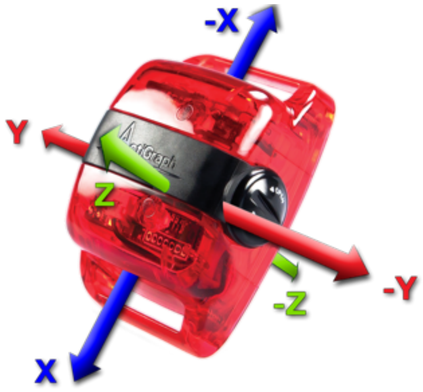
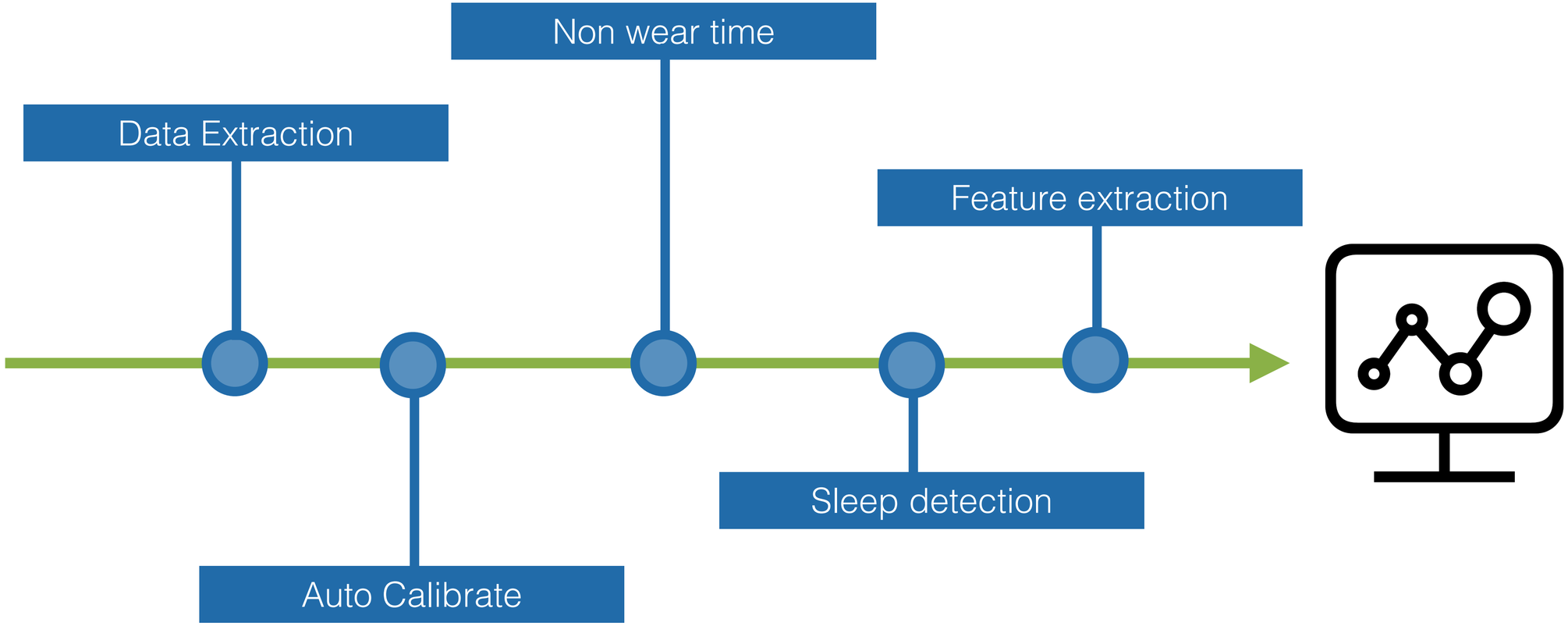
Non-wear detection with high resolution accelerometer data

Shaheen Syed, PhD

THE ROADMAP





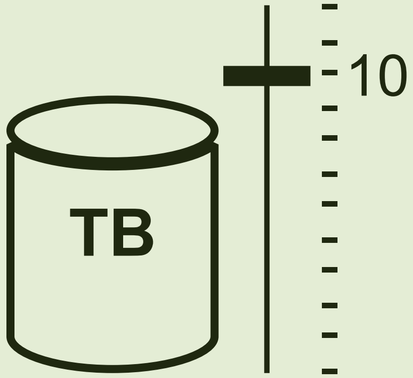
https://pypi.org/project/gt3x/

CHALLENGES ALONG THE WAY

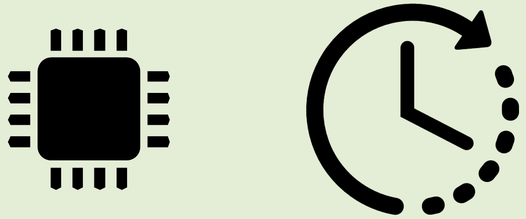
GT3X Proprietary format

6,500 Participants

10TB of Data

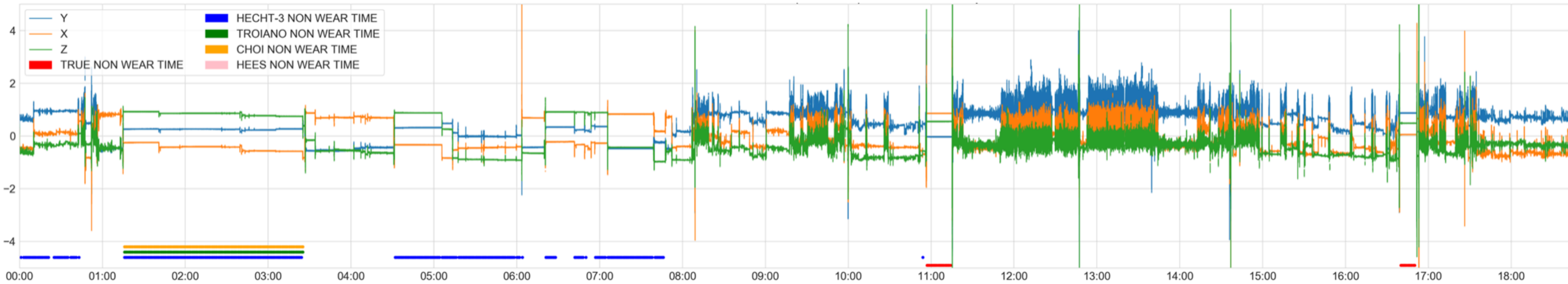


Low memory + computational time



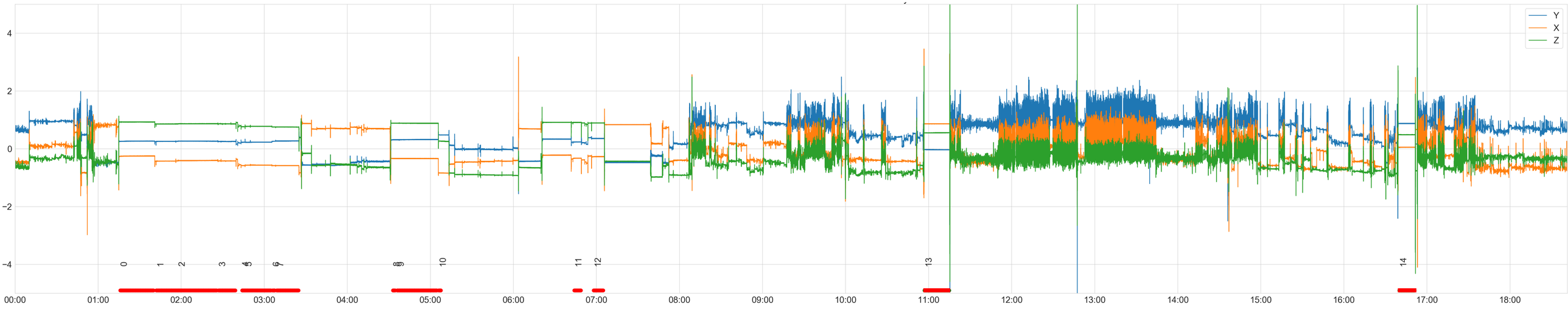
EXISTING ALGORITHMS TO DETECT NON-WEAR TIME

HECHT | TROIANO | CHOI | v. HEES



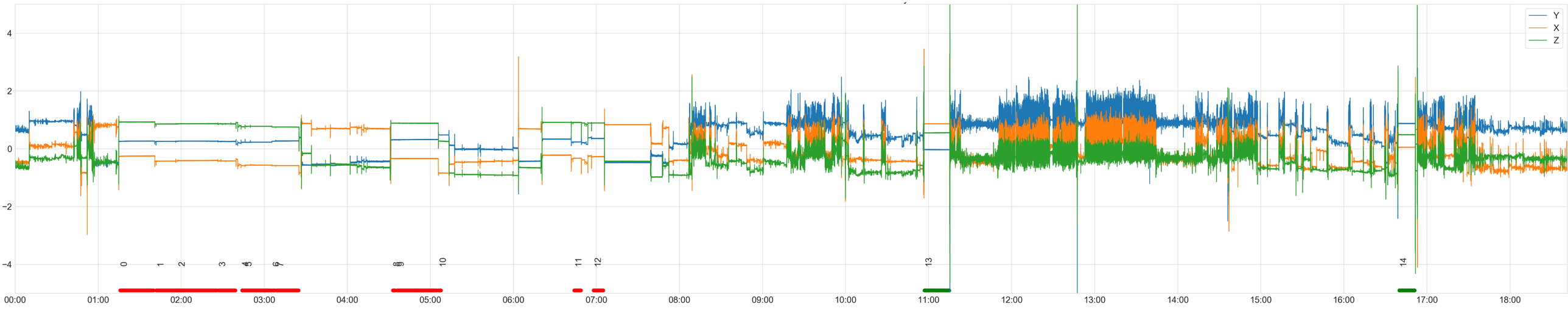
TRUE NON-WEAR TIME

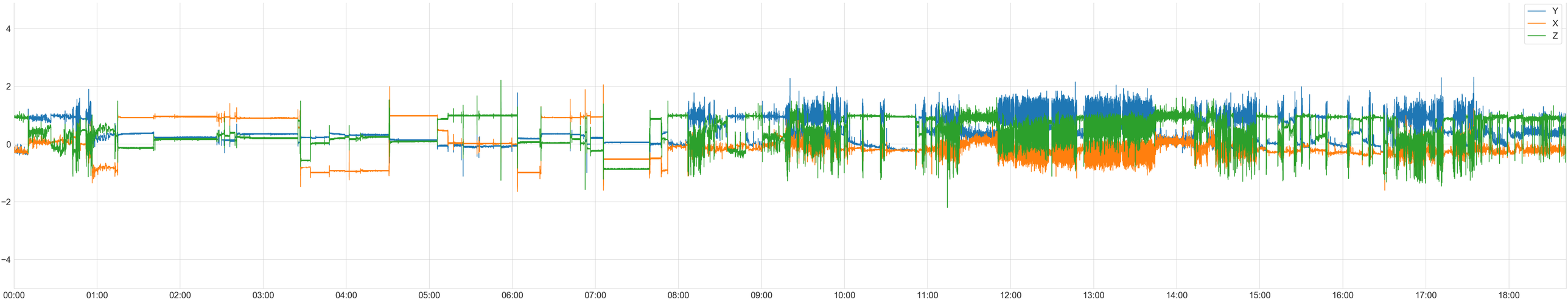
Candidate non-wear episodes



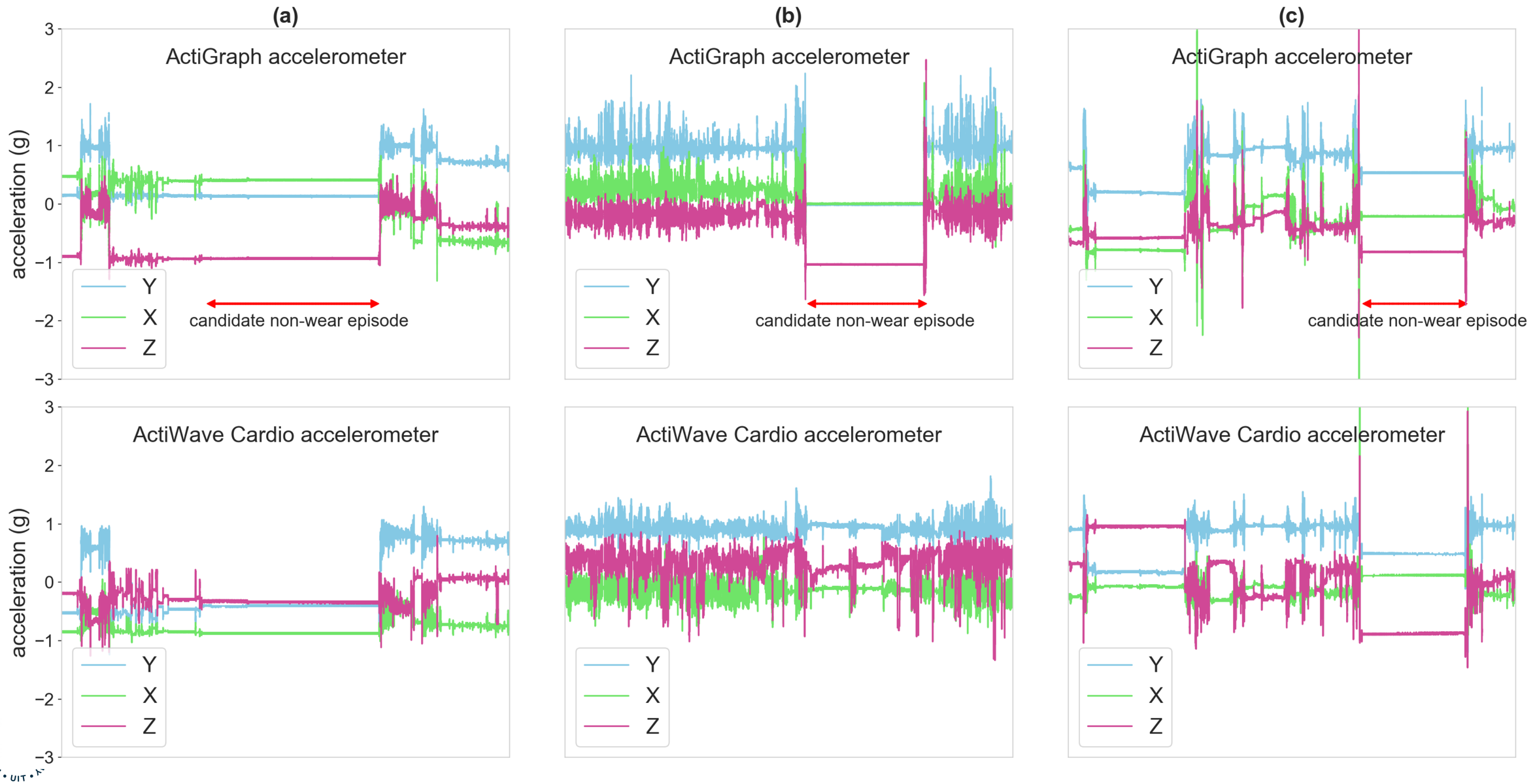
Standard deviation of all axes <= threshold value

What happened during candidate non-wear episodes

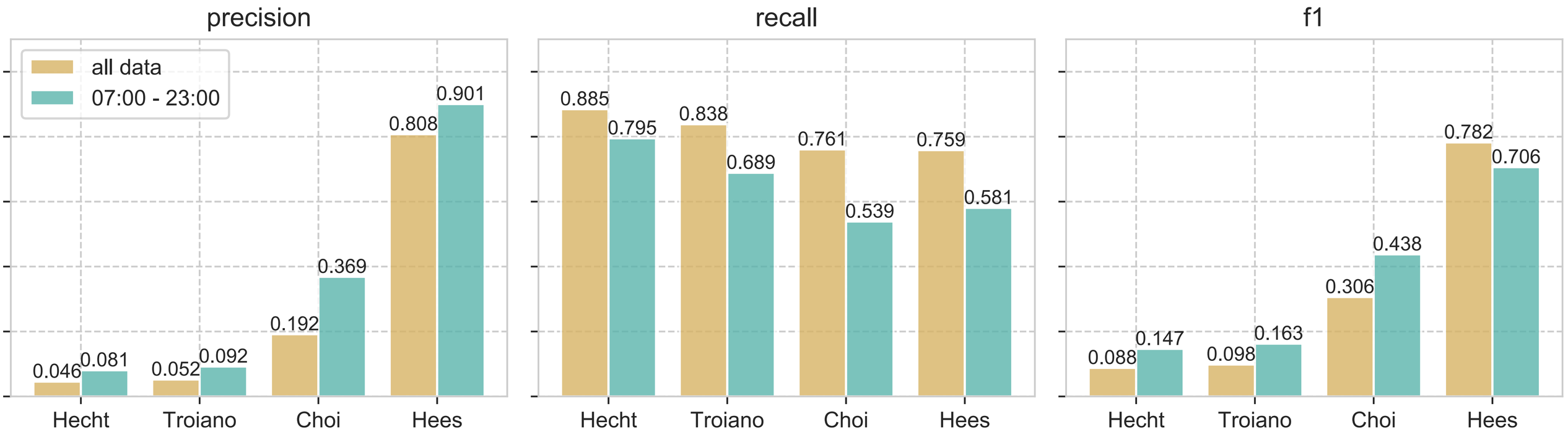




BUILDING A GROUND TRUTH DATASET



WHICH ALGORITHM TO USE?



Can we do better?
Hyperparameter Tuning

Hecht non-wear algorithm

The algorithm used to detect wear and non-wear time was based on the following three questions for each 1 minute epoch: (i) Is the VMU CMP (vector magnitude in counts per minute) value > 5.0; (ii) of the following 20 minutes, do at least 2 minutes have VMU CPM values > 5.0; and (iii) of the preceding 20 minutes, do at least 2 minutes have VMU CPM values > 5.0 (to allow for spikes or artificial movement). If at least two of the conditions were affirmed, then the 1 minute epoch was considered to be wear time; otherwise, it was considered to be non-wear time. The hyperparameters and corresponding default values of their algorithm included:

- The VMU CMP threshold; defaults to 5.0.
- The following ("upstream") and preceding ("downstream") activity window size; defaults to 20 minutes.
- The number of "spikes" in the following or preceding activity windows; defaults to 2.

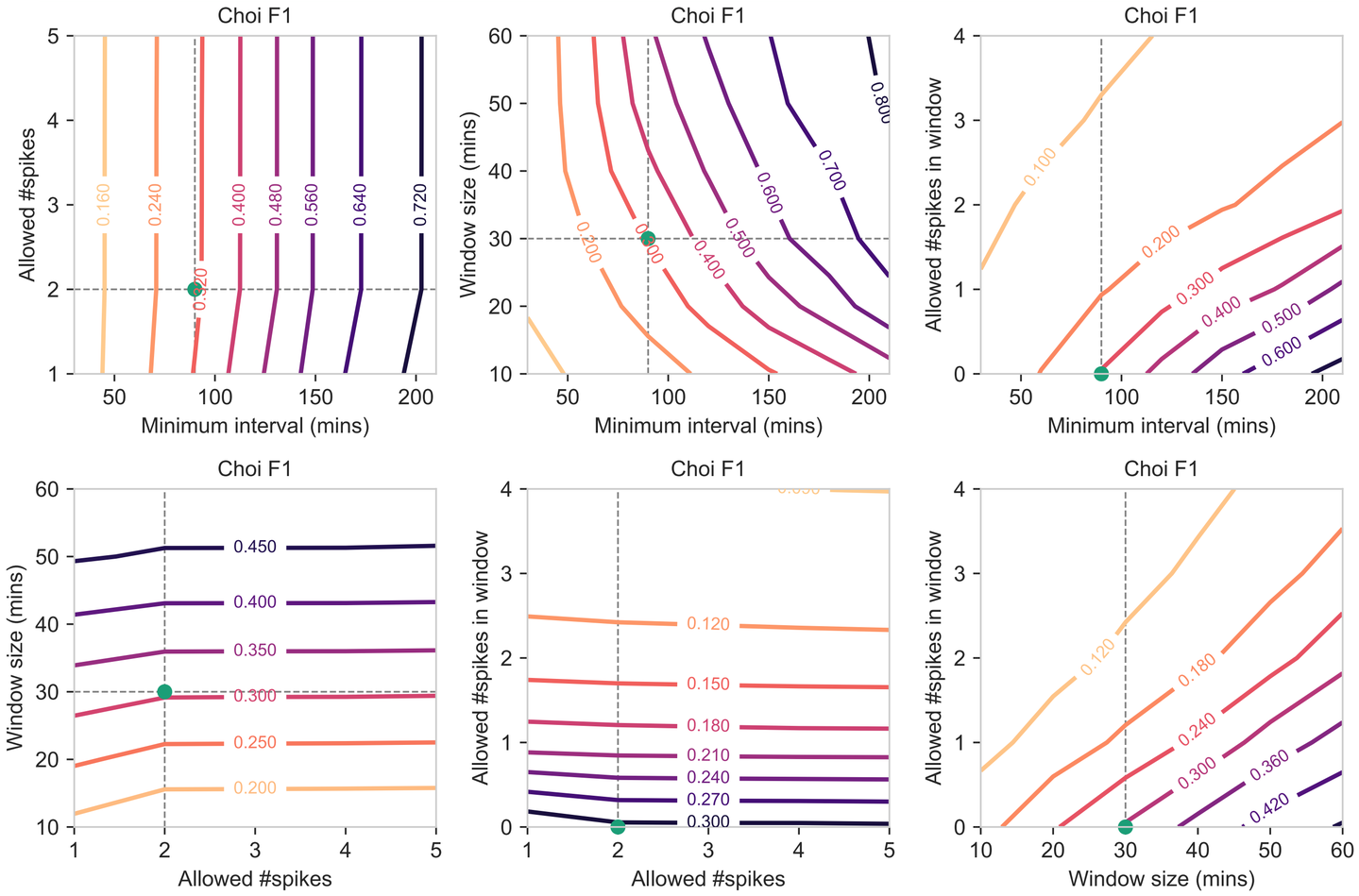
Troiano non-wear algorithm

The non-wear episodes were defined as intervals of at least 60 consecutive minutes of zero activity counts, with allowance for 1–2 minutes of counts between 0 and 100 (to allow for spikes or artificial movement). All remaining data were considered to be wear time. The hyperparameters and corresponding default values of their algorithm included:

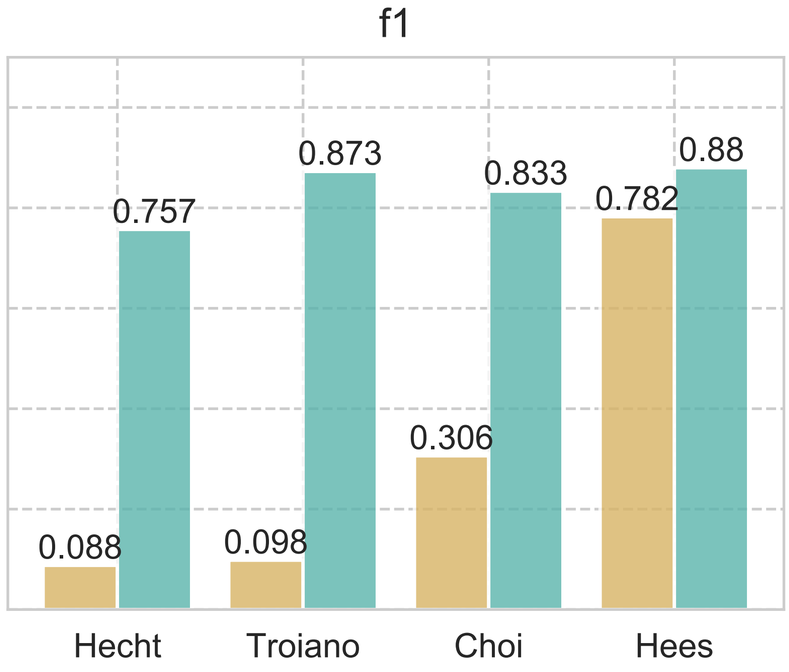
- The minimum length of the interval to classify an episode as non-wear time; defaults to 60 minutes.
- The number of spikes in the interval; defaults to 2.
- The lower bound of the spikes threshold within the interval; defaults to 0 counts.
- The upper bound of the spikes threshold within the interval; defaults to 100 counts.

What happens when we tweak the parameters

Sensitivity analysis for Choi on F1



Default vs optimised F1



Optimised hyperparameters (for older adults)

Hyperparameter	Default	Accuracy	Precision	Recall	F1	F1 (cv)
			Hecht			
VMU Threshold	5	1	1	50	1	1
Up/Down window size (mins)	20	100	100	5	100	100
Allowed #spikes in window	2	1	1	5	1	1
			Troiano			
Minimum spike threshold (counts)	0	0–100	0–100	50–100	0–100	0
Allowed #spikes	2	1–10	1–10	1–10	1–10	1
Maximum spike threshold (counts)	100	1	1	50–200	1	1
Use VMU	no	yes	yes	yes	yes	yes
Minimum interval (mins)	60	140	200	1	140	140
			Choi			
Minimum spike threshold (counts)	0	0–100	0–100	100	0–100	0
Allowed #spikes	2	1–5	1–5	5	1–5	1
Use VMU	no	yes	yes	no	yes	yes
Up/Down window size (mins)	30	20–60	20–60	10	20–60	20
Allowed #spikes in window	0	0–1	0–1	4	0–1	0
Minimum interval (mins)	90	210	210	30	210	210
			Hees			
Minimum interval (mins)	60	135	90	15	135	135
Std. threshold (mg)	3	8	2	7–8	8	7
Std. min. #axes	2	2	1	1	2	1
Value threshold (mg)	50	1–100	50	1–100	1–100	1
Value min. #axes	2	1–3	2	1–3	1–3	1

Results published in SR

www.nature.com/scientificreports

SCIENTIFIC REPORTS

natureresearch

OPEN

Evaluating the performance of raw and epoch non-wear algorithms using multiple accelerometers and electrocardiogram recordings

Shaheen Syed[1*], Bente Morseth[2], Laila A. Hopstock[3] & Alexander Horsch[1]

Can we do *even* better?

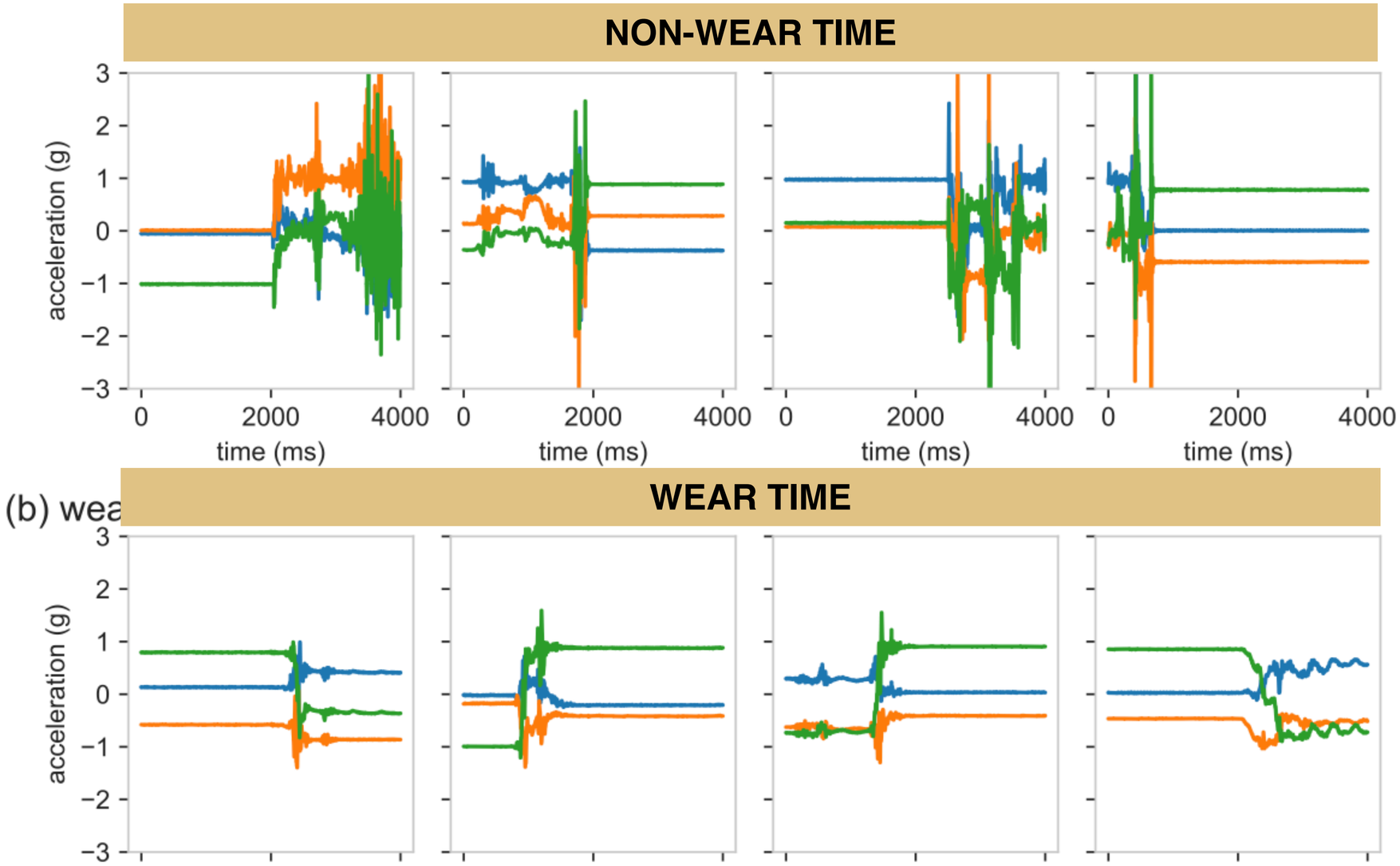
Deep Neural networks

Drawback current algorithms

Intervals need to be long enough to prevent false positives (type I errors),

while short enough to prevent false negatives (type II errors)

What happens before and after a candidate non-wear episode



(b) wea

Using Convolutional Neural Networks (CNNs)

INPUT
3-10 sec segments

v1

- **1D CNN** Kernel size : 10 Filters: 10
- **Fully Connected** Neurons : 50
- **Fully Connected** Neurons : 1

V2

- **1D CNN** Kernel size : 10 Filters: 10
- **1D CNN** Kernel size : 10 Filters: 10
- **1D CNN** Kernel size : 10 Filters: 10
- **Fully Connected** Neurons : 50
- **Fully Connected** Neurons : 50
- **Fully Connected** Neurons : 50
- **Fully Connected** Neurons : 1

V3

- **1D CNN** Kernel size : 50 Filters: 20
- **1D CNN** Kernel size : 50 Filters: 20
- **1D CNN** Kernel size : 50 Filters: 20
- **Fully Connected** Neurons : 50
- **Fully Connected** Neurons : 50
- **Fully Connected** Neurons : 50
- **Fully Connected** Neurons : 1

V4

- **1D CNN** Kernel size : 10 Filters: 10
- **Max Pooling** Pool size : 2
- **1D CNN** Kernel size : 10 Filters: 10
- **Max Pooling** Pool size : 2
- **1D CNN** Kernel size : 10 Filters: 10
- **Max Pooling** Pool size : 2
- **Fully Connected** Neurons : 50
- **Fully Connected** Neurons : 50
- **Fully Connected** Neurons : 50
- **Fully Connected** Neurons : 1

CNN classification results

CNN architecture v1

window size (sec)

	2	3	4	5	6	7	8	9	10
accuracy train	0.942	0.949	0.952	0.977	0.951	0.979	0.990	0.980	0.967
precision train	0.942	0.948	0.951	0.971	0.947	0.975	0.987	0.979	0.959
recall train	0.942	0.949	0.952	0.984	0.955	0.982	0.992	0.982	0.976
F1 train	0.942	0.948	0.952	0.977	0.951	0.979	0.990	0.980	0.967
accuracy val	0.927	0.934	0.942	0.968	0.936	0.970	0.981	0.975	0.950
precision val	0.937	0.932	0.938	0.952	0.920	0.957	0.966	0.952	0.933
recall val	0.911	0.932	0.943	0.985	0.952	0.982	0.996	1.000	0.968
F1 val	0.924	0.932	0.941	0.968	0.936	0.969	0.981	0.975	0.950
accuracy test	0.939	0.945	0.949	0.970	0.947	0.974	0.986	0.981	0.954
precision test	0.946	0.946	0.951	0.960	0.938	0.969	0.980	0.964	0.943
recall test	0.933	0.947	0.948	0.981	0.959	0.981	0.993	1.000	0.969
F1 test	0.939	0.947	0.949	0.971	0.948	0.975	0.987	0.982	0.956

CNN architecture v2

window size (sec)

	2	3	4	5	6	7	8	9	10
accuracy train	0.992	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
precision train	0.997	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
recall train	0.988	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
F1 train	0.992	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
accuracy val	0.987	0.995	0.995	0.994	0.994	0.993	0.993	0.991	0.992
precision val	0.982	0.990	0.990	0.988	0.988	0.987	0.985	0.982	0.983
recall val	0.991	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
F1 val	0.986	0.995	0.995	0.994	0.994	0.993	0.992	0.991	0.991
accuracy test	0.987	0.995	0.994	0.994	0.994	0.996	0.991	0.990	0.994
precision test	0.983	0.990	0.988	0.989	0.989	0.992	0.983	0.981	0.989
recall test	0.991	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
F1 test	0.987	0.995	0.994	0.995	0.995	0.996	0.992	0.991	0.994

CNN architecture v3

window size (sec)

	2	3	4	5	6	7	8	9	10
accuracy train	0.874	0.882	0.869	0.888	0.890	0.795	0.855	0.853	0.894
precision train	0.888	0.906	0.950	0.945	0.907	0.841	0.939	0.806	0.953
recall train	0.854	0.851	0.775	0.822	0.868	0.723	0.758	0.925	0.827
F1 train	0.871	0.878	0.854	0.879	0.887	0.777	0.839	0.862	0.885
accuracy val	0.876	0.875	0.837	0.874	0.889	0.807	0.858	0.846	0.897
precision val	0.895	0.881	0.954	0.901	0.901	0.725	0.963	0.796	0.912
recall val	0.846	0.860	0.699	0.833	0.868	0.972	0.737	0.919	0.872
F1 val	0.870	0.870	0.807	0.866	0.884	0.831	0.835	0.853	0.892
accuracy test	0.896	0.889	0.853	0.896	0.909	0.808	0.869	0.851	0.912
precision test	0.913	0.898	0.973	0.925	0.922	0.735	0.971	0.809	0.931
recall test	0.880	0.884	0.731	0.867	0.898	0.976	0.766	0.927	0.894
F1 test	0.896	0.890	0.835	0.895	0.910	0.839	0.857	0.864	0.912

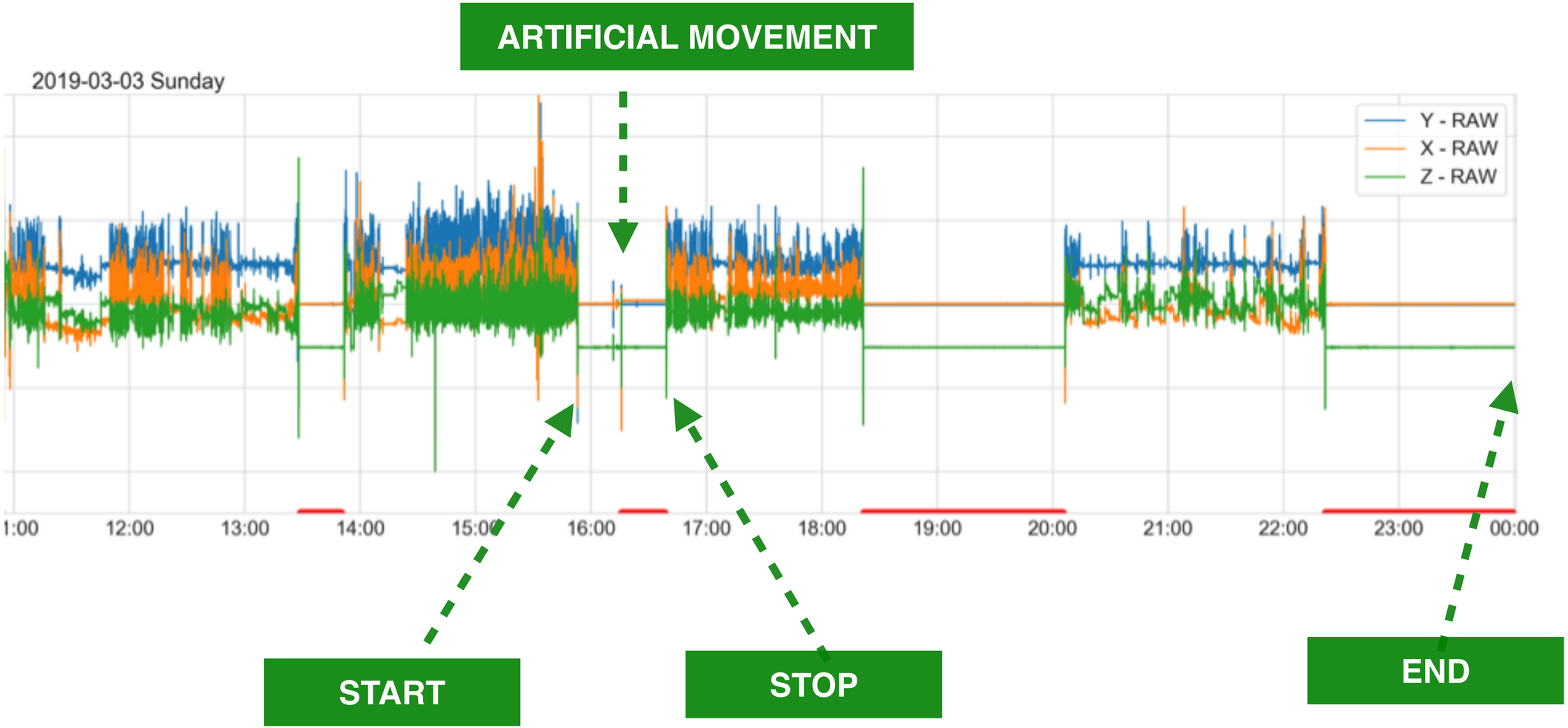
CNN architecture v4

window size (sec)

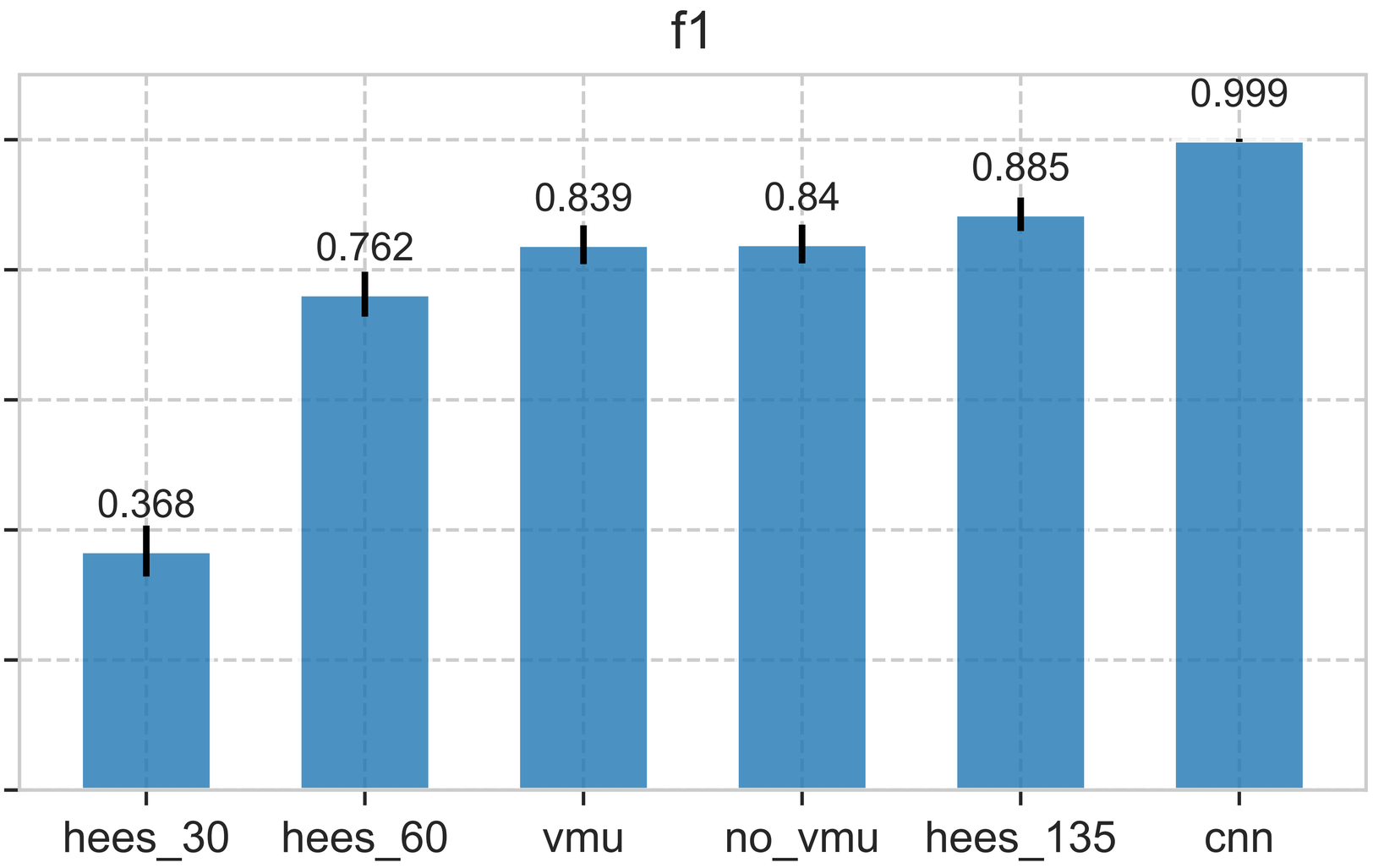
	2	3	4	5	6	7	8	9
accuracy train	0.907	0.923	0.915	0.922	0.932	0.928	0.916	0.930
precision train	0.916	0.930	0.931	0.927	0.937	0.937	0.926	0.941
recall train	0.896	0.913	0.896	0.914	0.924	0.916	0.902	0.917
F1 train	0.905	0.922	0.913	0.920	0.931	0.926	0.914	0.929
accuracy val	0.905	0.917	0.912	0.919	0.913	0.917	0.915	0.924
precision val	0.893	0.901	0.930	0.925	0.932	0.951	0.897	0.920
recall val	0.915	0.931	0.886	0.909	0.886	0.874	0.933	0.924
F1 val	0.904	0.916	0.907	0.917	0.908	0.911	0.914	0.922
accuracy test	0.919	0.932	0.925	0.933	0.932	0.928	0.922	0.938
precision test	0.914	0.923	0.948	0.944	0.952	0.962	0.906	0.935
recall test	0.928	0.944	0.904	0.924	0.913	0.895	0.945	0.944
F1 test	0.921	0.934	0.925	0.934	0.932	0.927	0.925	0.940

F1 test set = 0.995

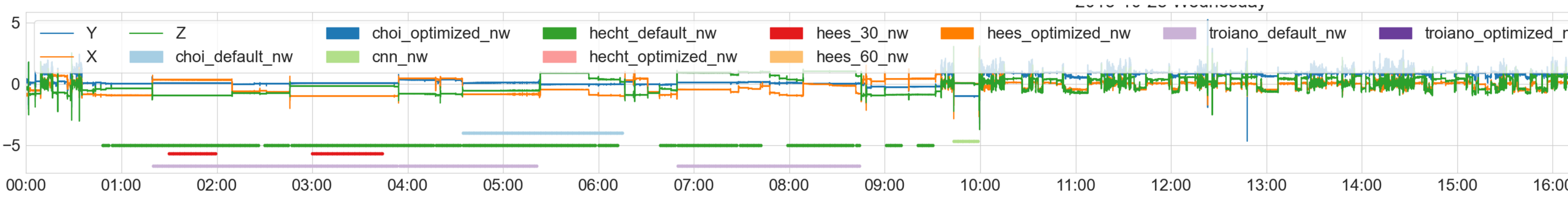
CREATING THE ALGORITHM



Classification Results on gold-standard dataset



Comparing Algorithms



Under review in SR

A novel algorithm to detect non-wear time from raw accelerometer data using convolutional neural networks

Shaheen Syed[1,*], Bente Morseth[2], Laila A. Hopstock[3], and Alexander Horsch[1]

[1]Department of Computer Science, UiT The Arctic University of Norway, Tromsø, Norway
[2]School of Sport Sciences, Faculty of Health Sciences, UiT The Arctic University of Norway, Tromsø, Norway
[3]Department of Community Medicine, Faculty of Health Sciences, UiT The Arctic University of Norway, Tromsø, Norway
[*]corresponding author: shaheen.syed@uit.no

https://doi.org/10.1101/2020.07.08.20148015

THANK YOU

Shaheen Syed, PhD

shaheen.syed@uit.no

https://github.com/shaheen-syed