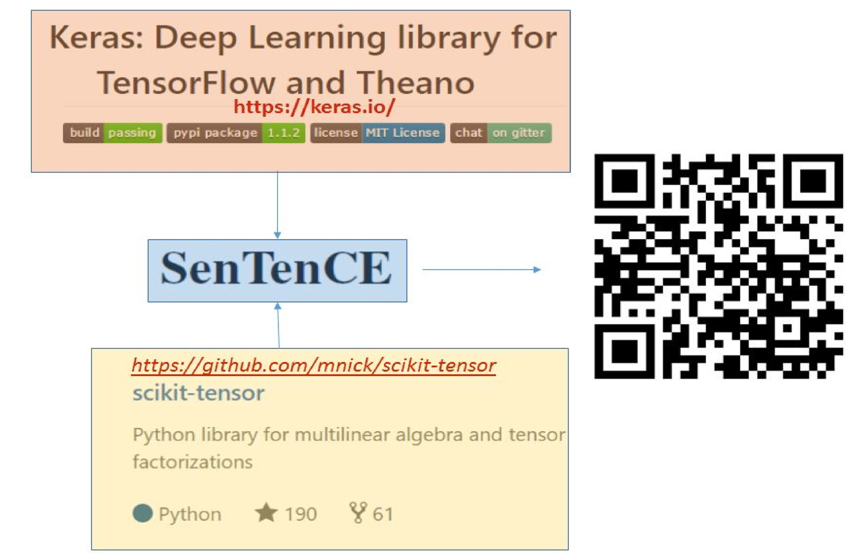
Multi-sensor data compression using tensor decompositions: A SenTenCE and more

Vinay Uday Prabhu
UnifyID
San Francisco CA 94107
vinay@unify.id

Paulo Arantes
UnifyID
San Francisco CA 94107
paulo@unify.id

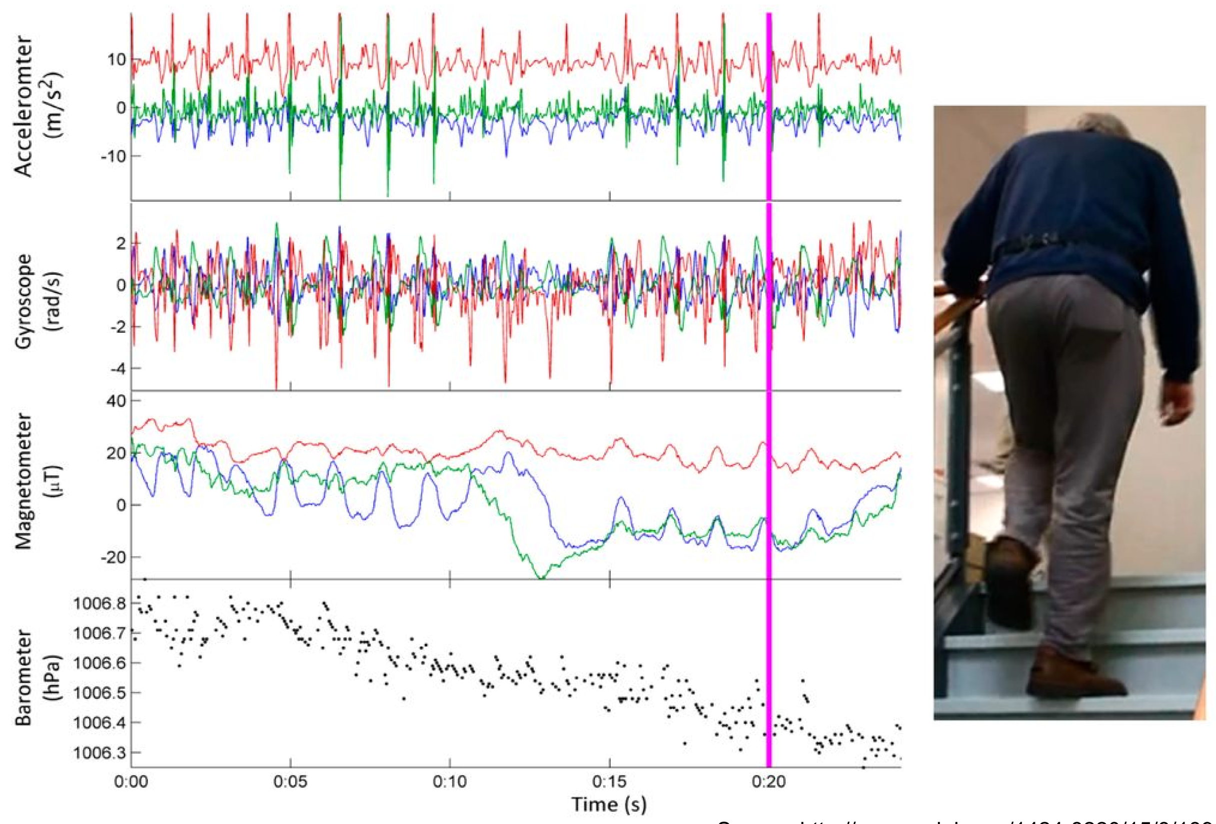
John Whaley
UnifyID
San Francisco CA 94107
john@unify.id

CODE AND DATA:



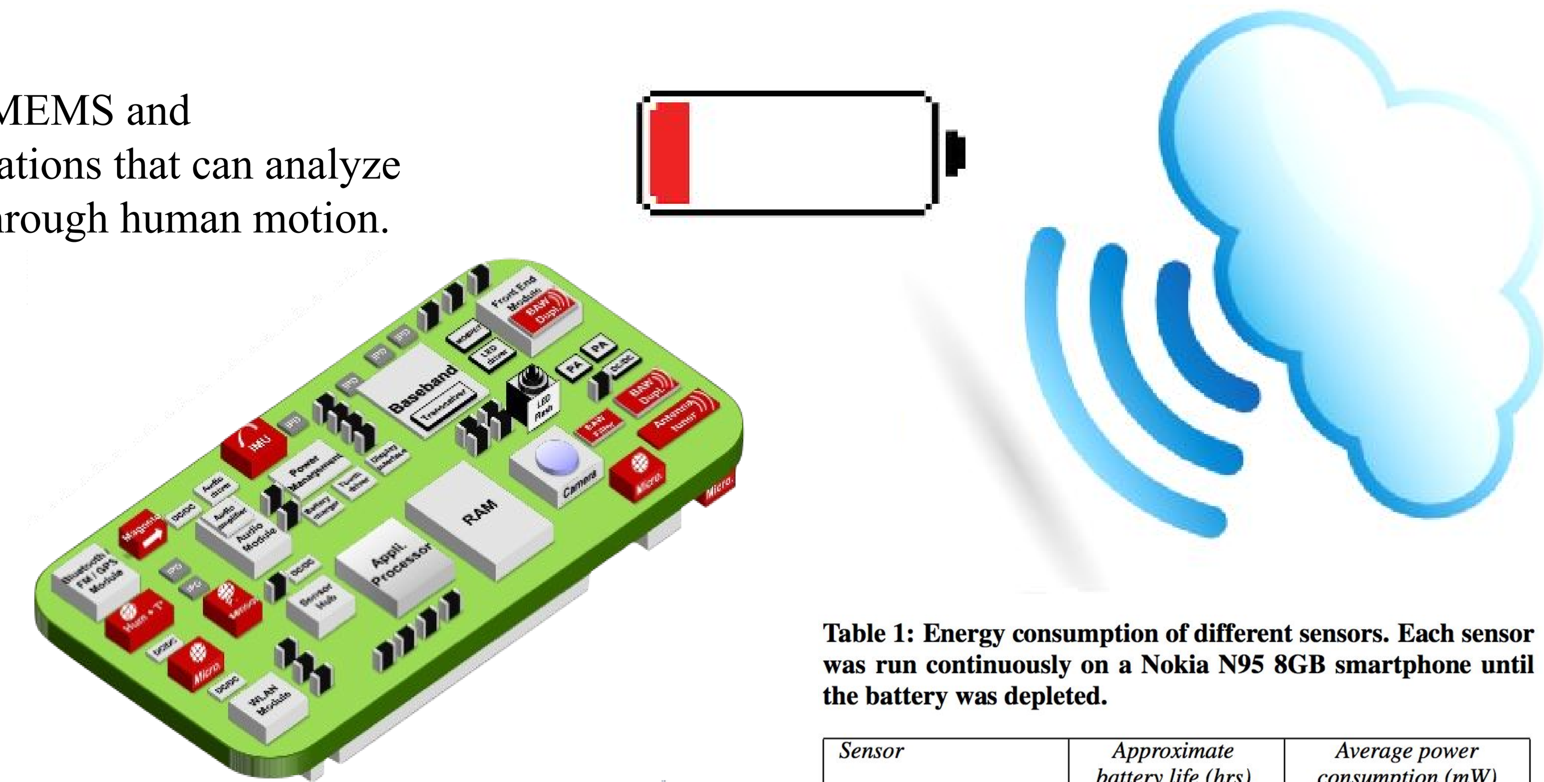
Motivation and Introduction:

Sensors are what make smartphones smart. In recent years, the proliferation of MEMS and advancements in Machine Learning have allowed for the development of applications that can analyze sensor data to solve many problems, such as classifying activities and devices through human motion.



Source: http://www.mdpi.com/1424-8220/15/8/18901/htm

Time_Stamp	Ax	Ay	Az	Gx	Gy	Gz	Mx	My	Mz	User
1.36E+12	0.23154591	-9.32994	-3.02372	0.275195	-0.11179	0.019242	-19.68	29.64	-6.6	Participant 1
1.36E+12	1.0487667	-8.85323	-1.22583	0.80115	1.128879	0.011912	-19.619999	29.4	-6.72	Participant 1
1.36E+12	0.939804	-9.507	-2.04305	0.807258	1.709201	-0.03238	-19.68	29.34	-6.96	Participant 1
1.36E+12	0.6401563	-9.83389	-2.15202	0.602008	2.062892	0.003665	-19.859999	29.22	-7.14	Participant 1
1.36E+12	0.027240695	-9.24822	-1.32117	0.629802	2.191174	0.159741	-19.68	29.039999	-7.98	Participant 1
1.36E+12	0.47671217	-9.34356	-1.45738	0.718988	1.960267	0.295659	-19.5	28.92	-8.4	Participant 1
1.36E+12	1.0896279	-9.61597	-0.8036	0.718683	1.903456	0.347277	-19.32	28.74	-8.76	Participant 1
1.36E+12	1.6889231	-9.45252	0.40861	0.725403	2.049147	0.331394	-18.779999	28.74	-9.78	Participant 1
1.36E+12	2.4380422	-9.60235	0.612916	0.627359	2.261729	0.308792	-18.48	28.74	-10.44	Participant 1
1.36E+12	3.091819	-9.87475	0.40861	0.564745	2.422386	0.235489	-18.119999	28.56	-11.28	Participant 1



Source:https://www.slideshare.net/Yole_Developpement/yole-mems-formobilejune2013reportsample

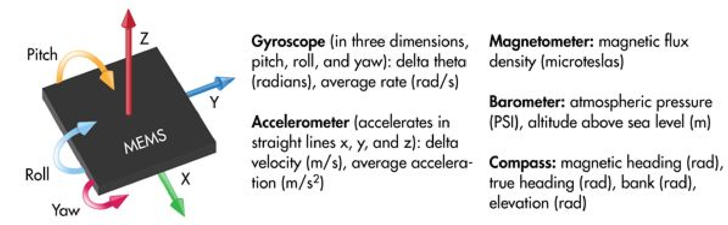


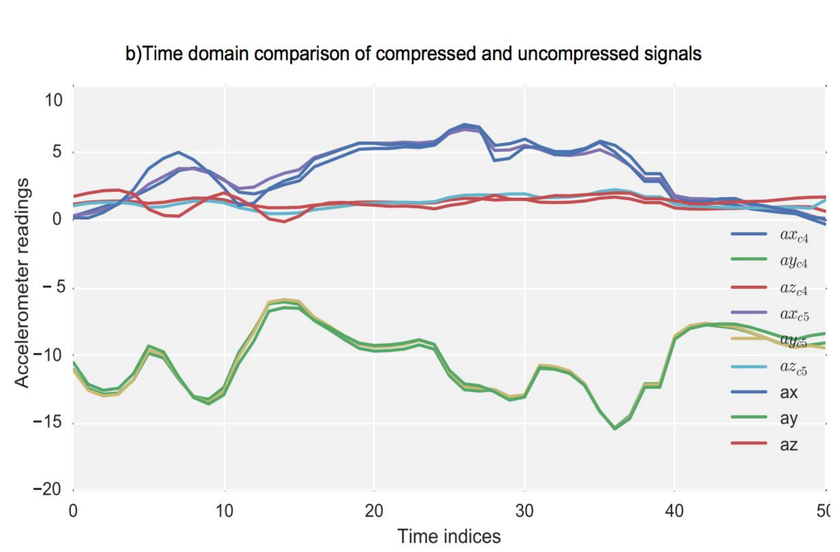
Table 1: Energy consumption of different sensors. Each sensor was run continuously on a Nokia N95 8GB smartphone until the battery was depleted.

Sensor	*Approximate battery life (hrs)*	*Average power consumption (mW)*
Video camera	3.5	1258
IEEE 802.11	6.7	661
GPS (outdoors)	7.1	623
GPS (indoors)	11.6	383
Microphone	13.6	329
Bluetooth	21.0	211
Accelerometer	45.9	96
All sensors turned off	170.6	26

Source: http://conferences.sigcomm.org/sigcomm/2009/workshops/mobiheld/papers/p61.pdf

Activating sensors and collecting its data can be very expensive and battery intensive. To use deep learning or other computationally intensive algorithms, it's also necessary to transfer the data, which poses another limitation on memory.
The Solution: Smart sensor data compression via SenTenCE.

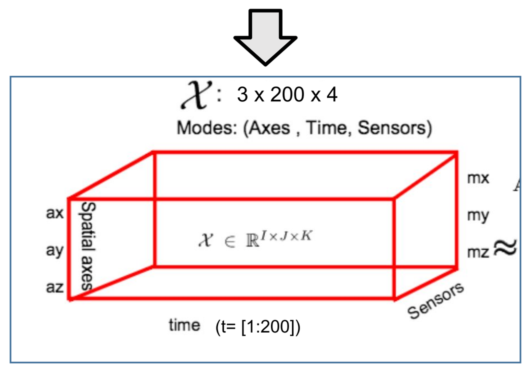
Dataset and implementation:

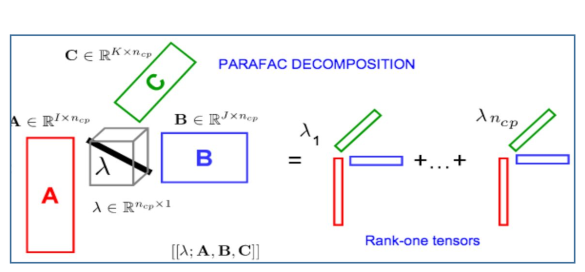


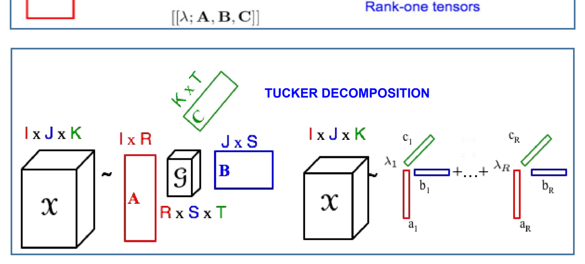
Six different datasets were fused, resulting in 9 activity labels, and 10.3 million data points. The datasets were converted into a common format and resampled to 200Hz. The input to the DCNN is a 200×4 image of raw sensor data. We used the 70-10-20 training - validation - testing split and 100 folds cross validation for each input data format. By performing the tasks on Nvidia-GPUs, the training time decreased by an order of **~18 times** in comparison to using CPUs.

Ax	Ay	Az	Gx	Gy	Gz	Mx	My	Mz
0.23154591	-9.329938	-3.0237172	0.2751948	-0.11178834	0.019242255	-19.68	29.64	-6.6
1.0487667	-8.853226	-1.2258313	0.8011498	1.128879	0.011911872	-19.619999	29.4	-6.72
0.939804	-9.507003	-2.0430522	0.8072584	1.709201	-0.032375857	-19.68	29.34	-6.96
0.6401563	-9.833891	-2.152015	0.6020077	2.062892	0.003665192	-19.859999	29.22	-7.14
0.027240695	-9.248216	-1.3211738	0.62980205	2.1911736	0.15974127	-19.68	29.039999	-7.98
0.47671217	-9.343558	-1.4573772	0.7189884	1.9602666	0.29565877	-19.5	28.92	-8.4
1.0896279	-9.615966	-0.8036005	0.71868294	1.9034561	0.3472769	-19.32	28.74	-8.76
1.6889231	-9.452521	0.40861043	0.7254025	2.0491474	0.3313944	-18.779999	28.74	-9.78

Tensorization







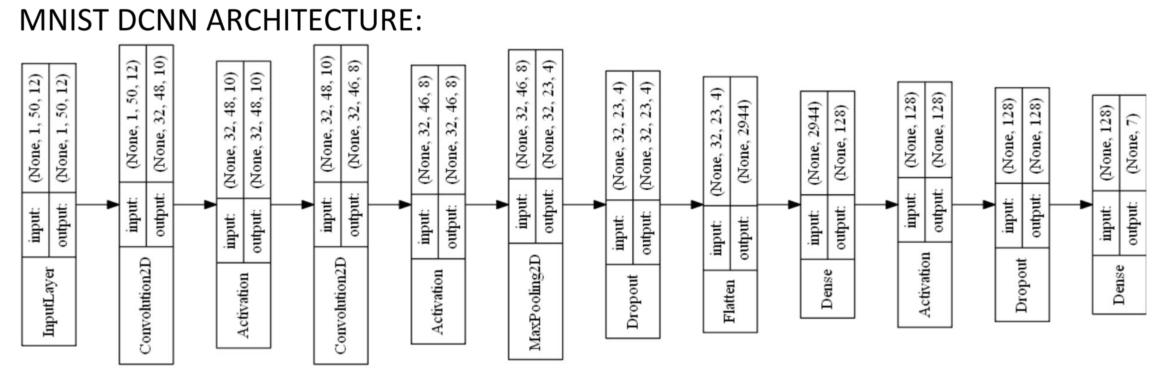
Compress on-Phone

Transmit only compressed data

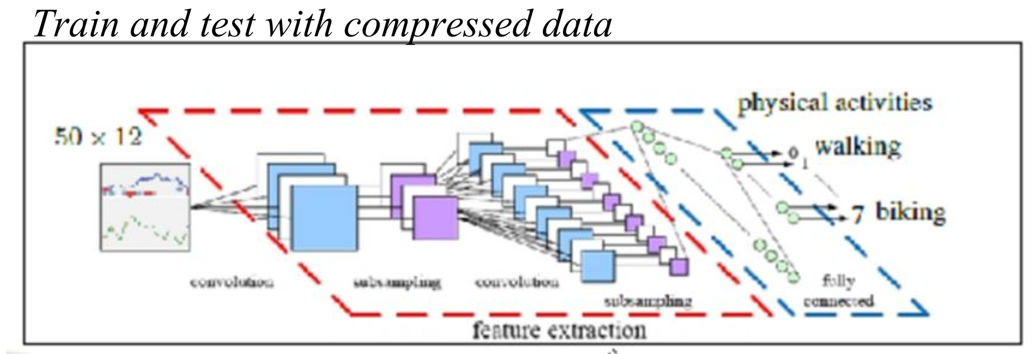
$$\mathcal{X} \approx \mathcal{G} \times_1 \mathbf{A} \times_2 \mathbf{B} \times_3 \mathbf{C} = \sum_{p=1}^{P}\sum_{q=1}^{Q}\sum_{r=1}^{R} g_{pqr}\, \mathbf{a}_p \circ \mathbf{b}_q \circ \mathbf{c}_r = [\![\mathcal{G}\,;\mathbf{A},\mathbf{B},\mathbf{C}]\!].$$

$$\mathcal{X} \approx [\![\lambda\,;\mathbf{A},\mathbf{B},\mathbf{C}]\!] = \sum_r \lambda_r\, \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r$$

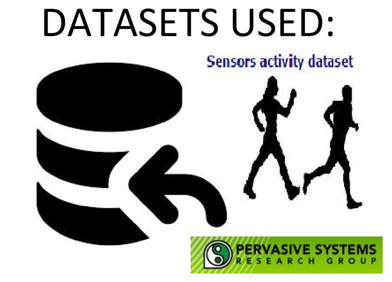
MNIST DCNN ARCHITECTURE:



Train and test with compressed data



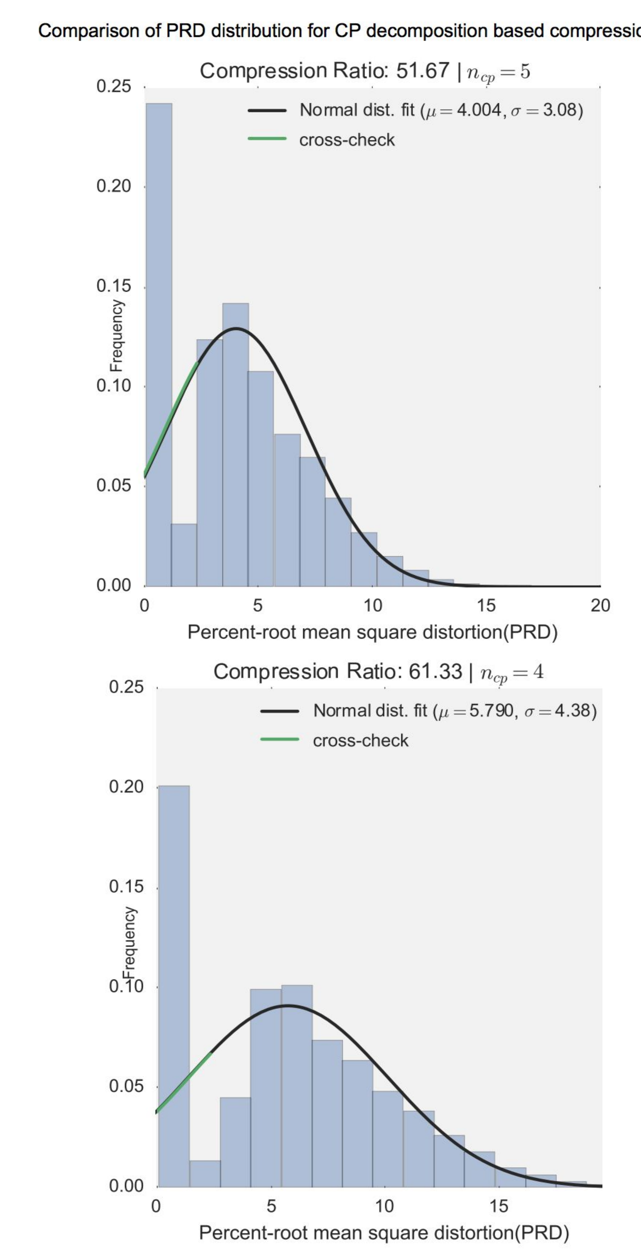
DATASETS USED:

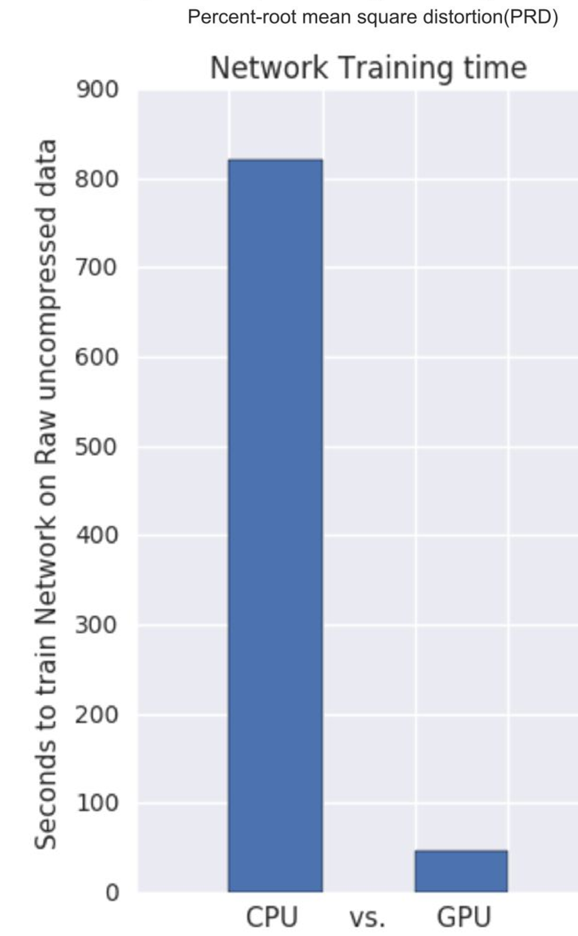


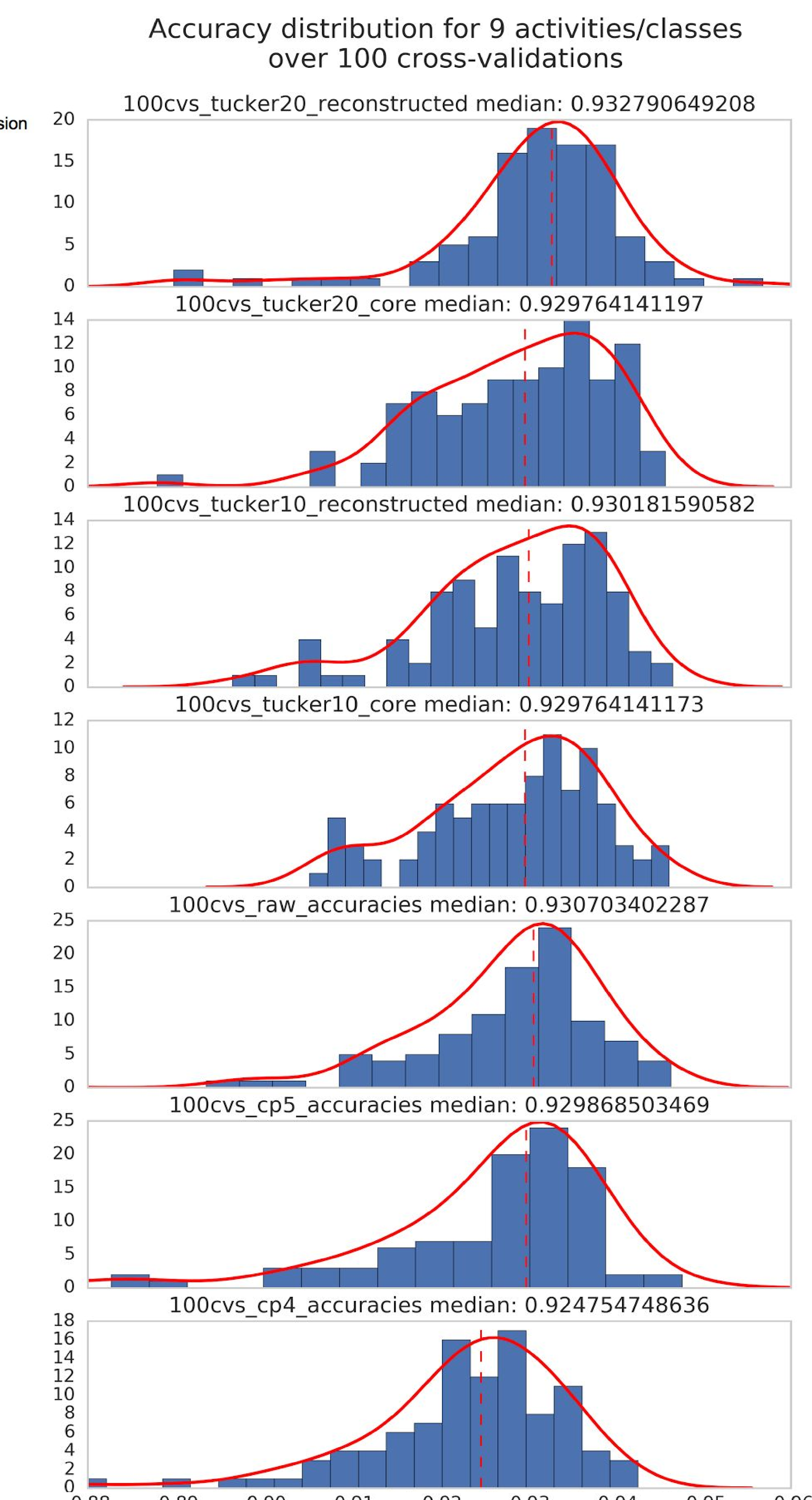
Dataset of walking, running, sitting, standing, jogging, biking, walking upstairs and walking downstairs activities
Citation notice: (a) Shoaib, M. and Bosch, S. and Incel, O.D. and Scholten, H. and Havinga, P.J.M. (2014) Fusion of Smartphone Motion Sensors for Physical Activity Recognition. Sensors, 14, 10146-10176; (b) This data collector is supported by the Dutch national program COMMIT in the context of the SWELL project.

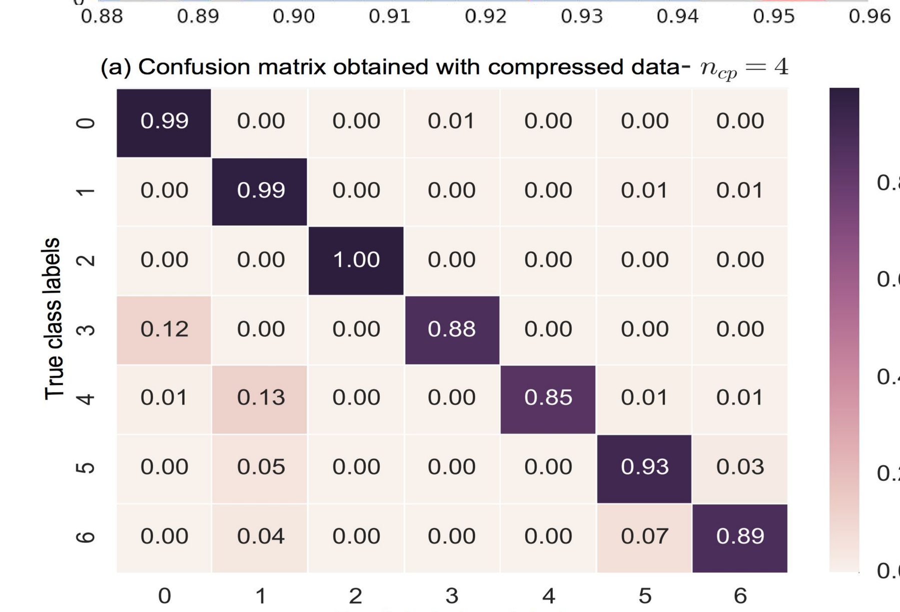
- Shoaib et al. (2013 and 2014)
- McCall et al.,
- 2011 Opportunity Activity Recognition Challenge
- Kwapisz et al. (2011), and
- Chen et al. (2015)

Results:









Input Data for Model	Data Compression	# of Parameters	Network params "Compression"	Mean Accuracy ⬇
Tucker ranks [20, 2, 4] reconstruction	0%	672,669	0%	93.104 ± 0.205%
Tucker ranks [20, 2, 4] core	90%	49,269	92.7%	92.886 ± 0.204%
Tucker ranks [10, 2, 4] reconstruction	0%	672,669	0%	92.850 ± 0.207%
Tucker ranks [10, 2, 4] core	95%	41,269	93.9%	92.842 ± 0.198%
Raw uncompressed	0%	672,669	0%	92.778 ± 0.222%
CP rank 5 reconstruction	61%	672,669	0%	92.513 ± 0.289%
CP rank 4 reconstruction	51%	672,669	0%	92.328 ± 0.217%

Via Canonical Polyadic we achieve a 61% data compression, less than 6% distortion rate (PRD), and lose less than 2% accuracy in the human activity classification task using a CNN deep classifier. Further, we also harness Tucker decomposition and achieve ~95% compression rate with a consistent improvement of about 0.5% in accuracy while also decreasing the number of training parameters by ~94%.