Predicting Activity Based on Device Accelerometer Data

ECM3420 Coursework

Ethan Hofton

University of Exeter

eh736@exeter.ac.uk

December 4, 2023

Research Question

Can the type of activity people are doing be inferred from device accelerometer data?

Why should anyone care about this?

- Health and fitness monitoring
- Safty and wellbeing
- Environmental monitoring

Real world applications:

- Fitness and health trackers
- Emergency response applications

The Dataset [1]

- attitude (roll, pitch, yaw)
- gravity (x, y, z)
- rotationRate (x, y, z)
- userAcceleration (x, y, z)
- Time series data

- 24 participants
- 6 activities (walking, jogging, upstairs, downstairs, sitting, standing)
- 15 trials, 9 long (3 mins each), 6 short (30 secs each)
- 50Hz sampling rate

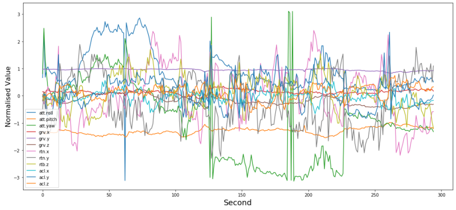


Figure: A sample from the dataset [2]

Machine Learning Methods

Predicting Activity

- Classification Problem
- Supervised Learning
- Multi-Class (DWS, UPS, SIT, STD, WLK, JOG)

Feature Selection

- PCA for dimensionality reduction
- Unsupervised Learning

Data Cleaning

- Data is already clean
- The data can be split into train test sets
    - The long trails (1-9) are used for training
    - The short trails (10-15) are used for testing
    - For NN models, the train data will be split 80/20 into train and validation sets

Feature Engeneering

User acceleration and gravity are combined to get the total acceleration. This reduces the dimensions of the data.
The norm of the total acceleration is also calculated. This is used as a feature for the NN models

Data Preprocessing
Dimentsionality Reduction

Combine gravity and user acceleration and remove the original columns. Is this a good idea?

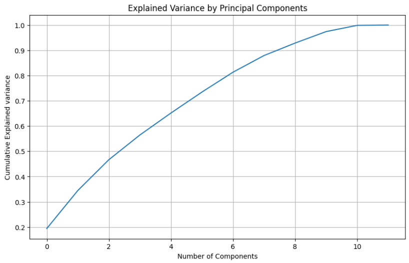


Figure: PCA on the data

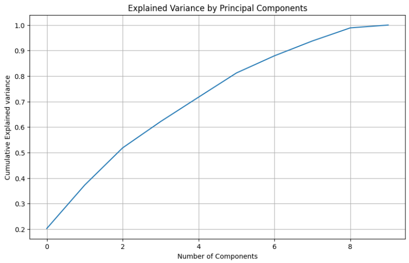


Figure: PCA on the data with gravity and user acceleration combined

- Decision Tree
- Random Forest
- Neural Network
- Convolutional Neural Network
- Simple Recurrent Neural Network
- Long Short Term Memory

Data Analysis

Model Architecture

Decision Tree

Standard decision tree with no hyperparameter tuning

Random Forest

10 estimators

Neural Network

- 10 neuron input layer
- 1 hidden layer with 64 neurons, relu activation
- 1 hidden layer with 32 neurons, relu activation
- 1 output layer with 6 neurons, softmax activation

Data Analysis

Early Results

Model	**DWS**	**UPS**	**SIT**	**STD**	**WLK**	**JOG**	Total
Decision Tree	48.1	51.7	94.7	90.3	59.2	60.5	76.7
Random Forest	62.7	59.1	94.9	92.6	70.4	71.7	82.2
Neural Network	49.4	46.4	89.4	92.2	76.2	70.0	79.4

Table: Results of the models on the test dataset

Results

These results could be better. The models do not take into account the time series nature of the data.

Data Preprocessing

Time Series Data

Windowing

The input data is slid over in windows of size 150 (3 seconds). The stride of the window is 10 (0.2 seconds). The resulting window is a 150x10 matrix. This data can be passed to a neural network model suited to time series/2 dimensional data like CNN, RNN.

Rolling Features

Rolling features involves calculating a rolling mean, standard deviation and median for each feature of the data. This allows the data to be passed to a model that is not suited to time series data like a decision tree or random forest without losing the time series nature of the data.

Data Analysis
Rolling Features

Model	**DWS**	**UPS**	**SIT**	**STD**	**WLK**	**JOG**	Total
Decision Tree	90.1	88.1	91.6	99.7	85.4	89.3	91.9
Random Forest	98.6	94.6	99.9	96.3	90.4	96.0	96.3
Neural Network	94.6	89.8	90.5	95.6	91.5	96.4	92.7

Table: Results of models on the rolling features dataset

Data Analysis
Rolling Feature Results

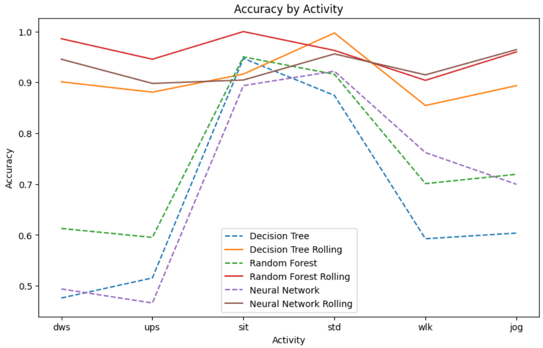
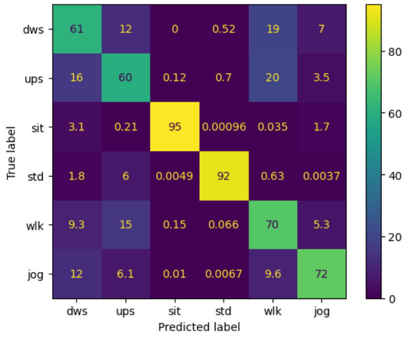


Figure: Results before and after rolling features are applied

- Time series data is not important for sitting and standing (no movement)

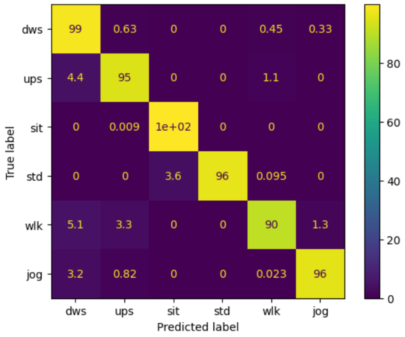
Data Analysis

Confusion Matrix



True \ Predicted	dws	ups	sit	std	wlk	jog
dws	61	12	0	0.52	19	7
ups	16	60	0.12	0.7	20	3.5
sit	3.1	0.21	95	0.00096	0.035	1.7
std	1.8	6	0.0049	92	0.63	0.0037
wlk	9.3	15	0.15	0.066	70	5.3
jog	12	6.1	0.01	0.0067	9.6	72

Figure: Random Forest confusion matrix before rolling features



True \ Predicted	dws	ups	sit	std	wlk	jog
dws	99	0.63	0	0	0.45	0.33
ups	4.4	95	0	0	1.1	0
sit	0	0.009	1e+02	0	0	0
std	0	0	3.6	96	0.095	0
wlk	5.1	3.3	0	0	90	1.3
jog	3.2	0.82	0	0	0.023	96

Figure: Random Forest confusion matrix after rolling features

Model Architecture

CNN Architecture

- 1D Convolutional Layer, 64 filters, 3x3 kernal
- Max Pooling Layer
- Dropout layer, 0.2 dropout rate
- 1D Convolutional Layer, 32 filters, 3x3 kernal
- Max Pooling Layer
- Dropout layer, 0.2 dropout rate
- Flatten Layer
- Dense layer, 64 neurons, L2 regularization, relu activation
- Dense layer, 32 neurons, L2 regularization, relu activation
- 6 neuron output layer, softmax activation

LSTM and RNN followed the same Architecture but with a single LSTM or RNN layer instead of the CNN layers

Data Analysis

Sliding Window Results

Model	**DWS**	**UPS**	**SIT**	**STD**	**WLK**	**JOG**	Total
CNN	95.3	90.4	95.2	95.9	92.9	97.8	94.7
LSTM	96.8	92.7	95.3	92.4	92.9	95.3	94.0
RNN	94.0	91.7	95.3	94.6	93.7	89.4	93.9

Table: Results of models on the windowing features dataset

Data Analysis

Adjusting the side of the window

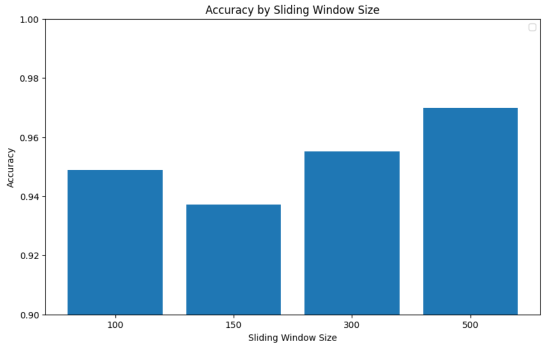


Figure: Results of the CNN model with different window sizes

Data Analysis
2s vs 10s Windows

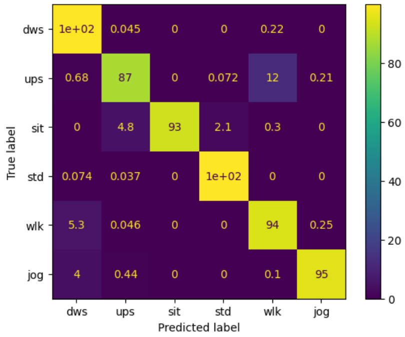


Figure: CNN confusion matrix with 2s windows

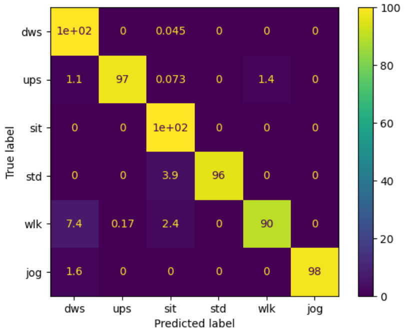


Figure: CNN confusion matrix with 10s windows

Data Analysis

Rolling Features vs Windowing

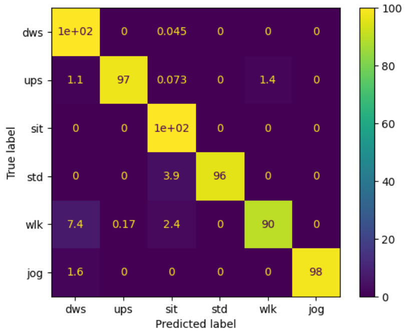


Figure: CNN confusion matrix with 10s windows

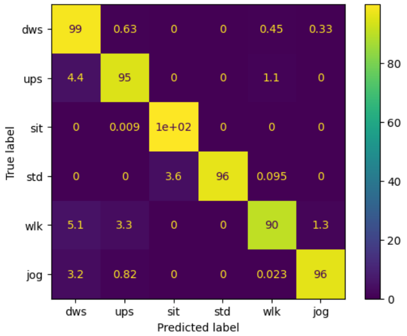


Figure: CNN confusion matrix Random Forest with rolling features

Conclusion
Results Overview

- Random Forest with rolling features achived 96.3% accuracy
- CNN with 10s windows achieved 97% accuracy
- CNN training time was approximately 13 minuets
- Random Forest training time was approximately 1.5 minuets

Concluding Remarks

Results

- Time series data is important for activities with movement
- Data with time series features performs better than data without
- Windowing and rolling features are both good ways to deal with time series data
- Windowing lead to better results than rolling features
- Windowing is more computationally expensive than rolling features
- Rolling features were quicker to train and predict than windowing

Limitations

The Data

- The dataset is small
- The dataset is not diverse
- The dataset is not real world

The Models

- Overfitting
- Computational cost
- Black box

References I

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)*, Montreal, Quebec, Canada, 2019. ISBN: 978-1-4503-6283-2. Pages 49–58. `http://doi.acm.org/10.1145/3302505.3310068`

motion-sense/materials/desc.png at master · mmalekzadeh/motion-sense. `https://github.com/mmalekzadeh/motion-sense/blob/master/materials/desc.png`. Accessed on 12/02/2023.