Sensor-based Fitness Activity Recognition

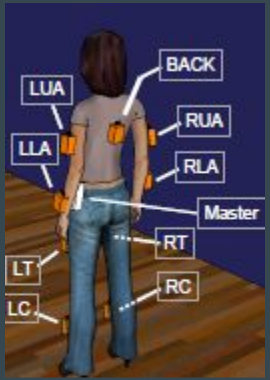
Mao Li, Songyuan Hai, Siyang Zhang,

Outline

- Introduction
- Methodology
- Data preprocessing
- Experiment
- Result
- Conclusion

Introduction

- Dataset: REALDISP Activity Recognition Dataset
    - Select 11 activities (including NULL)
        - Running, Jump up, Jump front & back, Jump sideways, Jump leg/arms open/closed
        - Knees bend forward, Rotation on the knees, Rowing, Elliptic bike, Cycling
    - Select 4 Xsens units:
        - RC (Right calf), RT (Right thigh), RLA (Right lower arm), RUA (Right upper arm)
- Motion sensors (50HZ)
    - Accelerometer
        - Measures the acceleration force that is applied to a device on x, y, z physical axes
    - Gyroscope
        - Measures a device's rate of rotation in rad/s around each of x, y, z physical axes



Methodology

- Weka
  - Decision Tree
  - Random Forest
  - Support Vector Machine
  - Multilayer Perceptron
- TensorFlow
  - LSTM Recurrent Neural Network

Preprocess

- Segmentation
  - Sliding window length = 400ms, Step = 200ms
  - Sliding window length = 600ms, Step = 300ms
  - Sliding window length = 1000ms, Step = 500ms
  - 50% overlap for better performance
- Feature Extraction
  - Mean
  - Standard deviation
  - Energy (magnitude)
  - Median absolute deviation
- Normalization

	1	2	3	4	5	6	7
1	-5.2532	10.0110	-6.4256	-0.1929	0.6241	0.2409	-11.1000
2	-4.6485	9.1648	-5.9846	0.6644	-0.2166	-0.2182	-9.7915
3	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
4	-6.0771	9.3891	-4.6835	-0.4156	0.2177	0.0683	[illegible]
5	-6.9881	10.1920	-3.8788	0.2494	-0.2887	-0.2674	[illegible]
6	-5.9963	5.9727	-3.6940	0.0643	0.3209	-0.0500	[illegible]
7	-5.3917	8.9056	-5.8380	-0.2100	0.2806	0.1762	[illegible]
8	-6.2063	9.0918	-6.4614	0.3259	-0.3309	-0.2943	[illegible]
9	-5.6226	6.2409	-4.3044	-0.0673	0.3182	0.0204	[illegible]
10	-3.9599	8.7727	-4.6042	-0.0717	0.1166	0.2677	[illegible]
11	-5.7575	9.3163	-5.2614	0.4029	-0.6298	-0.1497	[illegible]
12	-5.4607	7.1528	-4.2228	-0.2220	0.3636	0.0926	[illegible]
13	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
14	-6.4079	8.3144	-6.7067	0.6657	-0.4778	-0.3014	[illegible]
15	-6.3199	5.9035	-5.4782	0.0135	0.3719	-0.0195	[illegible]
16	-4.1408	7.8419	-5.3372	-0.1663	0.1064	0.2288	[illegible]
17	-6.6228	8.0498	-6.3931	0.4201	-0.3546	-0.1037	[illegible]
18	-5.8547	6.5200	-5.2412	-0.1528	0.6182	-0.0258	[illegible]
19	-3.4552	8.4666	-4.7176	-0.3402	0.0765	0.2058	-8.7890
20	-6.2233	8.8633	-4.5941	0.2469	-0.6868	-0.1874	-10.1580
21	-6.1965	6.9804	-3.4555	-0.0235	0.3487	-0.0711	-8.4537
22	-4.5422	8.4084	-4.7787	0.1026	0.3118	0.1492	-8.6942
23	-6.4813	8.3226	-6.0727	0.3719	-0.2556	-0.2204	-10.2320
24	-5.4833	6.3334	-4.6870	-0.3010	0.5256	-0.0347	-8.7653
25	-3.6267	8.8091	-5.4990	-0.1281	0.1919	0.1849	-9.2984
26	-5.4046	8.8628	-6.3296	0.6531	-0.4352	0.0826	-10.5040
27	-4.5868	5.9704	-5.0378	-0.0186	0.4386	-0.0198	-8.4384

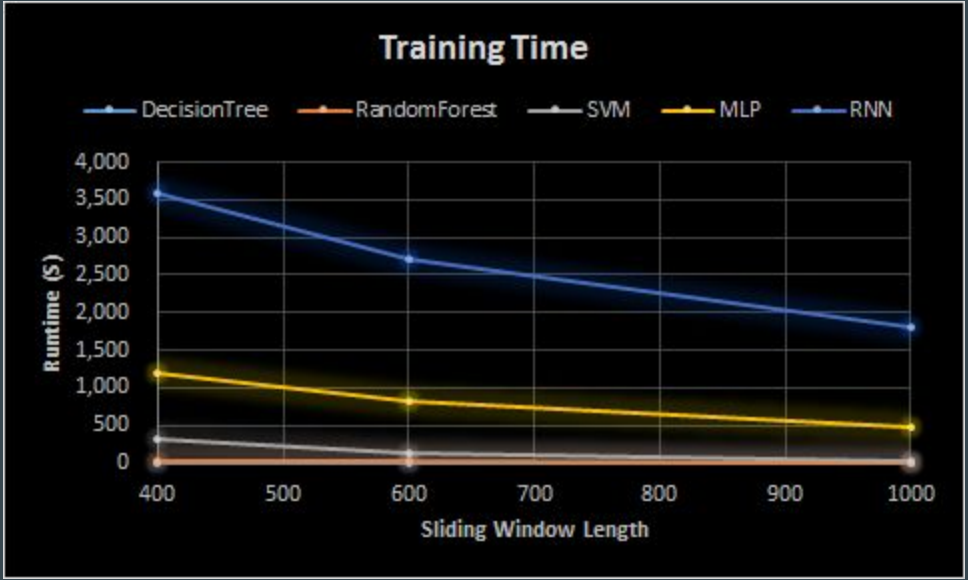
2	88.2800	18.6150	27.2280	2.2837	1.4970
3	96.7770	26.0880	21.8280	1.8250	1.3818

Experiment

- Environment: Intel Core i7-6700 CPU 3.40GHz, win 10
- Raw data: 409692 raw samples(about 2.5 hours)x4(units)x6(channels)
- Instances: 40969(slide 20) |27312(slide 30) | 16386(slide 30)
- Attributes: 96 features (24 x 4) + 1 label
- Training time:
  - Decision Tree < Random Forest << Support Vector Machine << Mutilayer Perceptron << LSTM RNN

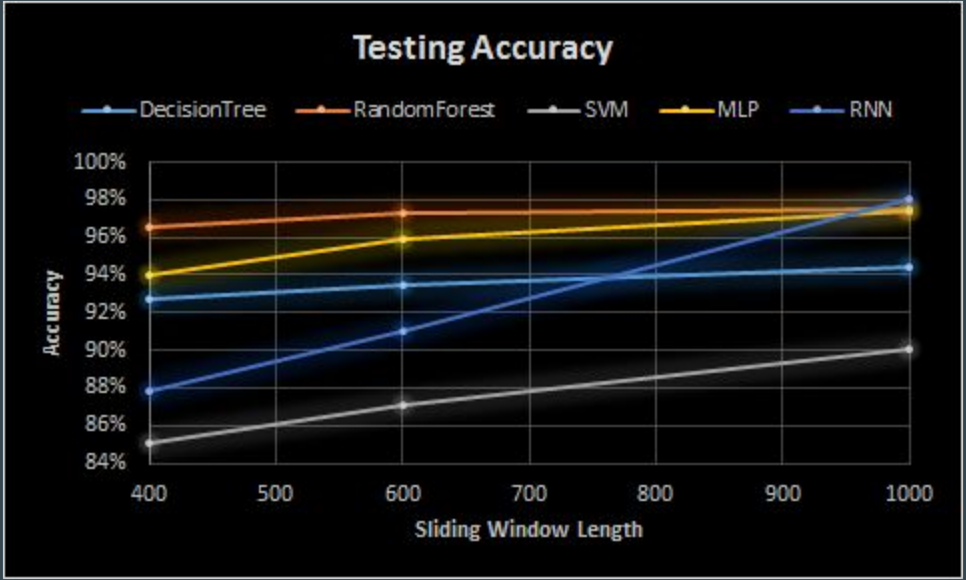
Result

- Training time
  - DecisionTree and RandomForest cost the least training time (<30s)
  - SVM costs 30s - 300s
  - MLP costs 8m - 20m
  - LSTM RNN costs the most training time (0.5h - 1h)



Result cont’d

- Testing accuracy
  - RandomForest has overall the best performance (96.5% - 97.5%)
  - LSTM RNN has greatly improved performance with increasing sliding window length (87.7% - 98.0%)
  - SVM has the worst performance (85.0% - 90.0%), the reason could be imbalanced dataset or multiple classes

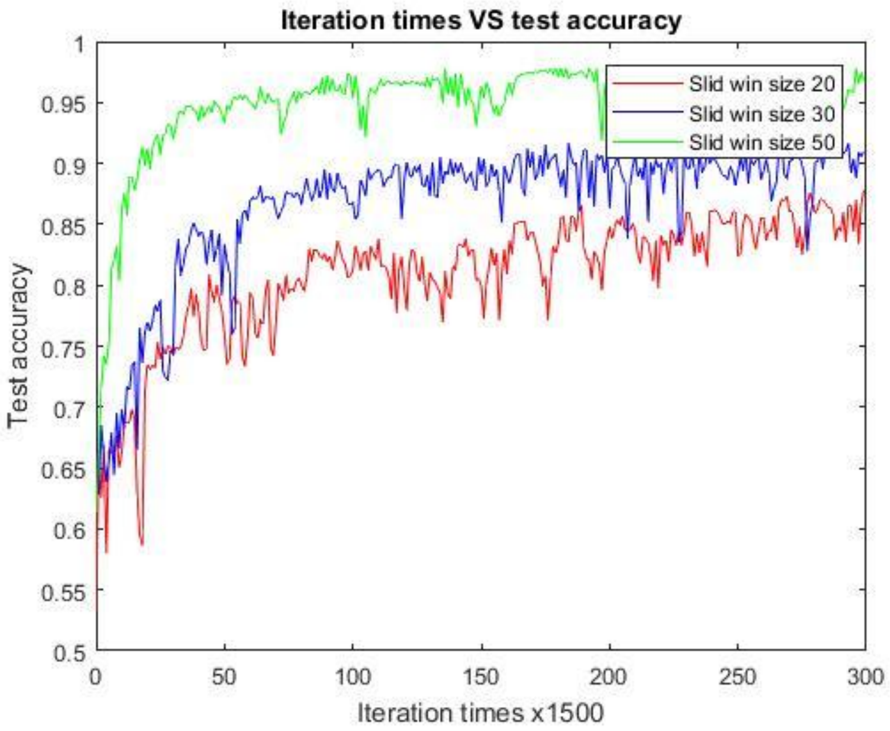


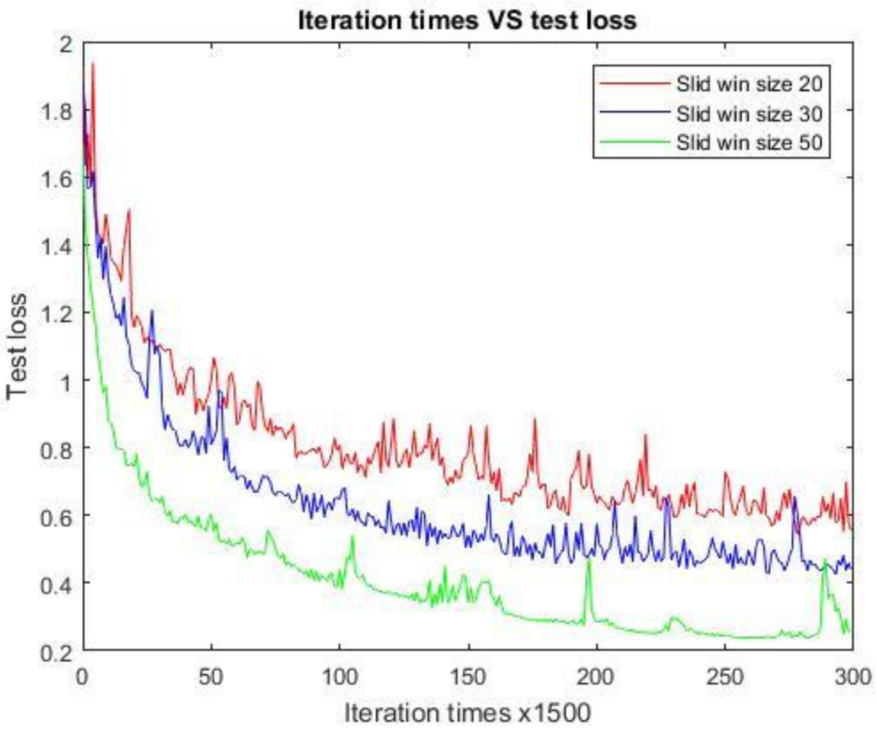
LSTM RNN

- 2 stacked LSTM cells
- 32 neurons in hidden layer
- Learning rate = 0.0025
- Lambda = 0.0015

```
Confusion Matrix:
[[3390    0    5    0    0    0    0    0    0    0    0]
 [   3  552    3    2    0    0    0    0    0    0    0]
 [   3    0  100    3    0    0    0    0    0    0    0]
 [   0    0   25  152   14    0    1    0    0    0    0]
 [   0    0    0    3  192    0    1    0    1    0    0]
 [   0    0    0    1    1  196    0    0    1    0    0]
 [   0    0    0    0    0    1  253    0    0    0    3]
 [   0    0    0    0    0    0    0  129    0    0    0]
 [   0    0    0    0    0    0    3    0  310    0    1]
 [   0    0    0    0    1    7  107    0    0  199    0]
 [   0    0    0    0    0    0    3    0    3    2  715]]
```

LSTM RNN Cont’d





Conclusion

- RandomForest has great performance with a large set of features and cheap time cost with a large amount of instances
- LSTM RNN has the best performance with appropriate sliding window length but requires high computational resources (GPU support)
- For real-time application, either choose a cheap classifier such as RandomForest or J48 DecisionTree, or train model and classify results on the cloud server

References

[1] Mukhopadhyay, S. C. (2015). Wearable Sensors for Human Activity Monitoring: A Review. IEEE Sensors Journal, 15(3), 1321-1330.

[2] Shoaib, M., Bosch, S., Incel, O., Scholten, H., & Havinga, P. (2016). Complex Human Activity Recognition Using Smartphone and Wrist-Worn Motion Sensors. Sensors, 16(4), 426.

[3] Yang, R., & Wang, B. (2016). PACP: A Position-Independent Activity Recognition Method Using Smartphone Sensors. Information, 7(4), 72.

[4] Ziaeefard, M., & Bergevin, R. (2015). Semantic human activity recognition: A literature review. Pattern Recognition, 48(8), 2329-2345.

[5] G. (2017, February 03). Guillaume-chevalier/LSTM-Human-Activity-Recognition. Retrieved April 22, 2017, from https://github.com/guillaume-chevalier/LSTM-Human-Activity-Recognition