sensors

MDPI

Article

Sliding-Window Normalization to Improve the Performance of Machine-Learning Models for Real-Time Motion Prediction Using Electromyography

Taichi Tanaka [1,*], Isao Nambu [2], Yoshiko Maruyama [3] and Yasuhiro Wada [2]

[1] Department of Science Technology of Innovation, Nagaoka University of Technology, Nagaoka 940-2188, Japan

[2] Department of Electrical, Electronics and Information Engineering, Nagaoka University of Technology, Nagaoka 940-2188, Japan; inambu@vos.nagaokaut.ac.jp (I.N.); ywada@vos.nagaokaut.ac.jp (Y.W.)

[3] Department of Production Systems Engineering, National Institute of Technology, Hakodate College, Hakodate 042-8501, Japan; yoshiko@hakodate-ct.ac.jp

* Correspondence: taichi_tanaka@stn.nagaokaut.ac.jp

Abstract: Many researchers have used machine learning models to control artificial hands, walking aids, assistance suits, etc., using the biological signal of electromyography (EMG). The use of such devices requires high classification accuracy. One method for improving the classification performance of machine learning models is normalization, such as z-score. However, normalization is not used in most EMG-based motion prediction studies because of the need for calibration and fluctuation of reference value for calibration (cannot re-use). Therefore, in this study, we proposed a normalization method that combines sliding-window and z-score normalization that can be implemented in real-time processing without need for calibration. The effectiveness of this normalization method was confirmed by conducting a single-joint movement experiment of the elbow and predicting its rest, flexion, and extension movements from the EMG signal. The proposed method achieved 77.7% accuracy, an improvement of 21.5% compared to the non-normalization (56.2%). Furthermore, when using a model trained by other people's data for application without calibration, the proposed method achieved 63.1% accuracy, an improvement of 8.8% compared to the z-score (54.4%). These results showed the effectiveness of the simple and easy-to-implement method, and that the classification performance of the machine learning model could be improved.

Keywords: electromyography; EMG; z-score; signal normalization; machine learning; classification model

check for updates

Citation: Tanaka, T.; Nambu, I.; Maruyama, Y.; Wada, Y. Sliding-Window Normalization to Improve the Performance of Machine-Learning Models for Real-Time Motion Prediction Using Electromyography. *Sensors* **2022**, *22*, 5005. https://doi.org/10.3390/s22135005

Academic Editors: Nebojsa Malesevic and Anders Björkman

Received: 10 June 2022
Accepted: 29 June 2022
Published: 2 July 2022

1. Introduction

Electromyography (EMG) is a biological signal whose amplitude fluctuates when exercising or contracting muscles. Many researchers have used this property to research and develop devices that are aimed at expanding and recovering human motor function [1–5]. Due to its easy design, which does not need a dynamics model and any physical parameters and only uses data, machine learning models have been used in many studies including motion control for artificial hands and gesture recognition using classifiers, and control of walking aids and assistance suits by predicting joint angles, joint angular velocities, or joint torque using regressors [2–5]. Linear models such as logistic regression and support-vector machines were first used around 2000, with an emphasis on improving classification performance by the feature extraction method such as mean absolute value, waveform length, and short-time Fourier transform [5–8]. However, as classification performance significantly improved with the development of deep learning [9] that occurred in 2012, research was also conducted to improve classification performance by changing the configuration of the deep neural network [10–12]. However, improving classification performance it is limited by the study of the feature and machine learning model alone. Therefore, methods other than feature-extraction and machine learning models are required to improve classification performance.

Data normalization is one of the methods for improving the classification performance of machine learning models and is used in fields such as imaging and biometrics [13–15]. Methods that are often used include min–max normalization [15,16], which normalizes the value range of the dataset from 0 to 1; and z-score, which normalizes the dataset mean to 0 and standard deviation to 1 [15,17]. Even in the field of EMG, the classification performance of models is improved by normalizing signals and features with z-score and min–max normalization [16–20]. Although the EMG fields use normalization such as maximum voluntary contraction (MVC) [21] or maximum voluntary isometric contraction (MVIC) [22] to enable motor analysis and motor performance evaluation between muscles and subjects, normalization is hardly used in EMG-based motion prediction research. Normalization is thought to be rarely used in motion prediction research for two possible reasons. The first reason is the need to measure the reference value (e.g., min, max, mean, or standard deviation of each EMG channel) to carry out the calibration. It can take 30 s–3 min to use the application, depending on the measurement method. The second reason is that reference values such as the max and mean of each EMG channel fluctuate that are due to various factors such as muscle fatigue, electrode position, and fluctuations in skin impedance [23–27]. Therefore, a reference value, once measured, cannot be re-used. This reduces the practical applications of normalization and makes it unsuitable for real-time processing (i.e., online processing). Therefore, we aimed to devise a normalization method that does not require calibration (i.e., measurement of reference values) and that is suited for real-time processing to enable normalization to be used as a means of improving the prediction performance of machine learning models.

We propose a normalization method that uses the sliding-window [28] and z-score normalization [15,17] shown in Section 2.1. The z-score is a simple normalization method that sets the dataset mean to 0 and standard deviation to 1. Compared to min-max normalization, which uses the minimum and maximum values in the entire dataset, the z-score uses the mean and standard deviation of the entire data, making it less susceptible to outliers. However, the z-score is usually not suitable for real-time processing because the entire dataset needs to be used for normalization, which incurs a time delay. Therefore, we considered combining the sliding-window analysis (SWA) that is used for signal analysis with time-varying parameter analysis. SWA involves analyses that use the signal within a specified window length. It is thought that using a signal of a sufficient length can achieve the same effect as the z-score that uses the entire dataset.

In recent years, research has focused toward enabling other people's machine learning models to exhibit the same classification performance as machine learning models trained from their own data (i.e., improving generalizability) [29]. Studies that solve the problem of requiring individually specialized machine learning models by measuring a large amount of data for each user because of individual differences in myoelectric amplitudes have been reported. Methods have been proposed to reduce the required amount of own data by using other people's data with domain adaptation, which technology enables the use of models that were trained in different datasets, even in datasets with different data attributes [29–31]. Such methods include geodesic flow kernel (GFK) [32], correlation alignment (CORAL) [33], and transfer component analysis (TCA) [34], which conducts motion prediction using a machine learning model trained from a different dataset after projecting one's own data onto the data space, and domain adversarial neural networks (DANN) [35], which is a kind of deep learning method that trains the model to extract common features across different datasets.

The proposed method normalizes the standard deviation of myoelectric amplitude with individual differences, so it is thought that the influence of individual differences in myoelectric amplitude can be reduced, and the classification performance in the model learned from a different subject's dataset can be improved. Compared to previous research such as DANN, the proposed method trains machine learning models using data other than one's own data, so it is superior in that the models do not need to be trained for each user.

2. Methods

2.1. Proposed Sliding-Window Normalization

We propose a normalization method using sliding-window analysis (SWA) and z-score to improve the classification performance of the machine learning model and generalizability (i.e., exhibiting the same classification performance as the own machine learning model in the other's machine learning model). SWA is used for signal analysis and time-varying parameter analysis using the signal within a specified window length [28]. SWA enables time series analysis by sliding the window so that when a new sample is obtained, the sliding window replaces the oldest sample with the new sample. The z-score is a kind of normalization method that is used to improve the classification performance of models in machine learning. The features are normalized by setting the feature mean to 0 and the standard deviation to 1 [15,17].

The proposed method is a combination of these two concepts and is called sliding-window normalization (SWN). As shown in Equation (1), the mean and standard deviation of the samples in the sliding window are set to 0 and 1, respectively.

$$SWN\ EMG_{t,\ n-t+L_{\mathrm{norm}}} = (EMG_n - m_t)/s_t \ \ (t - L_{\mathrm{norm}} < n \leq t) \tag{1}$$

where t is the current discrete time, L_{norm} is the sliding window length, n is the discrete time number in the sliding window, EMG_i is the ith processed EMG, $SWN\ EMG_{t,\ n\text{-}t\text{+}L\mathrm{norm}}$ is the myoelectric signal to which the $n\text{-}t\text{+}L_{\mathrm{norm}}$th proposed method (SWN) is applied at the tth, and m_t and s_t are the myoelectric mean and standard deviation on the tth sliding window, respectively. We used the "mean" and "std" functions in numpy in Python.

2.2. Comparison Methods

As comparison methods to SWN, applying z-score and none (without normalization).

2.2.1. Z-Score

Z-score sets mean to 0 and standard deviation to 1 on a dataset [15,17]. Here, normalizing train and test dataset are based on train data like Equation (2).

$$Z - Scored\ EMG_{t,\ \ d,s} = (EMG_{t,\ \ d,\ \ s} - \mu_{train,\ \ s})/\sigma_{train,\ \ s} \tag{2}$$

where t is the current discrete time, d means the train data or test data, s is the subject number, $EMG_{t,d,s}$ is the tth processed EMG on sth subject, $Z\text{-}Scored\ EMG_{t,d,s}$ is the myoelectric signal to which tth z-score is applied at the tth processed EMG on sth subject, and $\mu_{train,\ s}$ and $\sigma_{train,\ s}$ are the myoelectric mean and standard deviation on sth subject's training data.

2.2.2. None (Without Normalization)

None apply nothing in the normalization process (Section 2.4).

2.3. Evaluation Method

This paper evaluates three types of items. The first is the improvement of the classification performance of machine learning models when the proposed method (SWN) is applied (Section 2.3.1), the second is the improvement of generalizability of machine learning models when the proposed method (SWN) is applied (Section 2.3.2), and the third is the improvement of the classification performance of the machine learning model when the number of subjects of the model that was trained with different data is increased by applying the proposed method (SWN) (Section 2.3.3).

Two types of machine learning models need to be trained. The first is the model trained with one's own data (model type of OWN). The second is the model trained with another person's data (model type of OTHER). Sections 2.3.1 and 2.3.2 used models OWN and OTHER. Section 2.3.3 used only OTHER. Performance (OWN) involved dividing the data into training and testing datasets and calculating the performance using the model trained with one's own training data and own test data. Performance (OTHER) involved calculating the performance using the model trained with another person's training data and one's own test data. The training data and test data was created by randomly dividing them into a 1:1 ratio every 10 consecutive trials.

2.3.1. Normalization Evaluation

The evaluation of model classification performance improvement by the proposed method (SWN) was conducted by comparing the "classification performance in the model with SWN (OWN or OTHER)" and "classification performance of the model with z-score or None (OWN or OTHER)".

Improvements in model classification performance that were due to the proposed model will be indicated by higher performance and lower standard deviation in performance. We consider the model classification performance improved and the research objective achieved when the performance (OWN or OTHER) with SWN applied is equal to or greater than the performance (OWN or OTHER) with z-score or None (no normalization).

2.3.2. Generalizability Evaluation

The evaluation of generalizability was conducted by comparing the "classification performance in the model trained with one's own data (OWN)" with the "classification performance in the model trained with another person's data (OTHER)".

Better generalizability is indicated by higher performance and lower standard deviation. Generalizability is considered improved and the research objective achieved when the performance (OTHER) with SWN applied is equal to or greater than the performance (OWN) without normalization (None) applied, and the performance (OTHER) with SWN applied is equal to or greater than the performance (OWN) with SWN applied.

2.3.3. Evaluation on SWN Increased Number of Subject to Train Model

We investigated whether the classification performance of the model could be improved by increasing the number of subjects used for learning the model. We compared a model trained with nine subjects (OTHER) with a model trained with one subject (OTHER). The model trained with nine subjects (OTHER) was considered better if its performance was higher and its performance had a lower standard deviation.

2.3.4. Evaluation Index

The accuracy shown in Equation (3) was used as the evaluation index for the classification performance of the machine learning model. Accuracy is an evaluation index that can simply compare results with multiple targets.

$$\text{Accuracy} = \frac{\text{Success Predictions}}{\text{Success Predictions} + \text{Failure Predictions}} \tag{3}$$

The Wilcoxon rank-sum test was used for significance tests. The significance level was set for the p-value less than 0.05. The "ranksums" function in scipy.states in Python was used for implementation. The "multipletests" function in statemodels.sandbox.stats.multicomp in Python was used for multiple comparisons. We used the Bonferroni correction as the correction method for the p-value.2.3.5. Machine Learning Model

We chose multi-class logistic regression for the machine learning model, which allows multi-class classification and short training time, to easily confirm the improvement by the proposed SWN. The "LogisticRegression" function in scikit-learn in Python was used for implementation. The parameters were as follows: penalty = "none", class_weight = "balanced", and max_iter = 6000. This model transforms the feature that is extracted from EMG (Session 2.4) to elbow-joint movement: rest, flexion, or extension (Session 2.6). The number of models trained was calculated by ${}_{\text{number of subject}}C_{\text{number of subject to train}}$.

2.4. EMG Processing

Before training the machine learning model, the measured EMG underwent preprocessing, normalization, feature extraction, and decimation.

Preprocessing involved the application of a low-pass Butterworth filter (3rd order, 500 Hz), decimation (2000 → 500 Hz), and a high-pass Butterworth filter (3rd order, 30 Hz). We used the scipy.signal "butter" function and "sosfilt" in Python for implementation. Normalization involved the application of either SWN, z-score, or no normalization (i.e., None). The window length for SWN was set at between 100 and 500 ms, with 100 ms intervals, because too long a window length decreases the amount of data. To adjust the amount of data, the data near the beginning of the trial are reduced based on the longest window length. In the case of "z-score and None", the obtained features did not change even when the normalization window length was changed.

Feature extraction involved the calculation of the following six features to investigate window length for normalization and feature-extraction, for which high classification performance was obtained in previous studies: mean absolute value: MAV (Equation (4)) [6], mean waveform length: MWL (Equation (5)) [7], and difference root mean square: DRMS (Equation (6)) [7] as time-dimension features, short-time Fourier transform: STFT [5], and stationary wavelet transform: SWT [8] as frequency-dimension features, and combination of all five features: ALL. STFT involved averaging in the 1–70 Hz (low component), 60–100 Hz (middle component), and 100–250 Hz (high component) ranges and concatenating them (Equation (7)). SWT involved time-frequency conversion using Daubechies wavelet 2 (db2) as the mother wavelet and taking the absolute mean of the wavelet coefficient of level 3 frequency (cD3) as the feature.

$$MAV_t = 1/L_{\text{feature}} \sum_{n=0}^{L_{\text{feature}}-1} |EMG_{t-n}| \tag{4}$$

$$MWL_t = \frac{1}{L_{\text{feature}}-1} \sum_{n=0}^{L_{\text{feature}}-2} |EMG_{t-n} - EMG_{t-n-1}| \tag{5}$$

$$DRMS_t = \sqrt{\frac{1}{L_{\text{feature}}-1} \sum_{n=0}^{L_{\text{feature}}-2} (EMG_{t-n} - EMG_{t-n-1})^2} \tag{6}$$

$$\begin{aligned} STFT_t &= cat(Low, Mid, Hig) \\ Low &= \tfrac{1}{bin_{\text{low}}} \sum_{freq=1}^{70} \text{MeanSpec}_{freq}\left(EMG_{(t-L+1)-t}\right) \\ Mid &= \tfrac{1}{bin_{\text{mid}}} \sum_{freq=60}^{100} \text{MeanSpec}_{freq}\left(EMG_{(t-L+1)-t}\right) \\ Hig &= \tfrac{1}{bin_{\text{hig}}} \sum_{freq=100}^{250} \text{MeanSpec}_{freq}\left(EMG_{(t-L+1)-t}\right) \end{aligned} \tag{7}$$

where t is the current discrete time, $\text{L}_{\text{feature}}$ is the window length of feature extraction, cat(·) is the concatenation function, *freq* is the frequency, MeanSpec is the function that outputs the spectrogram averaged in the time direction, and *bin* is the number of discrete frequencies in each of the low/middle/high frequencies. A Hanning window with a window length of 64 samples was used for the STFT window function. The functions in scipy.signal in Python were used for implementation. SWT is a method that improves the position invariance, which was a problem of wavelet transforms (WT), and the same mother wavelet as in WT can be used. The "swt" function in the pywt module in Python was used for implementation. The window length of feature extraction was set between 100 and 500 ms, with 100 ms intervals, because too long a window length decreases the amount of data.

Finally, decimation involved reducing the sampling rate of the features from 500 Hz to 20 Hz to reduce the amount of data and shorten the training time of the machine learning model.

2.5. Data Acquisition

2.5.1. Subjects

The ethics board of the Nagaoka University of Technology approved this study according to the Declaration of Helsinki. The subjects were 10 right-handed 22- to 23-year-old men. The subjects were informed about the experiment in advance and consented to participate in the experiment.

2.5.2. Experiment

The positions of the hands, elbows, and shoulders, and the EMG of the forearm and upper arm muscles, were measured as in the experimental environment shown in Figure 1A. Subjects performed 12 types of elbow single-joint movements with four different start points and end points as tasks (Figure 1B). A task involves moving from one of the four points (start point) to one of the other three points (end point). Each trial consisted of pre-rest (2 s), task (2.5 s), and post-rest (0.1 s); 36 trials (12 movements × 3) were conducted in one session, for a total of 10 sessions (i.e., 360 trials). The tasks were randomly selected for each session. The following four rules were also set as the success conditions for the asks.

(1) No exercise during the rest period. Elbow joint angular velocity does not exceed 2 deg./s during the rest period.
(2) End the task during the task period. End the task between 0–2.5 s.
(3) Place the elbow joint angle at the start point (±2°.) during the rest period and at the end point (±6°.) at the end of the task.
(4) Place the shoulder and elbow joints within 3 cm of the initial position between the pre-rest and post-rest.

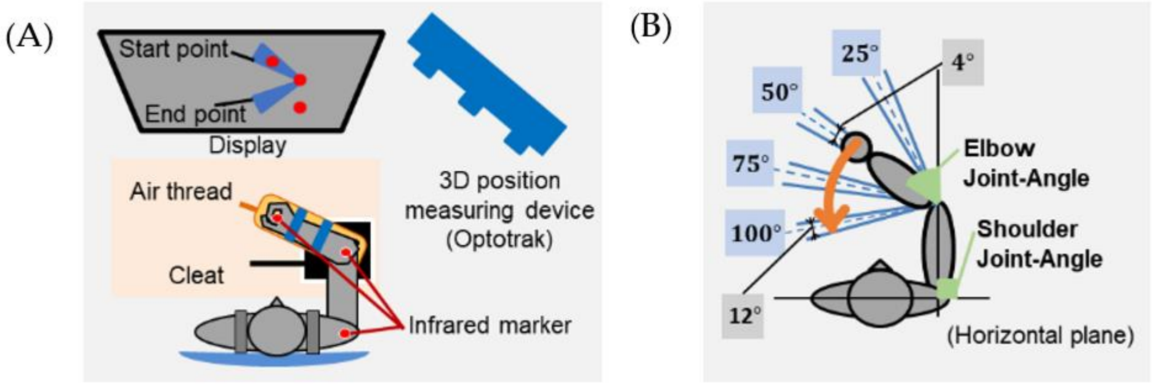


Figure 1. Experiment Condition: (**A**) state, (**B**) task.

The position data were measured at three locations, namely, the hand, elbow, and shoulder, using Optotrack Certus, (NDI Inc., Waterloo, Canada, sampling rate: 500 Hz). The EMG was measured at the biceps brachii (×4), brachialis (×1), brachioradialis (×1), anconeus (×1), triceps brachii (outside) (×2), triceps brachii (long head) (×2), and extensor carpi radialis longus (×1), totaling 12 locations, by using Trigno Lab Avanti (Delsys, Natick, MA, USA, sampling rate: 2000 Hz)

2.6. Position Processing

Position processing consisted of noise reduction, work space → joint angle space conversion, elbow joint angular velocity conversion, coding, and decimation to obtain the target (rest, flexion, and extension of elbow joint movement) from the positions of the hands, elbows, and shoulders obtained in the subject experiment.

For noise reduction, we applied a zero-phase low-pass Butterworth filter (2nd order 20 Hz). The Python scipy.signal “butter” and “sosfiltfilt” functions were used.

Work space → joint angle space conversion involved the conversion of the positions of the hand, elbow, and shoulder to the elbow and shoulder joint angles using Equation (8).

$$\begin{cases} \theta_{sld} = \text{atan2d}(a,\ b) - \text{atan2d}\left(\sqrt{a^2+b^2-c^2},\ c\right) \\ \theta_{elb} = \text{atan2d}\left(\sqrt{a^2+b^2-c^2}, c\right) + \text{atan2d}\left(\sqrt{a^2+b^2-d^2},\ d\right) \end{cases}$$
$$a = y_{hand} - y_{sld}$$
$$b = x_{hand} - x_{sld} \tag{8}$$
$$c = \left(a^2 + b^2 + {L_{sld}}^2 - {L_{elb}}^2\right)/2L_{sld}$$
$$d = \left(a^2 + b^2 - {L_{sld}}^2 + {L_{elb}}^2\right)/2L_{elb}$$

where atan2d(y, x) is the function that calculates the angle [deg.] from the two-dimensional coordinate position, x_{hand} and y_{hand} are the hand position [m], x_{sld} and y_{sld} are the shoulder position [m], and L_{sld} and L_{elb} are the upper arm and forearm length, respectively [m].

Elbow joint angular velocity conversion involved the conversion of the joint angle to the joint angular velocity using Equation (9).

$$\dot{\theta}_{elb,t} = (\theta_{elb,\ t+1} - \theta_{elb,\ t})f_s \tag{9}$$

where $\dot{\theta}_{elb,t}$ is the elbow joint angular velocity at the discrete time t, and f_s is the sampling frequency.

Coding involved the conversion of the elbow joint angular velocity to the target using Equation (10). This target was used as the teacher data for model training.

$$\text{target}_t = \begin{cases} \text{flexion } \left(\dot{\theta}_{elb,t} \geq 2.0\ [\text{deg.}/\text{s}]\right) \\ \text{extension } \left(\dot{\theta}_{elb,t} \leq -2.0\ [\text{deg.}/\text{s}]\right) \\ \text{rest (otherwise)} \end{cases} \tag{10}$$

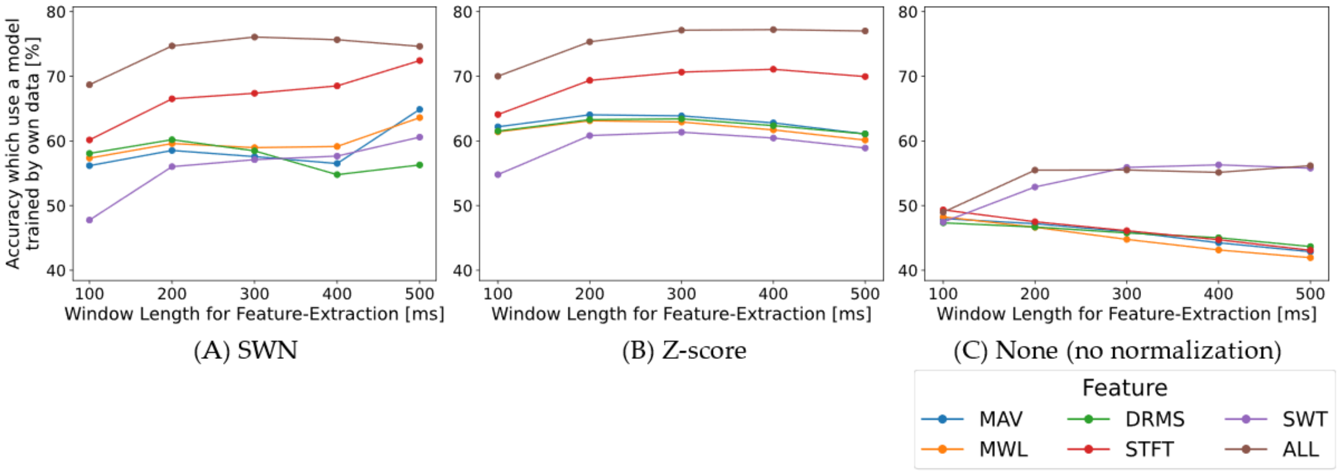
3. Results

Prior to evaluating the classification performance (Section 3.2) and generalizability (Section 3.3) of the machine learning model by the proposed SWN method, we investigated the effects of window length for feature extraction and normalization (Section 3.1). Thereafter, we investigated the effect of the number of subjects used in model OTHER (Section 3.4). The chance level of accuracy in all results was 33.3% (3 classes: rest, flexion, and extension).

3.1. Effect of Window Length

In model OWN, we investigated the effect of changing window length for feature extraction and normalization on accuracy.

First, the effect of window length for feature extraction on accuracy was investigated. Figure 2 shows the results of changing the window length for feature extraction between 100 and 500 ms in 100 ms intervals and comparing the proposed SWN (the window length fixed at 500 ms), z-score, and None (no normalization) for the six types of features. Figure 2A shows that applying SWN improved accuracy as the window length for feature extraction increased. In contrast, with z-score and no normalization, the accuracy decreased as the window length for feature extraction increased (Figure 2B,C). We surmise that applying SWN improves the classification performance of the model by lengthening the feature extraction window.



(A) SWN (B) Z-score (C) None (no normalization)

Figure 2. Effect of window length on feature-extraction (OWN). Window length for SWN is fixed at 500 ms. Horizontal axis indicates window-length for feature-extraction and vertical axis indicates accuracy using a model trained by data from own subject. Each color line shows results for the type of feature-extraction. (**A**) the results with the SWN (window length is fixed 500 ms), (**B**) the results with the z-score, and (**C**) the results with None (no normalization).

Next, the effect of window length for normalization on accuracy was investigated. We changed the window length for normalization between 100 and 500 ms at 100 ms intervals, and the window length for feature extraction was fixed at 500 ms. Figure 3 shows the results of calculating with all six feature types. The accuracy fundamentally increases with the window length for normalization. We recommend that the window length for normalization should be selected within the range of 200–500 ms, with the window length that maximizes accuracy being selected.

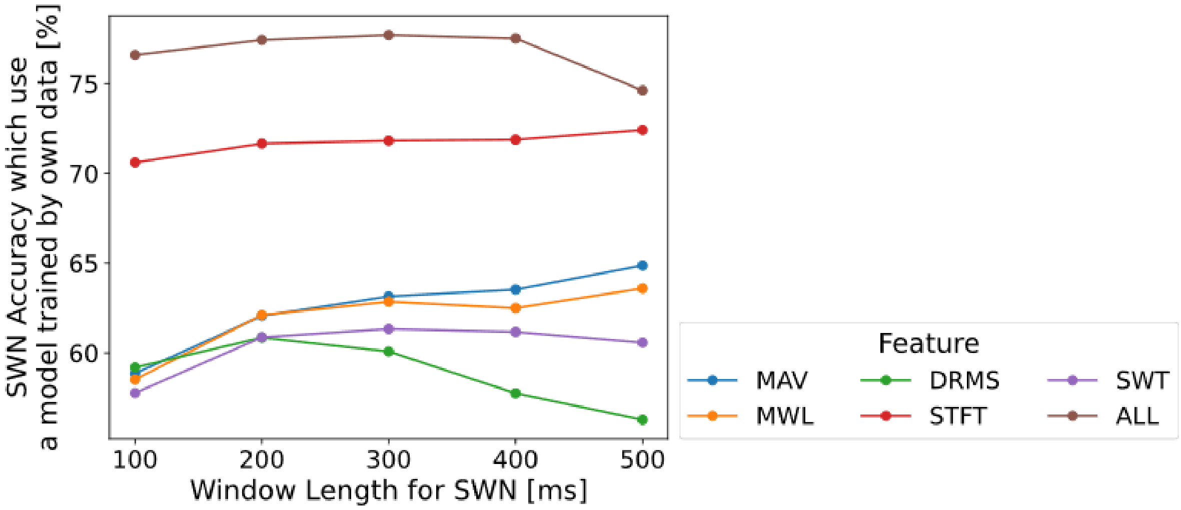


Figure 3. Effect of the window length for normalization (OWN). The window length for feature-extraction is fixed at 500 ms. Horizontal axis indicates window-length for normalization and vertical axis indicates accuracy using a model trained by data from own subject. Each color line shows results for the type of feature-extraction.

We also investigated whether there was any synergy between normalization and feature extraction window length, but no synergistic effects were observed. This result is shown in Appendix A.

3.2. Comparison of Normalization Methods

We investigated whether the proposed method (SWN) would improve the classification performance of the models using all the features (ALL). Research in recent years has been conducted to reduce the pre-data measurement of each user by enabling others' machine learning models to exhibit the same classification performance as one's own model (i.e., improving generalizability). Therefore, in this study, we compared the accuracy of SWN, z-score, and non-normalization (None) between a model learned from one's own data (OWN) and a model learned from other subjects' data (OTHER). The window lengths for normalization and feature extraction were changed between 100 and 500 ms in 100 ms intervals, and the maximum accuracy was compared. The number of subjects used when training model OTHER was set to nine people.

Figure 4 shows a result comparing the accuracy of the model (OWN or OTHER) with SWN, z-score, and no normalization (None). A comparison between SWN_OWN (accuracy: 77.7 ± 2.9%, blue bar) and None_OWN (accuracy: 56.2 ± 7.1%, orange bar) shows that the mean accuracy of SWN_OWN significantly increased by 21.5% (Wilcoxon rank-sum test, $p < 0.001$) and its standard deviation of accuracy decreased by 4.9%. A comparison between SWN_OWN (accuracy: 77.7 ± 2.9%, blue bar) and z-score_OWN (accuracy: 77.2 ± 2.4%, green bar) shows that SWN demonstrated the same performance as z-score on the model type OWN ($p > 0.05$). These results show that the proposed SWN can improve the accuracy of machine learning model, much like the z-score when using the machine learning model that was trained from one's own data. Furthermore, a comparison between SWN_OTHER (accuracy: 63.1 ± 5.1%, blue shaded bar) and None_OTHER (accuracy: 41.4 ± 11.3%, green shaded bar) shows that the accuracy of SWN_OTHER significantly increased by 21.6% ($p < 0.01$) and its standard deviation of accuracy decreased by 6.2%. A comparison between SWN_OTHER (accuracy: 63.1 ± 5.1%, blue shaded bar) and z-score_OTHER (accuracy: 54.4 ± 8.5%, orange shaded bar) shows that the accuracy of SWN_OTHER significantly increased by 8.8% ($p < 0.01$) and its standard deviation of accuracy decreased by 3.4%. These results show that the proposed SWN can improve the accuracy compared to the z-score when using other's machine learning models. These two results show the effectiveness of the proposed method.

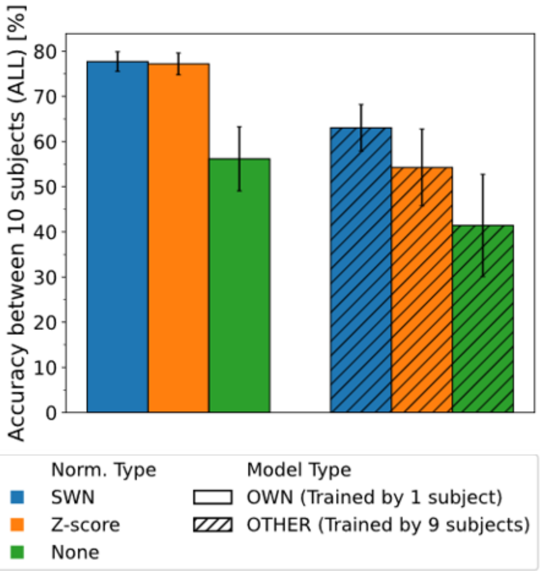


Figure 4. Performance comparison between normalization methods and model types using feature ALL. Horizontal axis indicates normalization methods and model types, and vertical axis indicates accuracy using a model trained by data from other or own subjects using feature ALL. SWN is the result of applying proposed normalization, z-score is the result of applying compared normalization and None is the result of not applying normalization.

3.3. Generalizability Comparison

We investigated whether the proposed SWN method would improve generalizability (i.e., other's machine model would exhibit the same classification performance as one's own model). A comparison was made between the accuracy of model OTHER with SWN applied that was used in Section 3.2 in Figure 4 and model OTHER where normalization was not applied.

From Figure 4, a comparison between SWN_OTHER (accuracy: 63.1 ± 5.1%, blue shaded bar) and None_OWN (accuracy: 56.2 ± 7.1%, green bar) shows that SWN_OTHER had an accuracy that was 6.9% higher ($p < 0.05$) and standard deviation of accuracy that was 2% lower. However, a comparison between SWN_OWN (accuracy: 77.7 ± 2.2%, blue bar) and SWN_OTHER (accuracy: 63.1 ± 5.1%, blue shaded bar) shows that SWN_OWN had an accuracy that was 14.7% higher ($p < 0.001$) and standard deviation of accuracy that was 3% lower. These results show that the classification performance of the machine learning model was improved by the proposed SWN, but even a model that used a large amount of other's data did not improve generalizability to the extent that it was similar to the classification performance using one's own data.

3.4. Number of Subjects to Train Model (OTHER)

It was shown in Section 3.2 that applying the proposed SWN method could improve the classification performance of not only the model trained from one's own data (OWN) but also the model trained from other user's data (OTHER). Therefore, investigating the extent to which the classification performance of the model (OTHER) could be improved by training the model by mixing the data of multiple other subjects. The number of subjects used for training the model was changed from 1 to 9. The window lengths for normalization and feature extraction were changed in the range of 100 to 500 ms in 100 ms intervals, and the maximum accuracy was compared. All the features (ALL) were used for the classification. From Figure 5, accuracy for feature ALL for cases of proposed SWN and z-score increased with subjects used in model training. In contrast, the accuracy did not either monotonically increase or decrease with respect to the number of subjects for cases without normalization (None). This implies that high classification performance can be achieved with an increase in the number of subjects by applying proposed SWN in cases that use others' data.

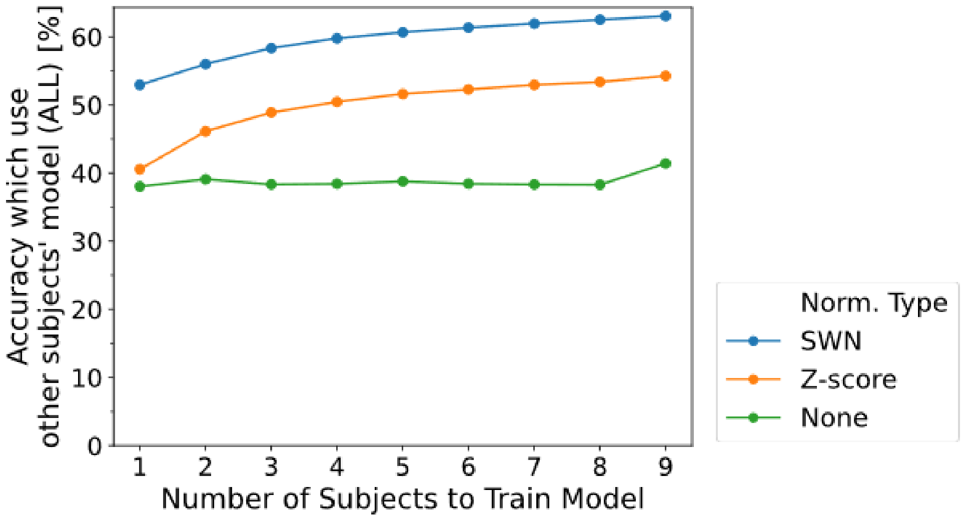


Figure 5. Effect of the number of subjects to train model (OTHER). Horizontal axis indicates number of subjects to train model (OTHER) and vertical axis indicates accuracy using a model trained by data from other subjects. Each color line shows results for the type of normalization.

Next, to investigate whether the increase in the number of subjects had a significant effect, we compared cases with either nine subjects (highest accuracy in Figure 5) and one subject (lowest accuracy in Figure 5) used in the training of the machine learning model. The results show $p < 0.01$ on proposed SWN, $p < 0.001$ on z-score, and $p \geq 0.05$ on None (no normalization). It implies increasing the accuracy by normalization (proposed SWN and z-score).

4. Discussion

In this study, we proposed a new normalization method, SWN, to improve the classification performance of machine learning models. We succeeded in increasing classification accuracy from 56.2% to 77.7%, an increase of 21.5%, by applying the SWN (blue and green bar with no line in Figure 4). Furthermore, the standard deviation of accuracy decreased from 7.1% to 2.9%, a decrease of 4.9%. The results show the effectiveness of the proposed method.

In this section, we discuss the performance of SWN compared with z-score and no normalization (Section 4.1), the parameters and features selection on SWN (Section 4.2), the factors that improve the model classification performance by the proposed method (Section 4.3), and the feasibility of real-time prediction (Section 4.4).

4.1. Performance of SWN

The proposed SWN can improve a classification accuracy (OWN) because the proposed SWN (77.7%) has a 21.5% higher accuracy than no normalization (56.2%) from Figure 4 in Section 3.2. However, the proposed SWN has the same accuracy as the z-score (77.2%) and is not better than the z-score. The advantage of the proposed SWN is that it normalizes EMG signals in each sliding window and does not need a reference value (e.g., min, max, mean, or standard deviation of each EMG channel). However, the z-score method normalizes the signal using all the data. If the measurement is done across days, the EMG signals may vary between days and the normalization could negatively affect the accuracy. The same effects are likely to occur in the case when the sensor placement changes and muscles fatigue. Therefore, we need to investigate whether the proposed SWN is better than the z-score using data that changes depending on the measurement day, sensor location, and muscle fatigue statement.

In recent years, research has focused on enabling other people's machine learning models (model type of OTHER) to exhibit the same classification performance as machine learning models trained from their own data (model type of OWN). Therefore, we investigated whether the SWN proposed in Section 3.3 could deliver the same or higher performance than the model trained on our own data. As a result, SWN_OTHER (63.1%) had a 14.7% lower accuracy than SWN_OWN (77.7%); however, it was higher than z-score_OTHER (54.4%) and None_OTHER (41.4%). The proposed SWN has better model accuracy when using other people's data than the z-score. This could be because SWN can normalize the myoelectric signal within the sliding window and the difference of data between subjects are reduced while z-score and None are influenced by such differences. This point is the advantage of the proposed SWN compared with the z-score. Furthermore, the accuracy increased with the number of subjects to train model (OTHER), much like the previous study [31,36,37]. Therefore, the proposed SWN has the same effect as the previous study's methods. However, similar to previous studies, a large amount of subjects' data is needed to obtain high classification accuracy when applying the proposed SWN to model OTHER.

4.2. Parameters and Features Selection of SWN

The parameters of SWN are the window length for normalization and feature extraction. They should be fundamentally set to long to improve the accuracy of the model when applying the proposed SWN. The window length for normalization should be set between 200–500 ms and the window length for feature extraction should be set at 500 ms from Figures 2 and 3. Furthermore, the effect of window length was investigated by using data with a short trial of 4 s in this paper. However, if the data length is more than 4 s, increasing the window length for normalization and feature extraction to more than 500 ms may improve the accuracy of the model. Therefore, we need to investigate the effect of the window length on normalization and feature extraction for data, where one trial of the measurement experiment is longer than 10 s.

The feature STFT has the highest accuracy on SWN from the five feature types: MAV, MWL, DRMS, STFT, and SWT, as shown Figure 2. Although SWT has the highest accuracy, the other four features have almost the same accuracy on no normalization (None). Thus, even though high accuracy was obtained in the previous study, it may not be possible to obtain it in the case of the proposed SWN. Additionally, using multiple features (feature ALL) has a higher accuracy than single features (MAV, MWL, etc.) as shown in Figures 2 and 3. Hence, classification accuracy can be enhanced by incorporating multiple features.

4.3. Analysis of SWN

We investigated the effect of dividing with the standard deviation of EMG, which was thought to have led to the improvement of the classification performance of machine learning models and is a feature of SWN. Visualizing the relationship of standard deviation of EMG and the feature of EMG by drawing a confidence ellipse with a standard deviation of 2. The “confidence_ellipse” function of matplotlib in Python was used for implementation. Figure 6 shows an example of the results of treating MAV as a representative of the features. The S.D. of EMG-MAV distribution in the case with normalization (SWN) had a weakly negative or no correlation, whereas the distribution in the case without normalization (None) had a strongly positive correlation.

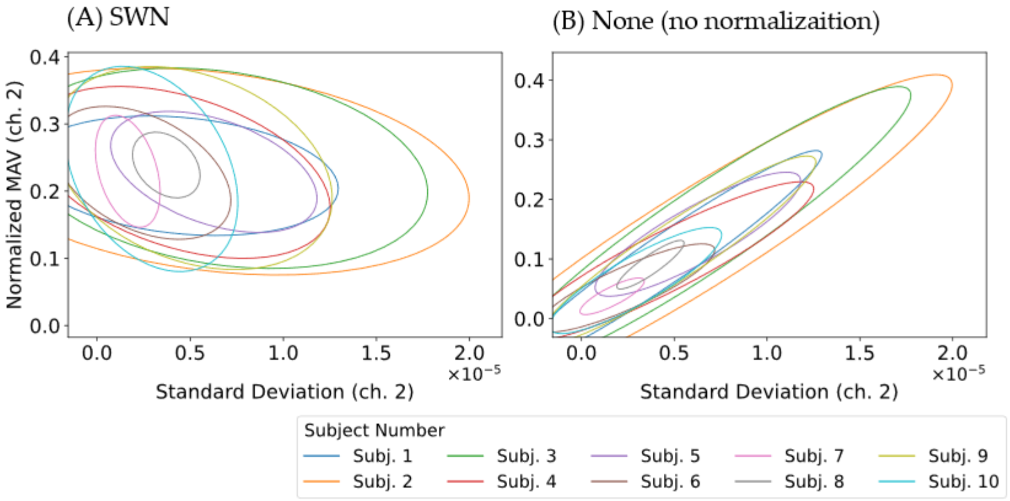


Figure 6. Example of the distribution analysis between standard deviation of EMG and feature each subject on the channel 2 (MAV). (**A**) Normalization (SWN), (**B**) No Normalization (None). Both MAV of SWN and None are normalized (max: 1.0) by maximum data in the 10 subjects.

The results obtained in Figure 6 are used as a basis for conducting a correlation analysis of cases with normalization (SWN) and without normalization (None). The representative feature was MAV, which was the same as in Figure 6. We calculated the correlation coefficient of the S.D. of EMG vs. MAV for each channel and subject, taking the mean value. The correlation coefficient was −0.33 on the with normalization (SWN) and 0.90 on the without normalization (None). Therefore, the S.D. of EMG and MAV had a weakly negative correlation for cases with normalization (SWN) and a strongly positive correlation for cases without normalization (None). These results imply that one of the factors that improved the classification performance of the machine learning model was the reduction of the influence of the standard deviation on the features by the proposed SWN method.

4.4. Comparison of Calculation Time

SWN was effective in improving the classification performance of the machine learning model. However, it is still unknown whether this can satisfy the required execution speed for real-time processing. Therefore, the preprocessing and normalization shown in Section 2.4 were executed at intervals of 20 ms (50 Hz), and the mean execution time was compared between cases with normalization (SWN) and without normalization (None). The execution environment was as follows: Intel(R)Core (TM) i7-9700K CPU @ 3.60 GHz, Python 3.8.12. The result shows 409 µs (SWN) and 333 µs (None). The normalization rate in 409 µs was 18.6%, a minimal effect. It implies the proposed SWN can be implemented on a low-computing-power device such as a microcomputer. These results indicate that the proposed SWN method can be implemented in real time.

5. Conclusions

In this paper, we proposed a normalization method (SWN) that used the sliding window and z-score to improve the classification performance of devices using EMG. Applying SWN improved the accuracy by 21.5% compared to the case without normalization. Even when a machine learning model that was trained with other's data was used, the accuracy improved by 21.6% compared to the case without normalization and 8.8% compared to the case with z-score. These results show that the classification performance of the machine learning model could be improved by the proposed method (SWN). Results of investigating the relationship between the standard deviation and features also show that applying the SWN changed the correlation between the standard deviation and features from that of a strongly positive one to a weakly negative one. This was assumed to be one of the factors that improved the classification performance of machine learning models.

The focus of future studies will be on the following two points. First, we found that the proposed SWN has a higher accuracy than the z-score on the model using other people's data. However, the proposed SWN has almost the same accuracy as the z-score in the case of using own data. To determine whether the proposed SWN is superior to the z-score, we need to analyze in detail whether it is robust to various data attributes such as measurement day, sensor location, and muscle fatigue. Second, we applied the proposed SWN to the classification model. We need to investigate whether the proposed SWN can improve the performance of the regression model that predicts kinematic parameters such as the joint angle and angular velocity.

Author Contributions: Conceptualization, T.T.; methodology, T.T.; software, T.T.; validation, T.T.; formal analysis, T.T.; investigation, T.T.; resources, I.N. and Y.W.; data curation, T.T.; writing—original draft preparation, T.T.; writing—review and editing, T.T., I.N., Y.M. and Y.W.; visualization, T.T.; supervision, I.N., Y.M. and Y.W.; project administration, T.T.; funding acquisition, I.N. and Y.W. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partly supported by JSPS KAKENHI: Grant Numbers 21H03287 and 21H03480, and Tateisi Science and Technology Foundation: Grant Number 2227008.

Institutional Review Board Statement: The study was conducted according to the guidelines of the Declaration of Helsinki and approved by the Ethics Committee of Nagaoka University of Technology (protocol code: R3-5, 30 April 2021).

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A Effect of Window Length for Feature-Extraction and Normalization

In Section 3.1, we investigated the effect of window length by fixing the window length for feature extraction or normalization at 500 ms. In this session, we investigate the effect of window length by not fixing the window length for feature extraction and normalization. The window length for feature extraction and normalization is changed in the range of 100–500 ms in 100 ms increments. We compare proposed SWN, z-score, and None (no normalization) in the six types of features by using model OWN (Figure A1).

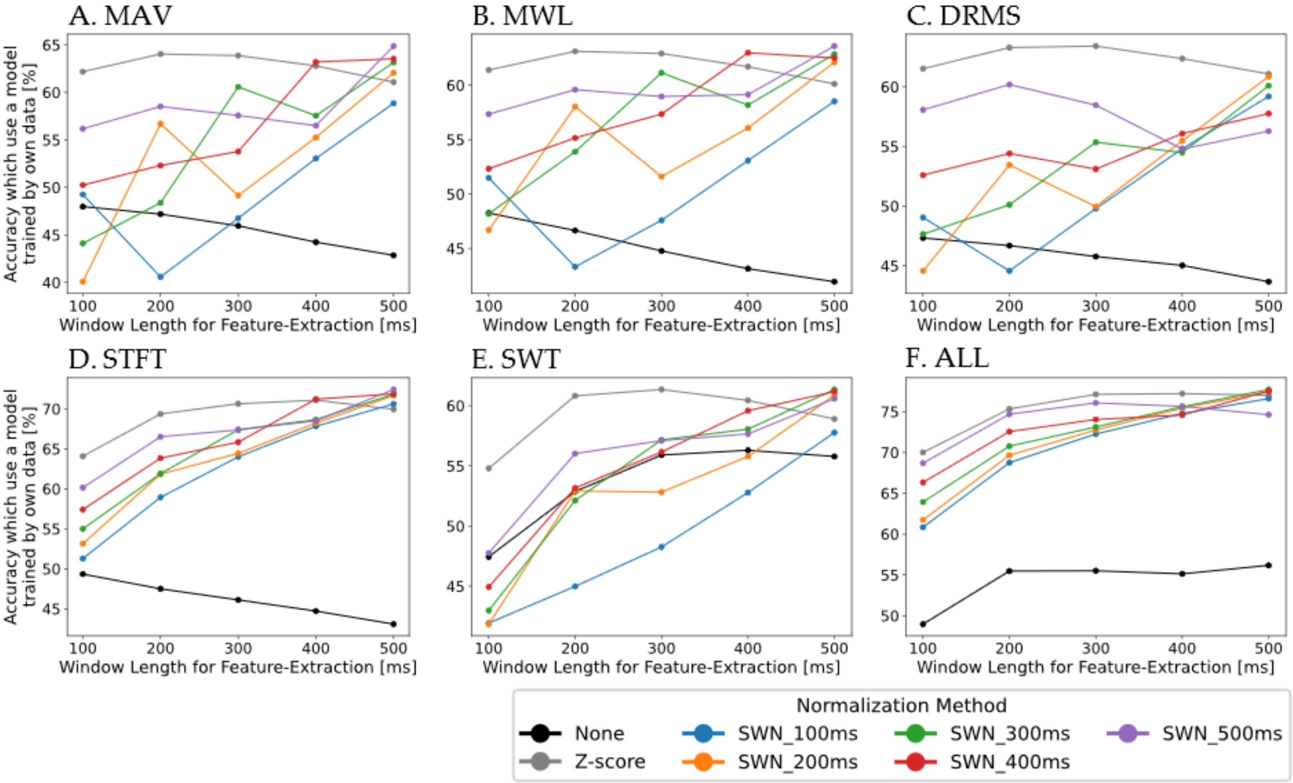


Figure A1. Effect of the window length for feature-extraction and normalization (OWN). Horizontal axis indicates window-length for feature-extraction and vertical axis indicates accuracy using a model trained by data from own subject. Each color line shows the normalization methods. The black line shows no-normalization (None), gray line shows z-score and other color line shows with normalization each window-length for normalization. (**A**–**F**) indicate feature-extraction methods.

From Figure A1, the accuracy increases with window length for feature extraction, and does not significantly change with window length for normalization, like in Section 3.1. Furthermore, there are no interactions with window length for feature extraction and normalization.

References

1. Fleischer, C.; Wege, A.; Kondak, K.; Hommel, G. Application of EMG signals for controlling exoskeleton robots. *Biomed. Tech.* **2006**, *51*, 314–319. [CrossRef] [PubMed]
2. Englehart, K.; Hudgins, B. A robust, real-time control scheme for multifunction myoelectric control. *IEEE Trans. Trans. Trans. Trans. Biomed. Eng.* **2003**, *50*, 848–854. [CrossRef]
3. Marcheix, B.; Gardiner, B.; Coleman, S. Adaptive Gesture Recognition System for Robotic Control Using Surface EMG Sensors. In Proceedings of the 2019 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT), Ajman, United Arab Emirates, 10–12 December 2019; IEEE: Piscataway, NJ, USA; pp. 1–6.
4. Laksono, P.W.; Kitamura, T.; Muguro, J.; Matsushita, K.; Sasaki, M.; Amri bin Suhaimi, M.S. Minimum Mapping from EMG Signals at Human Elbow and Shoulder Movements into Two DoF Upper-Limb Robot with Machine Learning. *Machines* **2021**, *9*, 56. [CrossRef]
5. Bi, L.; Guan, C. A review on EMG-based motor intention prediction of continuous human upper limb motion for human-robot collaboration. *Biomed. Signal Process. Control* **2019**, *51*, 113–127. [CrossRef]
6. Phinyomark, A.; Khushaba, R.N.; Scheme, E. Feature extraction and selection for myoelectric control based on wearable EMG sensors. *Sensors* **2018**, *18*, 1615. [CrossRef]
7. Phinyomark, A.; Quaine, F.; Charbonnier, S.; Serviere, C.; Tarpin-Bernard, F.; Laurillau, Y. Feature extraction of the first difference of EMG time series for EMG pattern recognition. *Comput. Methods Programs Biomed.* **2014**, *117*, 247–256. [CrossRef]

8. Ji, Y.; Sun, S.; Hong-Bo, X. Stationary wavelet-based two-directional two-dimensional principal component analysis for EMG signal classification. *Meas. Sci. Rev.* **2017**, *17*, 117. [CrossRef]
9. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25*; NIPS: San Diego, CA, USA, 2012.
10. Chen, Y.; Yu, S.; Ma, K.; Huang, S.; Li, G.; Cai, S.; Xie, L. A continuous estimation model of upper limb joint angles by using surface electromyography and deep learning method. *IEEE Access* **2019**, *7*, 174940–174950. [CrossRef]
11. Atzori, M.; Cognolato, M.; Müller, H. Deep learning with convolutional neural networks applied to electromyography data: A resource for the classification of movements for prosthetic hands. *Front. Neurorobot.* **2016**, *10*, 9. [CrossRef]
12. Buongiorno, D.; Cascarano, G.D.; Brunetti, A.; de Feudis, I.; Bevilacqua, V. A survey on deep learning in electromyographic signal analysis. In *Intelligent Computing Methodologies, Proceedings of the International Conference on Intelligent Computing, Nanchang, China, 3–6 August 2019*; Springer: Cham, Switzerland, 2019; pp. 751–761.
13. Ekenel, H.K.; Stiefelhagen, R. Analysis of local appearance-based face recognition: Effects of feature selection and feature normalization. In Proceedings of the 2006 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'06), New York, NY, USA, 17–22 June 2006; IEEE: Piscataway, NJ, USA, 2006; p. 34.
14. Wu, W.; Xing, E.P.; Myers, C.; Mian, I.S.; Bissell, M.J. Evaluation of normalization methods for cDNA microarray data by k-NN classification. *BMC Bioinform.* **2005**, *6*, 191. [CrossRef]
15. Jain, A.; Nandakumar, K.; Ross, A. Score normalization in multimodal biometric systems. *Pattern Recognit.* **2005**, *38*, 2270–2285. [CrossRef]
16. Ma, Y.; Chen, B.; Ren, P.; Zheng, N.; Indiveri, G.; Donati, E. EMG-based gestures classification using a mixed-signal neuromorphic processing system. *IEEE J. Emerg. Sel. Top. Circuits Syst.* **2020**, *10*, 578–587. [CrossRef]
17. Khan, A.M.; Sadiq, A.; Khawaja, S.G.; Akram, M.U.; Saeed, A. Physical action categorization using signal analysis and machine learning. *arXiv* **2020**, arXiv:2008.06971.
18. Yang, K.; Xu, M.; Yang, X.; Yang, R.; Chen, Y. A Novel EMG-Based Hand Gesture Recognition Framework Based on Multivariate Variational Mode Decomposition. *Sensors* **2021**, *21*, 7002. [CrossRef]
19. Wahid, M.F.; Tafreshi, R.; Al-Sowaidi, M.; Langari, R. Subject-independent hand gesture recognition using normalization and machine learning algorithms. *J. Comput. Sci.* **2018**, *27*, 69–76. [CrossRef]
20. Kerber, F.; Puhl, M.; Krüger, A. User-independent real-time hand gesture recognition based on surface electromyography. In Proceedings of the 19th International Conference on Human-Computer Interaction with Mobile Devices and Services, New York, NY, USA, 4–7 September 2017; Association for Computing Machinery: New York, NY, USA; pp. 1–7.
21. Sousa, A.S.; Tavares, J.M.R. Surface electromyographic amplitude normalization methods: A review. Electromyography: New developments, procedures and applications. In *Electromyography: New Developments, Procedures and Applications*; Nova Science Publishers, Inc.: Hauppauge, NY, USA, 2012.
22. Meldrum, D.; Cahalane, E.; Conroy, R.; Fitzgerald, D.; Hardiman, O. Maximum voluntary isometric contraction: Reference values and clinical application. *Amyotroph. Lateral Scler.* **2007**, *8*, 47–55. [CrossRef] [PubMed]
23. Ameri, A.; Akhaee, M.A.; Scheme, E.; Englehart, K. A deep transfer learning approach to reducing the effect of electrode shift in EMG pattern recognition-based control. *IEEE Trans. Neural Syst. Rehabil. Eng.* **2019**, *28*, 370–379. [CrossRef]
24. Scheme, E.; Englehart, K. Electromyogram pattern recognition for control of powered upper-limb prostheses: State of the art and challenges for clinical use. *J. Rehabil. Res. Dev.* **2011**, *48*, 643–659. [CrossRef]
25. Khushaba, R.N.; Takruri, M.; Miro, J.V.; Kodagoda, S. Towards limb position invariant myoelectric pattern recognition using time-dependent spectral features. *Neural Netw.* **2014**, *55*, 42–58. [CrossRef]
26. Wang, J.; Pang, M.; Yu, P.; Tang, B.; Xiang, K.; Ju, Z. Effect of muscle fatigue on surface electromyography-based hand grasp force estimation. *Appl. Bionics Biomech.* **2021**, *2021*, 8817480. [CrossRef]
27. Triwiyanto, T.; Wahyunggoro, O.; Nugroho, H.A.; Herianto, H. Muscle fatigue compensation of the electromyography signal for elbow joint angle estimation using adaptive feature. *Comput. Electr. Eng.* **2018**, *71*, 284–293. [CrossRef]
28. Oda, Y.; Sato, T.; Nambu, I.; Wada, Y. Real-time reduction of task-related scalp-hemodynamics artifact in functional near-infrared spectroscopy with sliding-window analysis. *Appl. Sci.* **2018**, *8*, 149. [CrossRef]
29. Zhang, Y.; Chen, Y.; Yu, H.; Yang, X.; Lu, W. Dual layer transfer learning for sEMG-based user-independent gesture recognition. *Pers. Ubiquitous Comput.* **2020**, *26*, 575–586. [CrossRef]
30. Kim, K.T.; Guan, C.; Lee, S.W. A subject-transfer framework based on single-trial EMG analysis using convolutional neural networks. *IEEE Trans. Neural Syst. Rehabil. Eng.* **2019**, *28*, 94–103. [CrossRef] [PubMed]
31. Guo, Y.; Gu, X.; Yang, G.Z. MCDCD: Multi-Source Unsupervised Domain Adaptation for Abnormal Human Gait Detection. *IEEE J. Biomed. Health Inform.* **2021**, *25*, 4017–4028. [CrossRef] [PubMed]
32. Gong, B.; Shi, Y.; Sha, F.; Grauman, K. Geodesic flow kernel for unsupervised domain adaptation. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 16–21 June 2012; IEEE: Piscataway, NJ, USA; pp. 2066–2073.
33. Sun, B.; Feng, J.; Saenko, K. Correlation alignment for unsupervised domain adaptation. In *Domain Adaptation in Computer Vision Applications*; Springer: Cham, Switzerland, 2017; pp. 153–171.
34. Pan, S.J.; Tsang, I.W.; Kwok, J.T.; Yang, Q. Domain adaptation via transfer component analysis. *IEEE Trans. Neural Netw.* **2010**, *22*, 199–210. [CrossRef]

35. Ganin, Y.; Lempitsky, V. Unsupervised domain adaptation by backpropagation. In Proceedings of the International Conference on Machine Learning, Lille, France, 6–11 July 2015; PMLR; pp. 1180–1189.
36. Gu, X.; Guo, Y.; Deligianni, F.; Lo, B.; Yang, G.Z. Cross-subject and cross-modal transfer for generalized abnormal gait pattern recognition. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *32*, 546–560. [CrossRef]
37. Sheng, X.; Lv, B.; Guo, W.; Zhu, X. Common spatial-spectral analysis of EMG signals for multiday and multiuser myoelectric interface. *Biomed. Signal Process. Control* **2019**, *53*, 101572. [CrossRef]