Data augmentation of wearable sensor data for Parkinson's disease monitoring using convolutional neural networks

Terry T. Um, Franz M.J. Pfister, Daniel Pichler, Satoshi Endo, Muriel Lang, Sandra Hirche, Urban Fietzek, Dana Kulic

Presenter: Shovito Barua Soumma

Date: December 20, 2023

- Published In 19th ACM International Conference on Multimodal Interaction (2017)
- 624 citation as today
- https://github.com/terryum/Data-Augmentation-For-Wearable-Sensor-Data

Challenges

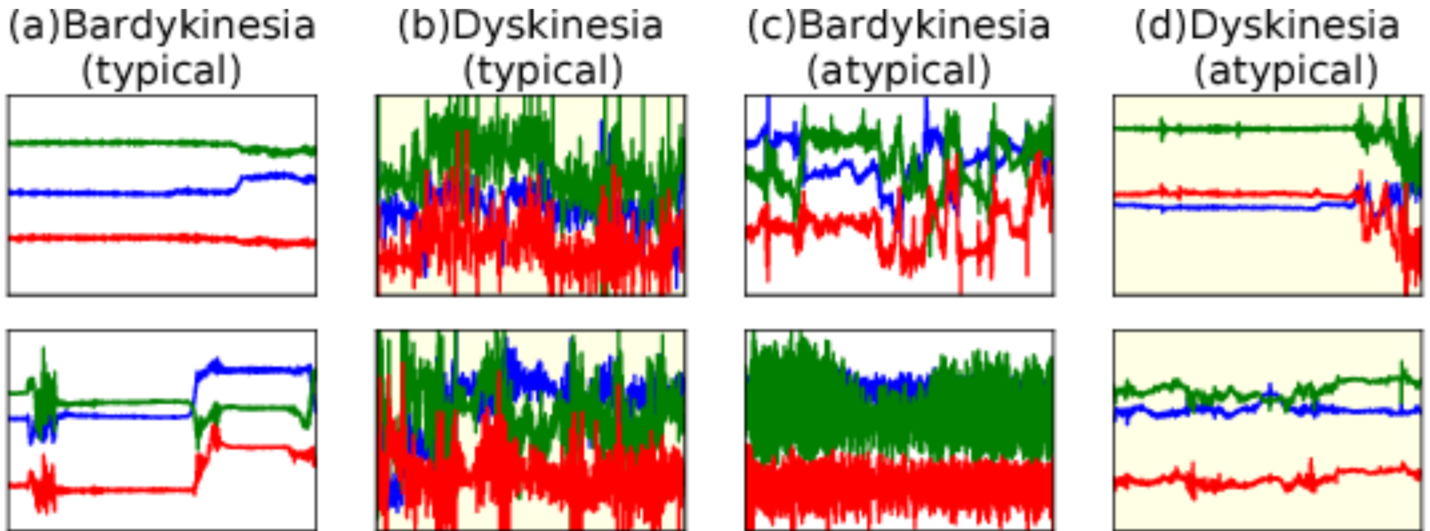
- Small medical dataset
- Noisy label
- high intra-class variability: sensor data for a single motor state of PD can vary greatly from patient to patient

Challenges in PD data

- Bradykinesia:
  - Freezing of voluntary movement, (movement speed will be decreased)
  - constant signal (less movement)
  - Atypical Bradykinesia= Bradykinesia + Tremor => *Looks like dyskinesia!!*

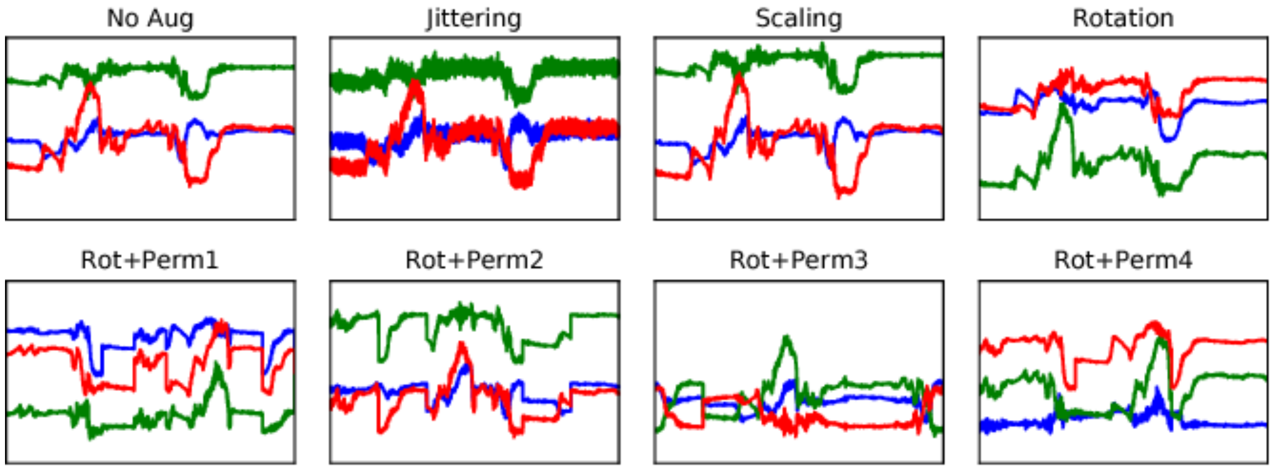
Dyskinesia:

- Extreme *involuntary* movements → fluctuation in sensor
- Atypical Dyskinesia = Dyskinesia + Voluntary suppression ➔



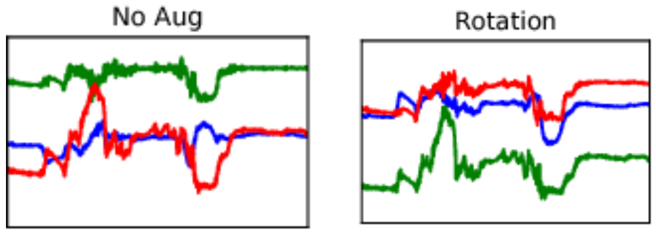
Data Augmentation

- Injecting prior knowledge
- Label preserving data augmentation for sensor is not intuitively recognizable like the image data
- Location Vs Magnitude



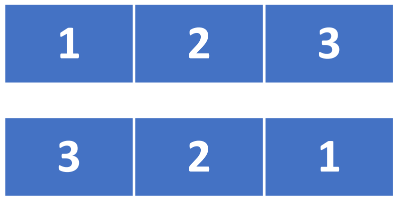
Data Augmentation (Temporal Location)

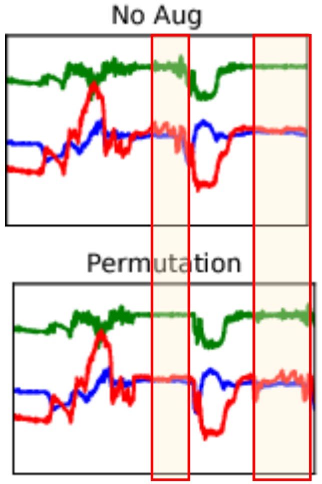
- **Rotation:** applying arbitrary rotations to the existing data, they mimic different sensor placements.
  - Changing the axes



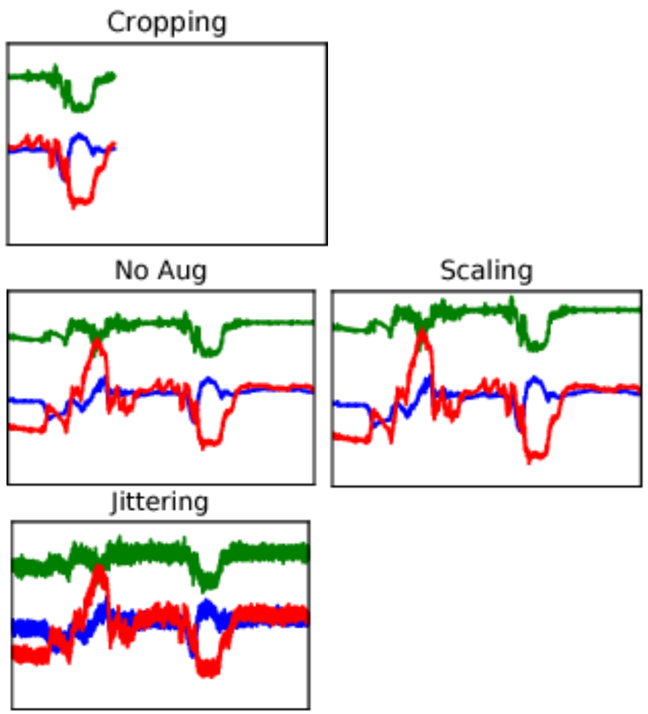
- **Permutation:**
  - Rearrange data sequence in a window = A new window!
  - Variability in the temporal location

- Slice the window data into N=3 segments
- Shuffle the n=1 segment





Data Augmentation (Magnitude)



- **Scaling:** (zoom in/out)
  - Multiplying the whole window with a random scaler
- **Jittering:** small random noise in the data
  - Simulate real world errors
- **Cropping:** (window slice)
  - Might loss important events/ capture event free region

Result

Table 1: The results of PD motor state classification with various data augmentation methods. R,P,T,M represent *Rot*, *Perm*, *TimeW*, *MagW*, respectively.

	SVM	CNN	Jitter	Scale	Crop	Rot	Perm
Train	98.82	99.92	99.78	99.84	65.77	100.0	99.33
Test	70.72	77.54	77.52	79.46	73.58	**82.62**	81.16
	MagW	TimeW	P,T	R,P	R,T	R,P,T	R,P,T,M
Train	100.0	94.67	96.63	99.08	94.70	94.43	94.20
Test	79.33	82.00	81.75	**86.76**	85.01	**86.88**	85.60

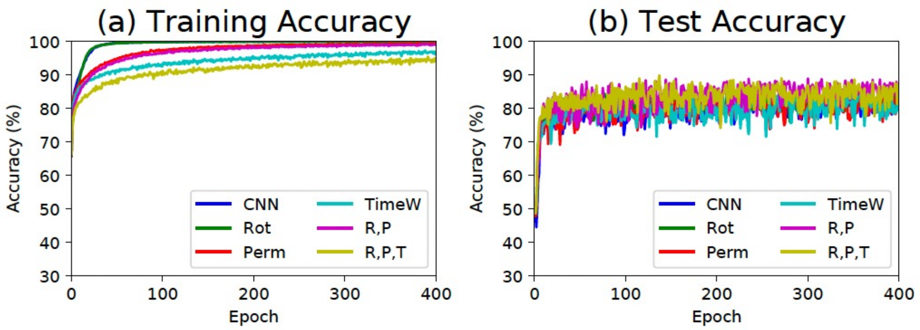


Figure 4: Training curves for *CNN*, *Rot*, *Perm*, *TimeW*, *Rot+Perm* and *Rot+Perm+TimeW* methods. The curves of *Rot+Perm+TimeW* shows slow training improvement and a better generalization performance.