CS205: Positioning Project Report

Team 1
Cyrus T
Akshay Raman
Akshay Shetty
Konark JS Kumar

Winter 2018

CONTENTS

INTRODUCTION

Fitness wearables include a wide range of products such as smartwatches, smart soles, smart bands and smartphones. Smartwatches occupy a majority of the market share in this space. The advent of Android wear has allowed a large number of companies to build fitness wearables. The most popular products among the smartwatches are Apple Watch, Sony SmartWatch and Samsung Gear. In addition to providing easy access to messages and emails, wearables can collect health vectors.

Today's smartwatches have accelerometers to measure the positioning and movement of the user. An accelerometer measures the acceleration it experiences relative to freefall and is the acceleration felt by people and objects.

Physical Activity is required to maintain good health and avoid several illnesses such as diabetes, early onset of heart attacks and cancers. The direct measurement of physical activity is helpful to determine the exact influence on the the health. There are several different methods to measure physical activity. The most convenient among them is the wristwatch technology. The accelerometer and gyroscope data from the watch helps to determine the current state of the user - whether he is active or idle. The use of smartwatches is again helped by low-cost, ease-of-use and wide audience. However, the data provided by smartwatches is raw and needs to be preprocessed and interpreted to understand the nature of physical activity of the user.

Machine Learning and deep learning are the currently the hot topics in the computer science world. The availability of huge data allows learning and understanding of the nature of the data. One example is that of a system diagnosis cancer based on the patients' symptoms. Neural networks and machine learning involve building a model that would classify a new object based on the analysis of the labelled dataset.

Neural networks provide us a better approach to classify data into different labels. With the availability of huge datasets and high computing power, neural networks are out performing machine learning models when it becomes difficult to select one particular list of features to build the model upon. Neural networks are typically organized in layers. Layers are made up of a number of interconnected 'nodes' which contain an 'activation function'. Patterns are presented to the network via the 'input layer', which communicates to one or more 'hidden layers' where the actual processing is done via a system of weighted 'connections'. The hidden layers then link to an 'output layer' where the answer is output.

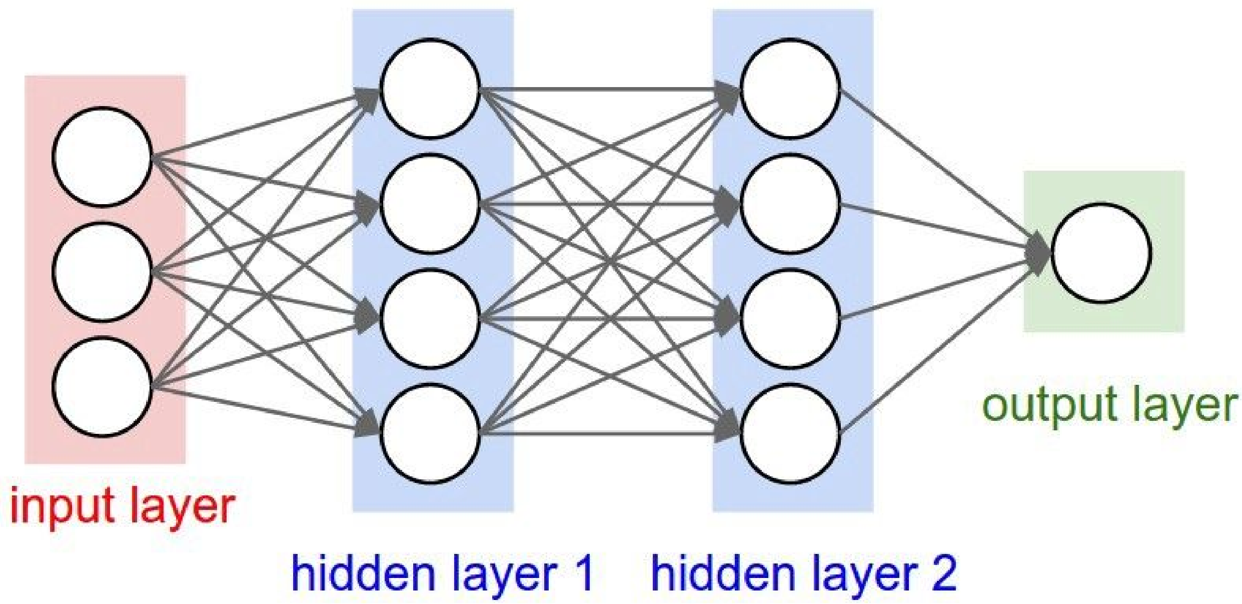


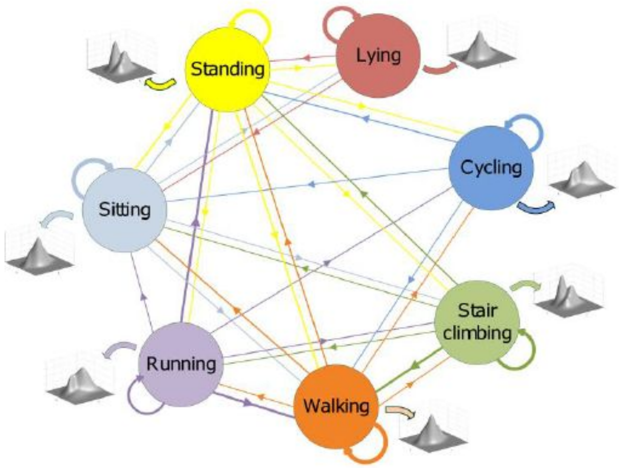
Figure: Simple illustration of neural network

Neural networks have been found to be particularly useful in scenarios when there is unlabeled or semi-labeled data. Neural networks are useful in the healthcare sector when we have lots of data about vitals (such as temperature, blood pressure) but do not know which features to select to give right output (predicting the disease).

LITERATURE REVIEW

To aid us as we worked on the project, our group read a few recommended papers that improved our understanding of existing systems that achieve similar goals as well as some of the techniques behind accurately estimating positioning based on sensors. Below, we summarize the ideas of two of the papers we found most relevant to our project as well as some recent news about technology to improve the battery life of wearable sensors.

The first paper, “Machine Learning Methods for Classifying Human Physical Activity from On-Body Accelerometers,” by Dr. Mannini and Dr. Sabatini[2], was very relevant to our project. The paper focused primarily on how accelerometers can be used to classify human physical activity into categories like standing, sitting, lying, walking, running, and cycling. The main motivation for accurate classification is that many methods to measure the metabolic energy expenditure of humans uses indirect methods that disregard the actual activity a person is doing. The information of the activity provides crucial context that can provide for more accurate estimation of energy expenditure. One of the most important parts to first consider is the sensor. As expected, wearable sensors should be small and lightweight so the user can partake in regular activities without feeling uncomfortable. One of the best in terms of cost, burden, and power consumption are microelectromechanical systems or MEMS sensors. They are being increasingly used due to their accuracy in capturing inertial motion while being unrestrained. With the sensor taken care of and accurate data being captured, it is important to embark on feature evaluation and feature extraction. Most machine learning algorithms do not work on raw data. There are also a series of transforms that occur that create a data representation in terms of feature variables. Choosing the right features is a crucial task and depends on the type of problem. Accelerometers measure the force being applied to a body as a function of the linear acceleration component and gravitational acceleration component projected along a particular axis. The high frequency component of the acceleration signal(AC component) is correlated with the dynamic motion a person is performing such as walking while the low frequency component(DC component) is correlated with gravitational influence so it can be used to classify static positions such as sitting. To measure features of interests, a window is used to capture measurements within a constant time frame. To get the DC component of acceleration, the signal is averaged from the data in each window. Each separate axis of the accelerometer provides a different window so the DC component indicates how a body is oriented in space with respect to gravity. Thus, the DC component is useful in the classification of static positions. Once the features have been evaluated, feature selection and extraction becomes the next task. As is well known, having too many features leads to the curse of dimensionality, reducing the tractability of classifiers. Thus, a good rule of thumb is to use as many features as needed for the problem. Identifying this minimal feature set is not trivial and depends on a variety of factors such as background and domain knowledge. A common practice is that the ratio of the number of training examples and the dimension of the features must be at least ten. As mentioned before, finding the optimal features to optimize the classification task is not easy as it requires searching through a large number of possible combinations. Today, feature selection usually involves approximate search algorithms such as branch-and-bound, sequential forward-backward selection, and the Pudil algorithm. The idea behind most of the above techniques is that features are added or removed and the effect is monitored by using k-NN to maximize the ratio between inter-class and intra-class distances across the possible feature subsets. Once the features are selected, feature extraction takes center stage to reduce the dimensionality of the data without losing any important information. The most popular technique is principal component analysis that transforms features into principal components through eigenvalue analysis. With the features extracted, it becomes time to apply classification. There are many possible models that can be used, and this paper explored the usage of Markov Models and Hidden Markov models to solve the task. The idea behind Marko Models is that the classifier takes into account decisions taken in the past for the current decision task. This makes sense because human activities are typically continuous. A person might walk for a bit, stand, and then sit. Thus, there is a wide window where the activity remains the same and it is helpful to incorporate past knowledge into the classifier. In particular, Hidden Markov Models, or HMMs, are used extensively in applications like speech recognition. However, they have rarely been used in human activity recognition. The HMM method is a mathematical approach to solving certain types of problems: (i) given the model, find the probability of the observations; (ii) given the model and the observations, find the most likely state transition trajectory; and (iii) maximize either *i* or *ii* by adjusting the model's parameters.



In particular, the paper developed a Gaussian cHMM which deals with continuous values and PDFs. The Gaussian means the Gaussian distribution is assumed for the parameters in the model. The diagram below shows the sequence of steps taken to classify sensor data into a class.

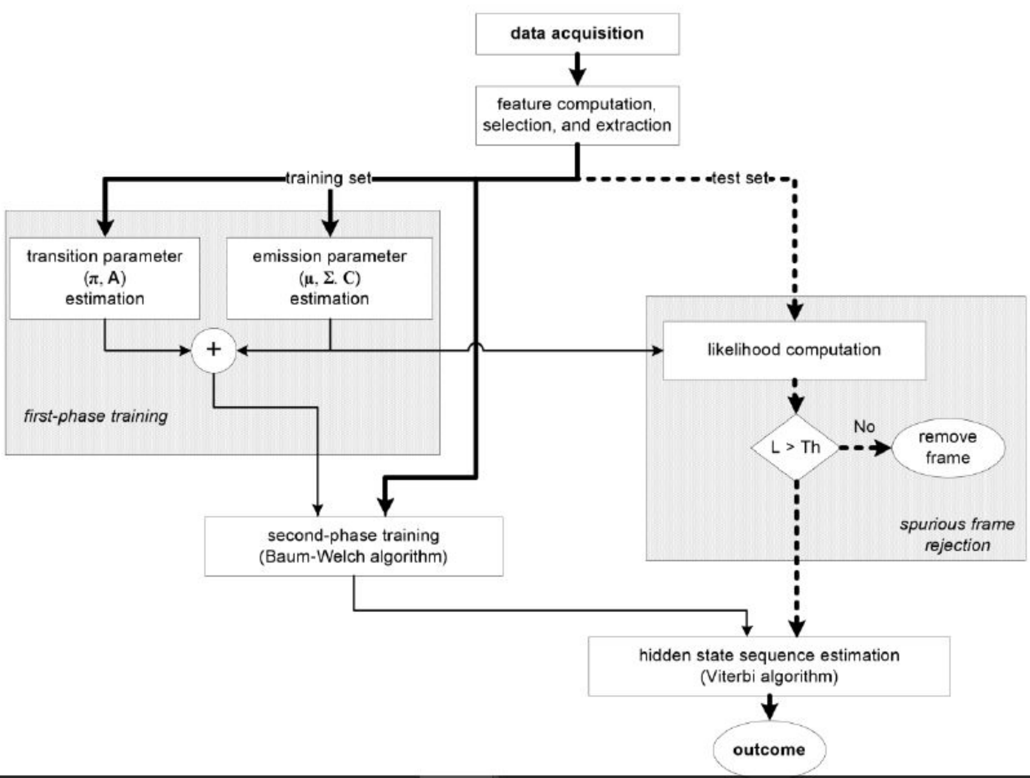


Figure 2: cHMM classifier(Mannini)

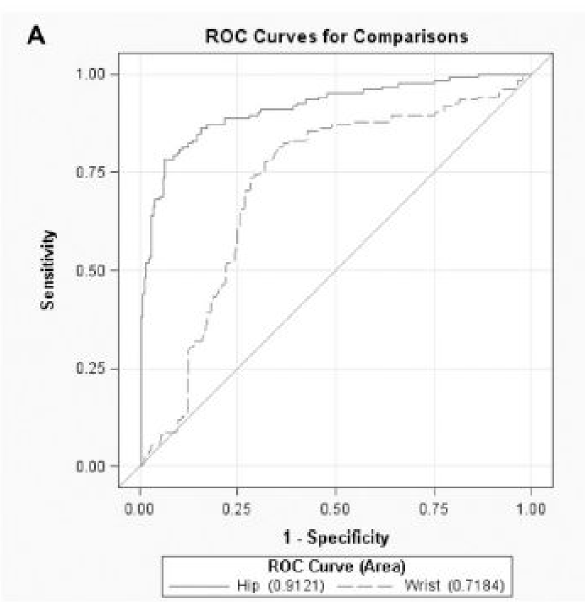
The results of the cHMM were very high, as can be seen in the below table. The two rows mean correspond the to accuracy after one phase of training and after two separate phases. The second figure below summarizes the accuracies of non-sequential classifiers.

Training	Classification accuracy, [%]
First-phase only	95.6
First and second-phase combined	98.4

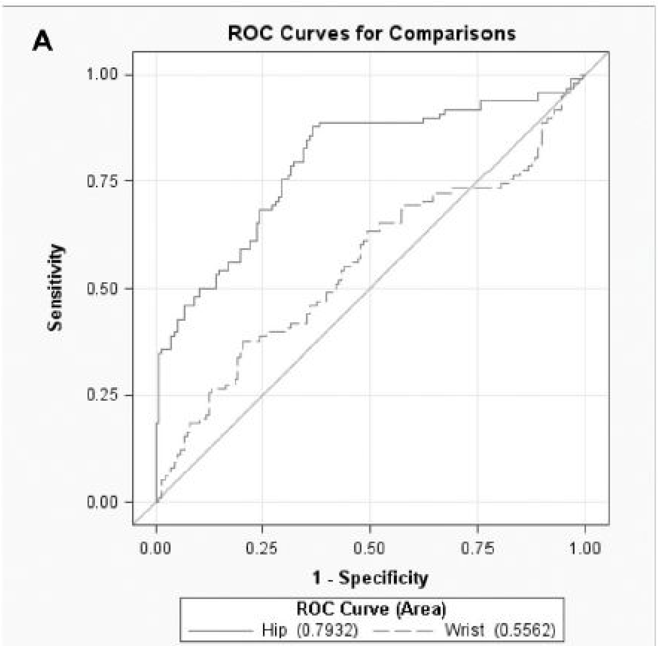
Classifiers	Classification accuracy, [%]
NB	97.4
GMM	92.2
Logistic	94.0
Parzen	92.7
SVM	97.8
NM	98.5
k-NN	98.3
ANN	96.1
C4.5	93.0

As you can see, the cHMM performed better with two phases of training than almost all the non-sequential classifiers. This shows the promise of sequential classifiers for activity recognition and even more research can be done to refine and improve algorithms and methodologies.

A second paper we read was “Estimating Activity and Sedentary Behavior From an Accelerometer on the Hip or Wrist” by Dr. Rosenberger et al.[4] from the Stanford Prevention Research Center at Stanford University. The goal of this paper was to measure physical activity intensity and type with an accelerometer worn on the hip and the wrist through a study. Thirty-seven adults participated in the study and wore Oxycon Mobile portable metabolic devices to measure energy expenditure of various activities through indirect calorimetry. The device is worn on the torso and measurements are made through lines attached rom the device to a mask worn over the subject's nose and mouth. After the data was collected, energy expenditure was calculated by measuring the carbon dioxide production and oxygen consumption. The accelerometer used was the Wockets system which is a device that contains a triaxial accelerometer sensor and communicates wirelessly to a phone through Bluetooth. The motion data from the accelerometer was captured at 90 Hz. The activities performed fell into four classes: sedentary, cycling, ambulation, and lifestyle. Sedentary activities include lying or sitting down. Cycling activities can mean riding a bike outdoors or exercise bikes. Ambulation activities include treadmills or walking. Lifestyle activities include folding laundry or sweeping the floor. All the participants performed at least 20 different activities during the study. All the activities were done for long enough to reach a metabolic steady state. The preprocessing of the data involved measuring the averages of the steady-state period for each activity while the coefficient of variation from the accelerometer measurement was estimated from the change in counts over time, summed across all three axes. The accelerometer raw data was converted to activity summary counts using the area under the accelerometer curve calculation(AUC) where a lowpass filter was applied to each axis using a moving average and then integrated to get the AUC for each axis. The techniques used to conduct statistical analysis involved using simple regression to compare the energy expenditure measurements for wrist and hip data, generating ROC curves through logistic regression, using mixed model analysis to create 2-regression models, and using leave-one-out cross validation to test the model on each subject. The results for identifying sedentary activities showed a specificity and sensitivity of 86% and 92% for the hip and 32% and 89% for the wrist. The below ROC plot for AUC from the paper shows the that the hip has a higher AUC and a higher AUC is preferable and indicates a better estimator for sedentary activities.



Similar results were achieved for identifying dynamic motion activities with the hip having a higher AUC than the wrist with a specificity of 78% and 49% for the hip and 65% and 40% for the wrist as can be seen in the below plot from the paper.



The main purpose of this paper was to compare hip worn accelerometers and wrist worn accelerometers and their ability to correctly identify classes for different activities. As shown above, hip accelerometers performed the best in identifying the correct classes and could be a more accurate way to improve energy expenditure measurement systems that rely on the correct classification for the best results. The effectiveness of hip accelerometers over wrist accelerometers in classifying sedentary and ambulatory activities was not surprising because many sedentary activities involve amounts of wrist movement. For example, when someone sits, they do not sit completely still. They may be writing something or typing on the computer. Using frequency domain features computed on short windows of 4 seconds improves the reliability of wrist accelerometers, but the hip accelerometer still performed 10% better. Further transformations like using wavelet features marginally improve the performance even more but are not worth the computational cost. In summary, hip accelerometers provide the best performance at activity classification.

Battery Optimization for Wearable Sensors

One of the main challenges in wearable sensors is battery life. What is the purpose of a high-end wearable if it only lasts a few hours? Improving the impact of these devices is contingent upon improving battery technology. At this years CES show, Bosch Sensortec unveiled the BMA400, an ultra-low power acceleration sensor for wearables and IOT devices that uses ten times less power than existing accelerometers, winning the 2018 CES innovation award[10]. According to Dr. Stefan Finkbeiner, CEO of Bosch Sensortec, “The BMA400 is the perfect solution for wearable and IoT applications that aim to extend battery lifetime without compromising on performance.” As accelerometers are often the most power-hungry component of wearable sensors, it is easy to see how this technology can be a game changer and enable a whole new possibility of applications.

ActLight, a startup based at EPFL in Lausanne, Switzerland, has developed a heart rate sensor that uses one-fifth of the energy of existing sensors[11] This is important because heart rate sensors take up around 80% of the battery life of many smartwatches. The secret to such optimizations lies in signal processing. Heart rate sensors in todays watches work with two diodes located on the back of the device, along the wrist that emit light that penetrates the upper layers of the skin, and blood flow determines how much light is reflected back. A sensor placed between the diodes detects these light waves and translates the information into electrical current, which is then converted into the pulse seen on the watch. However, instead of converting the light into a current and then measuring the amplitude, ActLight's dynamic photodiode sensors turn the current into light. The sensors use the pulse of light to identify the moment at which the current is triggered. A small reduction in energy consumption with every heartbeat, but repeated thousands of times per days, the energy savings add up to be considerable.

PROBLEM DESCRIPTION

The goal of this project is to develop a model to classify human activity and positioning of a user into one of the 4 labels: walking, standing, sitting or lying down. The project requires us to build a model that predicts the positioning of a user. The data is collected from a Sony SmartWatch using an app called “Decent Logger”. This collected data is used for training the models and building a system to classify unobserved data points. All four team members of this project collected data for each of these labels. Extracting features from the raw signal data is essential to improve our model. Using several machine learning such as Decision trees, SVM, Random forests and Fully Connected deep learning models, we are able to classify the activity on a test set.

SOLUTION

We have employed a fully connected neural network to build a model. We have collected about 5 hours of data with the labels “Sitting”, “Standing”,”Walking” and “Lying Down”. The data is preprocessed and divided into training and validation data in a 90:10 ratio. The training data is fed into a Fully Connected neural network and the model is trained to get the better network parameters. In order to get better generalization, the testing data was collected from a different member of the team and hidden from the training and used for evaluation only.

Sensor Data Features

We used data collected from the following sensors. These include both positional and rotational sensors.

Feature	**Use**	**Details**
Triaxial Accelerometer	Measure acceleration along gravity (m/s^2)	Calculated along x,y and z axis
Linear Accelerometer	Measures device acceleration excluding gravity(m/s^2)	Calculated along x,y and z axis
Gyroscope	Rate of rotation(rad/s)	Calculated along x,y and z axis
Gravity	Force of gravity(m/s^2)	Calculated along x,y and z axis
Orientation	Position sensor for orientation(degrees)	Pitch (angle around the x-axis),Roll (angle around the y-axis) and Azimuth (angle around the z-axis)
Rotation	Device rotation	Calculated along x,y and z axis

Data Processing

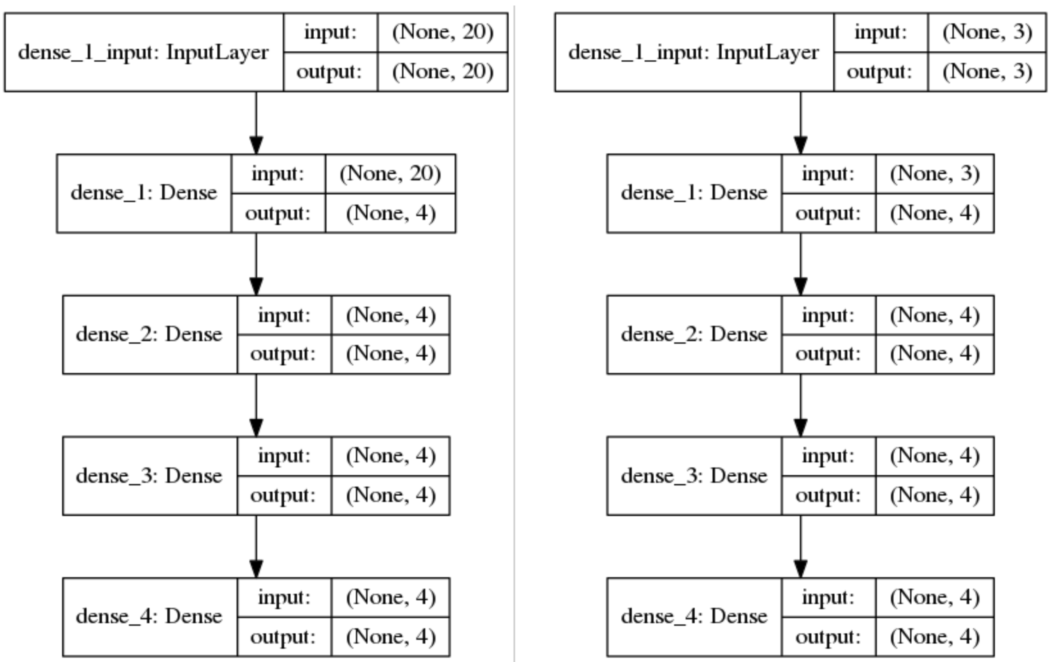
In the data processing step, we clean the data from the watch and convert it into .csv format. The data collected from the watch is stored in .gz format. First we uncompress all the files and then build the input data as a table with the first two columns representing the timestamp and label and the rest of columns represent the different sensor readings. The sensors include accelerometer, gravity, rotation_vector, orientation and gyroscope. Each row represents one reading recorded at that particular timestamp. We try to interpolate all missing timestamps (if possible) between two timestamps. This interpolation is carried out using the gradient between two existing data points. This helps us combine data from different sensors having different sampling rates. For instance, if we have data v_1 and v_2 from a sensor S at time t_1 and t_2 respectively, we can effectively compute the values for all timestamps between t_1 and t_2 using the gradient $(v_2-v_1)/(t_2-t_1)$. It might not be possible to interpolate all possible values for cases where we need data for timestamp t_a but we have data for that sensor only from t_b onwards and $t_b > t_a$. This makes it impossible to accurately compute a gradient since we do not have a lower bound (or upper bound) to go by. Fortunately, this is a very small percentage of cases so we can just prune out all rows that do not have timestamps. The data preprocessing script is agnostic to the number of sensors being used. The script can easily be configured to convert the watch into csv using as many sensors as the user wishes to train the model on.

Model Training and Validation

The preprocessed data is split into training and validation datasets in a 90:10 ratio. The dataset is labelled and correspond to walking(0), standing(1), sitting(2) and lying down(3) output labels. We have used a 3-layer Fully Connected neural network. In a fully connected layer, each neuron is connected to every neuron in the previous layer and each connection has its own weight. The three layers use 4 neurons each. The first 2 layers has ReLU activation. The output layer uses softmax activation to compute the probabilities for each of label. Adam is used as the optimizer for gradient descent and categorical cross entropy is used for loss calculation. The network is trained for 5 epochs with a batch size of 512. We have used k-fold cross-validation with 10 folds.

Model Testing

The testing data is provided in csv format. It will preprocessed and sent to the our neural network. In our case, we had reserved 90% of the dataset for training and 10% for validation. We have seen results of between 70-80% validation and test accuracy using our model and training data.



(a) (b)

Figure: Fully connected neural network model for (a) all sensor data, (b) accelerometer as input

Layer	**Output Shape**	**Number of parameters**
dense_1 (Dense)	(None, 4)	84
dense_2 (Dense)	(None, 4)	20
dense_3 (Dense)	(None, 4)	20
dense_4 (Dense)	(None, 4)	20
Total	Trainable parameters = 144	Non-trainable parameters = 0

Table: Parameters to the Fully Connected neural network model

Feature Extraction

Features can be defined as the abstractions of raw data as they represent reduced sets of original data(raw data) which highlight the main characteristics and behaviors of the signals. The feature extraction phase has a direct impact on recognizing human activity as the same activity can be performed in many different ways by people. A good feature set therefore should show little variation for repetition of same tasks performed by a particular individual but should vary between different people performing the same activity and it should also show considerable variations for different activities.

Keeping the above points in mind, On the collected data we used a sliding window of size 256 with an overlap of 50%. and captured various features for each sensor in each window. The features we extracted are listed below

- Mean
- Median
- Standard Deviation
- Interquartile range
- Root Mean Square
- Median Absolute Deviation

The above values are calculated for each of the x, y and z axis of the accelerometer. The same approach can be used on other sensors' data too.

- Pairwise correlation
- Autocorrelation

For the correlation pairs of axis were chosen and values obtained.

- Pearson correlation coefficient
- Fourier coefficient

FEATURE EXTRACTION

	min(x)	max(x)	mean(x)	mode(x)	median(x)	var(x)	sd(x)	iqr(x)	min(y)	max(y)	mean(y)
walking	-19.6	14.89	-8.71	-19.6	-8.61	13.2	3.63	4.18	-19.59	156.86	-3.33
standing	-19.6	11.13	-1.38	-8.83	0.65	24.95	5	9.98	-17.64	156.86	-6.1
sitting	-11.18	16	0	2.69	-0.23	7.5	2.74	5.18	-19.59	156.86	-3.08
lying down	-7.68	18.01	7.44	0	9.68	16.82	4.1	0.21	-7.31	10	1.24

	mode(y)	median(y)	var(y)	sd(y)	iqr(y)	min(z)	max(z)	mean(z)	mode(z)	median(z)	var(z)
walking	-2.34	-3.01	5.75	2.4	1.99	-19.6	18.08	3.16	2.83	3.03	4.54
standing	-7.5	-6.72	7.84	2.8	4.14	-9.85	17.53	-0.44	-6.61	1.03	25.34
sitting	-3.55	-3.17	7.12	2.67	2.36	-10.32	19.59	8.14	8.45	8.59	4.08
lying down	-0.11	-0.1	6.37	2.52	0.08	-11.31	16.15	-1.24	0.89	0.88	15.02

	sd(z)	iqr(z)	rms(x)	rms(y)	rms(z)	mad(x)	mad(y)	mad(z)	corr(x, y)	corr(y, z)	corr(z, x)
walking	2.13	2.23	9.43	4.11	3.81	2.75	1.48	1.48	0.03	0.24	0.07
standing	5.03	9.34	5.18	6.71	5.05	4.51	2.15	4.51	-0.55	0.48	-0.56
sitting	2.02	0.96	2.74	4.08	8.39	2.41	1.8	1.1	0.26	0.67	0.18
lying down	3.88	0.06	8.5	2.81	4.07	3.47	2.08	3.26	-0.96	-0.93	0.99

Feature normalization

The obtained features have different magnitudes. To avoid the case of features having higher magnitude to be given higher emphasis by the learning algorithms the feature set is normalized to zero mean and unit variance.

Feature selection

Not all features mentioned above are equally important for a specific activity and some of them maybe redundant or irrelevant. The feature selection phase chooses a smaller subset of the original feature set which is useful to identify informative features and also helps limit computational demands when running the learning algorithms. We performed Principal Component Analysis(PCA) which calculates the eigenvalues and eigenvectors of the covariance matrix of the calculated features with the eigenvalues sorted in descending order. We then performed Singular Value Decomposition to select the top 5 features where the eigenvectors with the highest eigenvalues are the principal components of our data set. The best features that PCA extracts can then be fed to our machine learning models.

Machine learning classifiers

In addition to the fully connected neural networks, we also created models based on SVM, Decision trees and Random forests to allows us to compare the accuracies between neural networks and machine learning techniques.

Support Vector Machines

SVMs are primarily a method that perform classification by constructing hyperplanes in a multidimensional space that separates cases of different class labels. SVM supports both regression and classification tasks and can handle multiple continuous and categorical variables. They learn weights in order to maximize the margin between classes by minimizing the hinge loss. The margin is a linear combination of the features and weights as the linear combination represents the distance between two points.

Decision Tree

Decision trees build classification or regression models in the form of a tree structure. They breaks down a dataset into continuously smaller sets while at the same time an associated decision tree is incrementally developed through the ID3 algorithm. The final result is a tree with decision nodes and leaf nodes. Each decision node has two or more branches while leaf nodes represent decisions, or in the case of classification, a class for the input.

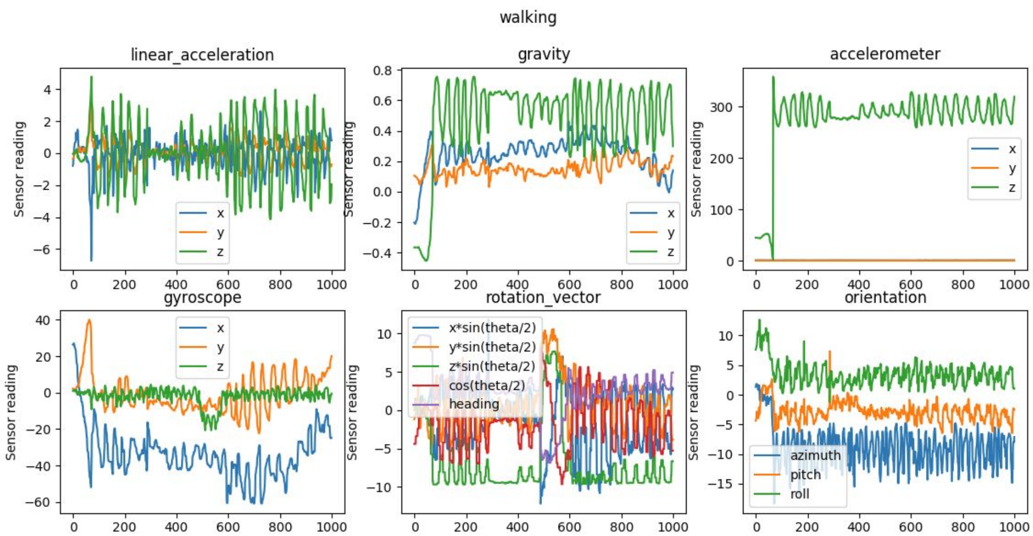
Random Forest

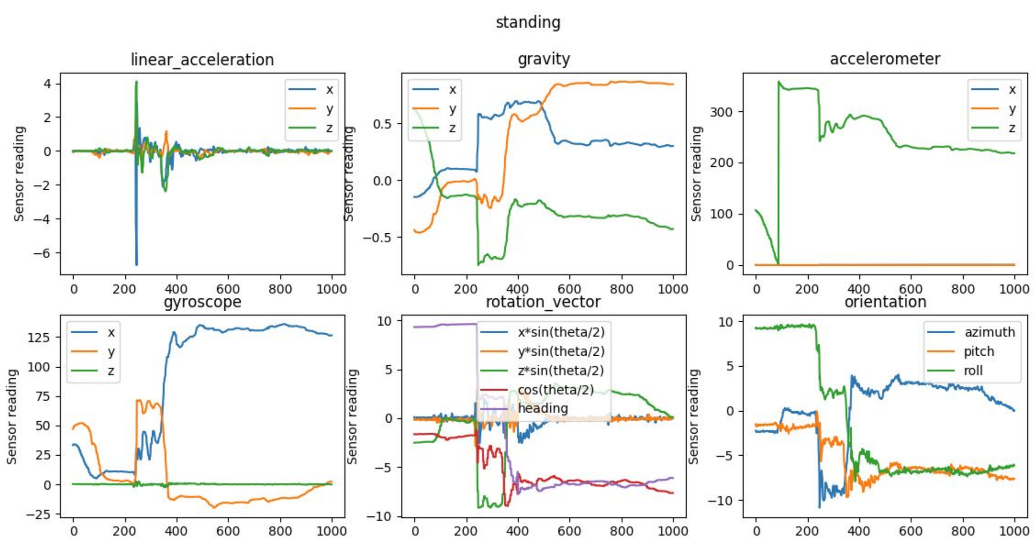
The random forest is an ensemble approach that can also be thought of as a form of nearest neighbor predictor. Ensembles are a divide-and-conquer approach used to improve performance. The main principle behind ensemble methods is that a group of weak learners can come together to form a strong learner. The random forest starts with a standard machine learning technique called a decision tree which, in ensemble terms, corresponds to our weak learner. The random forest is our strong learner. In a decision tree, an input is entered at the top and as it traverses down the tree the data gets put into increasingly smaller sets.When a new input is entered into the system, it is run down all of the trees. The result may either be an average or weighted average of all of the terminal nodes that are reached, or, in the case of categorical variables, a voting majority.

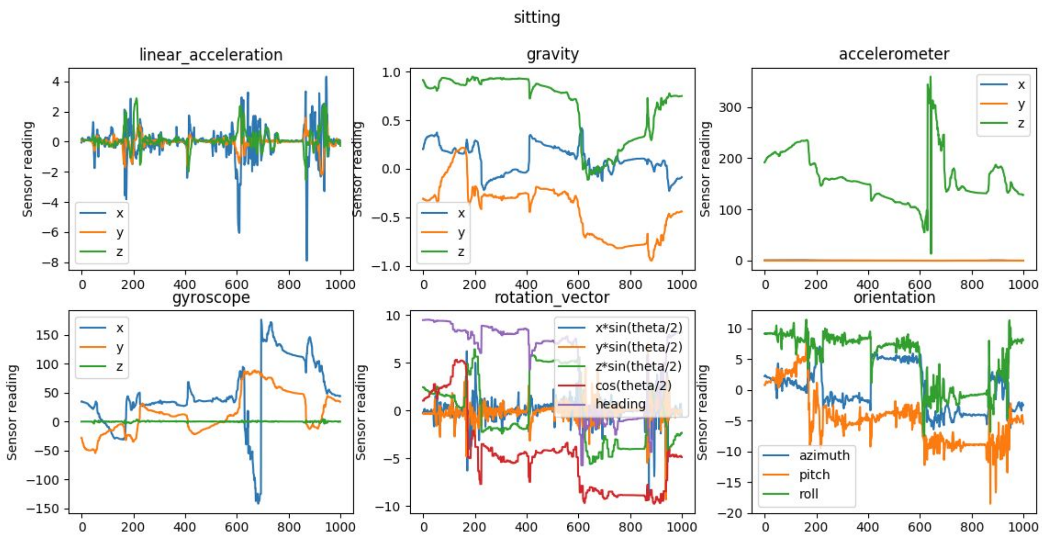
RESULTS

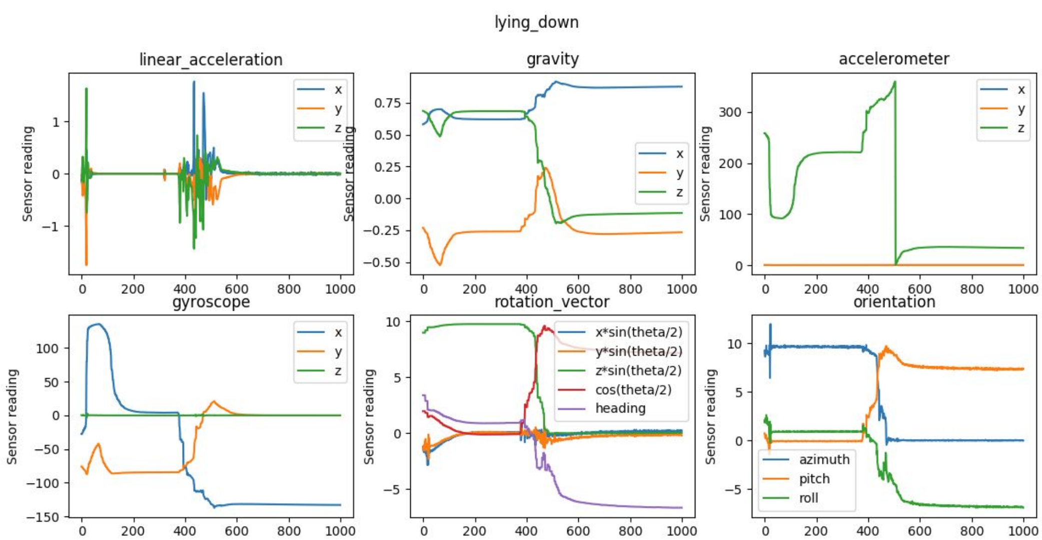
Raw data graphs

These graphs were plotted across 1000 timestamps (as opposed to using all the data) to make the plots more discernible.






sitting
linear_acceleration
gravity
accelerometer
gyroscope
rotation_vector
orientation
Sensor reading
x
y
z
x*sin(theta/2)
y*sin(theta/2)
z*sin(theta/2)
cos(theta/2)
heading
azimuth
pitch
roll


lying_down
linear_acceleration
gravity
accelerometer
gyroscope
rotation_vector
orientation
Sensor reading
x
y
z
x*sin(theta/2)
y*sin(theta/2)
z*sin(theta/2)
cos(theta/2)
heading
azimuth
pitch
roll

Deep Learning Models

The neural network model was first trained with all sensor data including linear_acceleration, gravity, accelerometer, gyroscope, rotation_vector and orientation. All four members of our team collected data for each of the labeled activities (standing, sitting, walking, lying down). The data from 3 members was merged to be our training dataset which we used to train our model. The performance was tested using the data collected by the fourth member.

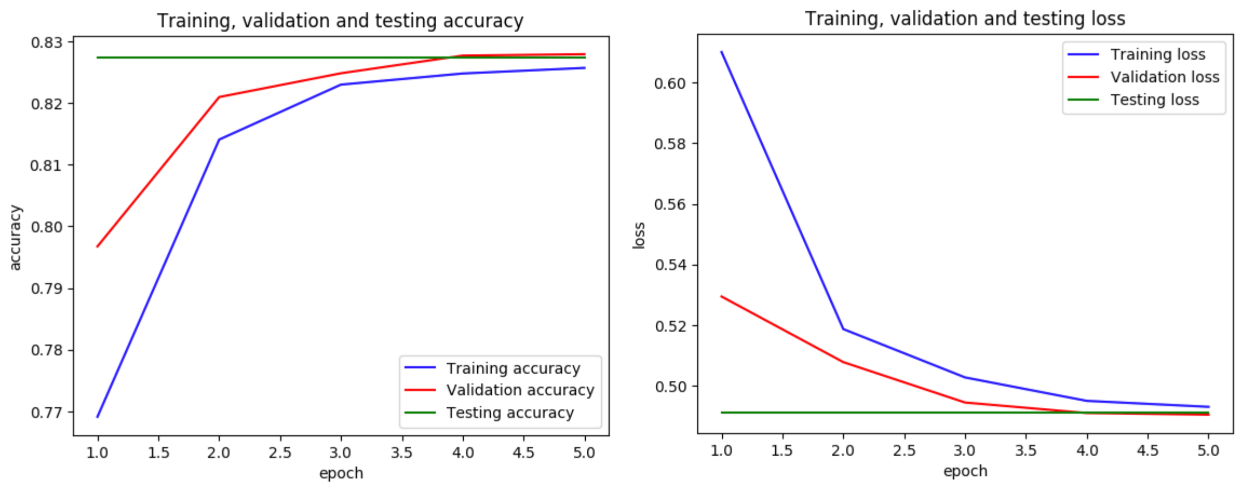


Figure : Training, Validation and Testing accuracy and loss using all sensors as features

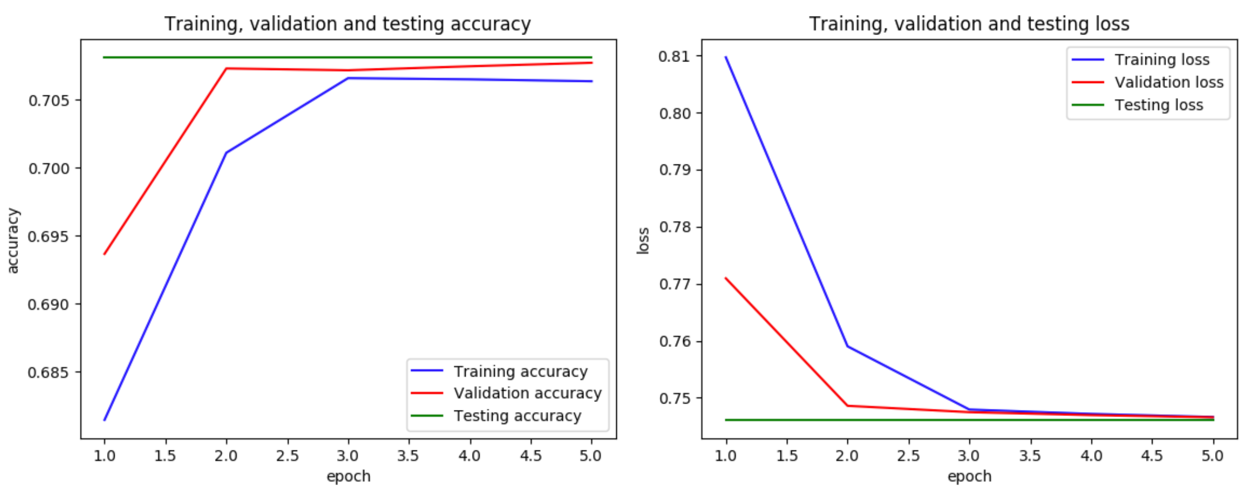


Figure: Training, Validation and Testing accuracy and loss using only the accelerometer

Confusion Matrix for the Fully Connected model

	Walking	**Standing**	**Sitting**	**Lying Down**
Walking	120847	47809	102112	273
Standing	183532	113302	43724	975
Sitting	91027	58439	971452	7494
Lying Down	40418	1298	310971	170

From the above confusion matrix, it is clear that the model does a reasonably good job of classifying walking, sitting and standing labels. The lying down label has room for improvement. Most misclassifications seem to be between walking-standing and sitting-lying down. This makes intuitive sense given that these activities have certain similarities between them.

Our model works with a window-size of 1 (by default), but this can easily be extended to detect activities for a range of timestamps. A time range with a large window will provide greater confidence in the output label when compared to a single timestamp. The window size is an optional command line argument that can be passed in the script.

We also ran SimpleRNN, LSTM and GRU models (4 hidden units each) on the accelerometer data for 3 epochs. All configurations used a stride size of 3. The pooling layers had a filter size of 3x3. For brevity, only the accuracy graphs are shown.

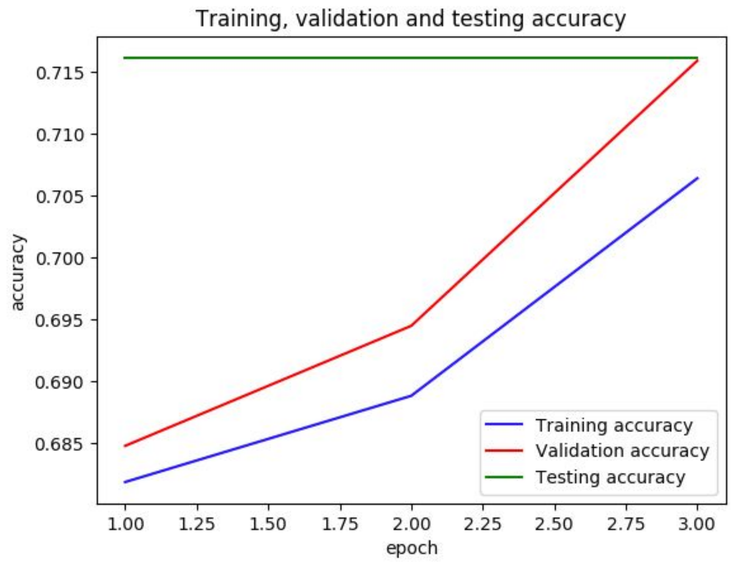


Figure: SimpleRNN accuracy graph

Confusion Matrix for the SimpleRNN model (only accelerometer)

	Walking	**Standing**	**Sitting**	**Lying Down**
Walking	596909	19387	97131	22098
Standing	205632	9657	118290	8728
Sitting	108204	14499	1272957	76071
Lying Down	200	5	159395	213700

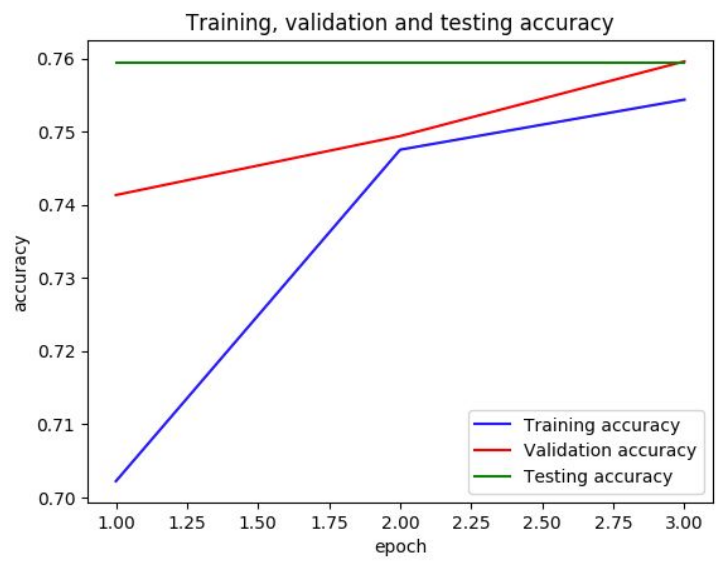


Figure: LSTM accuracy graph

Confusion Matrix for the LSTM model (only accelerometer)

	Walking	**Standing**	**Sitting**	**Lying Down**
Walking	575416	60556	102112	92202
Standing	46550	156642	135600	3515
Sitting	97871	47432	1262910	63518
Lying Down	507	12	148300	224481

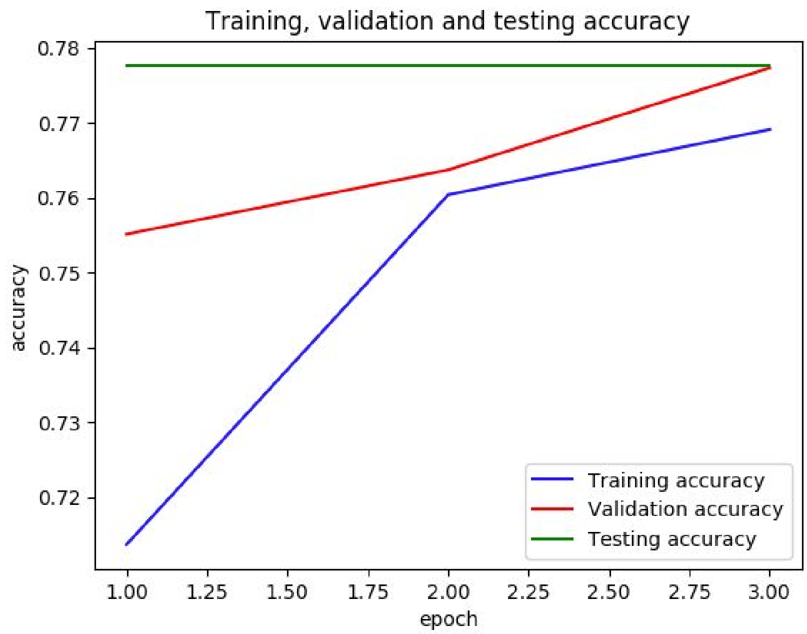


Figure: GRU accuracy graph

Confusion Matrix for the GRU model (only accelerometer)

	Walking	**Standing**	**Sitting**	**Lying Down**
Walking	585665	59928	86895	3037
Standing	70149	137813	129028	5317
Sitting	81387	44587	1291813	53944
Lying Down	959	546	114082	257713

Machine Learning Models

In addition, we compared our neural network with other classifiers like decision trees, support vector machines, and random forests.

Decision Tree Confusion Matrix

	Walking	**Standing**	**Sitting**	**Lying down**
Walking	82482	81366	15771	7990
Standing	4569	126992	52549	2959
Sitting	6568	46649	108991	25446
Lying Down	157	56415	56145	130529

SVM Confusion Matrix

	Walking	**Standing**	**Sitting**	**Lying Down**
Walking	131567	31887	16771	7384
Standing	10724	120799	51880	3666
Sitting	11253	40254	116824	19863
Lying Down	409	579	9539	177141

Random Tree Confusion Matrix

	Walking	**Standing**	**Sitting**	**Lying down**
Walking	186844	232	528	5
Standing	774	186016	275	4
Sitting	4212	1127	182251	65
Lying Down	34	25	117	187492

Result Summary

The results are summarized as follows(all accuracies are average after doing 10 fold cross validation):

Classifier	**Testing Accuracy**
Support Vector Machine	72%
Random Forest	99%
Decision Tree	95%
Fully Connected Neural Network	82%
GRU	78%
LSTM	76%
SimpleRNN	72%

CONCLUSION

The positioning project accurately identifies the activity of the person with high accuracy with the help of our neural network. The use of fully connected network allowed the weights to all the features. We tested different models using all sensor data and then using accelerometer data. Both the models almost gave the same accuracy. We have also tried modeling using SVM, Random Forest and comparing it against our neural network. While SVM uses only accelerometer data, the model gave a good prediction on the testing data.

We built an SVM model by using all sensor features. We achieved a total accuracy of 72% on the test set. The confusion matrix provides us a much more details on the performance. The misclassifications are quite frequent among sitting-lying down and standing-sitting pairs while there non-negligible misses among the other pairs too.

The Random forest model offered a better performance compared to the SVM model and this can be attributed to the fact that the model could use sensors other than accelerometer, feature selection is not required as the algorithm does this effectively on its own (Each tree is grown from a bootstrap sample of the training dataset and nodes are split by selecting the best among a randomly selected subset of features) and since each tree within the forest is grown independently using randomly selected features it ensures that model does not overfit the training data. Our observation while using random forest model was that, the best performance was observed when the model was trained and tested on individual data.

Deep learning architectures did not perform as well as traditional machine learning classifiers like random forest and decision tree due mainly to the limited data we had. If there was more data, it is possible these architectures would have been able to perform better. In addition, fully connected neural networks performed better than convolutional neural networks and recurrent neural networks as the latter two perform best on temporal and spatial data which is not necessarily the case for this project.

REFERENCES

1. Emmanuel Munguia Tapia, Stephen S. Intille, and Kent Larson. "*Activity recognition in the Home Using Simple and Ubiquitous Sensors*"
2. Andrea Mannini and Angelo Maria Sabatini. "*Machine Learning Methods for Classifying Human Physical Activity from On-Body Accelerometers*"
3. Shaopeng Liu, Robert Gao1, and Patty Freedson."*Computational Methods for Estimating Energy Expenditure in Human Physical Activities*"
4. Mary E. Rosenberger, William L. Haskell, Fahd Albinali, Selene Mota, Jason Nawyn and Stephen Intille. "*Estimating Activity and Sedentary Behavior From an Accelerometer on the Hip or Wrist*"
5. Andrea Mannini,, Stephen S. Intille, Mary Rosenberger, Angelo M. Sabatini, and William Haskell."*Activity recognition using a single accelerometer placed at the wrist or ankle*"
6. Alois Ferscha, Stefan Resmerita, Clemens Holzmann and Martin Reichor."*Orientation sensing for gesture-based interaction with smart artifacts*"
7. Tom White, Kate Westgate, Nicholas J. Wareham, Soren Brage. "*Estimation of Physical Activity Energy Expenditure during Free-Living from Wrist Accelerometry in UK Adults*"
8. Andrea Mannini, Mary Rosenberger, William L. Haskell,Angelo M. Sabatini, and Stephen S. Intille. "*Activity Recognition in Youth Using Single Accelerometer Placed at Wrist or Ankle*".
9. Deep neural networks by Ian Goodfellow. http://www.deeplearningbook.org
10. Kalnoska, Aimee. "4 μA accelerometer extends battery life in wearables and IoT applications". 2018
11. EPFL News. "A new sensor increases battery life by five times". 2017
12. Margarita Sordo, MSc, PhD. "*Introduction to Neural Networks in Healthcare*"