



Instance-based learning with prototype reduction for real-time proportional myocontrol: a randomized user study demonstrating accuracy-preserving data reduction for prosthetic embedded systems

Tim Sziburis^{1,2} · Markus Nowak² · Davide Brunelli³

Received: 4 November 2022 / Accepted: 21 August 2023 / Published online: 5 October 2023
© The Author(s) 2023

Abstract

This work presents the design, implementation and validation of learning techniques based on the kNN scheme for gesture detection in prosthetic control. To cope with high computational demands in instance-based prediction, methods of dataset reduction are evaluated considering real-time determinism to allow for the reliable integration into battery-powered portable devices. The influence of parameterization and varying proportionality schemes is analyzed, utilizing an eight-channel-sEMG armband. Besides offline cross-validation accuracy, success rates in real-time pilot experiments (online target achievement tests) are determined. Based on the assessment of specific dataset reduction techniques' adequacy for embedded control applications regarding accuracy and timing behaviour, decision surface mapping (DSM) proves itself promising when applying kNN on the reduced set. A randomized, double-blind user study was conducted to evaluate the respective methods (kNN and kNN with DSM-reduction) against ridge regression (RR) and RR with random Fourier features (RR-RFF). The kNN-based methods performed significantly better ($p < 0.0005$) than the regression techniques. Between DSM-kNN and kNN, there was no statistically significant difference (significance level 0.05). This is remarkable in consideration of only one sample per class in the reduced set, thus yielding a reduction rate of over 99% while preserving success rate. The same behaviour could be confirmed in an extended user study. With $k = 1$, which turned out to be an excellent choice, the runtime complexity of both kNN (in every prediction step) as well as DSM-kNN (in the training phase) becomes linear concerning the number of original samples, favouring dependable wearable prosthesis applications.

Keywords Prosthetic control · EMG · Machine learning · Embedded systems · Data reduction

1 Introduction and motivation

The development of prosthetics has been continuously improving since the twentieth century. After being solely a cosmetic replacement for amputees, prostheses evolved to body-driven functional devices and, especially beginning in the 1940s, to powered myoelectric systems ([102], p. 32).

From the existing myographic methods to collect data from muscular activity, this paper focuses on surface elec-

tromyography (sEMG), which comprises the capturing, processing and analysis of electromyographic signals, i.e. the changes over time in electric potential originating from skeletal muscles (cf. [63, 68]), measured by electrodes on the skin surface.

Prosthetic control describes the general concept behind the process from capturing signal data (sensor side) via processing and analyzing it to forwarding the data interpretation to a prosthetic device (actuator side)—with potential feedback loops. The (closed-loop) motor-control of the prosthetic actuators themselves, i.e. the single joints of the prosthetic device by, e.g. direct force control, impedance or admittance control will not be considered in the scope of this work.

There has been a variety of control schemes presented for EMG-based prosthetic devices in the area of open-loop myocontrol [81, p. 252]. As a consequence of their diverse fundamental characteristics, they differ in the achieved granularity, precision and stability of movements (dexterity).

✉ Tim Sziburis
tim.sziburis@alumni.cern

¹ Institute for Neuroinformatics (INI), Ruhr University Bochum, Universitätsstr. 150, Bochum 44801, Germany

² German Aerospace Center (DLR), Robotics and Mechatronics Center (RMC), Münchener Str. 20, 82234 Weßling, Germany

³ Department of Industrial Engineering, DII, University of Trento, Via Sommarive, 9, 38123 Trento, Italy

Pattern-recognition-based myoelectric control schemes utilize machine learning methods to correctly detect gestures by means of classification or regression. Specific features can be calculated from the raw or filtered signal in time-, frequency- and time-frequency domain [81, p. 250]. The basic principle of action (intent/intention) detection is to predict a specific action from the sensed biological signal.

This work examines and extends the k -nearest neighbour learning scheme (kNN). The respective methods are based upon detecting specific gestures defined beforehand, leading in the application phase to their real-time recognition. For not explicitly learned, intermediate gestures proportionality scaling techniques are introduced.

Commonly applied pattern recognition methods in myocontrol often show disadvantages in terms of generalizability, intuitive control and robustness regarding “electrodes shift, varying force levels” [50] (e.g. overshooting) and others. To cope with these limitations, an extended kNN learning scheme seemed promising due to its simplicity, incrementality and good results in exemplary tests.

Referring to the results of kNN in the context of EMG-based prosthetics in Sect. 2.1, it can be seen that in most cases, kNN delivers high performance in terms of accuracy and success rate respectively. [59] states that “excellent performance can be achieved if sufficient training data is available”.

kNN is shown to be relatively insensitive to noise [4], electrode displacement [11] and sampling frequency variation [16] which speaks in favour of robustness.

Its low complexity is a main advantage of kNN specifically in the context of implementation for embedded systems. As for all instance-based machine learning techniques, it does not require an explicit model training. However, it has to be coped with high computational demands during the prediction. Thus, this paper introduces mechanisms for dataset reduction, combines them with kNN and finally analyzes a selected method to be called DSM-kNN.

The applicability to embedded, wearable systems is specifically important since “users of modern prosthetics are now given access to applications that can run on external devices capable of fine tuning and setting up gestures or gesture patterns [...] [to] allow[] high-level customization” [102] and comes with non-functional constraints regarding energy consumption, portability, timeliness, safety and dependability.

Precise sensor placement on specific muscles is usually considered essential for achieving high detection performance from sEMG signals [83]. This placed-sensor approach is not suitable for wearable devices, but could be compensated by (time-)frequency-domain feature analysis, although it “cannot be realised in real-time using the simple embedded processors housed in EMG wearables”, as mentioned by [83]. They present an operational comparison of applicable features, projection techniques and classifiers. On this

basis, they introduce a time-domain algorithm “suitable for deployment on embedded processors for real-time inference in a portable, battery-operated device” [83] by reducing clock cycle and therefore power consumption without impairing accuracy. However, their promising approach neither introduced proportionality schemes to classification nor conducted online user studies and can be seen as a complement.

Although kNN has been applied in several EMG studies, there was no in-depth examination of the strategy’s parameters nor has it been particularly combined with data reduction techniques as in our work.

2 Related work

A variety of machine learning strategies has been followed over the years in the context of myocontrol (cf. [81, p. 251]). This includes neural networks in different compositions [4, 6, 8, 23, 30–32, 36, 39, 42, 45, 47, 82, 86, 90, 93], including such based on adaptive resonance theory [91]; support vector machines and variants [16, 39, 48, 49, 66, 84, 86, 90, 108]; decision trees [30]; (naïve) Bayesian classification [52, 70, 90, 103]; fuzzy logic approaches [3, 69]; Gaussian mixture models [44, 46, 106]; logistic regression [30]; logistic model trees [30]; classification via independent component analysis (ICA) [93], canonical discriminant analysis [71]; linear discriminant analysis (LDA) [6, 16, 20, 22, 23, 31, 46, 50, 53, 98, 108]; quadratic discriminant analysis (QDA) [46, 53, 90, 98]; random forest [86]; extreme learning machines (ELM) [90]; hidden Markov models [13]; evolvable hardware (EHW) with embedded Cartesian genetic programming [37]; and kNN (see Sect. 2.1).

The following features and transformations have proven well in the context of pattern-recognition-based myoelectric control (cf. [81, p. 250–251]): linear envelope [107], [10, 76, 104, p. 271]; zero crossings and variance [87]; integral absolute value, variance and zero crossing [94]; mean absolute value [6], its slope, wave form length, number of waveform slope sign changes and number of waveform zero crossings (Hudgins set of features) [45]; frequency spectrum via Fourier transform [26, 39, 93], random Fourier features [34, 35] and local frequency and phase content via short-time Fourier transform [22, 23, 41, 91]; autoregressive coefficients [14, 55, 103]; cepstral coefficients [14, 103]; wavelet decomposition coefficients [8, 22, 23, 36, 47, 48, 67, 84] and their eigenvalues [66]; wavelet packet feature sets [22, 23], motor unit action potentials (MUAPs) via wavelet packet transform and fuzzy C-means clustering [85]; signal energy (overall, within Hamming windows, within trapezoidal windows) as temporal features and spectral magnitude as well as spectral moments from short-time Thompson transform [91]; moving approximate entropy [2]; and contraction factors from fractal modelling [55] and fractal dimensions [7, 43].

A review of classification techniques for forearm prostheses is given in [77, p. 725], along with information about features, performed experiments, selected subjects and achieved results. A review of the multitude of features and an evaluation thereof on EMG data with significance analysis was conducted in [79, p. 4834–4838].

The following subsections particularly summarize the utilization of the kNN learning scheme in the same context, as well as applicable data reduction techniques.

2.1 Nearest neighbour techniques

kNN was firstly proposed for the parameter k (numbers of neighbours to consider) set to 1 as “nearest neighbor decision rule” in 1967 [18].

The basic principle of kNN consists of comparing new arriving data (instances) with all instances that were captured as reference data in an initial step and considering a subset of them (number of reference instances k) for a prediction decision. Although, this initial step does not comprise the generation of a generalized model (training of a model), it is usually called *training* (also in this paper).

The comparison of instances refers to the comparison of distances between a new instance and the training instances by using a specified distance measure.

After calculating the distance, kNN selects a number k of nearest instances to the new instance. If their labels are immediately averaged (which would have to be specified, usually

arithmetic mean), this leads to a prediction in the form of a regression method (kNN regression). If instead of averaging a majority vote is applied on the k nearest instances, the label with the most votes is yielded as categorical label prediction (kNN classification). In this sense, the focus of this paper is on kNN classification which will be extended by a proportionality scaling scheme (see Sect. 4.1).

In place of directly majority voting after a set of neighbouring instances has been selected (uniform weights), distance-based weighting factors can be calculated for all instances of the selection. For each class, these weightings are summed up, so that the prediction is determined as the class with the maximum sum. This distance weighting is typically introduced to avoid that a majority of class labels from instances which are farther away (but still within the neighbourhood) influences the prediction at the expense of instances which are closer but in minority.

kNN has been applied for EMG-based pattern recognition a variety of times. In 1983, a nearest neighbour classifier ($k = 1$) was already chosen in the context of prosthetics [21]. Additionally, a variant of prototype reduction (see Sect. 2.2) was introduced. There was no decrease of performance when reducing the number of samples per gesture from 100 to 2–6 (for power grasp, flexion, extension, pronation) and to 40–46 (for rest and supination), respectively.

Since then, nearest-neighbour-based methods have been evaluated in various EMG-based gesture detection studies rather unsystematically, mainly for comparison to other classifiers, both for able-bodied subjects and amputees (Table 1).

Table 1 kNN-based gesture detection in EMG control applications, sorted by year, in studies compared with other classifiers and their accuracies. *n/s* means not specified

Ref.	kNN k	Configuration Distance	Subjects #	Health	Actions	Accuracy (%)	Compared classifiers (accuracy (%), w.r.t. kNN)	Real time control
[21]	1	-	1	able-b.	6	72	-	No
[37]	5	<i>n/s</i>	1	<i>n/s</i>	8	95.5	SVM (similar), EHW (91–95.1), DT (91)	No
[52]	5	<i>n/s</i>	30	able-b.	4	94	Bayes (92)	Yes
[32]	31	-	<i>n/s</i>	able-b.	6	59–100	NN (similar)	No
[4]	7–17	Euclidean	<i>n/s</i>	<i>n/s</i>	<i>n/s</i>	83–100 ($k=15$)	NN (50–83)	No
[6]	16	Euclidean	10	able-b.	13	81	LDA, NN (similar)	Yes
[17]	8	Euclidean	5+5	both	7	89/79 (amp.)	-	yes
[53]	1–10	Euclidean	30	able-b.	5	85 ($k=5$)	QDA (82), LDA (81, signif. diff.)	No
[51]	8	Euclidean	8	able-b.	10	90	SVM (similar)	Yes
[98]	<i>n/s</i>	Euclidean	5	able-b.	15	88–100	LDA: 51–100, QDA: 82–100	No
[46]	1	Euclidean	28	able-b.	5	95	Gaussian mixture (similar); signif. diff.: LDA (92), QDA (94)	No
[90]	5	<i>n/s</i>	20/8	able-b.	10/15	87/89	NB, QDA, SVM, NN, ELM (similar)	No
[42]	1–7	various	4	able-b.	9	80–86 ($k=6$)	NN (93)	No
[30]	9	Euclidean	10	able-b.	6	88	LR (91), LMT (91), DT (84), NN (90)	No
[82]	<i>n/s</i>	<i>n/s</i>	10	able-b.	6	78–89	NN (70–88)	Yes
[50]	1	<i>n/s</i>	6	amput.	11	74	LDA (similar)	No
[86]	7	Manhattan	11	able-b.	17	85	NN (35), RF (90), SVM (73)	No
[16]	<i>n/s</i>	<i>n/s</i>	5	able-b.	8	60–100	SVM (56–100), LDA (40–98)	No

For example, kNN showed to perform as good as multi-layer neural networks. In this context, it was mentioned that the “kNN classifier may be considered to be a better choice for classification of continuous EMG signals to actuate the prosthetic drive” [82, p. 5].

Usually, kNN also exposed similarly good detection accuracy as QDA, SVMs, Gaussian as well as Bayesian methods, and performed comparably or better than LDA (see Table 1). In some cases, kNN was shown to perform statistically significantly better than LDA [46, 53] and even QDA [46]. Further work pointed out that “there was no significant [difference] between weak-load algorithms (NB, KNN, QDA, and ELM) and heavy-load algorithms (SVM and MLP) after applying the dimension reduction” [90]. An experiment with $k = 9$ showed that the “kNN classifier [was] better at classifying the EMG signal with PCA transformed statistical data compared to other classifiers in accuracy, sensitivity and specificity” [30], namely logistic regression, decision and logistic model trees as well as a neural network classifier.

Besides these comparisons of different classifiers without external factors, a study of noise influence on kNN exposed a high stability of detection accuracy even for reduced signal-to-noise ratios if k is chosen properly. For $k = 15$, the accuracy decreased from 100 to 83% for an increase of simulated noise from 25 to 5 dB SNR. With this, it showed a higher robustness than a neural network classifier.

The mid-term performance for discovering the influence of electrode shift on kNN showed a basically constant average performance from one day to another [62]. This confirmed a former analysis of performance over time [37] and is important for prosthetic devices since repositioning regularly introduces an electrode displacement which otherwise would require immediate retraining.

Finally, kNN has shown to be comparably insensitive to the reduction of recording sample rate. In an experiment, kNN achieved higher accuracies than SVM and LDA at all sampling rates, and the performance reduction for frequency reduction was not as steep as for other classifiers [16]. Instead, for a change from 1000 to 200 Hz, the accuracy reduced only minimally from 99.9 to 99%. At 20 Hz, it still provided 78% (vs. 71%/56% for SVM/LDA). This behaviour of the kNN method is highly advantageous for embedded systems scenarios, since lower sampling rates lead to lower CPU clock frequencies and therefore reduced powering requirements, which in the end support a low-cost approach and increase portability by requiring smaller dimensions for the prosthetic controller.

2.2 Training dataset reduction algorithms

An important drawback of instance-based learning schemes is the necessity of comparing new arriving instances whose labels are meant to be predicted to all already stored ones

(“training” data). In order to do so, all instances have to be iterated which leads to potentially—depending on the amount of data—high computational effort in the prediction phase.

Typically, two main approaches to improve the performance of nearest neighbour classifiers are pointed out [9]. The first is the utilization of efficient, optimized data structures (“ball-tree data structures, hashing” [58], “kd-tree” [9]). The second approach (thinning) can be seen both in a horizontal (feature-space) as well as in a vertical dimension (sample-space). Aside from that, there are techniques using an approximation of the kNN classification rule, for example large margin nearest neighbour [58].

In terms of horizontal thinning, the concept of feature selection has been applied in the context of pattern-recognition-based prosthetic control for large feature set dimensions, for instance biologically inspired methods such as genetic algorithms and particle swarm optimization [81, p. 251]. Horizontal thinning can be generalized (to horizontal data reduction) when feature projection, positioning [58] and discretization [64] techniques are also considered. These schemes come along with dimensionality reduction algorithms. Examples are principle component analysis (PCA) [39] and adaptations thereof [71] as well as variants of linear discriminant analysis (LDA) [73].

However, the examinations made in this work cope with vertical data reduction techniques. The general idea is to reduce the computational effort of prediction steps in instance-based learning by decreasing the number of instances within the training set. This process is usually referred to as instance reduction or prototype reduction [29, 99]. In principle, *prototype* stands synonymously for data instance or sample. Nevertheless, it already indicates that it refers to specific instances which represent a larger amount of instances to a certain extent.

Prototype reduction methods can be divided into prototype selection (vertical thinning) on the one hand [29] and prototype generation on the other hand [99]. While the former selects a subset of instances from the existing ones, the latter creates new instances based on the existing ones to represent the whole dataset.

3 Requirements and concept

The experimental studies and the developments which they are based on are driven by the requirements of Sect. 3.1 and composed of different parts:

- First, a pilot dataset of several (full-intensity) gesture exertions is captured from the authors in order to conduct an offline cross-validation analysis of kNN parameters on gesture classification without real-time application.

- Second, the obtained kNN parameter configuration is applied in a real-time scenario, in which new (full-intensity) gesture data is gathered. Additionally, an approach of proportionality scaling is introduced here. With that, real-time gesture detection performance is measured in an online target achievement test with just one subject. The success rate in this pilot study is utilized to analyze the influence of proportionality scaling parameters while testing three levels of exertion intensity (but just training on full intensity).
- Third, the two determined parameter configurations (kNN and proportionality scaling) are tested in a real-time user study with 12 subjects (and in an extended user study with 4 subjects). Again, target achievement tests are conducted, including three levels of gesture exertion intensity for detection (but just full-intensity for training). No further parameter optimization takes place in this step. Moreover, a data reduction technique is introduced and applied to each subject’s data. The success-rate performance of the non-reduced and the reduced data approach are compared.

3.1 Requirements

The requirements listed in Table 2 are to be met by the learning strategies developed in this work. While R1–R4 represent general prerequisites, R5 constitutes an additional constraint for embedded system implementations.

R1 and R2 are considered the minimum standard for myocontrol, while R3 targets the transfer from offline to online scenarios. R4 is motivated by the benefit of home recalibration for prosthetic users [57, 74].

Table 2 Requirements for the learning method, providing embedded applicability

No.	Description
R1	High accuracy in classification of actions
R2	Providing proportional control
R3	Besides the static accuracy, the users should be satisfied by the method when using it in real scenarios (success rate, robustness, stability, reaction time)
R4	Incrementality, i.e. providing the possibility for extending the reference data by new instances
R5	Applicability to control on embedded systems (non-functional requirements such as portability, battery consumption)
R5.1	Coping with real-time demands in prediction
R5.2	Low and deterministic memory requirements
R5.3	Low and deterministic training (reduction) time
R5.4	Simplicity of the algorithm for transparency and dependability

For R5, specific sub-requirements have been defined. The general motivation of providing an algorithm suited for embedded systems and still delivering high performance is the tendency of developing wearable systems that are usable stand-alone without the necessity of connecting standard computers.

3.2 Sensor hardware

A product widely used in research—also in this work—is the *Myo* wireless armband, produced from 2013 to 2018 by the Canadian company Thalmic Labs Inc. which is characterized by the following (cf. [101]):

- Eight EMG electrodes with ST 78589 operational amplifier per electrode.
- Maximum sampling frequency of 200 Hz.
- 9-axes IMU with 3-axis gyroscope, 3-axis accelerometer and 3-axis magnetometer (InvenSense MPU-9150).
- Freescale Kinetis ARM Cortex M4 120Mhz MK22FN1M microcontroller.
- Communication via BLE with Nordic nRF51822 to HM-11 BLE dongle.
- Vibration motor and LEDs for signalling.
- Two lithium batteries (3.7 V, 260 mAh), USB-charged.

No IMU information is utilized in the context of this work.

3.3 Signal processing and nearest-neighbour-based methods

In general, a kNN-based classification approach will be given priority over kNN regression, as the latter exposed a high extent of instability in preliminary experiments.

To keep the computational demands as low as possible for an embedded prosthetic control system, we aimed at utilizing time-domain features due to their lower complexity. Specifically, the linear envelope of the signal will be used as input feature. It can be shown that the majority of the discriminatory effect in the widely used Hudgins EMG feature set stems from the mean absolute value [88]. In this sense, the reduced demands for obtaining the amplitude data by calculating the absolute value are combined with a window length of 1 to not induce further calculations. As in similar publications [74], this is followed by low-pass filtering with a cut-off frequency of 1 Hz by a second-order Butterworth filter, as “at least 90% of the power in the power spectral density estimates were found to be below 1 Hz” [76] in the rectified signal.

The gestures chosen to evaluate the performance are selected among rest state (*rs*), power grasp (*pw*), pointing (*pn*), wrist flexion (*fl*), wrist extension (*ex*), wrist pronation (*pr*) and wrist supination (*su*).

In order to evaluate the static performance of the algorithm and specifically to match requirement R1, the cross-validation accuracy on a variety of *Myo* armband EMG datasets captured by the authors will be examined. For this purpose, these datasets comprise four repetitions per gesture. One repetition contains the filtered eight-channel-EMG data when exerting one specific gesture for 2 s at the maximum sample rate of the *Myo* armband, namely 200 Hz. Multiple repetitions are necessary as the gathered samples within one repetition cannot be considered independent and identically distributed. The stochastic dependence is abolished across multiple repetitions since there are interruptions in time, specifically because of training other gestures in between, before capturing the next repetition. With that, a block-wise (group-wise) cross-validation is possible so that samples within one block (group/repetition) are not validated against samples within the same block. In this way, a preventive measure against overfitting is established. In particular, a leave-one-group-out cross-validation will be applied, i.e. selecting one block as testing set while the others form the training set, for all possible combinations. In the end, the arithmetic mean of the single accuracies (i.e. correct classifications relative to all classifications) is used to characterize the accuracy of the whole dataset.

The parameters which can be altered in kNN for static cross-validation are the number of neighbours to be considered (k), the distance metric for comparing sample differences and the weighting of selected samples' data values. It is known that kNN's "performance is critically dependent on the selection of k and a suitable distance measure" [59, p. 3] so that these will be subject to a specific analysis.

A problem with kNN classification is the fundamental characteristic that no intermediate states can be predicted. Therefore, kNN classification will be extended by proportionality scaling schemes to provide proportional control.

The following concepts are applicable for kNN classification based on majority-voting regarding the occurrences of individual class labels.

It is assumed that the intensity of an exerted action/gesture is proportional to the amplitude of the EMG signal's linear envelope [24] (averaged for all channels). By analyzing this magnitude, a proportionality scaling can be applied as soon as a gesture has been detected [45, 89]. To obtain a correct gesture classification from samples of a specific gesture at lower intensity levels, the samples are normalized (assuming that the signal shape is similar when comparing signals of the same gesture at different intensity levels).

Furthermore, a threshold for the rest action, i.e. the state where no gesture is exerted, will be introduced (rest magnitude thresholding). The motivation for this is that, if samples are closer to the rest state than to the specific real gesture, it would be classified as rest, until the transition point in the signal amplitude is reached. The rest state usually resides

at around zero signal amplitude, unless distinct postures are considered where this might differ due to the limb position effect.

The concept of the rest magnitude thresholding consists of measuring the average rest activity and basing a threshold of signal amplitude on this value, possibly altered by further parameters. If this threshold amplitude is exceeded when executing the prediction on a new sample, the classification takes place and a class label of the available ones except rest is assigned. Otherwise, the new sample is considered as rest.

Requirement R3 will be evaluated by means of target achievement tests. First, the presented concepts will be evaluated in pilot experiments without being statistically representative. The tendencies obtained are used as a baseline for a user study with multiple subjects following afterwards. For both versions of experiments, several gestures will be tested on different signal intensity levels (for instance exerting just one third of a full wrist flexion), after training solely took place on full intensity level. The single gesture has to be reached and held for a certain period of time without deviating too much within some error range in order to consider the task as successful. For this purpose, the subject will see a visual stimulus in the form of a hand model to be followed, as well as another hand model visualizing the current gesture prediction (as in Fig. 10). The results will be compared to those obtained from state-of-the-art ridge regression methods.

The final user study will be conducted in a double-blind manner in order to provide comparability of the algorithms. Therefore, the selection of gestures and intensity levels during one experiment will be randomized. For the purpose of not favouring a single method over another (if there should be a time-dependency of success), the occurrences of methods and levels will be equally distributed across the available time slots.

Requirement R4 is met inherently by the standard kNN approach since in every prediction step, each instance of the training set is compared with the sample to be predicted for obtaining the particular distance. This means if there are new samples to be stored in the training set, they are directly taken into account during prediction, thus leading to incrementality.

3.4 Assessment of embedded applicability

The applicability on embedded systems is specified in requirement R5 with its sub-components R5.1–R5.4.

To meet requirement R5.1, it is necessary to reduce the kNN computation effort in the prediction phase. As an instance-based learning technique, kNN suffers from the computational disadvantage mentioned in Sect. 2.2. Specifically, for each new instance, the prediction step comprises the calculation of the distance from the new instance to the

n stored ones (runtime complexity of $\mathcal{O}(n)$) and the sorting of these distances to obtain an ascending order of nearest neighbours. Depending on the sorting algorithm, the overall time complexity can reach $\mathcal{O}(n \log n)$ (being the proven lowest possible bound for comparison-based data sorting). Nevertheless, if the number of neighbours to consider is set to $k = 1$, no sorting is necessary anymore. Thus, with a minimum search being sufficient instead, the complexity reduces to $\mathcal{O}(n)$.

Possibilities to reduce the computational effort in terms of the number of training samples n to specifically achieve requirement R5.1 have been introduced in Sect. 2.2. In particular, the concept of prototype reduction is chosen. As presented in [96], an assessment of the variety of these algorithms has to be made in order to lower the number of instances in the training set for kNN. To meet requirement R5.2, it is necessary that the particular algorithm to be chosen provides a possibility to specify the number of prototypes in the final set or accordingly the reduction rate beforehand. When it comes to prototype selection algorithms reviewed in [29], only random mutation hill climbing (RMHC, [92]) inherently possesses this characteristic as it is the only method with fixed reduction. Nevertheless, RMHC is a wrapper method which means that in each step the decision if to select a prototype or not, a complete kNN evaluation for all instances has to take place. For this reason, long computational times during the reduction process have to be expected. In [29, p. 425–427], it is shown that this assumption holds in real use cases for both small and medium-sized datasets. Exemplary tests on EMG datasets confirmed that behaviour so that RMHC was excluded from consideration.

Besides the fixed reduction prototype selection algorithms, there might be also mixed reduction methods which provide the property of determinism with respect to the number of samples contained in the final training set. However, the algorithms of that category described in [29] are all wrapper methods, too. Due to the respective high execution times as mentioned before, these methods are not considered within the scope of this work.

In terms of prototype generation, there is a variety of fixed reduction algorithms. They can be summarized in the following way:

- *Positioning adjustment, condensation approaches:* Learning vector quantization (LVQ)-based methods [33, 56].
- *Positioning adjustment, hybrid approach:* Particle swarm optimization (PSO [72]).
- *Centroid-based condensation approaches:* Bootstrap technique for nearest neighbour (BTS3 [40]) and adaptive condensing algorithm based on mixtures of Gaussians (MGauss [65]).
- *Space-splitting:* Chen algorithm [15].

While the Chen and BTS3 algorithms are not incremental in the sense of requirement R3, in PSO, MGauss and the LVQ-based methods, each step in the reduction process only depends on the former step (where a certain model or prototype configuration is obtained) but not on the instances themselves from the initialization of the whole process. Usually, this leads to the characteristic that the reduction process does not depend on the order of decisions, i.e. the order of instances being considered.

The LVQ-based algorithm LVQTC (LVQ with Training Counter, [75]) turned out to not provide determinism with regard to the final set's size and was therefore not taken into account for further evaluation.

Again, there are mixed algorithms which may also provide the final set size determinism like the fixed ones are supposed to. Some of them are in turn wrapper methods (evolutionary nearest prototype classifier (ENPC) [27], adaptive Michigan particle swarm optimization (AMPSO) [12]) and hence not considered with respect to the previously mentioned reason.

Filter and semi-wrapper methods which might be applicable in principle are gradient descent and deterministic annealing (MSE [19]), hybrid LVQ3 (HYB [54]), integrated concept prototype learner (ICPL2 [60]), LVQ with pruning (LVQPRU [61]) and prototype selection clonal selection algorithm (PSCSA [28], artificial immune system model).

The reason why the first three of these algorithms were not chosen for the evaluation in the end are their non-determinism with respect to the final set size. The remaining algorithms are to be compared. Since they vary with regard to the time needed for the reduction process, this is examined in experiments that are based on datasets of captured rectified and filtered EMG signals (linear envelope). Besides the amount of reduction (requirement R5.2) and the runtime behaviour (R5.3), the achieved accuracies when using the reduced sets in block-wise cross-validation will be assessed. The choice for specific algorithms will be further guided by requirement R5.4, i.e. taking into account the implementation complexity of the algorithms.

4 Methods

In terms of the methodical realization of the algorithm, several characteristics will be pointed out in the following, regarding both the kNN scheme and data reduction techniques.

4.1 Methodological considerations for the kNN approach

The kNN training process is structured as follows (see also Fig. 1): capturing training data, calculating class magnitude averages for proportionality scaling and rest magnitude

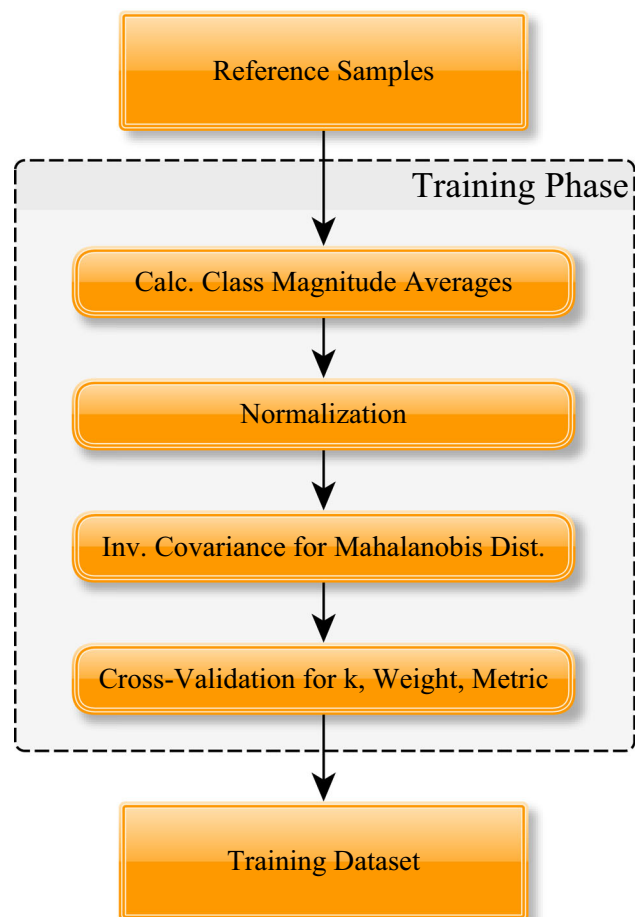


Fig. 1 General structure of training phase

threshold (if enabled), executing normalization of this data (if enabled), calculating the inverse covariance matrix of the data if the Mahalanobis distance is activated, executing block-wise cross-validation for obtaining the optimal k , weighting and metric in terms of accuracy.

The prediction process comprises the following: applying rest magnitude thresholding (if enabled), executing normalization of sample (if enabled), calculating k nearest neighbours of the sample whose label is supposed to be predicted and their distances, applying distance weighting on the k selected neighbouring samples (if enabled), executing direct averaging of neighbour samples in kNN regression, or calculating the proportionality scaling factor by analysis of the signal amplitude, before executing kNN classification by majority voting on the (potentially weighted) samples, i.e. the class with the highest weight sum will be selected for predicting the full gesture, which will be scaled by applying magnitude proportionality scaling (if enabled).

These steps are also pointed out in Fig. 2. Additionally, different windowing schemes could be applied (cf. [95]).

4.1.1 Nearest neighbour parameter configurations

The number of next neighbouring samples (k) to consider in a prediction step is varied from $k = 1$ (just nearest neighbour) to higher numbers. For each case, the particular block-wise cross-validation accuracy is computed, if enabled. Due to the characteristics of this validation scheme, the maximum k cannot exceed the total number of samples minus the size of one repetition block.

The examined distance measures are the Minkowski-norm-based metrics Manhattan ($p = 1$), Euclidean ($p = 2$) and Chebyshev ($p \rightarrow \infty$), as well the Mahalanobis distance. For distance weighting, inversely distance-dependent factors are calculated for each sample and summed up for each class within the selected set of neighbouring samples. Hereby, a weighted majority vote is obtained as classification.

kNN inherently involves k minimum searches to obtain the k nearest neighbours. This is implemented by means of sorting the distances in a descending order and picking the k first entries. For this purpose, an appropriate sorting function is called. However, if k is selected to be 1, the sorting procedure can be replaced by a search for the minimum distance within the set.

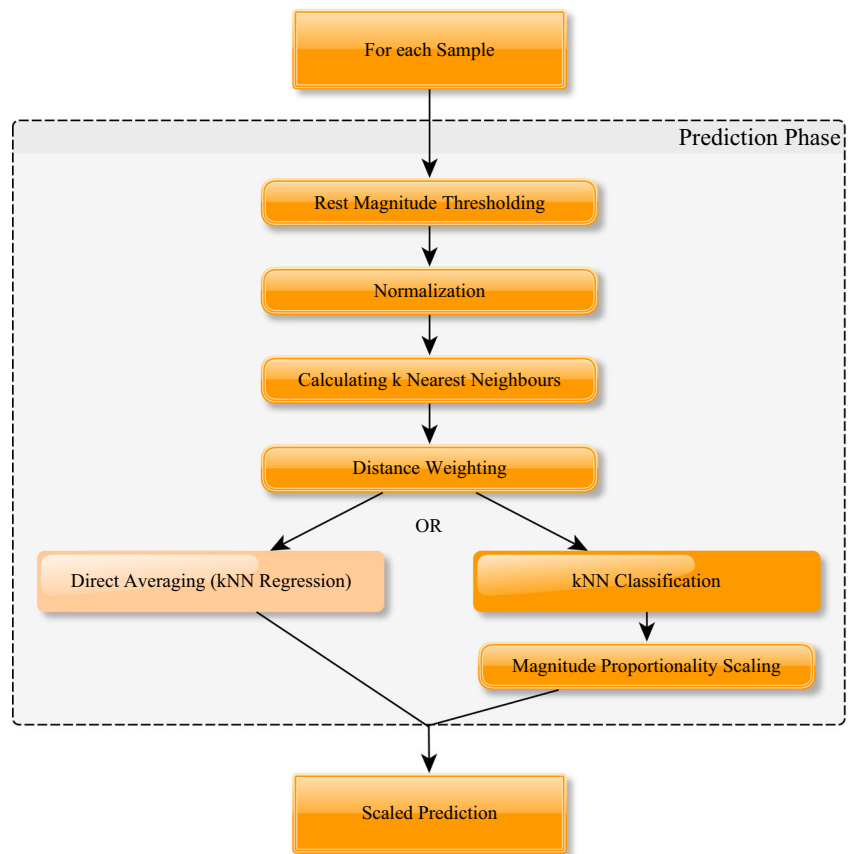
4.1.2 Proportionality scaling and rest thresholding

As presented in [97], the approach for rest magnitude thresholding is realized in a way that the magnitudes of the rest samples gathered during training are averaged and taken as a baseline for rest activity (t_0). The threshold of signal amplitude which has to be exceeded for not classifying a gesture as rest anymore is based on the obtained average: $t = g \cdot t_0$ (amplified by gain g). Although this enables to reduce unintended actuations, a higher thresholding level t results in a lower proportionality resolution by presuming higher activation forces. Another possibility for calculating a threshold could be to consider other functions applied on the rest activity instead of the mean, such as the median or the maximum (although the latter would require a specific consideration of outliers).

For the non-rest gestures, an approach of proportionality scaling is utilized [97]. This is implemented in a linear manner, i.e. intermediate gestures are assumed to be linearly scaled between rest activity and the average training magnitude of the particular gesture set as function maximum. Again, instead of the mean of the individual gestures' magnitudes, other functions might be used.

As mentioned, there is the need for a trade-off between the level of proportionality resolution and suppressing unintended activations. Therefore, a divisor v to scale the proportionality function offset $m_0 = \frac{t}{v}$ is moreover introduced. This does not scale the rest threshold t itself.

Fig. 2 General structure of prediction phase (regression path not chosen)



These relations between measured magnitude m and applied scaling factor s are depicted in Fig. 3: The blue function describes the theoretical linear proportionality scale, i.e. the scaling of the predicted gesture starts with 0 at 0

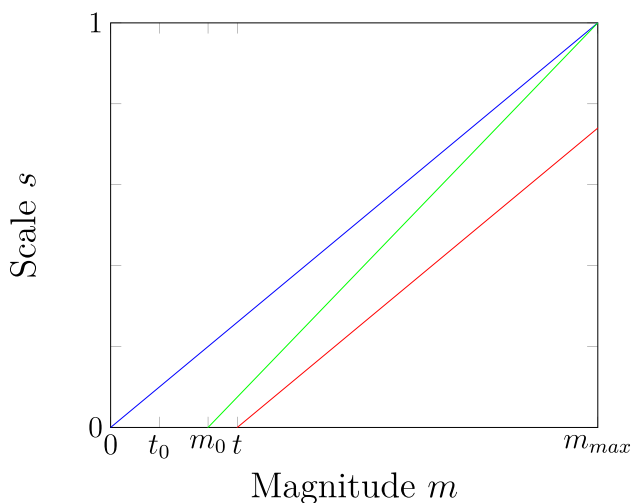


Fig. 3 Example of proportionality scaling: $t = g \cdot t_0$ marks rest threshold, $m_0 = \frac{t}{g}$ adjusted offset (for proportionality function, not for thresholding itself), m_{max} gesture magnitude maximum. Blue curve is theoretical ideal behaviour, red exceeds m_{max} and green adjusts function slope (implemented)

magnitude, assuming there is no baseline rest activity at all that could lead to wrong classifications. With introducing the rest threshold t as an offset for the scaling function, too, the average activity of the full gesture m_{max} would be required to be exceeded in prediction to reach the maximum scaling. This could be avoided by also adapting the scaling function maximum for $s = 1$. Since this would lead to a reduced magnitude resolution, the maximum is pertained and the slope of the function is modified (green curve) as follows:

$$s(m) = \frac{1}{m_{max} - m_0} \cdot (m - m_0).$$

An alternative approach could be to use piecewise linear functions or modelling non-linear relationships.

4.2 Dataset reduction algorithms

For the evaluation of prototype reduction algorithms, the open-source (GPLv3) software tool KEEL (*Knowledge Extraction based on Evolutionary Learning* [100, p. 1239]) was chosen and extended, in which the particular prototype reduction algorithms from [29] and [99] have been implemented.

Special focus of this work will be put on reduction methods based on learning vector quantization (LVQ). They are composed of the following basic steps:

1. Initialization by choosing random samples.
2. Or selecting the classes' centres of masses as initial prototypes and potentially adding more samples randomly (as long as the number of prototypes to be chosen is not exceeded, distribute the selection equally over all classes while choosing randomly within each class).
3. Repeating the correction process for a specified number of iterations: for each sample, decide if it has a rewarding and/or a penalizing effect on particular prototypes and employ this effect.

The idea of the standard LVQ-based approaches, originally proposed in the context of self-organizing maps in [56] (with prototypes being called codebook vectors) in three different variants (LVQ1, LVQ2, LVQ3), is to represent the probability distribution behind the dataset. An exception is the decision surface mapping (DSM) strategy which instead aims at appropriately modelling the class borders (decision boundaries/surfaces) [78, p. 335].

As [38] points out, standard “LVQ corresponds to what is usually known as SCL [Simple Competitive Learning] in the neural network literature”. [1] defines it as “a single-layer neural network in which the outer layer is made of distance units, referred to as prototypes”.

Two specific variants implemented in the scope of this work are the aforementioned methods DSM and LVQ3. Their correction steps are realized as in Algorithm 1.

For DSM and LVQ3, the specific conditions as well as the prototype adjustment actions are defined in Table 3, which refer to the rewarding and penalization terms in Algorithm 2 and 3 (the learning rate parameter is set to a fixed number of 0.01).

Algorithm 1 LVQ-based reduction methods, specific substitutions given in Table 3

```

Input: original samples
Result: reduced prototype set
initialization: prototypes as centres of masses; if there are more:
randomly chosen;
foreach iteration i do
    foreach sample x in the original training set do
        if condA then
            | procA();
        end if
        if condB then
            | procB();
        end if
    end foreach
end foreach
    
```

Algorithm 2 *reward()* for LVQ3 and DSM

```

Input: prototype p, sample x
Output: adjusted prototype  $p_s$ 
 $p_s = p + \alpha(x - p)$ 
    
```

Algorithm 3 *penalize()* for LVQ3 and DSM

```

Input: prototype p, sample x
Output: adjusted prototype  $p_s$ 
 $p_s = p - \alpha(x - p)$ 
    
```

Based on these algorithmic descriptions, a specific runtime complexity analysis of DSM is conducted in Sect. 5.4.

5 Evaluation and results

This section presents the experimental outcomes to evaluate the developed strategies. These results were obtained from conducting the following experiments:

1. Offline tests with datasets from one subject.
2. Online tests with real-time data from one subject (pilot experiments).

Table 3 Specific properties of LVQ3 and DSM

	<i>condA</i>	<i>procA</i>	<i>condB</i>	<i>procB</i>
<i>LVQ3</i>	<i>x</i> is inside window of the 2 nearest prototypes (p_0, p_1)	If <i>x</i> has same label as p_0 but not as p_1 : $reward(p_0, x)$ and $penalize(p_1, x)$. If <i>x</i> has same label as p_1 but not as p_0 : $reward(p_1, x)$ and $penalize(p_0, x)$.	(p_0, p_1) have the same class label as <i>x</i>	$reward(p_0, x)$ and $reward(p_1, x)$
<i>DSM</i>	Label of <i>x</i> and nearest prototype p_0 differ	$penalize(p_0, x)$	<i>condA</i> true and nearest prototype with same class label as <i>x</i> exists (p_s)	$reward(p_s, x)$

3. Online tests with 12 subjects (basic user study).
4. Online tests with 4 subjects (extended user study).

While the offline tests were primarily evaluated by cross-validation accuracy, the main criterion for the online experiments was the success rate (see also requirements Table 2).

Some experiments include specific gestures in one case but do not include these in another. This applies to the pointing gesture to consider and analyze the assumption that it is not as well separable from rest, power grasp, wrist flexion and wrist extension, as these four are from each other. Furthermore, it applies to wrist pronation and supination (again, with and without the pointing gesture) which were chosen to extend the system by a rotational dimension in order to observe the development of performance with an increasing number of degrees of freedom.

5.1 Offline cross-validation accuracy

The offline experiments described in this section are based on several series of EMG data captured from the authors. They provide a rational measure of the applicability by means of cross-validation accuracy. In the datasets, one training repetition consisted of 400 samples per gesture (2 s capturing with 200 Hz sample rate). In each set, each gesture was recorded in several repetitions. For each configuration of considered gestures, several sets of data were recorded; see Table 4 for the resulting number of samples.

The main parameter of kNN, the number of neighbours to consider (k), is varied throughout all cross-validation accuracy experiments. In order to guarantee comparability of the results concerning specific numbers of k across datasets of different sizes, k is not employed as an absolute number of samples. Instead, k is compared in the sense of a relative value k_{rel} , i.e. as the proportion of k relative to the maximum number of samples in the set (n): $k_{rel} = \frac{k}{n}$.

Since the cross-validation is applied block-wise, the maximum k cannot exceed the total number of samples in the set minus the number of samples in a block.

Table 4 Datasets used for the offline tests with overall number of sample vectors (2 s capturing at 200 Hz)

Classes	Sets	Repetitions	Samples
rs, pw, fl, ex	3	4	19,200
rs, pw, fl, ex, pn	3	4	24,000
rs, pw, fl, ex, pr, su	3	4	28,800
rs, pw, fl, ex, pr, su, pn	3	4	33,600

5.1.1 Influence of distance weighting

For the evaluation of cross-validation accuracy when changing the distance weighting factor, the different datasets showed the same qualitative behaviour. The distance from the current to the particular other samples is denoted by d .

Independent of the weighting factor used, it could be observed that high numbers of k usually decreased the cross-validation accuracy. Considering a dataset of four gestures (rs, pw, fl, ex), the accuracy stays at about 99% until k_{rel} is at around 15% in Fig. 19a (using Euclidean distance) for all weighting factors. This threshold value of k_{rel} is even higher in Fig. 19b, namely at about 30% (using Chebyshev distance). Furthermore, in this case, the threshold only applies for weightings of 1 or $\frac{1}{\sqrt{d}}$. For $\frac{1}{d}$ and $\frac{1}{d^2}$, the accuracy stays above 99%. In all cases, the highest accuracy can be noticed with a weighting of $\frac{1}{d^2}$, followed by $\frac{1}{d}$, $\frac{1}{\sqrt{d}}$ and 1. The effect of decreasing accuracy is the most apparent in the case no weighting is applied (decreasing until 0 at about 40–50% relative k). All accuracies stabilize at some point.

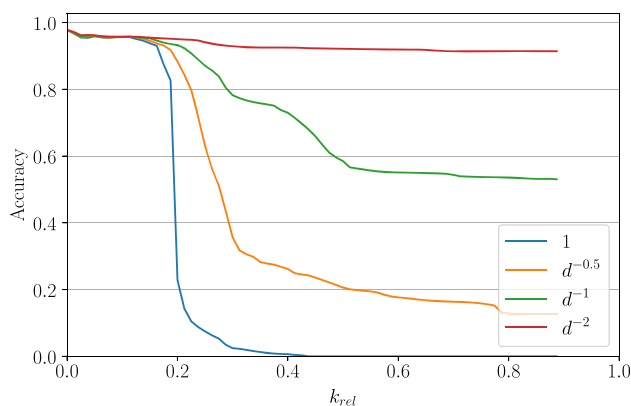
When also including the pointing gesture into the comparison, the behaviour is principally similar. Starting at around 98% accuracy in Fig. 4a and even 100% in Fig. 4b respectively for all weightings at $k = 1$, it drops to 0 for higher k s when using no weighting. Again, the decrease at weighting 1 is the highest, followed by $\frac{1}{\sqrt{d}}$, $\frac{1}{d}$ and finally $\frac{1}{d^2}$. The above-mentioned threshold level of decrease lies at about $k_{rel} = 20\%$.

It can be stated that as soon as a low number of k is meant to be used ($k = 1$ seems suitable in all cases), the weighting scheme does not matter. This means that in this case, for the sake of computation resources, even no weighting could be applied. Nevertheless, if higher numbers of k should be necessary, a higher exponent in the weighting factor’s divisor should be introduced. $\frac{1}{d^2}$ seems to be a good choice for that, without increasing the computation effort notably.

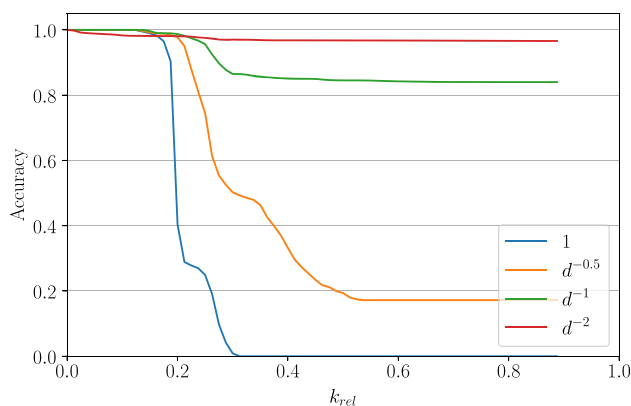
This observation is also confirmed in further tests: Fig. 20c shows this for the Chebyshev distance while additionally including pronation and supination (obtaining a threshold of about $k_{rel} = 15\%$), and Fig. 20b for the Manhattan distance with pronation and supination included without pointing (threshold value some percents higher). In the latter case, the even better performance of a distance weighting of $\frac{1}{d^3}$ is additionally depicted, although the difference only appears after reaching a relative k of 25% and is neglectable due to its small value (0.25%).

5.1.2 Influence of distance metric

The variation of the distance metric showed almost no effect in the case of the four gestures (rs, pw, fl, ex); see Fig. 20a in the Appendix (using a weighting factor of $\frac{1}{d^2}$). An excep-



(a) Euclidean distance



(b) Chebyshev distance

Fig. 4 Distance weighting influence on cross-validation accuracy for sets of five gestures (rs, pw, pn, fl, ex), k relative to number of samples in the set

tion is the Mahalanobis distance which only provided about 96% of accuracy at low numbers of k while the other metrics achieved 100%. Furthermore, in the case of the Mahalanobis distance, the accuracy dropped fast when increasing k until it stabilized at around 69% for $k_{rel} > 50\%$. The accuracy when using the other metrics stayed constant at about 100% (Euclidean drops slightly to 99%).

The observed behaviour in the case of data which included the pointing gesture exposed the following differences (Fig. 5): While the accuracy when applying the Mahalanobis distance showed the same tendency (starting from at about 98% going down to 89%), it also dropped for the other distance metric cases when increasing k_{rel} over 10%. This was mostly noticeable when looking at the Manhattan metric as the accuracy started at about 99% for low numbers of k and decreased until 93%. For the other metrics, it went down from almost 100 to 99% (Chebyshev) and 98% (Euclidean) respectively.

Including the wrist rotation gestures without pointing (weighting factor of $\frac{1}{d^2}$, Appendix Fig. 20b) showed qualitatively the same behaviour as in the already described case

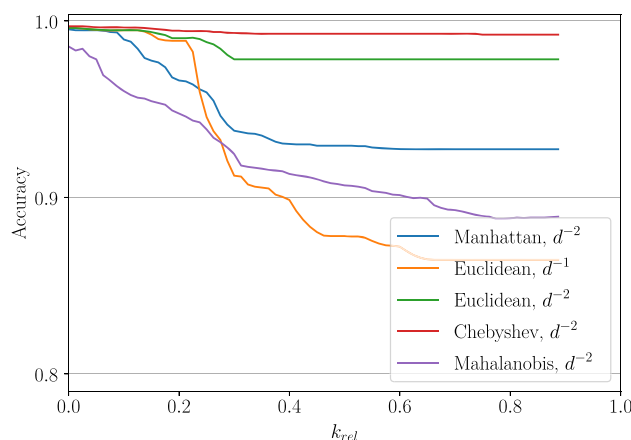


Fig. 5 Distance metric influence on cross-validation accuracy, gestures (rs, pw, pn, fl, ex), k relative to number of samples in the set

where pointing was not included. This means that the Mahalanobis distance started at lower accuracy values than the others (98% instead of 100%) and dropped until it stabilized at 84% (the Chebyshev norm dropped to 99.5%, the Euclidean norm to 99.6% and the Manhattan norm to 99.7%).

When additionally including the pointing gesture again, the effect was comparable, although pointing influenced the Minkowski-norm-based distances slightly more. For the Mahalanobis distance, the accuracy dropped from 97% until it reached a stabilization level of about 88%. The Minkowski-based norms started at 100% accuracy for low numbers of k and decreased at a relative k of about 15% until they reached an accuracy of 96% (Chebyshev), 98.4% (Euclidean) and 99.3% (Manhattan) respectively.

Figure 20c also shows that the behaviour is the same when applying a distance weighting of $\frac{1}{d}$ instead of $\frac{1}{d^2}$ for the cases of Euclidean and Chebyshev norm, although the drop in accuracy is higher.

The evaluation of the distance metrics showed that differences are not evident in all cases. It can be summarized that the Mahalanobis distance is not recommended to be used for the present data. Due to the necessary calculation of the covariance matrix and its inverse, it is also of disadvantage with respect to computational resources.

The Minkowski-distance-based metrics differ regarding the chosen order of norm, especially for high numbers of k . In some cases, the accuracy gets better the higher the order of norm gets (Chebyshev ($p \rightarrow \infty$) is best, followed by Euclidean ($p = 2$) and Manhattan ($p = 1$) in the end). However, when pronation and supination are included, the effect is reversed (both with and without pointing). In fact, this reversed effect is lower than original effect. Nevertheless, the Euclidean norm seems to be a good trade-off to compensate both effects.

5.1.3 Summary

All in all, it could be observed that for the cross-validation accuracy in the case of low numbers of k (relative k until about 5–10%), neither the weighting factor nor the distance metric is of essential importance as long as a Minkowski-based distance norm is applied. However, Fig. 21d shows an exceptional case where there was a clear accuracy difference between the Euclidean (99%) and the Chebyshev (91%) distance even at low numbers of k . In the sense of computational demands, for the lower range of k_{rel} , a distance weighting of 1 (i.e. no further arithmetic operations) is recommended. If also considering k_{rel} higher than 5–10%, a weighting factor of $\frac{1}{d^2}$ might be the best choice, together with the Euclidean norm. These recommendations hold for all tested sets of gestures. Further evaluations of other datasets which confirm this observation are depicted in the Appendix in Fig. 21 with respect to the Euclidean and the Chebyshev distance as well as several weighting factors.

With that, the Euclidean distance and a weighting of $\frac{1}{d^2}$ can be seen as a general recommendation in terms of accuracy for a broad range of k . However, with regard to requirement R5.1, the Euclidean distance might also not be preferred since its calculation (8 subtractions, 8 multiplications, 7 additions in each prediction step due to 8 EMG channels) is more computationally expensive than both the Manhattan distance calculation (8 subtractions, 8 absolute value calculations, 7 additions) and the one of the Chebyshev distance (8 subtractions, 8 absolute value calculations, 7 comparisons for maximum search) which do not require multiplication operations. The individual requirements must be balanced with respect to the specific use case.

5.2 Real-time pilot experiments

The pilot study experiments described in this section were only evaluated on one subject. Although the results obtained from these target achievement tests are therefore not representative, they may give insights on how different means and adaptations in the used algorithms can affect the achieved online success rates in gesture recognition with kNN (with k set to 1 and 10 respectively, equally distributed, results averaged), especially when it comes to intermediate intensity levels of gestures. Following the results from Sect. 5.1 for a broad range of k , for kNN, the Euclidean norm was chosen as distance metric with a weighting of $\frac{1}{d^2}$.

For each pilot experiment, the user first trained the system by capturing data from the exertion of the full-intensity gestures. Each gesture had to be held for 2 s—as in the offline training, resulting in 400 training samples per gesture and repetition. This time, five training repetitions were gathered, i.e. 2000 8-value sample vectors per gesture (see Table 5).

Table 5 Captured training data for online pilot experiments and online user studies with overall number of sample vectors resulting from numbers of participants, repetitions and 2 s capturing at 200 Hz

	Classes	Part.	Rep.	Samples
<i>Pilot Experiments</i>				
	rs, pw, fl, ex	1	5	8000
	rs, pw, fl, ex, pn	1	5	10,000
	rs, pw, fl, ex, pr, su	1	5	12,000
	rs, pw, fl, ex, pr, su, pn	1	5	14,000
<i>User Studies</i>				
<i>Basic</i>	rs, pw, pn, fl, ex	12	3	72,000
<i>Ext.</i>	rs, pw, pn, fl, ex, pr, su	4	3	33,600

In the prediction phase of each pilot experiment, all gestures (apart from rest) were not only tested on full-intensity exertion, but on three different intensity levels ($\frac{1}{3}$, $\frac{2}{3}$, full gesture). For this proportional control, proportionality scaling as described in Sect. 4.1.2 was implemented. To consider a trial a success, the user had to mimic a virtual stimulus, while the real-time continuous prediction was shown in a hand model, and provide spatial matching within a time margin of several seconds. Each combination of gesture and exertion level was tested twice. In this way, the number of prediction samples was several magnitudes higher than the number of training samples, so that issues of overfitting can be further excluded.

As a measure of comparison, the accuracy of ridge regression with random Fourier features (RR-RFF) as state-of-the-art gesture recognition method was also evaluated in each test run.

5.2.1 Rest class thresholding: rest magnitude threshold

The rest magnitude threshold was introduced to cope with the problem of separating intermediate gestures from the rest class in the proposed proportional control. In order to evaluate the influence on the user success rate, multiple tests were conducted with the gesture sets (rs, pw, fl, ex) and (rs, pw, pn, fl, ex).

Figure 6 shows that the standard approach without any rest thresholding yielded averaged success rates of 65% on average for both types of dataset. While the success rates in the variant with pointing could not be considerably increased (only by 4%), it was beneficial for the variant without pointing. Ninety-two percent success rate could be achieved for two times the mean rest signal magnitude ($g = 2$) as well as three times mean rest magnitude ($g = 3$) as threshold. Furthermore, it is noticeable that even without thresholding kNN performed better than RR-RFF when including pointing (63% vs. 46%). When not including point, kNN without thresholding performed worse than RR-RFF (67% vs. 83%).

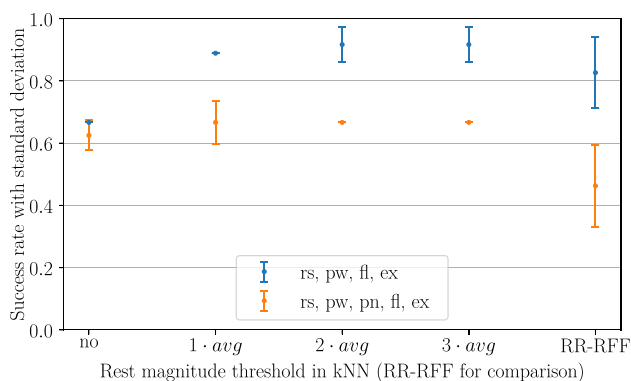


Fig. 6 Influence of rest magnitude thresholding, pilot experiments

But with thresholding in the latter case, kNN's success rate could exceed RR-RFF's (92% vs. 83%).

Since there was no difference recognizable between the success rates of $g = 2$ and $g = 3$, $g = 2.5$ was chosen as a compromise for further experiments. The expected performance of this choice could be confirmed in Fig. 7. Ninety-eight percent success rate could be achieved for kNN without including pointing (in comparison to 80% for RR-RFF) and 64% when including pointing (56% for RR-RFF).

It has to be noted that the results of RR-RFF yielded larger standard deviations than in all kNN cases. This could signify that kNN performs more stable and robust with less non-determinism in the algorithm's behaviour.

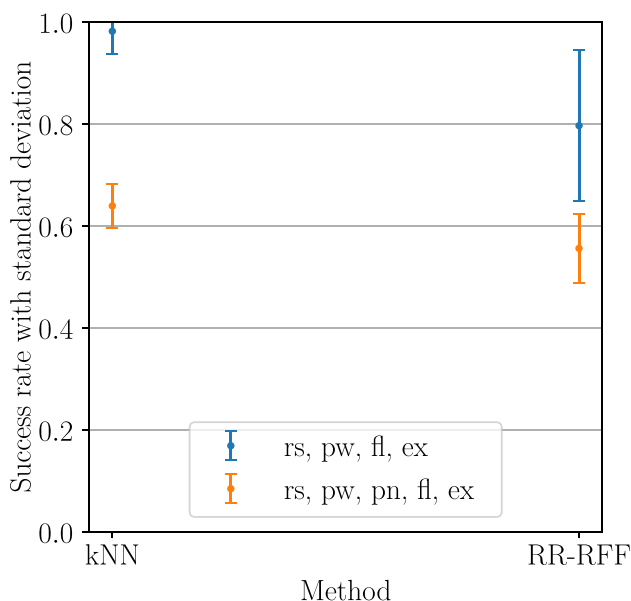


Fig. 7 Influence of selected gestures, introducing rest magnitude threshold ($g = 2.5$), pilot experiments

5.2.2 Proportionality offset scaling: scale offset divisor

As described, besides the rest magnitude threshold, a proportionality offset was introduced. This offset is divided by the scale offset divisor v with the purpose of adjusting the proportionality scaling for intermediate gestures. The target achievement tests described in the following refer to a variety of runs with datasets comprising (rs, pw, fl, ex, pr, su) and (rs, pw, pn, fl, ex, pr, su), respectively. Besides kNN with different scale offset divisors, the performance of RR-RFF and standard ridge regression (RR) was also captured. For the evaluation, a rest magnitude threshold with $g = 2.5$ was chosen, as motivated in Sect. 5.2.1.

Figure 8 depicts the particular results. It is observable that the increase of v could initially improve the average success rate for the used datasets. After reaching a maximum around 5–10, the success rates started to decrease again, probably because of low-intensity levels of gestures getting less reachable due to misclassification with rest. Nevertheless, the approach without any offset (corresponding to an infinite scale offset divisor) still performed clearly better than RR-RFF and standard RR.

Higher averaged success rates were achievable for all scale offset divisors in kNN than for RR and RR-RFF. The best averaged performance when the pointing gesture was included could be achieved for $v = 5$ (94% vs. 43% for RR-RFF); and for $v = 10$ (95% vs. 51% for RR-RFF) when pointing was not included.

With higher scale offset divisors, low-intensity level gestures ($\frac{1}{3}$) get less reachable. This property has been assessed as more severely influencing the motivation of subjects than a reduced magnitude value range, since jumps between rest condition and low-intensity gestures appeared rather difficult than reduced sensitivity perceived as “missing damping”.

Because of this, 5 was favoured over 10, although their performance appeared to be comparable (with 5 providing a slightly better performance when averaging over all dataset configurations, i.e. 93% vs. 91% with a comparable standard deviation).

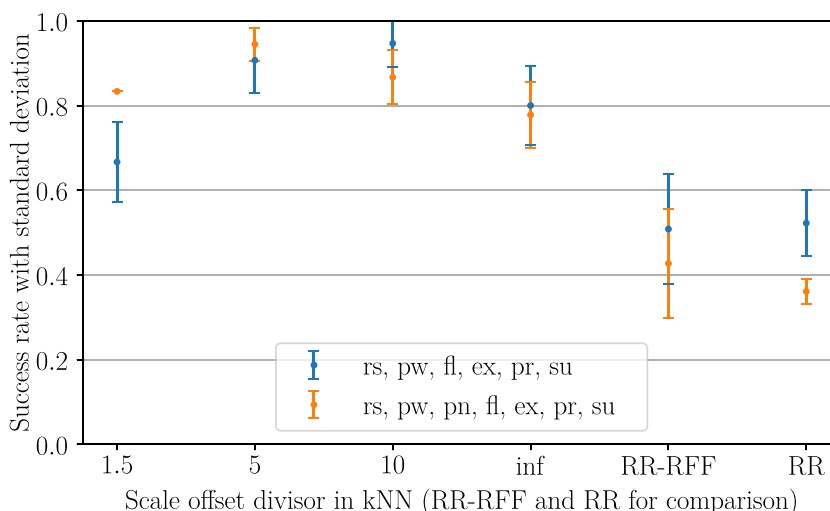
5.3 Evaluation of prototype reduction algorithms

In order to evaluate the performance of the chosen prototype reduction algorithms (see Sect. 3.4), the datasets captured for offline tests (Table 4) were transferred to KEEL and utilized as baseline.

These were pilot results to test the algorithms' accuracy and processing times with reduced datasets.

The reduction was executed on each cross-validation fold of the dataset individually, followed by the actual validation. As the considered algorithms comprise kNN-calculations inside, specifically for obtaining the validation accuracy, its parameters had to be defined. Following the recommenda-

Fig. 8 Influence of scale offset divisor (with rest thresholding, $g = 2.5$), pilot experiments



tions in Sect. 5.1.3, a k of 1, using $\frac{1}{d^2}$ weighting, and the Euclidean distance as metric were configured.

The detailed examinations and results for varying datasets were presented in previous work [96]. From this data, it could be seen that BTS3 and VQ were the lowest performing algorithms in terms of cross-validation accuracy so that these algorithms were disregarded. It also described the exclusion of PSCSA due to slow timing characteristics. Further conclusions drawn in that paper regarding timing can be representatively seen in Fig. 9, where the time needed for reduction to 20 prototypes is depicted. This reduction time adds up with the cross-validation time to constitute the easily measurable overall runtime. Since the validation is the same process for each fold, the validation time can be disregarded so that the runtime qualitatively describes the algorithms’ reduction times for comparison.

With respect to reduction time, MGauss, Chen and LVQPRU exposed a broad variance, leading to the presump-

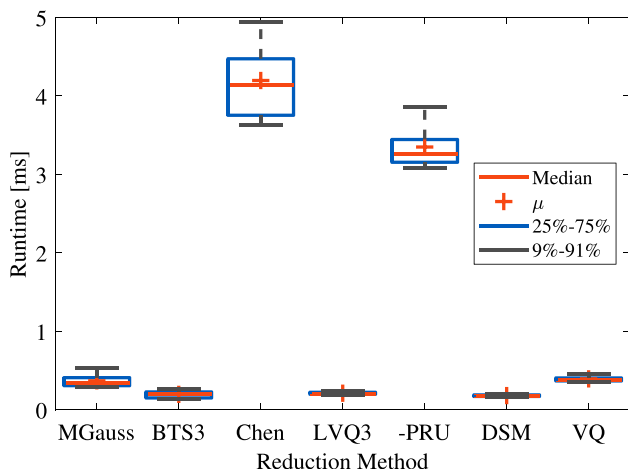


Fig. 9 Average reduction times for seven gestures (rs, pw, pn, fl, ex, pr, su)

tion of reduced time determinism. Furthermore, these showed the highest means and medians of runtime, so that MGauss, LVQPRU and Chen were disregarded, too.

With that, LVQ3 and DSM (also based on LVQ) were the techniques to be chosen for a real-time implementation. With a low runtime of about 0.2 ms in most cases and a low time variance [96], they turned out to be suitable for real-time scenarios, thus fulfilling requirement R5.1. For the present study, particularly, DSM was selected to be examined in any further steps and proved itself as appropriate.

In order to deeper analyze DSM’s suitability for embedded systems, an assessment of the runtime complexity will be made in the subsequent section.

5.4 Runtime complexity of DSM

To analyze the DSM prototype generation algorithm with regard to its runtime complexity, two phases can be distinguished, namely initialization and actual reduction. The phases will be analyzed on their worst-case runtime.

The following conventions are made: $N \equiv$ number of samples in the original training set, $M \equiv$ number of prototypes in the final reduced set, $C \equiv$ number of gestures/classes and $I \equiv$ number of iterations.

The results of this analysis are shown in Table 8. All operations are considered per EMG channel. The initialization process is designed in a way that there is at least one prototype per class by using the class centres as initial prototypes which become adjusted later on by penalizing or rewarding them in the reduction phase.

Summarizing Table 8, this yields the following running time complexity in initialization:

$$\mathcal{O}(C \cdot N + (M - C) \cdot N) = \mathcal{O}(M \cdot N)$$

and in reduction:

$$\mathcal{O}(I \cdot N \cdot (M + M \cdot \log M)) = \mathcal{O}(I \cdot N \cdot M \cdot \log M).$$

When assuming the number of classes to be constant with e.g. $C = 7$ for (rs, pw, pn, fl, ex, pr, su) and also thinking of the number of iterations as a constant, e.g. $I = 20$, an overall running-time complexity of $\mathcal{O}(N \cdot M \cdot \log M)$ can be derived.

It has to be noted that the time complexity in reduction can principally be reduced from $\mathcal{O}(N \cdot M \cdot \log M)$ to $\mathcal{O}(N \cdot M)$ since no complete sorting of the distances between the currently selected sample and the single prototypes is necessary. Instead, a minimum search for the closest sample (1NN approach) and another minimum search for the closest sample with an identical class label would be sufficient—thus leading to two times iterating the full prototype set at most (comparing the distances in the first case and comparing both distance and class label in the second case).

Generally speaking, if $k = 1$ is used in kNN, the runtime complexity can be reduced to linear instead of logarithmic-linear (quasilinear).

Due to the fact that the number of prototypes M is selected small and configured as a constant for the purpose of final prototype set size determinism (e.g. $M = 20$), it might also be disregarded with regard to runtime, leading to an overall complexity of $\mathcal{O}(N)$ in the best case.

Interestingly, this would mean that DSM has the same runtime complexity in reduction (which is only performed once) as standard kNN in each single prediction step (or even better if a higher number of k is used in kNN which would require sorting). Depending on the number of prototypes, the computational effort in a prediction step of DSM-reduced kNN is neglectable, in particular if $k = 1$ is set inside prediction.

5.5 Real-time user studies with multiple subjects

In order to analyze if requirement R4 can be fulfilled by the proposed algorithms, online user studies with multiple subjects were conducted for the evaluation of suitability in practical scenarios. The setup of the experiment is shown in Fig. 10. In the basic user study, it was chosen to compare the following four methods:

- kNN parameterized according to the configuration obtained in the pilot experiments.
- kNN after training dataset reduction by means of DSM.
- Ridge regression with random Fourier features (RR-RFF).
- Standard ridge regression (RF).

In the extended user study, RR was not examined due to a higher number of analyzed gestures.

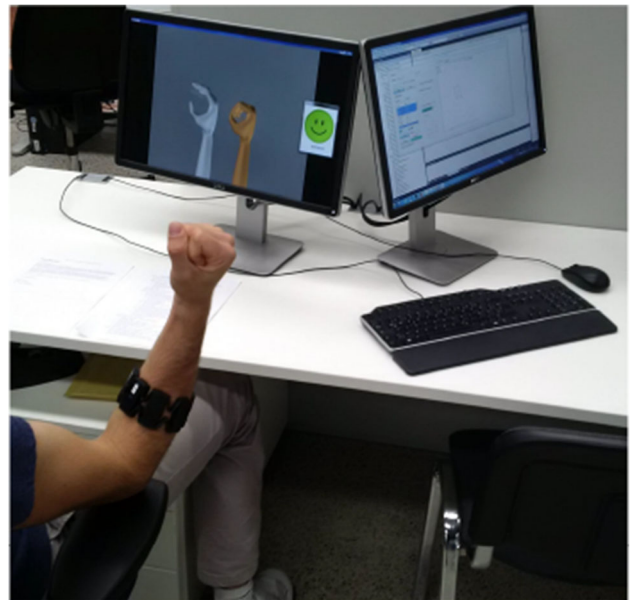


Fig. 10 Experimental setup of the user study, the subject is positioned to only view the left screen while the experimenter can control the experiment on the right screen, e.g. for pausing if needed (without seeing the currently used method)

Following the general recommendations from Sects. 5.1 and 5.2, the configuration of the standard kNN algorithm was set to $k = 1$, the Euclidean distance metric, a weighting of $\frac{1}{d^2}$, a rest magnitude threshold of $g = 2.5$ and proportionality scaling with $v = 5$.

For DSM-kNN, the same parameters were used within the prediction phase. For the reduction phase, DSM was configured to generate 7 prototypes in the final set with 40 iterations enabled. The results obtained will be explained in the following.

All statistical tests conducted in the following refer to a significance level of $\alpha = 0.05$.

5.5.1 Basic user study (five classes)

The subjects provided informed consent and statistical information as follows:

- Age range from 21 to 34 (mean 25, median 24).
- Three female and 9 male.
- One left-handed and 11 right-handed.
- Four already participated in many EMG experiments, 3 in a few and 5 without any EMG experience.

The experimental procedure for the real-time user study followed the same structure as the pilot experiments. The participants put on the Myo armband on their dominant forearm. Afterwards, for the training phase, they followed the visual stimulus (as in Fig. 10) by performing a repetitive series of

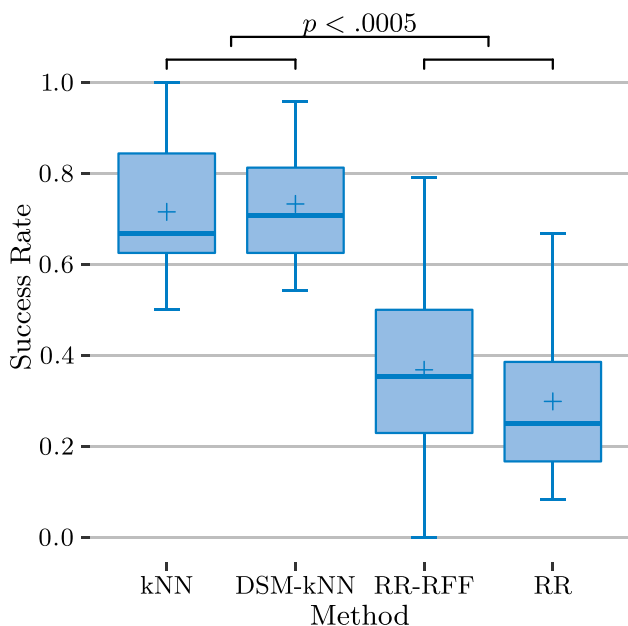
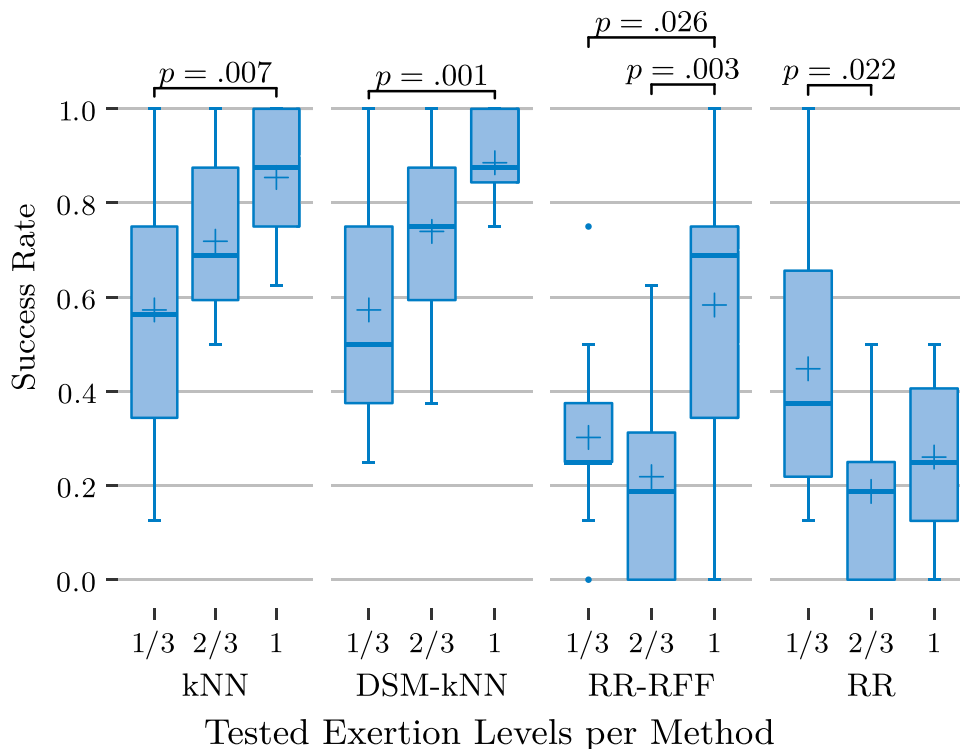


Fig. 11 Basic user study: success rates (over subjects) and significance from ANOVA ($\alpha = 0.05$), showing significant difference in success rates for kNN-based compared to RR-based methods

hand and wrist movements (classes rs, pw, pn, fl, ex) one after another in three repetitions for 2 s each (all with full-intensity exertion). At maximum sampling rate of 200 Hz, this results in 6000 training sample vectors (8 channels) per person ($= 5 \cdot 3 \cdot 2s \cdot 200 \frac{1}{s}$), see Table 5.

Fig. 12 Basic user study: performance of the examined methods for individual gesture exertion levels and significance from ANOVA ($\alpha = 0.05$)



In the prediction and test phase, they were asked to follow the stimulus again, in a total of 96 tasks (randomized but equally distributed among the subjects: 4 gestures (the rest class was not tested), 3 exertion levels, 4 methods, 2 repetitions) with breaks after a quarter, the half and three quarters of tasks. In this phase, they furthermore saw the prediction of the currently exerted gesture in a second hand model. The goal was to match the stimulus and the predicted gesture within some spatial margin and time frame. Success was signalled by a green visualization. Otherwise, a yellow visualization was shown as visual feedback.

The summarized performance of each examined method for the 12 subjects is depicted in Fig. 11, after first averaging the per-level- and -gesture-performances for each subject-method combination. This yields the variance and median of the success rates in a subject-based manner. Overall, it is observable that the success rates achieved with kNN-based methods exceeded the ones from RR-based methods. DSM-reduced kNN performed as good as standard kNN (success rates of 73% and 71% mean, 71% and 67% median respectively), while RR-RFF and RR showed success rates at a lower level (37% and 30% mean, 25% and 25% median). An ANOVA pointed out significance between the groups of kNN-based methods and the group of RR-based methods ($p < 0.0005$), while there is no significant difference within each of the groups.

Figure 12 splits the achieved success rates additionally per gesture exertion level, after first averaging the per-action-performances for each level-subject-combination.

The subject- and level-based variances and medians are depicted for each of the methods. Again, kNN and DSM-kNN do not show major differences, despite the intensive dataset reduction of DSM-kNN. It is apparent that these methods performed better at higher intensity levels (median of 87.5% for full intensity). An exertion level of $\frac{2}{3}$ exhibits intermediate performance, while gestures with only $\frac{1}{3}$ of intensity yielded a mean success rate of 57% for each with high variance. In both methods, the difference between the lowest and the highest level success rate was significant.

When looking into the results from the RR-based methods, it can be observed that there is no level of intensity where those would have outperformed the kNN-based methods in median and mean of the achieved success rate. Interestingly, standard RR yielded a higher number of successes for low-intensity signal amplitudes than when incorporating random Fourier features. In contrast, RR-RFF performed better than RR for gestures of full intensity. For gestures of $\frac{2}{3}$ exertion level, both had a similar performance, with a mean of 19% success rate, the lowest across the intensity levels. This resulted in significance between lowest and intermediate level for RR, as well as between intermediate and highest level for RR-RFF.

At the level of $\frac{1}{3}$, there was no significance between any of the methods ($\alpha = 0.05$). At intermediate level, both kNN and DSM-kNN performed significantly better than RR and RR-RFF ($p < 10^{-5}$). For the full intensity gestures, both

kNN-based methods were significantly better than RR and RR-RFF, while RR-RFF also exposed significantly better performance than RR ($p < 0.01$).

The relation between individual gestures and success rate is presented in Fig. 13, basing on first averaging the per-level-performances for each action-subject-combination. The subject- and action-based variances and medians are depicted for each of the methods, again. It is noticeable that the performance trends were similar for kNN and reduced kNN. For them, the best success rates could be achieved for wrist extension (median of 100% for both, mean of 90% for kNN and 99% for DSM-kNN with small standard deviation).

Wrist flexion was the second best detected gesture for the kNN-based methods (about 76% mean for both), followed by power grasp (67% median), and concluded by the pointing gesture with the worst performance (about 55% mean).

For both kNN and DSM-kNN, the performance difference between wrist extension and pointing was significant. For DSM-kNN, the comparison of wrist extension and power grasp also yielded significance.

While RR exposed the same tendency of gesture performances as the kNN-based methods (on a lower baseline), for RR-RFF, the pointing gesture yielded the best success rate on average (median 58%, mean 51%). Interestingly, wrist extension exposed the worst performance of gestures for RR-RFF (33% median, 31% mean). Wrist flexion and power grasp revealed the same tendency as described for the other meth-

Fig. 13 Basic user study: performance of the examined methods for individual gestures and significance from ANOVA ($\alpha = 0.05$)

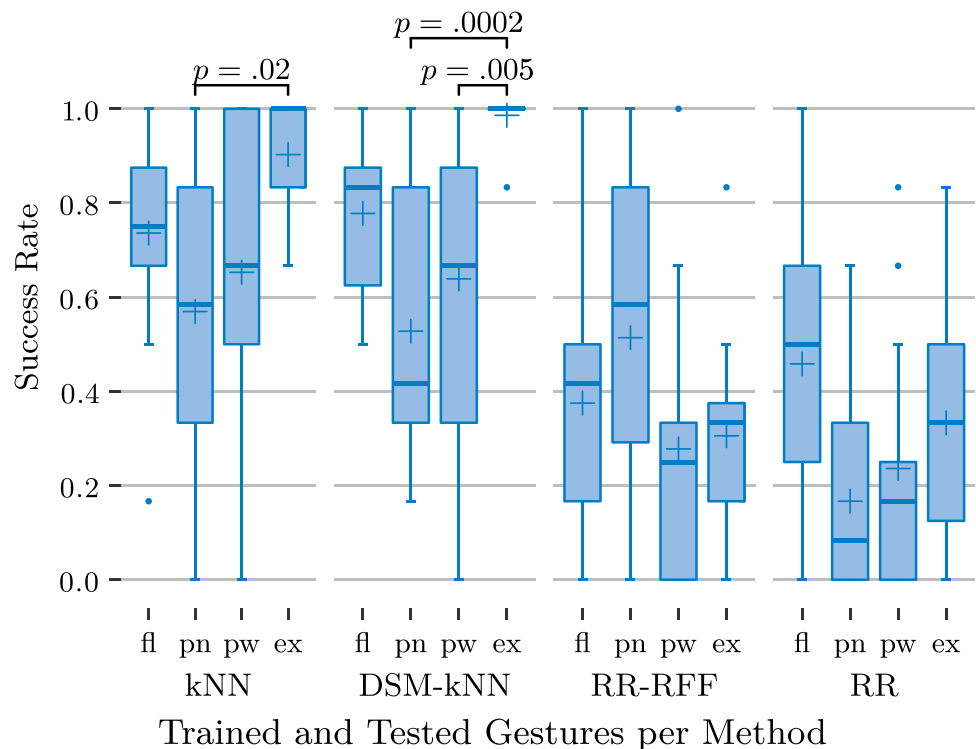
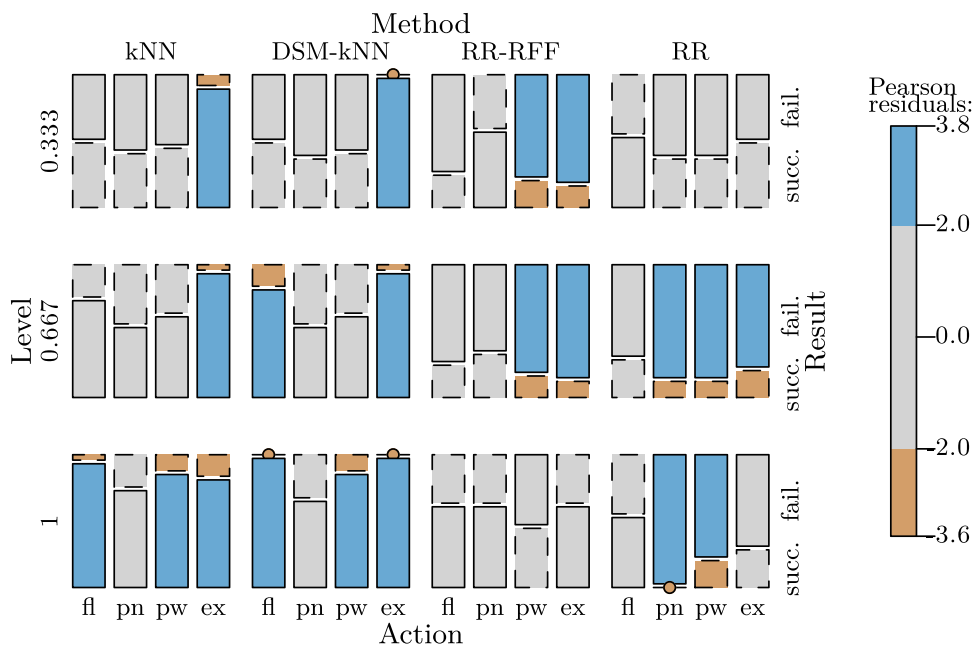


Fig. 14 Basic user study: mosaic plot illustrating significantly more successes for several gestures of different exertion levels for kNN-based methods with significantly more failures in certain cases for RR-based methods



ods. For the RR-based methods, no significance could be shown between different gestures.

With regard to the individual gestures, the group of kNN-based methods performed significantly better than the RR-group for power grasp ($p < 0.05$) as well as wrist extension ($p < 10^{-6}$). For wrist flexion, the same holds ($p < 0.05$) with the exception of the difference between standard kNN and RR not being significant ($\alpha = 0.05$). Concerning the pointing gesture, the kNN-based schemes as well as RR-RFF performed better than standard RR ($p < 0.05$).

The overall relations are summarized in Fig. 14, where the contribution of factor combinations to significance are illustrated.

In Table 6, the online classification times are given for the participants of the user study (ARM Cortex-A72), averaged for all classifications executed at sample rate during prediction, confirming the real-time control properties.

5.5.2 Extended user study (seven classes)

For the purpose of investigating the suitability of the developed methods when including even more gestures in the training, further experiments were conducted as an extension of the described user study. Four subjects who had no EMG experience before but participated in the basic user study were selected again (subjects S5, S7, S9 and S10). On the one hand, the previous participation in the main part of the study might have influenced the impartiality. On the other hand, this might give interesting insights in the algorithms’ performances in the case of low experience with EMG-based control.

Since standard RR showed to not perform well in the main part of the study, this was excluded in the extended evaluation in order to avoid participants’ demotivation. Instead, the wrist rotation gestures pronation and supination were added.

For training the system, data were again gathered for 2 s per gesture at maximum sampling rate of 200Hz with three repetitions each. This was done for all considered classes (rs, pw, pn, fl, ex, pr, su) at full-intensity exertion. This results in 8400 training sample vectors (8 channels) per person ($= 7 \cdot 3 \cdot 2s \cdot 200\frac{1}{s}$), see Table 5.

Table 6 Basic user study (5 classes, 12 subjects), online classification time per subject and method, averaged for all prediction samples, confirming real-time performance

[ms]	kNN	DSM-kNN	RR	RR-RFF
	3.692	0.004	0.001	0.028
	3.775	0.004	0.001	0.035
	3.663	0.004	0.002	0.036
	3.673	0.004	0.001	0.034
	4.343	0.005	0.001	0.035
	3.717	0.004	0.001	0.034
	3.731	0.004	0.001	0.034
	3.733	0.004	0.001	0.035
	4.337	0.005	0.001	0.034
	3.647	0.004	0.001	0.035
	3.678	0.004	0.001	0.033
	3.674	0.004	0.001	0.036
Mean	3.805	0.004	0.001	0.034

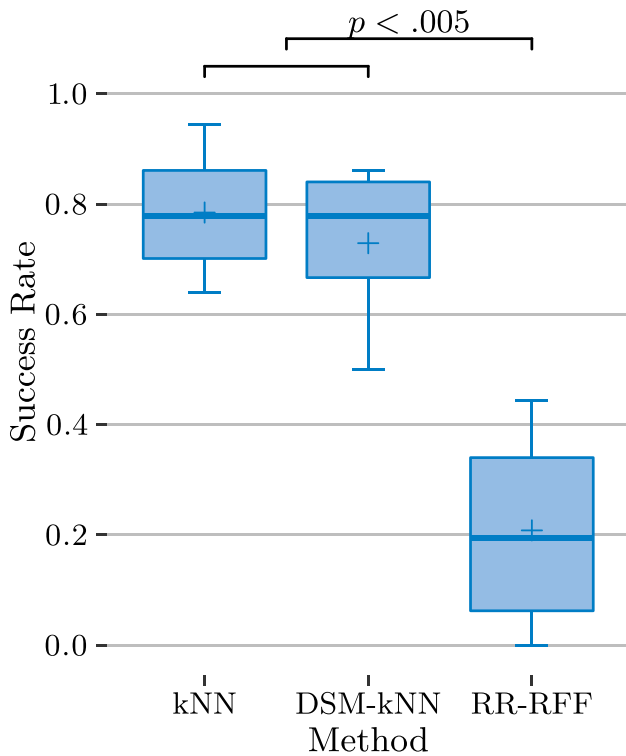


Fig. 15 Extended user study: success rates (over subjects) and significance from ANOVA ($\alpha = 0.05$), showing significant difference in success rates for kNN-based methods compared to RR-RFF

By again repeating each task two times, the number of tasks performed per subject was 108 in total (6 gestures, 3 intensity levels, 3 learning methods, 2 repetitions). Besides these aspects, this part of the study was identical to the previous part. Again, the rest class was not explicitly tested.

The results of the extended user study’s evaluation are summarized in Fig. 15, after the per-level- and -gesture-

performances were averaged for each subject-method combination to obtain the variance and median of the success rates based on the subjects. The kNN-based methods achieved success rate means and medians of over 70%, while RR-RFF performed significantly worse (median 19%, mean 21%) with $p < 0.005$. This time, the DSM-reduced kNN yielded slightly lower values than standard kNN (both medians and kNN mean at 78%, but kNN-DSM mean at 73%).

In Fig. 16, it is observable that the lowest exertion levels did not show the worst performance for any of the methods. Instead, the success rates at $\frac{1}{3}$ exertion level were similar to the $\frac{2}{3}$ level but had slightly higher means and medians. The best behaviour could be reached at full intensity (kNN: 92% median, 90% mean; DSM-kNN: 88% median and mean). All three tested methods showed the same tendency in terms of performance for individual levels—with RR-RFF’s success rates shifted towards a lower baseline (e.g. for full intensity median 42%, mean 40%). RR-RFF could not outperform kNN or DSM-kNN at any level. Between the different levels of a single method, there is no significance.

For each individual level, the success rates of RR-RFF and the group of kNN-based methods differ significantly ($p < 0.05$), while there is no significance between kNN and DSM-kNN ($\alpha = 0.05$).

The examination of the individual gestures (see Fig. 17) exposes a behaviour that was comparable between kNN and DSM-kNN. Wrist flexion and extension achieved the highest success rates (100% median for both methods). Pointing and pronation performed worst here (medians of 75% as well as 58% for kNN and 50% as well as 67% for DSM-kNN). In contrast, for RR-RFF, pronation performed the best with similar success rates (median 50%) as kNN and DSM-kNN, while power exposed severe issues (median 0%, mean 4%). The analysis of variances between the success rates of ges-

Fig. 16 Extended user study: performance of the examined methods for individual gesture exertion levels and significance from ANOVA ($\alpha = 0.05$)

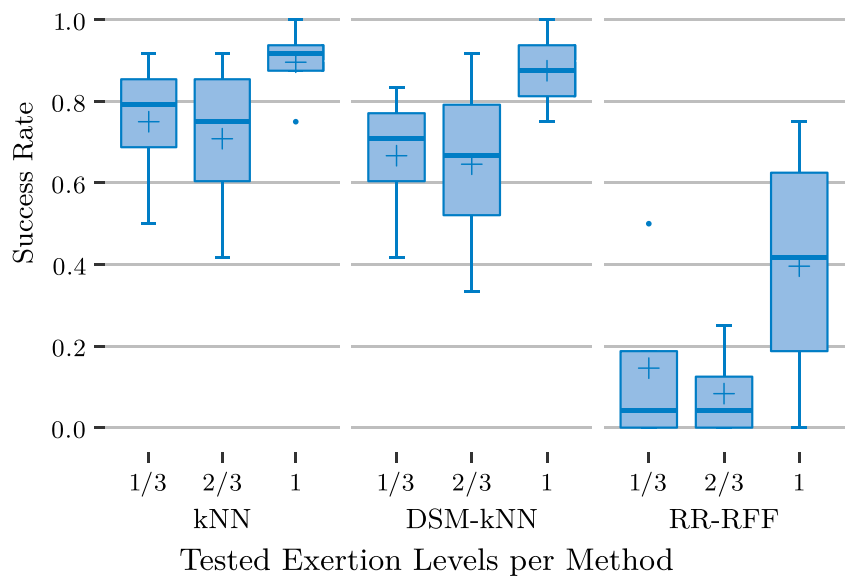
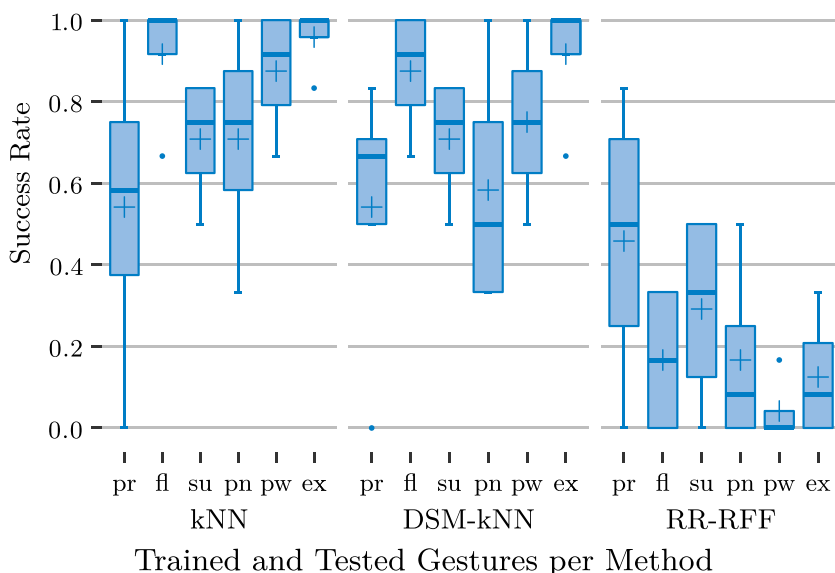


Fig. 17 Extended user study: performance of the examined methods for individual gestures and significance from ANOVA ($\alpha = 0.05$)



ture performances for a single method could not show any significance within the method.

However, significant differences could be found between the methods for individual gestures: For power grasp ($p < 0.0005$), extension ($p < 0.0001$), flexion ($p < 0.001$) and supination ($p < 0.05$), RR-RFF was significantly worse than both kNN and DSM-kNN. There was no significant success rate difference for neither pronation nor pointing ($\alpha = 0.05$).

In Fig. 18, the effects of the combinations of factors on significance are summarized. When examining the success rates for individual gestures at specific exertion levels, it can be seen that RR-RFF contributed to significantly more failures than the group of kNN-based methods at levels of $\frac{1}{3}$ and $\frac{2}{3}$, specifically for power grasp and wrist extension.

As for the basic user study, the computation times needed for each classification were measured and averaged per user

and method. These are presented in Table 7, again providing a confirmation for the real-time capability of the proposed methods.

6 User study discussion

Overall, it could be shown in the user studies that both the standard kNN scheme as well as the DSM-reduced technique yielded significantly higher success rates than RR-RFF and RR in most of the scenarios. The behaviour that kNN-based methods performed significantly better at higher exertion levels in the basic user study could be due to the fact that gestures of low intensity are more often subject to misclassification. This might result from a too high rest magnitude threshold which causes movements with low signal amplitudes being

Fig. 18 Extended user study: mosaic plot illustrating significantly more failures for several gestures of different exertion levels for RR-RFF

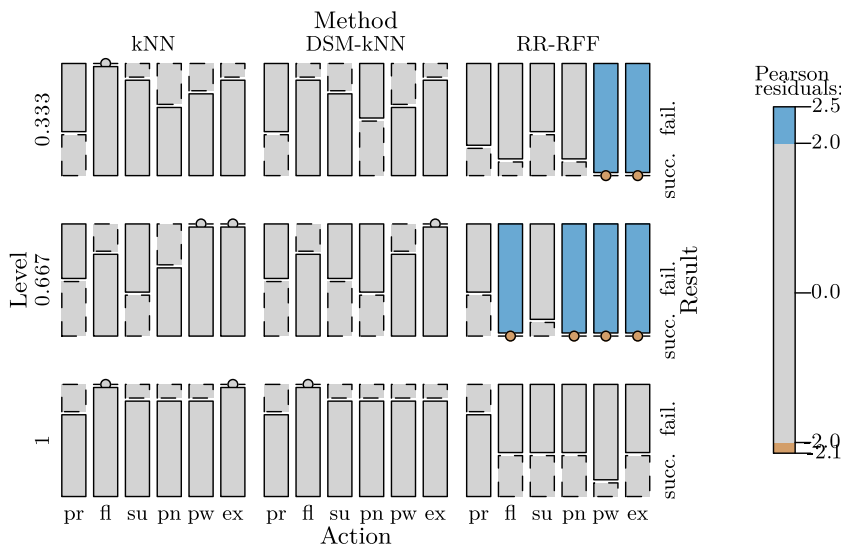


Table 7 Extended user study (7 classes, 4 subjects), online classification time performance per subject, averaged for all prediction samples

[ms]	kNN	DSM-kNN	RR-RFF
	6.010	0.006	0.033
	6.039	0.007	0.040
	6.051	0.006	0.040
	6.048	0.007	0.037
Mean	6.037	0.006	0.037

classified as rest state. In the extended user study, there was no significance for this difference. However, the effect could probably be curtailed in general by a learning process where the subjects would get used to the specific behaviour of the algorithm and adapt to it. Furthermore, the limb position effect might influence the results with respect to the average rest signal magnitude, although the subjects sat in a standardized pose.

Since the principle idea behind the use of random Fourier features is to fit cosine functions in the regression space, this might lead to unwanted behaviour at intermediate levels while showing better performance for especially the full gesture exertion (and slightly also for low levels). For RR, a similar principle holds, with the exception of using linear instead of cosine functions. Probably, the assumption of linear dependency is valid for small intensities, hence showing better success rates for $\frac{1}{3}$ exertion level when comparing RR to RR-RFF. For increasing intensities, the proportionality behaviour might change to other functional dependencies. Nevertheless, the regression approach should fit the level of full intensity since it was trained on that. This means a higher number of successes for full intensity gestures. Regarding the basic user study, it has to be noted that RR-RFF performed significantly better than RR at full gesture exertion. There, kNN and DSM-kNN also had significantly higher success rates than the RR-based methods at intermediate and full gesture level. In the extended user study, kNN and DSM-kNN were significantly better than RR-RFF at all gesture levels.

For the kNN-based methods, the highest success rates were achieved for wrist extension (and in the second user study also for wrist flexion). The reason for this could be that power grasp and pointing gesture are most probably mainly exerted by the same group of muscles, but the wrist gestures are not—thus leading to better separability of those classes. In the basic user study, precisely, the success rate difference between pointing gesture and wrist extension was significant; for DSM-kNN, power grasp also differed significantly from wrist extension. This might point towards the described explanation. In the extended user study, there was no significance for that.

Since the muscle groups activated in pointing gesture and power grasp are spatially close to each other from a biomechanical point of view, it could be explained that these two yielded the lowest success rates in the first user study, probably due to misclassifications between the two classes (they differed not significantly). The gestures are only distinguished in one degree of freedom (index finger), while the other degrees of freedom are the same. In the extended user study, pronation and supination also performed at the same lower level; however, there was no significance provable. Standard RR showed the same behaviour in terms of individual gesture performances as the kNN-based methods (at a generally reduced success rate baseline), with the exception of wrist extension performing worse than flexion, although this difference was not significant in any case. Extension and flexion address the same degree of freedom which might therefore cause smaller deviations. With that, the differences between the wrist movements in RR-RFF could also be explained. One reason for pointing yielding the most successes in RR-RFF (although not significantly) could be that it was exerted by the subjects in a different manner than when the other methods were tested. Basically, the exertion can take place by also using the muscle group used for extension (stressing the index finger movement), instead of the muscle group used for flexion and power (where the activity patterns of the flexed fingers are stressed). If this was the case, for RR-RFF, this might also explain the reduced performance of wrist extension. Nevertheless, the question would be why this would have been only the case for RR-RFF. A reason therefore might be traced back to the specific properties of random Fourier features when subjects try to reduce overshooting or similar. However, for the success rate differences between individual gestures in RR and RR-RFF, no significance could be substantiated.

In the extended study, pointing and pronation performed with the lowest success rates for the kNN-based methods, probably due to potentially addressing the same groups of muscles by these gestures, although there was no significance. The observation that RR-RFF performed worst in the extended user study (with the main exception of pronation where there was no significance between RR-RFF and the other schemes) could potentially be related to its capability of predicting multiple degrees of freedom in parallel. This might be leading to unstable predictions when it comes to predicting only a single degree of freedom. In order to realize a higher extent of comparability, only single degrees of freedom might be checked in the target achievement tests instead of all of them in future experiments. All in all, the extended user study could confirm the general observations made in the study's previous main part so that further in-depth experiments including more than the originally tested four gestures are recommended. Except for RR-RFF, there

was no drop of success rate in comparison to the basic user study.

The fact that the DSM-reduced kNN did not perform worse (and in some cases even better) than the non-reduced kNN could be attributed to a possibly better subject's adaptability to the algorithm since there are less samples available whose decision borders are more clearly defined than when making use of the whole non-reduced dataset with possibly more abrupt changes in the decision borders the user can hardly learn. Another reason might be that noisy instances are discarded in the reduction process so that samples leading to misclassifications and worsening the performance are not considered anymore. Nevertheless, the slightly better performance of DSM-kNN might also be just resulting from stochastic factors. There is no statistical significance for that observation.

The generally better performance of the chosen kNN-based techniques can be also addressed to the fact that they are classification-based methods—extended by proportionality schemes. This is why in the presented manner, they are not suited for simultaneous control, i.e. predicting mixed states of different gestures. In contrast, the RR-based methods also consider these states as an inherent property of regression. This means, as soon as multiple degrees of freedom are trained, RR and RR-RFF can get influenced by multiple degrees of freedom in parallel, although in the prediction tasks, only a certain degree is tested at once. Therefore, the advantage of simultaneous control is at the expense of stability and robustness and vice versa for kNN.

In order to find a good measure of comparability with regard to the development of the success rate over time for the individual methods, the available time slots during one experiment for a subject have been split into eight time subgroups. The differences over time are rather minor. The main perceivable difference originated from the choice of method. However, some interpretation of minor tendencies is given in the following.

It could be observed for the kNN-based methods that the performance in the first time slot was below the ones in the later timeslots. This could potentially be explained by a learning effect, i.e. the subject adapting to the specific behaviour of the algorithm. The same effect could explain that in DSM-kNN, the highest success rates were achieved towards the end of the experiment. At the very end, the performance decreased again, potentially due to muscle fatigue setting in. It could be seen that RR-RFF and RR exposed the same monotonicity when it comes to the mean performance over time, namely a sequence of possible learning effect, muscle fatigue and learning effect again. The learning effect might have set in again after each break. Apart from this, the RR-based methods showed a constant median of 33% success rate over time (with the exception of the first time slot in RR).

However, to gain representative insights, it might be useful to look into the time performance of individual gestures.

7 Conclusions and outlook

In this work, a detailed examination of kNN-based learning techniques in the context of electromyographically controlled prostheses was conducted.

Summing up, with the proposed and implemented algorithms, all requirements stated in Table 2 could be fulfilled.

First, the influence of several parameters on the block-wise cross-validation was examined for kNN classification. This showed that setting $k = 1$ yielded excellent results, sometimes causing a ceiling effect. Accuracies often close to 100%, always higher than 95% for gesture subsets from (rs, pw, pn, fl, ex, pr, su), could be achieved, thus satisfying requirement R1.

The analysis of numbers k on a higher scale was based on the inspiration that for increasing ks , the upper probability bound of classification error decreases from about twice to once the Bayes probability of error [105]. Furthermore, previous work observed that “the standard deviations tend to decrease as k -values increase” [53]. Independent of the mentioned bounds, the experiments showed that the overall performance did not enhance for increasing ks —as opposed to the expectations. All in all, relative ks until 5% can be used without explicit drops in accuracy.

With the choice of $k = 1$, the runtime complexity of the algorithm is reduced to linear time since instead of sorting distances (with logarithmic-linear time in the best case), a minimum search is sufficient, favouring the applicability on embedded systems.

In contrast to the Mahalanobis distance, the distance metrics based on the Minkowski norm proved well. In some cases, a higher order of norm yielded better results. This was the case when not considering the wrist rotation gestures where the Chebyshev distance performed the best. With pronation and supination included, there was the reverse effect, i.e. the Manhattan norm performing best. The Euclidean distance seems to be a good compromise to equalize both effects, although its calculation (multiplications) is slightly more computationally expensive than those of the Manhattan and Chebyshev norms.

The chosen factor of distance weighting seemed to not heavily influence the classification accuracy if k was low. Nonetheless, higher exponents in the factor's divisor showed drastic improvements for a high k so that this might be considered when choosing k_{rel} (i.e. the proportion of k and the total sample size) over 5%. A weighting factor of $\frac{1}{d^2}$ seemed to be sufficient in any case. When specifically referring to computational requirements (R5.1), a weighting of

$\frac{1}{d}$ might be of preference due to less multiplication operations. For $k_{rel} < 5\%$, applying no weighting would be even more advantageous due to the reduced computational demand without loss in accuracy.

Regarding the pilot experiments, the offset scaling showed the best effect in optimizing the trade-off between a high value range of exertion levels and low-intensity gestures still being reachable. A scale offset divisor set to 5 could increase the success rate up to over 90% both for the gesture set (rs, pw, pn, fl, ex) as well as the specifically problematic set of additionally including another degree of freedom in the form of pronation and supination. Nevertheless, possibilities for a usage of more sophisticated proportionality schemes have to be evaluated. Instead of using linear dependencies, other functions to realize interpolation might be tested.

The approach of rest magnitude thresholding could overcome the problem related to gestures exerted with less than half the full intensity being detected as rest state. A value of about 2.5 times the average magnitude across all rest training samples showed the best results.

The adaptations made to kNN have no influence on kNN's original incrementality so that requirement R4 is inherently guaranteed.

The motivation behind investigating prototype reduction mechanisms was to cope with the inherent issue of instance-based learning manifesting in very high computational demands during prediction. The concept of prototype reduction promised to reduce these demands by preponing calculations to the training phase where the amount of data to be processed in prediction is reduced, thus accomplishing both requirement R5.1 and requirement R5.2.

From the multitude of algorithms proposed in literature, the DSM algorithm singled out as highly appropriate. It is deterministic with regard to the size of the final reduced prototype set to be generated (memory determinism, requirement R5.2), yielded high cross-validation accuracies using EMG datasets captured with the Myo armband (requirement R1) at a low amount of time needed for reduction (requirement R5.3), and is considered to be incremental (requirement R4).

So far, the best results for DSM were shown when using the centres of the classes as initial prototypes. This was implemented by means of calculating the class-wise means. Nevertheless, this can lead to misclassification in the case of overlapping classes, specifically if they are concentric [78, p. 335], why it is proposed to use the median instead. This might be examined in further research.

Furthermore, requirement R5.4 is also met due to DSM's elementary composition of an initialization of prototypes in the class centres and a correction phase shifting them by either penalizing or rewarding them depending on different basic criteria. As it is meant to be used with the proposed

kNN implementation which guarantees this requirement, R2 is also fulfilled.

In the final user study, requirement R3 was evaluated for both the standard kNN approach (extended by the introduced adaptations) and kNN applied on the dataset reduced by DSM. It could be shown that the kNN methods performed significantly better than the ridge regression methods. Within the groups themselves, there was no statistical significance determinable.

Interestingly, DSM-kNN and kNN performed equally well; DSM-kNN sometimes even better, even though a reduction of over 99% was achieved by relying on only seven prototypes in total. By this, requirement R3 as to user satisfaction in real scenarios is fulfilled. The extended user study on additionally including the wrist rotation gestures, achieving very good success rates, might give further motivation to deeper analyze this influence in the context of representative studies.

Regarding the measured online timing behaviour in the proposed setup, it could be shown that DSM reduces kNN's classification times by three orders of magnitude. With that, it achieves the same order as standard RR and is one order of magnitude faster than RR-RFF. This means, DSM-kNN is excellently suited for real-time control [25].

It could be shown that DSM-kNN is an appropriate method to be integrated into wearable prosthetic devices. Its properties lead to fulfilling non-functional requirements with respect to dependability and energy consumption, among other properties, favouring battery-powered portable myocontrol implementations.

A limitation of the conducted user study is the combined comparison of simultaneous and non-simultaneous control. The advantage of higher stability and robustness in the kNN-based methods comes at the disadvantage of not allowing to predict multiple degrees of freedom in parallel. On the contrary, the RR-based methods are subject to instabilities because of their tendency towards simultaneous predicting multiple degrees of freedoms. Approaches of how to handle mixed states in the case of kNN could comprise explicit learning on mixed gestures or implicit learning by automatically creating mixtures of gestures. Another approach is suggested in [5]: if a non-simultaneous control method yields a low precision for the current gesture, it is switched to a simultaneous control scheme.

For a further evaluation of the kNN-based methods, a user study with handicapped subjects is of high importance. Additionally, besides using the visual feedback of the hand model, experiments with prosthetic devices have to be conducted to identify the potential of the methods in terms of helpfulness for amputees. Longer-term studies may provide information about the influence of potential electrode shift as well as how to counteract this effect (as in [80]).

Such experiments could also reveal further insights with regard to preprocessing, choice of the features (potentially combined with feature selection for horizontal data reduction), additional modalities or coping with the limb position effect (where the armband’s integrated IMU might be useful). Since we solely rely on the linear envelope, a combination of our work with an embedded feature analysis [83] seems promising to be investigated.

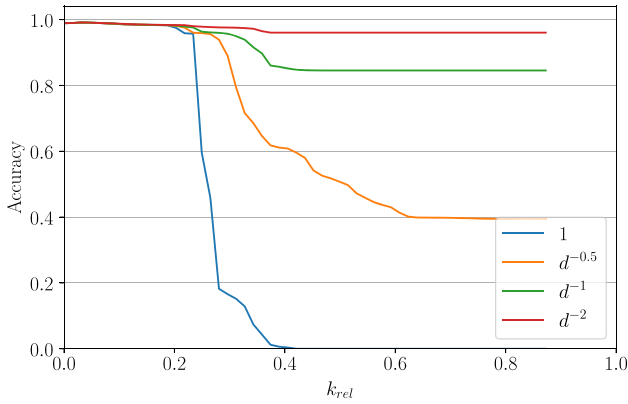
All in all, this paper confirmed the suitability of nearest neighbour learning techniques in the context of proportional myocontrol. Specifically, the results of using decision surface mapping at very high reduction rates (>99%) motivate to further look into this promising method.

Appendix 1. DSM runtime complexity

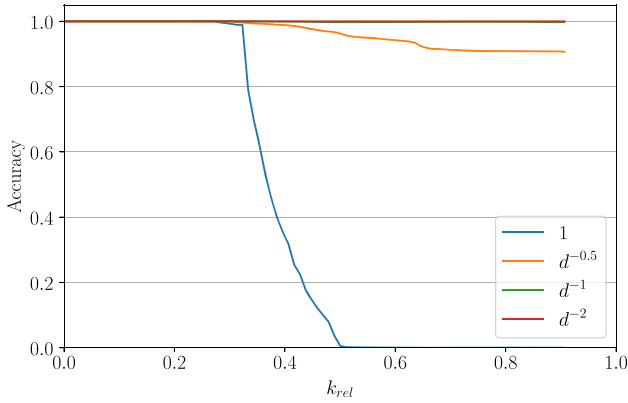
Table 8 Time complexity of DSM, training phase consists of initialization and reduction, see Sect. 5.4

Initialization			
	Step	Complexity	Comment
	Create new prototype set (signal in X /label in Y)	$\mathcal{O}(1)$	if static arrays used
C times:	Get samples of specific class	$\mathcal{O}(N)$	
	Average these samples	$\mathcal{O}(N)$	
	Set row in prototype set	$\mathcal{O}(1)$	if implemented as array with indexing
$(M - C)$ times:	If selection from specific class: get its samples	$\mathcal{O}(N)$	
	Get random number	$\mathcal{O}(1)$	
	Get sample with that random ID	$\mathcal{O}(1)$	if implemented as array with indexing
	Set row in prototype set	$\mathcal{O}(1)$	if implemented as array with indexing
Reduction			
	Step	Complexity	Comment
$I \cdot N$ times:	Calculate sample distances to prototypes	$\mathcal{O}(M)$	
	Sort these distances	$\mathcal{O}(M \cdot \log M)$	could be replaced by minimum search: $\mathcal{O}(M)$
	Penalize	$\mathcal{O}(1)$	
	Set row in prototype set	$\mathcal{O}(1)$	if implemented as array with indexing
	Get nearest within class	$\mathcal{O}(M)$	
	Reward	$\mathcal{O}(1)$	
	Set row in prototype set	$\mathcal{O}(1)$	if implemented as array with indexing

Appendix 2. Further offline cross-validation results

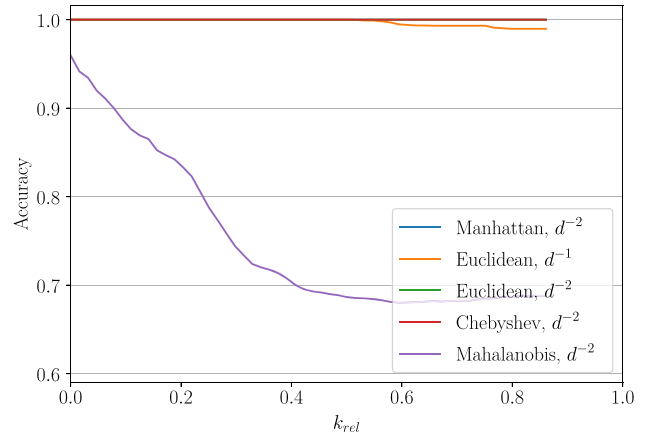


(a) Euclidean distance

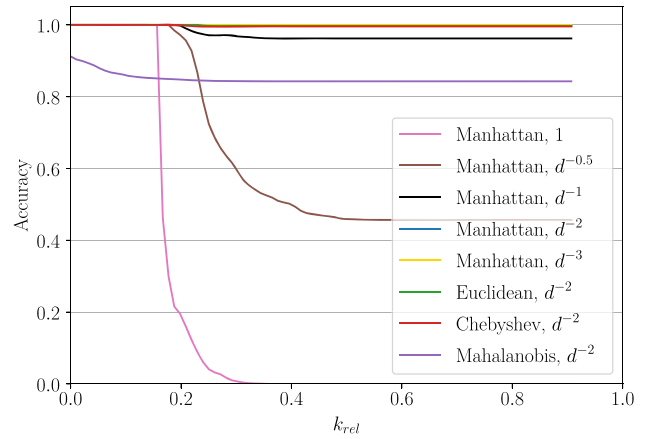


(b) Chebyshev distance

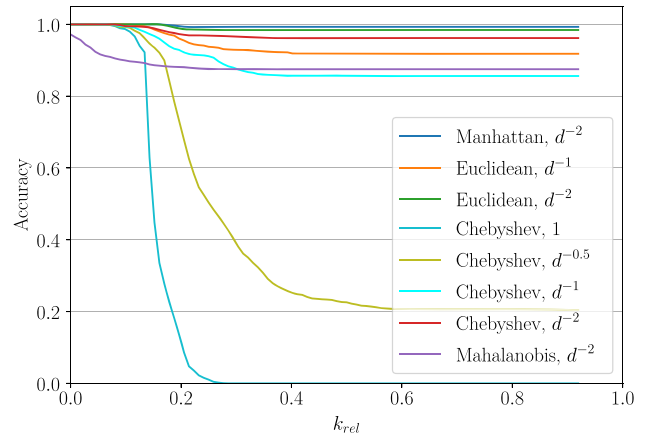
Fig. 19 Distance weighting influence on cross-validation accuracy for sets of four gestures (rs, pw, fl, ex)



(a) Gestures (rs, pw, fl, ex)



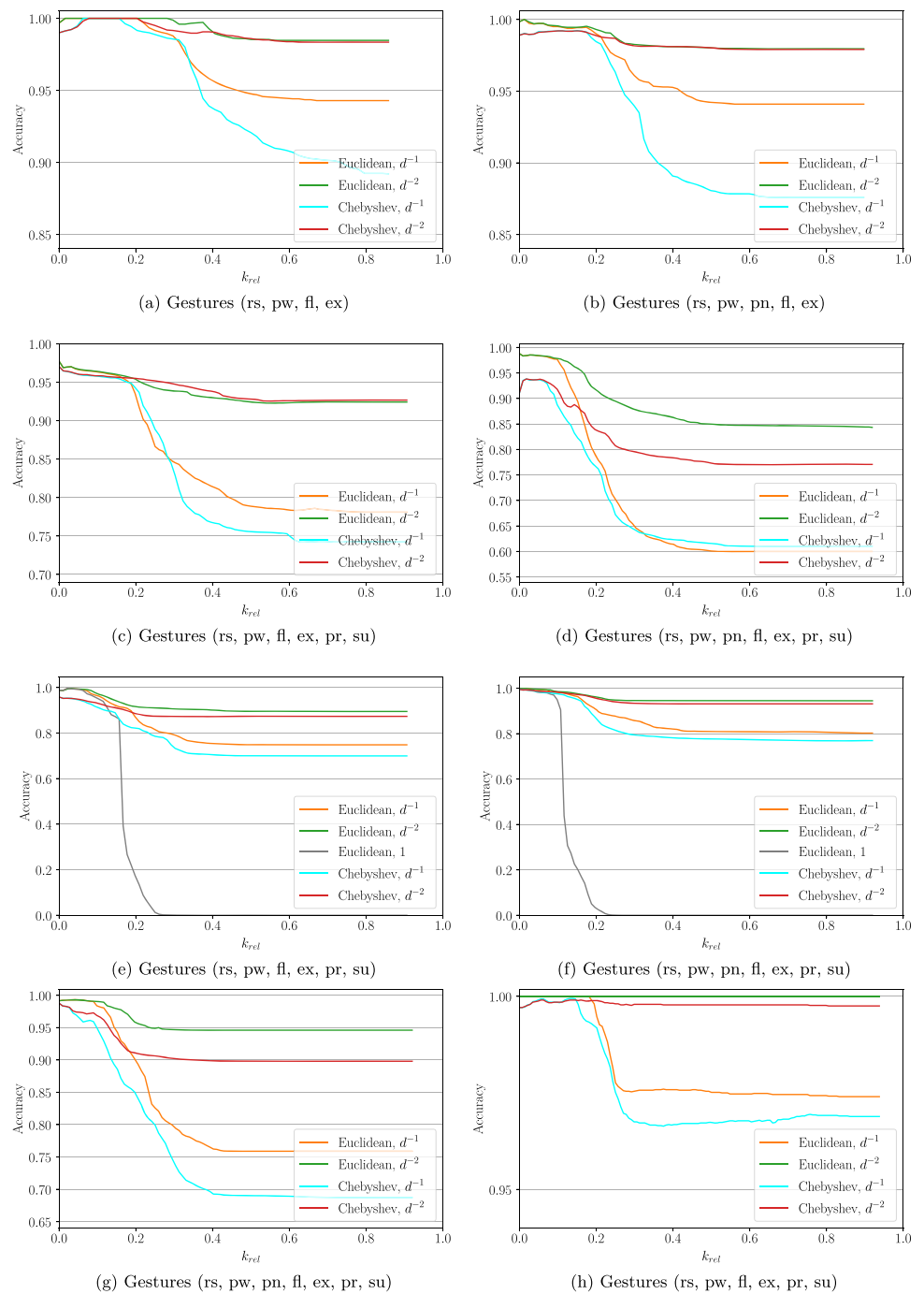
(b) Gestures (rs, pw, fl, ex, pr, su)



(c) Gestures (rs, pw, pn, fl, ex, pr, su)

Fig. 20 Distance metric and weighting influence on cross-validation accuracy

Fig. 21 Influence of metrics/weightings on cross-validation accuracy, further datasets



Funding Open Access funding enabled and organized by Projekt DEAL.

Declarations

Ethical approval and informed consent All procedures performed in the studies that involved human participants were approved by the internal committee for personal data protection of the German Aerospace Center (DLR) and followed the World Medical Association’s Declaration of Helsinki. Each participant was informed about the experimental process beforehand and signed an informed consent form.

Conflict of interest The authors declare no competing interests.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the

permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Aggarwal CC (2014) Data classification: algorithms and applications. Chapman and Hall/CRC, New York. <https://doi.org/10.1201/b17320>
- Ahmad, S.A., Chappell, P.H.: Surface EMG classification using moving approximate entropy. In: 2007 International Conference on Intelligent and Advanced Systems, pp. 1163–1167 (2007). <https://doi.org/10.1109/ICIAS.2007.4658567>
- Ajiboye A, Weir R (2005) A heuristic fuzzy logic approach to EMG pattern recognition for multifunctional prosthesis control. *IEEE transactions on neural systems and rehabilitation engineering?: a publication of the IEEE Engineering in Medicine and Biology Society* 13:280–91. <https://doi.org/10.1109/TNSRE.2005.847357>
- Al-Faiz, M.Z., Ali, A.A., Miry, A.H.: A k-nearest neighbor based algorithm for human arm movements recognition using EMG signals. In: 2010 1st International Conference on Energy, Power and Control (EPC-IQ), pp. 159–167 (2010)
- Amsuess S, Vujaklija I, Goebel P, Roche AD, Graitmann B, Aszmann OC, Farina D (2016) Context-dependent upper limb prosthesis control for natural and robust use. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 24(7):744–753. <https://doi.org/10.1109/TNSRE.2015.2454240>
- Antfolk, C., Sebelius, F.: A comparison between three pattern recognition algorithms for decoding finger movements using surface EMG. In: *MyoElectric Controls/Powered Prosthetics Symposium* (2011)
- Arjunan, S.P., Kumar, D.K.: Fractal based modelling and analysis of electromyography (EMG) to identify subtle actions. In: 2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, pp. 1961–1964 (2007). <https://doi.org/10.1109/IEMBS.2007.4352702>
- Arveti, M., Gini, G., Folgheraiter, M.: Classification of EMG signals through wavelet analysis and neural networks for controlling an active hand prosthesis. In: 2007 IEEE 10th International Conference on Rehabilitation Robotics, ICORR'07, pp. 531–536 (2007). <https://doi.org/10.1109/ICORR.2007.4428476>
- Bajramovic, F., Mattern, F., Butko, N., Denzler, J.: A comparison of nearest neighbor search algorithms for generic object recognition. In: Proceedings of the 8th International Conference on Advanced Concepts For Intelligent Vision Systems, ACIVS'06, pp. 1186–1197. Springer-Verlag, Berlin, Heidelberg (2006). https://doi.org/10.1007/11864349_108
- Barzilay O, Wolf A (2011) A fast implementation for EMG signal linear envelope computation. *Journal of Electromyography and Kinesiology* 21(4):678–682. <https://doi.org/10.1016/j.jelekin.2011.04.004>
- Boschmann, A., Platzner, M.: Reducing classification accuracy degradation of pattern recognition based myoelectric control caused by electrode shift using a high density electrode array. In: 2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society, pp. 4324–4327 (2012). <https://doi.org/10.1109/EMBC.2012.6346923>
- Cervantes A, Galván I, Isasi P (2007) An adaptive Michigan approach PSO for nearest prototype classification. In: Mira J, Álvarez JR (eds) *Nature Inspired Problem-Solving Methods in Knowledge Engineering*. Springer, Berlin Heidelberg, Berlin, Heidelberg, pp 287–296
- Chan ADC, Englehart KB (2005) Continuous myoelectric control for powered prostheses using hidden Markov models. *IEEE Transactions on Biomedical Engineering* 52(1):121–124. <https://doi.org/10.1109/TBME.2004.836492>
- Chang GC, Kang WJ, Luh JJ, Cheng CK, Lai JS, Chen JJJ, Kuo TS (1996) Real-time implementation of electromyogram pattern recognition as a control command of man-machine interface. *Medical Engineering & Physics* 18(7):529–537. [https://doi.org/10.1016/1350-4533\(96\)00006-9](https://doi.org/10.1016/1350-4533(96)00006-9)
- Chen C, Józwick A (1996) A sample set condensation algorithm for the class sensitive artificial neural network. *Pattern Recognition Letters* 17(8):819–823. [https://doi.org/10.1016/0167-8655\(96\)00041-4](https://doi.org/10.1016/0167-8655(96)00041-4)
- Chen, H., Zhang, Y., Zhang, Z., Fang, Y., Liu, H., Yao, C.: Exploring the relation between EMG sampling frequency and hand motion recognition accuracy. In: 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp. 1139–1144 (2017). <https://doi.org/10.1109/SMC.2017.8122765>
- Cipriani C, Antfolk C, Controzzi M, Lundborg G, Rosen B, Carrozza MC, Sebelius F (2011) Online myoelectric control of a dexterous hand prosthesis by transradial amputees. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 19(3):260–270. <https://doi.org/10.1109/TNSRE.2011.2108667>
- Cover TM, Hart PE (2006) Nearest neighbor pattern classification. *IEEE Transactions on Information Theory* 13(1):21–27. <https://doi.org/10.1109/TIT.1967.1053964>
- Decaestecker C (1997) Finding prototypes for nearest neighbour classification by means of gradient descent and deterministic annealing. *Pattern Recognition* 30(2):281–288. [https://doi.org/10.1016/S0031-3203\(96\)00072-6](https://doi.org/10.1016/S0031-3203(96)00072-6)
- Dellacasa Bellingegni A, Gruppioni E, Colazzo G, Davalli A, Sacchetti R, Guglielmelli E, Zollo L (2017) NLR, MLP, SVM, and LDA: a comparative analysis on EMG data from people with trans-radial amputation. *Journal of NeuroEngineering and Rehabilitation* 14(1):82. <https://doi.org/10.1186/s12984-017-0290-6>
- Dening D, Gray F, Haralick R (1983) Prosthesis control using a nearest neighbor electromyographic pattern classifier. *Biomedical Engineering, IEEE Transactions on* 30:356–360. <https://doi.org/10.1109/TBME.1983.325138>
- Englehart K, Hudgins B (2003) A robust, real-time control scheme for multifunction myoelectric control. *IEEE Transactions on Biomedical Engineering* 50(7):848–854. <https://doi.org/10.1109/TBME.2003.813539>
- Englehart K, Hudgins B, Parker P, Stevenson M (1999) Classification of the myoelectric signal using time-frequency based representations. *Medical Engineering & Physics* 21(6):431–438. [https://doi.org/10.1016/S1350-4533\(99\)00066-1](https://doi.org/10.1016/S1350-4533(99)00066-1)
- Esposito, D., Andreozzi, E., Fratini, A., Gargiulo, G.D., Savino, S., Niola, V., Bifulco, P.: A piezoresistive sensor to measure muscle contraction and mechanomyography. *Sensors* 18(8) (2018). <https://doi.org/10.3390/s18082553>
- Farrell TR, Weir RF (2007) The optimal controller delay for myoelectric prostheses. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 15(1):111–118. <https://doi.org/10.1109/TNSRE.2007.891391>
- Farry KA, Walker ID, Baraniuk RG (1993) Myoelectric teleoperation of a complex robotic hand. *IEEE Trans. Robotics and Automation* 12:775–788
- Fernández F, Isasi P (2004) Evolutionary design of nearest prototype classifiers. *Journal of Heuristics* 10(4):431–454. <https://doi.org/10.1023/B:HEUR.0000034715.70386.5b>
- Garain U (2008) Prototype reduction using an artificial immune model. *Pattern Anal. Appl.* 11:353–363. <https://doi.org/10.1007/s10044-008-0106-1>
- García S, Derrac J, Cano J, Herrera F (2012) Prototype selection for nearest neighbor classification: taxonomy and empirical study. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34(3):417–435. <https://doi.org/10.1109/TPAMI.2011.142>

30. Geethanjali P (2015) Comparative study of PCA in classification of multichannel EMG signals. *Australasian Physical & Engineering Sciences in Medicine* 38(2):331–343. <https://doi.org/10.1007/s13246-015-0343-8>
31. Geethanjali P, Ray KK (2011) Identification of motion from multi-channel EMG signals for control of prosthetic hand. *Australasian Physical & Engineering Sciences in Medicine* 34(3):419–427. <https://doi.org/10.1007/s13246-011-0079-z>
32. Geethanjali, P., Ray, K.K., Shanmuganathan, P.V.: Actuation of prosthetic drive using EMG signal. In: TENCON 2009 - 2009 IEEE Region 10 Conference, pp. 1–5 (2009). <https://doi.org/10.1109/TENCON.2009.5396091>
33. Geva S, Sitte J (1991) Adaptive nearest neighbor pattern classifier. *IEEE transactions on neural networks / a publication of the IEEE Neural Networks Council* 2:318–22. <https://doi.org/10.1109/72.80344>
34. Gijsberts A, Bohra R, Sierra Gonzalez D, Werner A, Nowak M, Caputo B, Roa M, Castellini C (2014) Stable myoelectric control of a hand prosthesis using non-linear incremental learning. *Frontiers in Neurobotics* 8:8. <https://doi.org/10.3389/fnbot.2014.00008>
35. Gijsberts, A., Metta, G.: Incremental learning of robot dynamics using random features. pp. 951–956 (2011). <https://doi.org/10.1109/ICRA.2011.5980191>
36. Gini G, Arveti M, Somlai I, Folgheraiter M (2012) Acquisition and analysis of EMG signals to recognize multiple hand movements for prosthetic applications. *Appl. Bionics Biomechanics* 9(2):145–155. <https://doi.org/10.3233/ABB-2011-0024>
37. Glette, K., Gruber, T., Kaufmann, P., Torresen, J., Sick, B., Platzner, M.: Comparing evolvable hardware to conventional classifiers for electromyographic prosthetic hand control. In: 2008 NASA/ESA Conference on Adaptive Hardware and Systems, pp. 32–39 (2008). <https://doi.org/10.1109/AHS.2008.12>
38. Gonzalez AI, Grana M, D'Anjou A (1995) An analysis of the GLVQ algorithm. *IEEE Transactions on Neural Networks* 6(4):1012–1016. <https://doi.org/10.1109/72.392266>
39. Güler NF, Koçer S (2005) Classification of EMG signals using PCA and FFT. *J. Med. Syst.* 29(3):241–250. <https://doi.org/10.1007/s10916-005-5184-7>
40. Hamamoto Y, Uchimura S, Tomita S (1997) A bootstrap technique for nearest neighbor classifier design. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 19(1):73–79. <https://doi.org/10.1109/34.566814>
41. Hannaford, B., Lehman, S.: Short time Fourier analysis of the electromyogram: fast movements and constant contraction. *IEEE Transactions on Biomedical Engineering* **BME-33**(12), 1173–1181 (1986). <https://doi.org/10.1109/TBME.1986.325697>
42. Haris, M., Chakraborty, P., Rao, B.V.: EMG signal based finger movement recognition for prosthetic hand control. In: 2015 Communication, Control and Intelligent Systems (CCIS), pp. 194–198 (2015). <https://doi.org/10.1109/CCIntelS.2015.7437907>
43. Hu X, Wang Z, Ren X (2005) Classification of surface EMG signal with fractal dimension. *Journal of Zhejiang University. Science. B* 6:844–8. <https://doi.org/10.1631/jzus.2005.B0844>
44. Huang, Y., Englehart, K., Hudgins, B., Chan, A.D.C.: Optimized gaussian mixture models for upper limb motion classification. *The 26th Annual International Conference of the IEEE Engineering in Medicine and Biology Society* **1**, 72–75 (2004)
45. Hudgins B, Parker P, Scott RN (1993) A new strategy for multi-function myoelectric control. *IEEE Transactions on Biomedical Engineering* 40(1):82–94. <https://doi.org/10.1109/10.204774>
46. Jeong, E.c., Kim, S.j., Song, Y.r., Lee, S.m.: Comparison of wrist motion classification methods using surface electromyogram. *Journal of Central South University* **20**(4), 960–968 (2013). <https://doi.org/10.1007/s11771-013-1571-2>
47. Jiang, M.W., Wang, R.C., Wang, J.Z., Jin, D.W.: A method of recognizing finger motion using wavelet transform of surface EMG signal. In: 2005 IEEE Engineering in Medicine and Biology 27th Annual Conference, pp. 2672–2674 (2005). <https://doi.org/10.1109/IEMBS.2005.1617020>
48. Kakoty, N.M., Hazarika, S.M.: Classification of grasp types through wavelet decomposition of EMG signals. *2009 2nd International Conference on Biomedical Engineering and Informatics* pp. 1–5 (2009)
49. Kartsch, V., Benatti, S., Mancini, M., Magno, M., Benini, L.: Smart wearable wristband for EMG based gesture recognition powered by solar energy harvester. In: 2018 IEEE International Symposium on Circuits and Systems (ISCAS), pp. 1–5 (2018). <https://doi.org/10.1109/ISCAS.2018.8351727>
50. Khushaba, R.N., Al-Timemy, A., Al-Ani, A., Al-Jumaily, A.: Myoelectric feature extraction using temporal-spatial descriptors for multifunction prosthetic hand control. In: 2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), pp. 1696–1699 (2016). <https://doi.org/10.1109/EMBC.2016.7591042>
51. Khushaba RN, Kodagoda S, Takruri M, Dissanayake G (2012) Toward improved control of prosthetic fingers using surface electromyogram (EMG) signals. *Expert Systems with Applications* 39(12):10731–10738. <https://doi.org/10.1016/j.eswa.2012.02.192>
52. Kim, J., Mastnik, S., André, E.: EMG-based hand gesture recognition for real time biosignal interfacing. In: Proceedings of the 13th International Conference on Intelligent User Interfaces, IUI '08, pp. 30–39. ACM, New York, NY, USA (2008). <https://doi.org/10.1145/1378773.1378778>
53. Kim KS, Choi HH, Moon CS, Mun CW (2011) Comparison of k-nearest neighbor, quadratic discriminant and linear discriminant analysis in classification of electromyogram signals based on the wrist-motion directions. *Current Applied Physics* 11(3):740–745. <https://doi.org/10.1016/j.cap.2010.11.051>
54. Kim SW, Oommen BJ (2003) A brief taxonomy and ranking of creative prototype reduction schemes. *Pattern Analysis & Applications* 6(3):232–244. <https://doi.org/10.1007/s10044-003-0191-0>
55. Kirlangic, M.E., Denizhan, Y.: Fractal modelling for pattern recognition via artificial neural networks. In: 2000 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.00CH37100), vol. 6, pp. 3610–3613 vol.6 (2000). <https://doi.org/10.1109/ICASSP.2000.860183>
56. Kohonen T (1990) The self-organizing map. *Proceedings of the IEEE* 78(9):1464–1480. <https://doi.org/10.1109/5.58325>
57. Kuiken TA, Miller LA, Turner K, Hargrove LJ (2016) A comparison of pattern recognition control and direct control of a multiple degree-of-freedom transradial prosthesis. *IEEE Journal of Translational Engineering in Health and Medicine* 4:1–8. <https://doi.org/10.1109/jtehm.2016.2616123>
58. Kusner, M.J., Tyree, S., Weinberger, K., Agrawal, K.: Stochastic neighbor compression. In: Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32, ICML'14, pp. II–622–II–630. JMLR.org (2014). <https://doi.org/10.5555/3044805.3044962>
59. Kuzborskij, I., Gijsberts, A., Caputo, B.: On the challenge of classifying 52 hand movements from surface electromyography. In: 2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society, pp. 4931–4937 (2012). <https://doi.org/10.1109/EMBC.2012.6347099>
60. Lam W, Keung CK, Liu D (2002) Discovering useful concept prototypes for classification based on filtering and abstraction. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 24:1075–1090. <https://doi.org/10.1109/TPAMI.2002.1023804>

61. Li, J., Manry, M.T., Yu, C., Wilson, D.R.: Prototype classifier design with pruning. *International Journal on Artificial Intelligence Tools* **14**(01n02), 261–280 (2005). <https://doi.org/10.1142/S0218213005002090>
62. Li, Q.X., Chan, P.P.K., Zhou, D., Fang, Y., Liu, H., Yeung, D.S.: Improving robustness against electrode shift of SEMG based hand gesture recognition using online semi-supervised learning. In: 2016 International Conference on Machine Learning and Cybernetics (ICMLC), vol. 1, pp. 344–349 (2016). <https://doi.org/10.1109/ICMLC.2016.7860925>
63. Library, U.S.N.: Electromyography mesh descriptor data 2019 (1999). <https://meshb.nlm.nih.gov/record/ui?name=Electromyography>
64. Liu H, Hussain F, Tan CL, Dash M (2002) Discretization: an enabling technique. *Data Min. Knowl. Discov.* 6:393–423. <https://doi.org/10.1023/A:1016304305535>
65. Lozano M, Sotoca JM, Sánchez JS, Pla F, Pkalska E, Duin RPW (2006) Experimental study on prototype optimisation algorithms for prototype-based classification in vector spaces. *Pattern Recogn.* 39(10):1827–1838. <https://doi.org/10.1016/j.patcog.2006.04.005>
66. Luo Zhizeng, Gao Jian: Using singular eigenvalues of wavelet coefficient as the input of SVM to recognize motion patterns of the hand. In: 2005 International Conference on Neural Networks and Brain, vol. 3, pp. 1477–1481 (2005). <https://doi.org/10.1109/ICNNB.2005.1614910>
67. Maitrot A, Lucas MF, Doncarli C, Farina D (2005) Signal-dependent wavelets for electromyogram classification. *Medical & Biological Engineering & Computing* 43(4):487–492. <https://doi.org/10.1007/BF02344730>. Erratum. In: *Med Bio Eng Comput.* 2007. 45(8):807
68. Merriam-Webster: Definition of electromyograph (1944). <https://www.merriam-webster.com/dictionary/electromyography>
69. Micera S, Sabatini AM, Dario P (2000) On automatic identification of upper-limb movements using small-sized training sets of EMG signals. *Medical engineering & physics* 22(8):527–33
70. Min Lei, Zhi-Zhong Wang, Li-Yu Cai, Hai-Hong Zhang, Hua Cai: An EMG classifying method based on Bayes' criterion. In: Proceedings of the 20th Annual International Conference of the IEEE Engineering in Medicine and Biology Society. Vol.20 Biomedical Engineering Towards the Year 2000 and Beyond (Cat. No.98CH36286), vol. 5, pp. 2625–2626 vol.5 (1998). <https://doi.org/10.1109/IEMBS.1998.744998>
71. Nagata, K., Adno, K., Magatani, K., Yamada, M.: A classification method of hand movements using multi channel electrode. In: 2005 IEEE Engineering in Medicine and Biology 27th Annual Conference, pp. 2375–2378 (2005). <https://doi.org/10.1109/IEMBS.2005.1616944>
72. Nanni L, Lumini A (2009) Particle swarm optimization for prototype reduction. *Neurocomputing* 72:1092–1097. <https://doi.org/10.1016/j.neucom.2008.03.008>
73. Negi, S., Kumar, Y., Mishra, V.M.: Feature extraction and classification for EMG signals using linear discriminant analysis. In: 2016 2nd International Conference on Advances in Computing, Communication, Automation (ICACCA) (Fall), pp. 1–6 (2016). <https://doi.org/10.1109/ICACCAF.2016.7748960>
74. Nowak, M., Bongers, R.M., van der Sluis, C.K., Albu-Schäffer, A., Castellini, C.: Simultaneous assessment and training of an upper-limb amputee using incremental machine-learning-based myocontrol: a single-case experimental design. *Journal of NeuroEngineering and Rehabilitation* **20**(1) (2023). <https://doi.org/10.1186/s12984-023-01171-2>
75. Odorico R (1997) Learning vector quantization with training count (LVQTC). *Neural networks?: the official journal of the International Neural Network Society* 10(6):1083–1088. [https://doi.org/10.1016/s0893-6080\(97\)00012-9](https://doi.org/10.1016/s0893-6080(97)00012-9)
76. Paek, A.Y., Brown, J.D., Gillespie, R.B., O'Malley, M.K., Shewokis, P.A., Contreras-Vidal, J.L.: Reconstructing surface EMG from scalp EEG during myoelectric control of a closed looped prosthetic device. In: 2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), pp. 5602–5605 (2013). <https://doi.org/10.1109/EMBC.2013.6610820>
77. Peerdeman B, Boere D, Witteveen H, Veld R, Hermens H, Stramigioli S, Rietman J, Veltink P, Misra S (2011) Myoelectric forearm prostheses: state of the art from a user-centered perspective. *Journal of rehabilitation research and development* 48:719–37. <https://doi.org/10.1682/JRRD.2010.08.0161>
78. Perez JC, Vidal E (1993) Constructive design of LVQ and DSM classifiers. In: Mira J, Cabestany J, Prieto A (eds) *New Trends in Neural Computation*. Springer, Berlin Heidelberg, Berlin, Heidelberg, pp 334–339
79. Phinyomark A, Quaine F, Charbonnier S, Serviere C, Tarpin-Bernard F, Laurillau Y (2013) EMG feature evaluation for improving myoelectric pattern recognition robustness. *Expert Systems with Applications* 40(12):4832–4840. <https://doi.org/10.1016/j.eswa.2013.02.023>
80. Prahm C, Schulz A, Paaßen B, Schoiswohl J, Kaniusas E, Dorffner G, Hammer B, Aszmann O (2019) Counteracting electrode shifts in upper-limb prosthesis control via transfer learning. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 27(5):956–962
81. Purushothaman G (2016) Myoelectric control of prosthetic hands: state-of-the-art review. *Medical Devices: Evidence and Research* 9:247–255. <https://doi.org/10.2147/MDER.S91102>
82. Purushothaman, G., Ray, K.K.: Motion control of drives for prosthetic hand using continuous myoelectric signals. *Journal of The Institution of Engineers (India): Series B* **97**(1), 55–60 (2016). <https://doi.org/10.1007/s40031-014-0172-2>
83. Raurale SA, McAllister J, del Rincon JM (2020) Real-time embedded EMG signal analysis for wrist-hand pose identification. *IEEE Transactions on Signal Processing* 68:2713–2723. <https://doi.org/10.1109/TSP.2020.2985299>
84. Rekhi, N.S., Singh, H., Arora, A.S., Rekhi, A.K.: Analysis of EMG signal using wavelet coefficients for upper limb function. In: 2009 2nd IEEE International Conference on Computer Science and Information Technology, pp. 357–361 (2009). <https://doi.org/10.1109/ICCSIT.2009.5234929>
85. Ren, X., Huang, H., Deng, L.: MUAP classification based on wavelet packet and fuzzy clustering technique. In: 2009 3rd International Conference on Bioinformatics and Biomedical Engineering, pp. 1–4 (2009). <https://doi.org/10.1109/ICBBE.2009.5163091>
86. Robinson, C.P., Li, B., Meng, Q., Pain, M.T.: Pattern classification of hand movements using time domain features of electromyography. In: Proceedings of the 4th International Conference on Movement Computing, MOCO '17, pp. 27:1–27:6. ACM, New York, NY, USA (2017). <https://doi.org/10.1145/3077981.3078031>
87. Saridis, G.N., Gootee, T.P.: EMG pattern analysis and classification for a prosthetic arm. *IEEE Transactions on Biomedical Engineering* **BME-29**(6), 403–412 (1982). <https://doi.org/10.1109/TBME.1982.324954>
88. Scheme E, Englehart K (2013) Training strategies for mitigating the effect of proportional control on classification in pattern recognition-based myoelectric control. *JPO Journal of Prosthetics and Orthotics* 25(2):76–83. <https://doi.org/10.1097/jpo.0b013e318289950b>
89. Scheme E, Lock B, Hargrove L, Hill W, Kuruganti U, Englehart K (2014) Motion normalized proportional control for improved

- pattern recognition-based myoelectric control. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 22(1):149–157
90. Shin S, Langari R, Tafreshi R (2014). A performance comparison of EMG classification methods for hand and finger motion. <https://doi.org/10.1115/DSCC2014-5993>
 91. Sijiang Du, Vuskovic, M.: Temporal vs. spectral approach to feature extraction from prehensile EMG signals. In: Proceedings of the 2004 IEEE International Conference on Information Reuse and Integration, 2004. IRI 2004., pp. 344–350 (2004). <https://doi.org/10.1109/IRI.2004.1431485>
 92. Skalak, D.B.: Prototype and feature selection by sampling and random mutation hill climbing algorithms. In: W.W. Cohen, H. Hirsh (eds.) *Machine Learning Proceedings 1994*, pp. 293 – 301. Morgan Kaufmann, San Francisco (CA) (1994). <https://doi.org/10.1016/B978-1-55860-335-6.50043-X>
 93. Sueaseanak, D., Wibirama, S., Chanwimalueang, T., Pintavirooj, C., Sangworasil, M.: Comparison study of muscular-contraction classification between independent component analysis and artificial neural network. In: 2008 International Symposium on Communications and Information Technologies, pp. 468–472 (2008). <https://doi.org/10.1109/ISCIT.2008.4700236>
 94. Sukhan Lee, Saridis, G.: The control of a prosthetic arm by EMG pattern recognition. *IEEE Transactions on Automatic Control* 29(4), 290–302 (1984). <https://doi.org/10.1109/TAC.1984.1103521>
 95. Sziburis, T.: Nearest-neighbour-based learning techniques for proportional myocontrol in prosthetics. Master's thesis, University of Trento, Università degli Studi di Trento (2019). <https://elib.dlr.de/133564>. At German Aerospace Center (DLR)
 96. Sziburis, T., Nowak, M., Brunelli, D.: Prototype reduction on SEMG data for instance-based gesture learning towards real-time prosthetic control. In: Proceedings of the 14th International Joint Conference on Biomedical Engineering Systems and Technologies - Volume 2: BIOSIGNALS., pp. 299–305. INSTICC, SciTePress (2021). <https://doi.org/10.5220/0010327500002865>
 97. Sziburis T, Nowak M, Brunelli D (2022) KNN learning techniques for proportional myocontrol in prosthetics. In: Torricelli D, Akay M, Pons JL (eds) *Converging Clinical and Engineering Research on Neurorehabilitation IV*. Springer International Publishing, Cham, pp 679–683. https://doi.org/10.1007/978-3-030-70316-5_109
 98. Tello, R.M.G., Bastos-Filho, T., Costa, R.M., Frizzera-Neto, A., Arjunan, S., Kumar, D.: Towards SEMG classification based on Bayesian and k-NN to control a prosthetic hand. In: 2013 ISSNIP Biosignals and Biorobotics Conference: Biosignals and Robotics for Better and Safer Living (BRC), pp. 1–6 (2013). <https://doi.org/10.1109/BRC.2013.6487520>
 99. Triguero, I., Derrac, J., Garcia, S., Herrera, F.: A taxonomy and experimental study on prototype generation for nearest neighbor classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 42(1), 86–100 (2012). <https://doi.org/10.1109/TSMCC.2010.2103939>
 100. Triguero, I., González, S., Moyano, J., García, S., Alcalá-Fdez, J., Luengo, J., Fernández, A., Del Jesus, M.J., Sanchez, L., Herrera, F.: Keel 3.0: An open source software for multi-stage analysis in data mining. *International Journal of Computational Intelligence Systems* 10(1), 1238–1249 (2017). <https://doi.org/10.2991/ijcis.10.1.82>
 101. Visconti, P., Gaetani, F., Zappatore, G., Primiceri, P.: Technical features and functionalities of Myo armband: an overview on related literature and advanced applications of myoelectric armbands mainly focused on arm prostheses. *International Journal on Smart Sensing and Intelligent Systems* 11, 1–25 (2018). <https://doi.org/10.21307/ijssis-2018-005>
 102. Vujaklija I, Farina D, Aszmann O (2016) New developments in prosthetic arm systems. *Orthopedic Research and Reviews* 20168:31–39. <https://doi.org/10.2147/ORR.S71468>
 103. Wen-Juh Kang, Jiue-Rou Shiu, Cheng-Kung Cheng, Jin-Shin Lai, Hen-Wai Tsao, Te-Son Kuo (1995) The application of cepstral coefficients and maximum likelihood method in EMG pattern recognition [movements classification]. *IEEE Transactions on Biomedical Engineering* 42(8):777–785. <https://doi.org/10.1109/10.398638>
 104. Winter DA (2005) *Biomechanics and motor control of human movement*, 4th edn. John Wiley & Sons, Hoboken, N.J
 105. Wu X, Kumar V, Ross Quinlan J, Ghosh J, Yang Q, Motoda H, McLachlan GJ, Ng A, Liu B, Yu PS, Zhou ZH, Steinbach M, Hand DJ, Steinberg D (2007) Top 10 algorithms in data mining. *Knowl. Inf. Syst.* 14(1):1–37. <https://doi.org/10.1007/s10115-007-0114-2>
 106. Yonghong Huang, Englehart, K.B., Hudgins, B., Chan, A.D.C.: A Gaussian mixture model based classification scheme for myoelectric control of powered upper limb prostheses. *IEEE Transactions on Biomedical Engineering* 52(11), 1801–1811 (2005). <https://doi.org/10.1109/TBME.2005.856295>
 107. Zhang LQ, Shiavi R, Hunt MA, Chen JJ (1991) Clustering analysis and pattern discrimination of EMG linear envelopes. *IEEE Transactions on Biomedical Engineering* 38:777–784
 108. Zhang, Z., Wong, C., Yang, G.Z.: Forearm functional movement recognition using spare channel surface electromyography. pp. 1–6 (2013). <https://doi.org/10.1109/BSN.2013.6575507>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Tim Sziburis is with the Institute for Neuroinformatics, Ruhr University Bochum, and the Ruhr West University of Applied Sciences. His research interests include machine learning, control engineering and embedded systems for biomedical data processing. He holds a Dott. Mag. (cum laude) in Computer Science from the University of Trento, an M.Sc. in ICT Innovation from TU Berlin/EIT Digital, as well as an M.Sc. in Computational Engineering Science from TU Berlin. He spent research periods at CERN and the German Aerospace Center (DLR), among others.

Markus Nowak received the M.Sc. degree in mechanical engineering from the Technical University of Munich. In his master's curriculum, he focused on medical engineering and numerical mechanics. His master thesis was carried in cooperation with the Robotics and Mechatronics Center of the German Aerospace Center (DLR), Oberpfaffenhofen. After his master's, he continued as a Research Fellow at the Robotics and Mechatronics Center with a focus on human-machine interfaces for controlling upper-limb prosthetics.

Davide Brunelli received the M.S. (cum laude) and Ph.D. degrees in electrical engineering from the University of Bologna, Italy, in 2002 and 2007, respectively. He is an Associate Professor with the University of Trento, Italy. He has published more than 260 research papers in international journals or proceedings of international conferences. His research interests include the development of new techniques of energy scavenging for low-power embedded systems and energy-neutral wearable devices.