Supplementary Information

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

Shaheen Syed^{1,*}, Bente Morseth², Laila A. Hopstock³, and Alexander Horsch¹

¹Department of Computer Science, UiT The Arctic University of Norway, Tromsø, Norway ²School of Sport Sciences, Faculty of Health Sciences, UiT The Arctic University of Norway, Tromsø, Norway

³Department of Community Medicine, Faculty of Health Sciences, UiT The Arctic University of Norway, Tromsø, Norway

*corresponding author: shaheen.syed@uit.no

December 21, 2020



Figure S1: Overview of ActiGraph and Actiwave Cardio data for a single participant. The candidate nonwear episodes are shown in the top figure and indicated by a red or green horizontal line and numbered. The candidate non-wear episodes are duration of activity not exceeding a standard deviation threshold of 4.0 mg (0.004 g). A major challenge is to infer what happened during those episodes, and here the second accelerometer (Actiwave Cardio) provides additional information. By focusing on the discrepancies between the two signals, made more clear in the bottom graph that shows the vector magnitude (VMU), we know that candidate nonwear episode 17 (colored green) is true non-wear time since there is recorded activity measured by the Actiwave Cardio accelerometer.



Figure S2: Graphical representation showing the correct and incorrect classifications of wear and non-wear time: (a) six minutes of labeled data is compared against six minutes of inferred data derived from a non-wear algorithm. Depending on the difference between the true labels and the inferred labels, each 1-minute interval is classified as a true negative or true positive if the labels match. Alternatively, intervals are classified as false negatives or false positives of the labels mismatch. (b) similarly as (a) but now shown as a confusion matrix.



Figure S3: Overview of training (blue) and validation (orange) loss, accuracy, precision, recall, F1, and area under the curve (AUC) while training the CNN V2 model with a window length of 3 seconds. This model was set to train for 250 epochs with early stopping enabled when the validation loss did not increase for 10 epochs. The model was trained for a total of 94 epochs. In addition, the weights of the model with the best validation loss were restored. More information on early stopping can be found in the TensorFlow documentation at https://www.tensorflow.org/api_docs/python/tf/keras/callbacks/EarlyStopping

	ACCUFACY minimum interval length (min) 15 30 45 60 75 90 105 120									precision minimum interval length (min) 15 30 45 60 75 90 10							
0.004	0.933	0.967	0.985	0.992	0.994	0.995	0.995	0.994	0.004	0.204	0.338	0.576	0.799	0.907	0.967	1	1
shold (g) 0.005	0.898	0.948	0.976	0.989	0.992	0.994	0.995	0.995	shold (g) 0.005	0.142	0.242	0.426	0.684	0.804	0.923	0.973	1
SD thre 0.006	0.877	0.936	0.971	0.987	0.991	0.994	0.994	0.994	SD thre 0.006	0.121	0.203	0.369	0.625	0.78	0.9	0.946	0.97
0.007	0.859	0.925	0.965	0.985	0.991	0.993	0.994	0.994	0.007	0.107	0.177	0.323	0.572	0.752	0.881	0.946	0.97
	recall minimum interval length (min)									f1 minimum interval length (min)							
			min	rec	call val length (i	min)						min	f imum inter	1 val length (i	min)		
	15	30	min 45	rec imum interv 60	call val length (i 75	min) 90	105	120		15	30	min 45	f imum interv 60	1 val length (i 75	min) 90	105	120
0.004	15 0.868	30 ' 0.803	min 45 0.783	rec imum inter 60 ' 0.765	call ral length (r 75 ' 0.742	min) 90 ' 0.742	105 ' 0.718	120 ' 0.708	0.004	15 ' 0.33	30 ' 0.476	min 45 0.663	f imum interv 60 0.782	1 ral length (i 75 0.816	min) 90 0.84	105 0.836	120 1 0.829
shold (g) 0.005 0.004	15 0.868 0.874	30 0.803 0.81	45 0.783 0.789	rec imum interv 60 0.765 0.775	call ral length (1 75 0.742 0.747	^{min)} 90 0.742 0.747	105 0.718 0.73	120 .708 0.72	shold (g) 0.005 0.004	15 0.33 0.244	30 - 0.476 0.372	45 0.663 0.554	f imum interv 60 0.782 0.727	1 val length (1 75 0.816 0.774	^{min)} 90 0.84 0.825	105 0.836 0.834	120 0.829 0.837
SD threshold (g) 0.006 0.005 0.004	15 0.868 0.874 0.88	30 0.803 0.81 0.813	45 - 0.783 0.789 0.789	rec imum interv 60 .765 0.775 0.775	call ral length (r 75 0.742 0.747 0.746	^{min)} 90 0.742 0.747 0.746	105 - 0.718 0.73 0.73	120 - 0.708 0.72 0.72	SD threshold (g) 0.006 0.005 0.004	15 0.33 0.244 0.212	30 - 0.476 0.372 0.325	45 0.663 0.554 0.503	f imum intern 60 0.782 0.727 0.692	1 val length (r 75 0.816 0.774 0.763	nin) ₉₀ 0.84 0.825 0.816	105 0.836 0.834 0.824	120 0.829 0.837 0.827

Figure S4: Classification performance metrics accuracy, precision, recall, and F1 for baseline models to detect non-wear time by using raw acceleration values in gravity units from all three orthogonal axes. This baseline model is referred to as the the XYZ non-wear algorithm.

	accuracy minimum interval length (min) 15 30 45 60 75 90 105 120									15	105	120					
0.004	0.817	0.911	0.963	0.984	0.991	0.994	0.995	0.994	0.004	0.0841	0.153	0.314	0.553	0.777	0.903	0.971	0.971
shold (g) 0.005	0.745	0.863	0.933	0.968	0.982	0.989	0.992	0.993	shold (g) 0.005	0.0618	0.103	0.192	0.35	0.52	0.692	0.831	0.877
SD thre 0.006	0.704	0.827	0.906	0.952	0.974	0.984	0.989	0.991	SD thre 0.006	0.0539	0.0837	0.143	0.25	0.403	0.563	0.701	0.761
0.007	0.67	0.797	0.882	0.935	0.963	0.976	0.985	0.989	0.007	0.0485	0.072	0.116	0.195	0.306	0.43	0.591	0.686
	recall									f1							
	15	30	45	60 1	75	90	105	120		15	30	45	60	75	90	105	120
0.004	0.876	0.818	0.791	0.777	0.754	0.754	0.738	0.728	0.004	0.153	0.258	0.45	0.646	0.765	0.822	0.839	0.832
shold (g) 0.005	0.879	0.818	0.791	0.777	0.754	0.754	0.738	0.728	shold (g) 0.005	0.115	0.184	0.31	0.482	0.616	0.721	0.782	0.796
SD three 0.006	0.886	0.819	0.791	0.777	0.754	0.754	0.738	0.728	SD three 0.006	0.102	0.152	0.242	0.379	0.525	0.645	0.719	0.744
0.007	0.886	0.819	0.791	0.777	0.754	0.754	0.738	0.728	2.007	0.092	0.132	0.202	0.312	0.436	0.548	0.657	0.706

Figure S5: Classification performance metrics accuracy, precision, recall, and F1 for baseline models to detect non-wear time by using the vector magnitude units (VMU) calculated as $\sqrt{acc_x^2 + acc_y^2 + acc_z^2}$, where acc_x , acc_y , and acc_z refer to each of the orthogonal axes. This baseline model is referred to as the VMU non-wear algorithm.