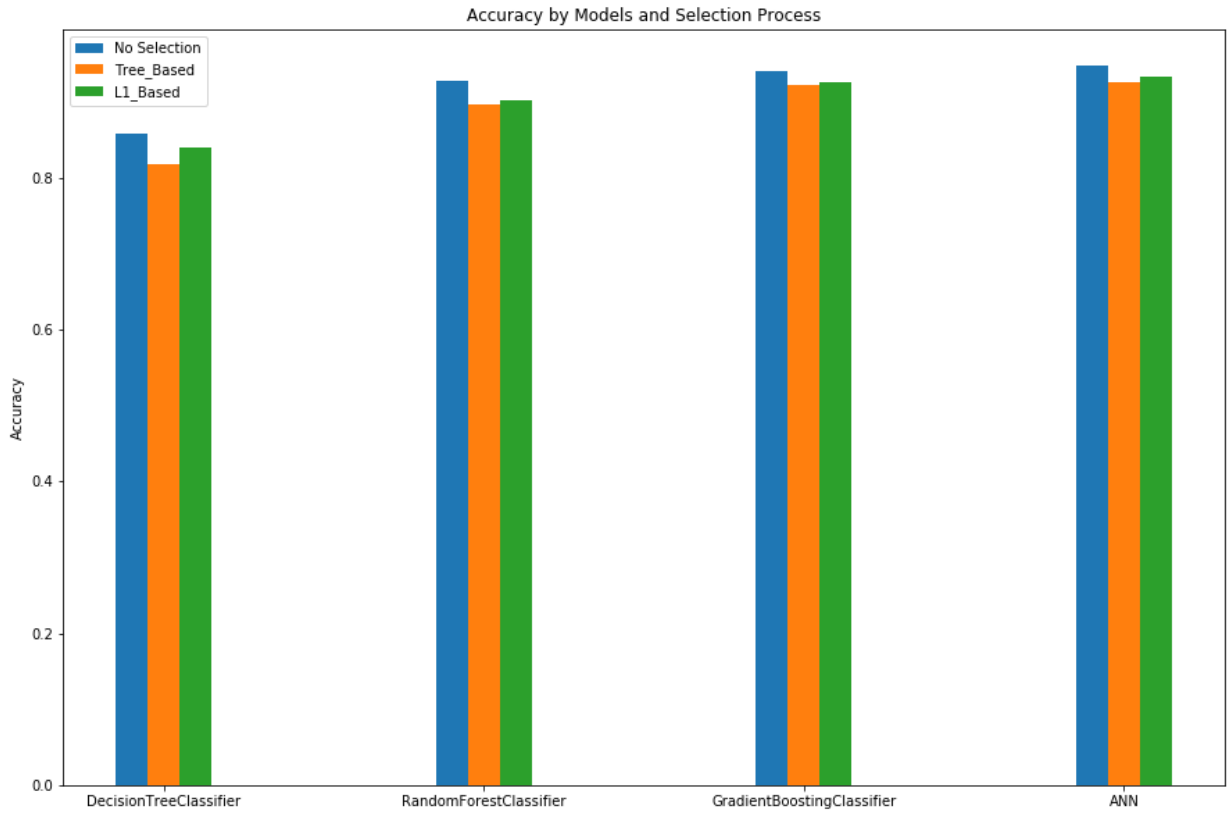

Fig 15. Accuracy by Models and Selection Process

The graph in Figure 15 indicates that with feature selection, the accuracy does not reduce considerably. This indicates that the overall accuracy of the model is not compromised by shrinking the size of the data set. The feature selection methods are sound techniques for selecting impactful features and also preventing overfitting. Improving and tuning the parameters of the feature selection techniques should provide the capabilities to improve performance too.

The graph in Figure 16 shows a decrease in time for building the model and predicting the values with the use of feature selection. This indicates an increase in efficiency, without endangering the accuracy of the model. For classifiers such as Decision Trees, there is not a drastic change in execution time. But for more complex models such as Gradient Boosting

Classifiers and Artificial Neural Network, there is a significant time difference with and without feature selection. Gradient Boosting Classifier works in a forward additive fashion with the features being randomly permuted and split at each stage. It continues the process till it improves on the loss function. The classifier build time reduces considerably when smaller set of the data containing the important features are inputted to the model. The model builds faster as it requires less iterations to improve accuracy. Similarly, for the artificial neural network, less misleading data would help improve on the loss function faster and a smaller input vector would require a smaller number of computations at each neuron. We do not see a critical time difference for Decision Trees as the model first calculates the information gain of each feature and splits the tree on those features where the information gain is the maximum. It carries out the same process for the feature selected dataset and hence we do not see a substantial improvement in build time.

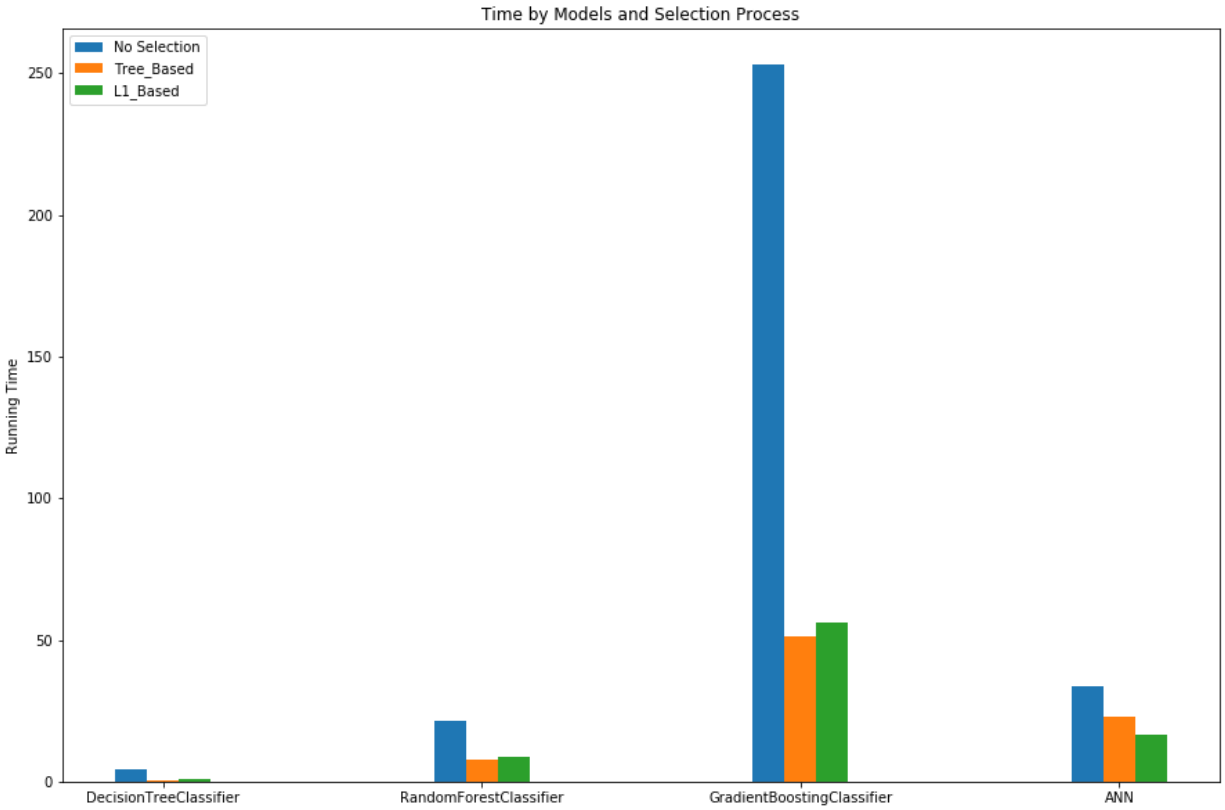

Fig 16. Time by Models and Selection Process

6.2.2 Long Short-Term Memory Network (RNN - LSTM):

An LSTM network model is developed to recognize human activity from the raw inertial signals instead of using feature engineered data. This experiment was carried out to test the ability of the network to learn features from a sequence of time series data and validate if it can recognize human activities from the features it extracts. The raw inertial signals are made of three main signals, body acceleration, total acceleration and body gyroscope with each attribute recording data over the three axes' (X, Y and Z). For a window of 2.56 seconds or 128 timesteps, 9 variables are recorded to give a total of 1152 elements (9*128) for each row of data. RNN-LSTM's are suitable for time series data as they have the ability to learn and remember over long sequences of data. The 'Smartphone Human Activity Recognition' dataset is a good fit for an LSTM network because it is intended to be used with sequences of data, up to 400 timesteps. Instead of manually engineering features, the RNN-LSTM learns from the time series signals directly and can achieve comparable results to models which are built on feature engineered data.

Each axis of each signal is stored in a different file, with a total of 9 files each for training and testing data. The training and testing data are loaded in a separate single three-dimensional NumPy array (samples, timesteps, features). The functionality of the NumPy library allows to stack features and create a single 3D array. The output variable, which is an integer representing one of the 6 activities, is one hot encoded to make it suitable to fit a multi-class neural network model.

The RNN-LSTM model is built using the Keras library and is defined as a Sequential Keras model. The model first has a single hidden LSTM layer which is used to extract features from the sequence of input data. A dropout layer is added to the model to reduce overfitting on

the training data. Next, a fully connected dense layer is added to the model which interprets the features followed by an output layer which gives out the final predictions of the human activity. 'Categorical Cross Entropy' is used as a loss function along with an 'Adam' optimizer to boost accuracy and optimize the network. 'Categorical Cross Entropy' is a commonly used loss function for classification tasks while the 'Adam' optimizer is well suited for large datasets as they are computationally efficient and require less memory. The 'ReLU' activation function is used in the dense layer with a 'Softmax' activation function for the output. The ReLU activation function overcomes the trouble of vanishing gradients and is hence preferred over Sigmoid and Tanh activation functions. The 'Softmax' activation function provides a probability distribution over all the classes and reacts well to low and high simulation. The model is run for a total of 15 epochs with a batch size of 64 samples. Generally, sequence data is not shuffled for an LSTM neural network, but for this particular experiment, the windows of time series data are shuffled. This is carried out as the focus is to learn features across time steps in a window and not across multiple windows.

As neural networks are stochastic, it is difficult to judge the evaluation from a single execution and hence the LSTM network is evaluated a total of 10 times. The results are summarized in Table 5. The model runs well, achieving an aggregate accuracy of 90.19% with a standard deviation of 0.994. The accuracy shows us the single-layered LSTM model evaluated over raw signal inputs is at par with other classification models which are built over feature engineered data. Table 6 provides a comparison chart of the accuracy achieved by different models built over the feature engineered dataset and accuracy achieved by the LSTM network over the raw inertial signal data. The LSTM model achieves greater accuracy than the original paper [26] which published an accuracy of 86% using a modified SVM classification algorithm.

RNN - LSTM Model Evaluation	
Evaluation Run	**Accuracy**
1	90.13%
2	88.70%
3	90.09%
4	91.42%
5	91.65%
6	89.41%
7	89.79%
8	89.72%
9	91.69%
10	89.31%
Aggregate	**90.19%**

Table 5. RNN-LSTM Model Evaluation over 10 runs.

Accuracy Comparison Chart				
Feature Engineered Dataset				**Raw Inertial Signals Dataset**
Decision Tree	**Random Forrest**	**Gradient Boosting Classifier**	**ANN**	**LSTM – RNN Network**
85.78%	92.80%	94.06%	94.77%	90.19%

Table 6. Accuracy Comparison Chart

7 Conclusion and Future Work

7.1 Conclusion:

In this research paper, we have presented the general architecture utilized to build human activity recognition systems and emphasized the design issues such as selection of sensors, obtrusiveness, flexibility, etc. which are independently evaluated based on the kind of system which is being developed. The paper further focuses on the importance of selecting important features from the data and provides a quantitative analysis of the metrics of execution time and accuracy. Tree-based and L1-based feature selection methods were utilized to select important features and were evaluated over four classification models. The results indicate that without a compromise in accuracy, the execution time and computational cost are greatly reduced with the use of feature selection methods. Better feature selection methods and improvement in tuning the parameters can assist further to improve accuracy and decrease computational cost. The research paper also provides a solution to reduce and eliminate the dependency of the requirement of domain knowledge to create hand-crafted features from the raw signals obtained from the sensor data. We have successfully shown that with the use of the LSTM network model, built to train on the sequences of raw inertial signals, features can be learned automatically by the network, and a significant accuracy is achieved. The accuracy achieved through the use of a recurrent neural network on raw signal data is at par with other classification models which are built on handcrafted features. Adding further layers to the network or increasing the complexity would further boost the recognition accuracy of the deep learning algorithm.

7.2 Future Work

For HAR systems to reach their full potential, more research is required. Comparison between HAR systems is hindered and becomes unquantifiable as each researcher uses a different dataset for activity recognition. A common public dataset would help researchers benchmark their systems and evolve the system altogether. Activities recognized in existing systems have been simple and atomic, which could be a part of more complex composite behaviors. Recognition of composite activities can enrich context awareness. There is also a great research opportunity to recognize overlapping and concurrent activities. Expanding on the work carried out on deep learning algorithms, one dimensional and two-dimensional convolutional neural networks, hybrids of convolutional networks and LSTMs should be further studied to determine their suitability to solve the problem of human activity recognition from raw signal data. Existing HAR systems are mainly focused on individual activities but could be extended further towards recognizing patterns and activity trends for a group of people with the use of social networks. Finally, recognition systems which could predict actions before they take place by the user could be a revolutionary development in certain applications.

References

[1] A. Perez, M. Labrador, and S. Barbeau, "G-Sense: A Scalable Architecture for Global Sensing and Monitoring," *IEEE Network*, vol. 24, no. 4, pp. 57–64, 2010.

[2] L. C. Jatoba, U. Grossmann, C. Kunze, J. Ottenbacher, and W. Stork, "Context-aware mobile health monitoring: Evaluation of different pattern recognition methods for classification of physical activity," in *30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 5250–5253, 2008.

[3] G. Tanaka, M. Okada, and H. Minemo, "GPS-Based Daily Context Recognition for Lifelog Generation Using Smartphone," in *(IJACSA) Int. Journal of Advanced Computer Science and Applications*, vol. 6, no. 2, 2015.

[4] M. Berchtold, M. Budde, D. Gordon, H. Schmidtke, and M. Beigl, "Actiserv: Activity recognition service for mobile phones," in *International Symposium on Wearable Computers*, pp. 1–8, 2010.

[5] D. Riboni and C. Bettini, "Cosar: Hybrid Reasoning for Context-Aware Activity Recognition," *Personal and Ubiquitous Computing*, vol. 15, pp. 271–289, 2011.

[6] T. Brezmes, J. Gorricho, and J. Cotrina, "Activity Recognition from Accelerometer Data on a Mobile Phone," in *Distributed Computing, Artificial Intell., Bioinformatics, Soft Computing, and Ambient Assisted Living*, vol. 5518 of Lecture Notes in Computer Science, pp. 796–799, Springer Berlin / Heidelberg, 2009.

[7] L. Bao and S. S. Intille, "Activity Recognition from User-Annotated Acceleration Data," in *Pervasive*, pp. 1–17, 2004.

[8] Y. Hanai, J. Nishimura, and T. Kuroda, "Haar-Like Filtering for Human Activity Recognition Using 3D Accelerometer," in *IEEE 13th Digital Signal Processing Workshop and 5th IEEE Signal Processing Education Workshop*, pp. 675–678, 2009.

[9] Z. He and L. Jin, "Activity Recognition from Acceleration Data Using AR Model Representation and SVM," in *Int. Conf. on Machine Learning and Cybernetics*, vol. 4, pp. 2245–2250, 2008.

[10] R. Olszewski, C. Faloutsos, and D. Dot, "Generalized Feature Extraction for Structural Pattern Recognition," in *Time-Series Data*. 2001.

[11] O. Lara, M. Labrador, "A survey on human activity recognition using wearable sensors", *IEEE Common. Surveys Tuts.,* vol. 15, no. 3, pp. 1192-1209, 2013.

[12] J. Kwapisz, G. Weiss and S. Moore, "Activity Recognition using Cell Phone Accelerometers," in *ACM SIGKDD Explorations Newsletter,* vol. 12, issue 2, Dec 2010

[13] C. Williams and J. Mathew, "An Architecture for Mobile Context Services," in *Lect. Notes in Electr. Eng.*, vol 313, Springer, Cham, 2015, pp. 61-68.

[14] Z. Wei, R.H. Deng, J. Shen, J. Zhu, K. Ouyang, and Y. Wu, "Multidimensional Context-Awareness in Mobile Devices," *MultiMedia Modeling: 21st Int. Conf. MMM 2015*, pp. 38-49, Jan. 2015.

[15] U. Maurer, A. Smailagic, D. P. Siewiorek, and M. Deisher, "Activity recognition and monitoring using multiple sensors on different body positions," in *Int. Workshop on Wearable and Implantable Body Sensor Networks*, (Washington, DC, USA), IEEE Computer Society, 2006.

[16] E. Tapia, S. Intille, W. Haskell, K. Larson, J. Wright, A. King, and R. Friedman, "Real-time recognition of physical activities and their intensities using wireless accelerometers and a heart monitor," in *Int. Symposium on Wearable Computers*, 2007

[17] J. Parkka, M. Ermes, P. Korpipaa, J. Mantyjarvi, J. Peltola, and I. Korhonen, "Activity classification using realistic data from wearable sensors," in *IEEE Trans. on Inf. Technol. in Biomedicine*, vol. 10, no. 1, pp. 119–128, 2006.

[18] A. Khan, Y.-K. Lee, S. Lee, and T.-S. Kim, "A triaxial accelerometer-based physical-activity recognition via augmented-signal features and a hierarchical recognizer," in *IEEE Trans. on Inf. Technol. in Biomedicine*, vol. 14, no. 5, pp. 1166–1172, 2010.

[19] S. Reddy, M. Mun, J. Burke, D. Estrin, M. Hansen, and M. Srivastava, "Using mobile phones to determine transportation modes," *ACM Trans. on Sensor Networks*, vol. 6, no. 2, pp. 1–27, 2010.

[20] O. D. Lara, A. J. Perez, M. A. Labrador, and J. D. Posada, "Centinela: A human activity recognition system based on acceleration and vital sign data," *Journal on Pervasive and Mobile Computing*, 2011.

[21] K. L. Huang, S. S. Kanhere, and W. Hu, "Preserving privacy in participatory sensing systems," *Computer Communications*, vol. 33, no. 11, pp. 1266–1280, 2010.

[22] I. J. Vergara-Laurens and M. A. Labrador, "Preserving privacy while reducing power consumption and information loss in lbs and participatory sensing applications," in *IEEE GLOBECOM*, 2011.

[23] M. Berchtold, M. Budde, H. Schmidtke, and M. Beigl, "An extensible modular recognition concept that makes activity recognition practical," in *Advances in Artificial Intelligence*, Lecture Notes in Computer Science, pp. 400–409, Springer Berlin / Heidelberg, 2010.

[24] O. D. Lara and M. A. Labrador, "A mobile platform for real-time human activity recognition," in *IEEE Conference on Consumer Communications and Networks*, 2012.

[25] C. Zhu and W. Sheng, "Human daily activity recognition in robot-assisted living using multi-sensor fusion," in *IEEE International Conference on Robotics and Automation*, pp. 2154–2159, 2009.

[26] D. Anguita, A. Ghio, L. Oneto, X. Parra, J. L. Reyes-Ortiz, "Human Activity Recognition on Smartphones Using a Multiclass Hardware-Friendly Support Vector Machine," in *Ambient Assisted Living and Home Care*, Lecture Notes in Computer Science, vol 7657. Springer, Berlin, Heidelberg, 2012

[27] J. Wang, Y. Chen, S. Hao, X. Peng, L. Hu, "Deep learning for sensor-based activity recognition: A survey," in *Pattern Recognition Letters*, vol. 119, pp. 3-11, Mar 2019.