Activity Classification Using Device Accelerometer Data

1 Introduction

In this project I will discover if device accelerometer data can be used to determine the activity a person is doing. This problem has many real world applications such as Health and Fitness monitoring. This application can all ready be seen in action in devices such as the Apple Watch [1], Fitbit [4] and Garmin [3] smart watches. This feature will give the users retrospective analysis of there overall daily activities which is very useful for health and fitness monitoring. On top of this, it is also applicable to safety monitoring, the device data can be analysed to detect falls, crashes and other things that will impact the users safety. If an accident is detected, the device can be programmed to call emergency services.å

2 The Dataset and Prepossessing

The data set is the motion sense data set [5]. The data set is from a study by Queen Mary University of London where the researchers got 24 participants to perform 6 different activities: walking, jogging, standing, sitting, walking upstairs and walking downstairs. Each activity was recorded 3 times over 15 different trials, 9 of witch were long trials, 6 of witch were sort. Each action of each trial for each participant recorded their motion data using the CrowdSense app [2], which recorded the users acceleration (x,y,z), gravity acceleration (x,y,z), rotation rate (x,y,z) and attitude (pitch, roll, yaw). This lead to big collection of CSV files all composed of time series data with 12 total features.

First, the big collection of CSVs was compiled into one big data set, with an additional column for the activity the user was doing for that entry. Next, I combined the user acceleration and the gravity acceleration to get the total acceleration column and removed the user and gravity acceleration columns. On top of this, I also added a column that was the norm of the total acceleration. In order to make sure that this was necessary, I performed PCA on both data sets. This lead me to find that the total variance for the original data set could be explained by only 10 principal components meaning the data set could be reduced by 2 features. PCA on the modified data set showed the total variance was explained by 10 principal components, the same number of features the data set has, meaning the data set could not be reduced any further without loosing information.

After the modified data set was created, I split the data between long trials (1-9) and short trials (10-15) into train and test data respectively. For training neural network based models, the train data was further split using a stratified train test split of 80%/20% into train and validation data respectively.

3 Initial Results

As an early test, I fit 3 different supervised machine learning models with the train data. Those models were: Decision Tree, Random forest with 10 estimators and a neural network with 2 hidden layers of 64 and 32 neurons each both with relu activation and a softmax output layer. These initial models performed okay, with a range of results from 70-80% overall accuracy, calculated using 5-fold stratified cross validation.

Breaking down the results into individual categories showed that the model performed significantly worse on all but two categories: sitting and standing. This is due to the fact that sitting and standing do not in vole any motion, contrary to the other categories where the overall action cannot be inferred with only a stationary point in time, which is all the model can see at the moment.

4 Time Series Data

In order to help the model see desecrate time points instead of only stationary points in time, I chose to employ two methods: sliding windows and rolling features. Sliding windows in voles sliding a window of a fixed size over the data set and storing these big matrices as inputs to the model. This comes with the drawback that this method is only useful for models that can understand data in 2 dimensions such as convolutional neural networks or recurrent neutral networks and will also have significantly more features, meaning the model is a lot more complex

Rolling features in voles calculating a rolling mean, standard deviation and median of a fixed step for each feature of the data set. This allows the data to remain 1 dimensional and can still be used with our original models without loosing any information, and will have significantly less parameters than the sliding window method

5 Rolling Features Results

Using the same models from the initial tests but trained to the rolling feature data, the models saw an average results of approximately 90% on average using 5-fold cross validation. Miss classification of walking decreased from 30% to 10% due to the access to the time series data. The comparison results can be seen in figure 1.

6 Sliding Window Results

In order to process the 2 dimensional layer I had to use a convolutional neural network with 2 convolutional layers both containing dropout and max pool layers and 2 hidden layers both with L2 kernel regularizer and relu activation function and finally a softmax output layer. As well as this I also tested results with the same architecture but with LSTM and RNN cells instead of the 2 convolutional layers. These 3 models performed with a minimum of 94% accuracy tested using 5-fold stratified cross validation. These testes were done on a sliding window with size 150 and stride 10. Additional tests were done with the CNN only, testing the window sizes of 100, 150, 300 and 500. Each increase in window size increased the accuracy with the window size of 500 reaching 97%, the highest accuracy achieved in this project. Since the sliding window data is more complex, it allows the models to be more complex without over fitting the data. This means there is still room for improvement in fine tuning the architecture of the CNN, RNN and LSTM models to get even higher results.

7 Limitations

The motion sense data set was relatively small, not diverse and not real world. This posed big limitations on what the models could do. The size of the data set made over fitting a big concern, L2 regularizers and dropout layers were used to mitigate this. The data not being real world also made this project unusable in real world situations where users will be changing activity rapidly, situations the models were not trained for. The data set not being diverse enough could mean that the models would not know how to classify activity recorded by people who were not one of the 24 participants as there might not be enough data to generalize the patterns found.

8 Conclusion

This project successfully showed that users device accelerometer data can be used to determine the type of activity they are doing with the most accurate model achieving up to 97% accuracy on the testing data set. The project also outlined a computationally cheaper method of rolling features which saw accuracy of up to 96% using a Random Forest with only 10 estimators, which would be more suitable for situations where computational power is limited, such as smart watches.

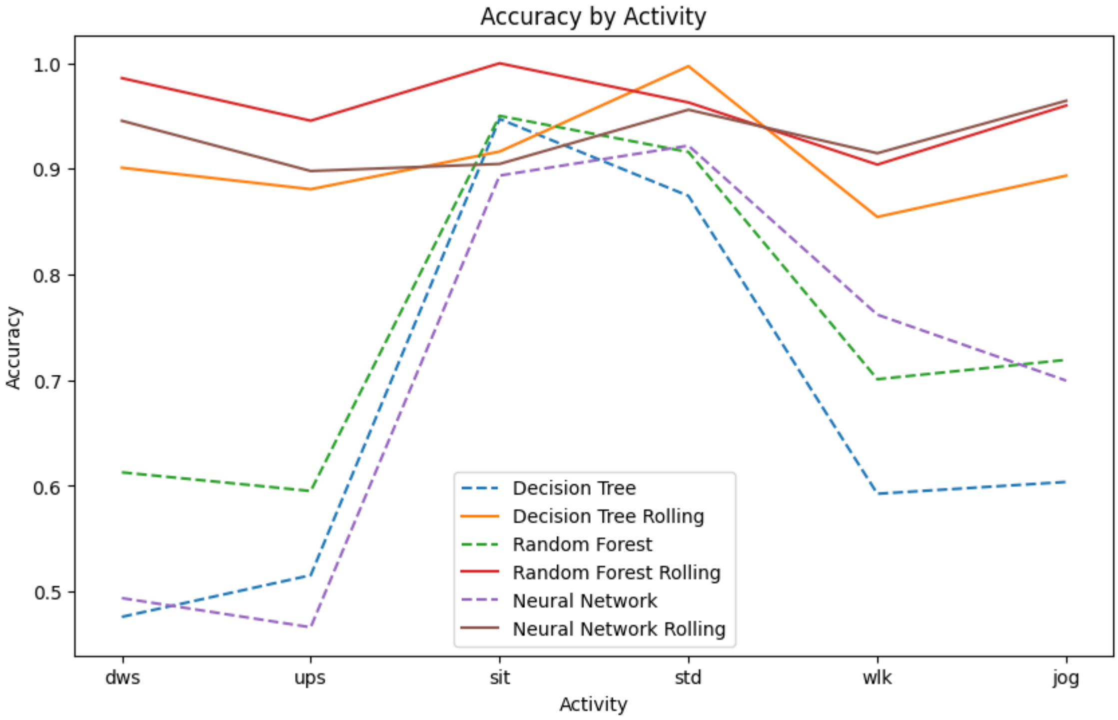

Figure 1: The figure shows the results of the initial models before and after rolling features were added.

References

[1] Buy apple watch - apple (uk). https://www.apple.com/uk/shop/buy-watch. (Accessed on 12/05/2023).

[2] Crowdsense on the app store. https://apps.apple.com/gb/app/crowdsense/id930853606. (Accessed on 12/05/2023).

[3] Garmin | united kingdom | home. https://www.garmin.com/en-GB/. (Accessed on 12/05/2023).

[4] What's new from fitbit. https://www.fitbit.com/global/uk/products/whats-new?utm$_medium = semutm_source = googleutm_campaign = UK_PF_ROASgad_source = 1gclid = CjwKCAiAjrarBhAWEiwA2qWdCPbGbtRqFOsY3KNi82Z9HovZa9G24Dny9XB_8VVcafjlb3BXz$ $aw.ds.(Accessedon12/05/2023)$.

[5] Mohammad Malekzadeh, Richard G. Clegg, Andrea Cavallaro, and Hamed Haddadi. Mobile sensor data anonymization. In *Proceedings of the International Conference on Internet of Things Design and Implementation*, IoTDI '19, pages 49–58, New York, NY, USA, 2019. ACM.